

Contents

Deploy and update Windows 10

- Deploy Windows 10 with Microsoft 365

- What's new in Windows 10 deployment

- Windows 10 deployment scenarios

- Windows Autopilot

Subscription Activation

- Windows 10 Subscription Activation

- Windows 10 Enterprise E3 in CSP

- Configure VDA for Subscription Activation

- Deploy Windows 10 Enterprise licenses

Resolve upgrade errors

- Resolve Windows 10 upgrade errors

- Quick fixes

- SetupDiag

- Troubleshooting upgrade errors

- Windows error reporting

- Upgrade error codes

- Log files

- Resolution procedures

- Submit Windows 10 upgrade errors

Deploy Windows 10

- Deploying Windows 10

- Windows Autopilot

- Windows 10 upgrade paths

- Windows 10 edition upgrade

- Windows 10 volume license media

- Windows 10 in S mode

- Switch to Windows 10 Pro/Enterprise from S mode

- Windows 10 deployment test lab

Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit

Deploy Windows 10 in a test lab using System Center Configuration Manager

Plan for Windows 10 deployment

Windows 10 Enterprise FAQ for IT Pros

Windows 10 deployment considerations

Windows 10 compatibility

Windows 10 infrastructure requirements

Volume Activation [client]

Plan for volume activation [client]

Activate using Key Management Service [client]

Activate using Active Directory-based activation [client]

Activate clients running Windows 10

Monitor activation [client]

Use the Volume Activation Management Tool [client]

Appendix: Information sent to Microsoft during activation [client]

Application Compatibility Toolkit (ACT) Technical Reference

SUA User's Guide

Using the SUA Wizard

Using the SUA Tool

Compatibility Administrator User's Guide

Using the Compatibility Administrator Tool

Managing Application-Compatibility Fixes and Custom Fix Databases

Using the Sdbinst.exe Command-Line Tool

Compatibility Fixes for Windows 10, Windows 8, Windows 7, and Windows Vista

Deploy Windows 10 with the Microsoft Deployment Toolkit

Get started with the Microsoft Deployment Toolkit (MDT)

Key features in MDT

MDT Lite Touch components

Prepare for deployment with MDT

Create a Windows 10 reference image

Deploy a Windows 10 image using MDT

Build a distributed environment for Windows 10 deployment

Refresh a Windows 7 computer with Windows 10

Replace a Windows 7 computer with a Windows 10 computer

Perform an in-place upgrade to Windows 10 with MDT

Configure MDT settings

- Set up MDT for BitLocker

- Configure MDT deployment share rules

- Configure MDT for UserExit scripts

- Simulate a Windows 10 deployment in a test environment

- Use the MDT database to stage Windows 10 deployment information

- Assign applications using roles in MDT

- Use web services in MDT

- Use Orchestrator runbooks with MDT

Deploy Windows 10 with System Center 2012 R2 Configuration Manager

- Integrate Configuration Manager with MDT

- Prepare for Zero Touch Installation of Windows 10 with Configuration Manager

- Create a custom Windows PE boot image with Configuration Manager

- Add a Windows 10 operating system image using Configuration Manager

- Create an application to deploy with Windows 10 using Configuration Manager

- Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager

- Create a task sequence with Configuration Manager and MDT

- Finalize the operating system configuration for Windows 10 deployment with Configuration Manager

- Deploy Windows 10 using PXE and Configuration Manager

- Monitor the Windows 10 deployment with Configuration Manager

- Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager

- Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager

- Perform an in-place upgrade to Windows 10 using Configuration Manager

Windows 10 deployment tools

- Windows 10 deployment scenarios and tools

- Convert MBR partition to GPT

- Configure a PXE server to load Windows PE

- Windows ADK for Windows 10 scenarios for IT Pros

Deploy Windows To Go in your organization

Windows To Go: feature overview

Best practice recommendations for Windows To Go

Deployment considerations for Windows To Go

Prepare your organization for Windows To Go

Security and data protection considerations for Windows To Go

Windows To Go: frequently asked questions

Volume Activation Management Tool (VAMT) Technical Reference

Introduction to VAMT

Active Directory-Based Activation Overview

Install and Configure VAMT

VAMT Requirements

Install VAMT

Configure Client Computers

Add and Manage Products

Add and Remove Computers

Update Product Status

Remove Products

Manage Product Keys

Add and Remove a Product Key

Install a Product Key

Install a KMS Client Key

Manage Activations

Perform Online Activation

Perform Proxy Activation

Perform KMS Activation

Perform Local Reactivation

Activate an Active Directory Forest Online

Activate by Proxy an Active Directory Forest

Manage VAMT Data

Import and Export VAMT Data

Use VAMT in Windows PowerShell

VAMT Step-by-Step Scenarios

[Scenario 1: Online Activation](#)

[Scenario 2: Proxy Activation](#)

[Scenario 3: KMS Client Activation](#)

VAMT Known Issues

User State Migration Tool (USMT) Technical Reference

User State Migration Tool (USMT) Overview Topics

[User State Migration Tool \(USMT\) Overview](#)

[Getting Started with the User State Migration Tool \(USMT\)](#)

[Windows Upgrade and Migration Considerations](#)

User State Migration Tool (USMT) How-to topics

[Exclude Files and Settings](#)

[Extract Files from a Compressed USMT Migration Store](#)

[Include Files and Settings](#)

[Migrate Application Settings](#)

[Migrate EFS Files and Certificates](#)

[Migrate User Accounts](#)

[Reroute Files and Settings](#)

[Verify the Condition of a Compressed Migration Store](#)

User State Migration Tool (USMT) Troubleshooting

[Common Issues](#)

[Frequently Asked Questions](#)

[Log Files](#)

[Return Codes](#)

[USMT Resources](#)

User State Migration Toolkit (USMT) Reference

[USMT Requirements](#)

[USMT Best Practices](#)

[How USMT Works](#)

[Plan Your Migration](#)

[User State Migration Tool \(USMT\) Command-line Syntax](#)

[USMT XML Reference](#)

Offline Migration Reference

[Install fonts in Windows 10](#)

[Update Windows 10](#)

[Update Windows 10 in enterprise deployments](#)

[Windows as a service](#)

[Windows as a service - introduction](#)

[Quick guide to Windows as a service](#)

[Servicing stack updates](#)

[Overview of Windows as a service](#)

[Prepare servicing strategy for Windows 10 updates](#)

[Build deployment rings for Windows 10 updates](#)

[Assign devices to servicing channels for Windows 10 updates](#)

[Get started](#)

[Get started with Windows Update](#)

[How Windows Update works](#)

[Windows Update log files](#)

[How to troubleshoot Windows Update](#)

[Common Windows Update errors](#)

[Windows Update error code reference](#)

[Other Windows Update resources](#)

[Optimize delivery](#)

[Optimize Windows 10 update delivery](#)

[Delivery Optimization for Windows 10 updates](#)

[Set up Delivery Optimization for Windows 10 updates](#)

[Delivery Optimization reference](#)

[Configure BranchCache for Windows 10 updates](#)

[Whitepaper: Windows Updates using forward and reverse differentials](#)

[Best practices](#)

[Best practices for feature updates on mission-critical devices](#)

[Deploy feature updates during maintenance windows](#)

[Deploy feature updates for user-initiated installations](#)

[Conclusion](#)

Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile

Use Windows Update for Business

Deploy updates using Windows Update for Business

Configure Windows Update for Business

Integrate Windows Update for Business with management solutions

Walkthrough: use Group Policy to configure Windows Update for Business

Walkthrough: use Intune to configure Windows Update for Business

Use Windows Server Update Services

Deploy Windows 10 updates using Windows Server Update Services

Enable FoD and language pack updates in Windows Update

Deploy Windows 10 updates using System Center Configuration Manager

Manage device restarts after updates

Manage additional Windows Update settings

Determine the source of Windows updates

Windows Analytics

Windows Analytics overview

Windows Analytics in the Azure Portal

Windows Analytics and privacy

Upgrade Readiness

Manage Windows upgrades with Upgrade Readiness

Upgrade Readiness architecture

Upgrade Readiness requirements

Get started

Get started with Upgrade Readiness

Upgrade Readiness deployment script

Use Upgrade Readiness

Use Upgrade Readiness to manage Windows upgrades

Upgrade overview

Step 1: Identify apps

Step 2: Resolve issues

Step 3: Deploy Windows

Step 4: Monitor deployment

[Additional insights](#)

[Targeting a new operating system version](#)

[Monitor Windows Updates](#)

[Monitor Windows Updates with Update Compliance](#)

[Get started with Update Compliance](#)

[Use Update Compliance](#)

[Need Attention! report](#)

[Security Update Status report](#)

[Feature Update Status report](#)

[Windows Defender AV Status report](#)

[Delivery Optimization in Update Compliance](#)

[Update Compliance Perspectives](#)

[Device Health](#)

[Device Health overview](#)

[Get started with Device Health](#)

[Using Device Health](#)

[Enrolling devices in Windows Analytics](#)

[Troubleshooting Windows Analytics and FAQ](#)

Deploy Windows 10 with Microsoft 365

6/18/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic provides a brief overview of Microsoft 365 and describes how to use a free 90-day trial account to review some of the benefits of Microsoft 365.

[Microsoft 365](#) is a new offering from Microsoft that combines [Windows 10](#) with [Office 365](#), and [Enterprise Mobility and Security](#) (EMS). See the [M365 Enterprise poster](#) for an overview.

For Windows 10 deployment, Microsoft 365 includes a fantastic deployment advisor that can walk you through the entire process of deploying Windows 10. The wizard supports multiple Windows 10 deployment methods, including:

- Windows Autopilot
- In-place upgrade
- Deploying Windows 10 upgrade with Intune
- Deploying Windows 10 upgrade with System Center Configuration Manager
- Deploying a computer refresh with System Center Configuration Manager

Free trial account

If you already have a Microsoft services subscription account and access to the Microsoft 365 Admin Center

From the [Microsoft 365 Admin Center](#), go to Billing and then Purchase services. In the Enterprise Suites section of the service offerings, you will find Microsoft 365 E3 and Microsoft 365 E5 tiles. There are "Start Free Trial" options available for your selection by hovering your mouse over the tiles.

If you do not already have a Microsoft services subscription

You can check out the Microsoft 365 deployment advisor and other resources for free! Just follow the steps below.

NOTE

If you have not run a setup guide before, you will see the **Prepare your environment** guide first. This is to make sure you have basics covered like domain verification and a method for adding users. At the end of the "Prepare your environment" guide, there will be a **Ready to continue** button that sends you to the original guide that was selected.

1. [Obtain a free M365 trial.](#)
2. Check out the [Microsoft 365 deployment advisor](#).
3. Also check out the [Windows Analytics deployment advisor](#). This advisor will walk you through deploying [Upgrade Readiness](#), [Update Compliance](#), and [Device Health](#).

That's all there is to it!

Examples of these two deployment advisors are shown below.

- [Microsoft 365 deployment advisor example](#)

- Windows Analytics deployment advisor example

Microsoft 365 deployment advisor example

The screenshot shows the Microsoft 365 deployment advisor interface. On the left, a vertical sidebar contains a progress bar with 9 steps:

- Step 1: Get started (green checkmark)
- Step 2: Prepare your domains and users (green checkmark)
- Step 3: Verify readiness to upgrade (red dot)
- Step 4: Prepare Configuration Manager (grey dot)
- Step 5: Add a Windows 10 operating system image (grey dot)
- Step 6: Create a task sequence (grey dot)
- Step 7: Edit the task sequence (grey dot)
- Step 8: Finalize the operating system configuration (grey dot)
- Step 9: Deploy and monitor Windows 10 (grey dot)

The main content area is titled "Verify readiness to upgrade Windows". It includes an introductory paragraph, a link to "What does USMT migrate?", and a section for "Review the following requirements before starting your Windows 10 deployment." This section contains three expandable categories:

- Windows editions eligible for upgrade (expanded)
- Supported devices (expanded)
- Deployment preparation (collapsed)

Below these categories, there is a section titled "Make sure you have the following before you start configuring the deployment:" followed by a list of requirements:

- PXE-enabled Windows PE boot image.** To use PXE to deploy an operating system, you must have both x86 and x64 PXE-enabled boot images distributed to one or more PXE-enabled distribution points. Use this information to enable PXE on a boot image and distribute the boot image to distribution points:
 - To enable PXE on a boot image, select **Deploy this boot image from the PXE-enabled distribution point** from the **Data Source** tab in the boot image properties.
 - If you change the properties for the boot image, redistribute the boot image to distribution points. For more information, see [Distribute content](#).
- Windows 10 installation media.** The installation media should be located on a separate drive, with the ISO already mounted. You can obtain the ISO from [MSDN Subscriber Downloads](#) or from the [Volume Licensing Service Center](#).
- Microsoft Deployment Toolkit (MDT).** MDT is a free, supported download from Microsoft that adds approximately 280 enhancements to Windows operating system deployment with System Center Configuration Manager. For more information, see [Microsoft Deployment Toolkit](#).
- Backups of user data.** Although user data will be migrated in the upgrade, best practice is to configure a backup scenario. For example, export all user data to a OneDrive account, BitLocker To Go-encrypted USB flash drive, or network file server. For more information, see [Back up or transfer data in Windows](#).

At the bottom of the main content area, there is a "Next" button with a right arrow icon, and a "Back" button with a left arrow icon. Below the "Back" button is a "Start Over" button. In the bottom right corner of the interface, there are "Need help?" and "Feedback" buttons.

Windows Analytics deployment advisor example

M365 Enterprise poster

Microsoft 365 Enterprise

A complete, intelligent solution that empowers everyone to be creative and work together securely.

Microsoft 365 Enterprise = Office 365 & Windows 10 Enterprise & Enterprise Mobility + Security

Unlocks creativity <ul style="list-style-type: none"> • Create compelling content with intelligent tools • Work effectively in the cloud and mobile • Discover information in new ways • Connect to important enterprise devices 	PowerPoint Designer, Editor, Smart Lookup, Excel Insights	Touch and ink support, 3D	
Built for teamwork <ul style="list-style-type: none"> • Email and calendar with Exchange • Connect to mobile devices, work with shared mail • Video, voice, and chat with Teams and Microsoft Teams • Network across the organization with Teams • Connect with other 3rd parties 	Exchange Online, SharePoint Online, Skype for Business, Microsoft Teams, Yammer, Office 365 ProPlus		
Integrated for simplicity <ul style="list-style-type: none"> • Microsoft 365 offers a single management & billing experience • Broad support for PC, Mac, iOS, & Android platforms • Configuration management & user self-management 	Office 365 ProPlus deployment	Windows 10 deployment with upgrade in place and Autopilot	Auto-enrollment of Windows PCs and devices
Intelligent security <ul style="list-style-type: none"> • Identity & access management • Information protection • Threat protection • Security management 	Office 365 Advanced Threat Protection, Office 365 Multi-Factor Authentication, SharePoint Online and Exchange Online conditional access policies, Office 365 Threat Intelligence, Azure Information Protection (AIP), Data Loss Prevention policies,	Windows Defender Advanced Threat Protection (ATP), Windows Hello for Business, Windows Information Protection (WIP)	Microsoft Intune device-based conditional access policies, Azure AD Privileged Identity Management (PIM), Advanced Threat Analytics, Azure Advanced Threat Protection, Microsoft Cloud App Security, Azure Multi-Factor Authentication, and others

Microsoft 365 Enterprise plans	Operating system Windows 10 Enterprise	Office Applications Word, Excel, PowerPoint, Outlook	Email & calendar Exchange Online	Chat based workplace Microsoft Teams	Schedule & Task Management Outlook, MyCalendar	Video, voice, & meetings Skype for Business, Microsoft Teams	Social & internet SharePoint Online	Threat protection Microsoft Defender for Office 365, Microsoft Defender for Endpoint, Microsoft Defender for Cloud	Identity & access management Azure Active Directory, Microsoft Entra ID	Device & app management Microsoft Intune	Information protection Microsoft Information Protection, Microsoft Purview	Advanced compliance Microsoft Purview, Microsoft Defender for Cloud	Analytics Power BI, Microsoft Dynamics 365
E3 An intelligent solution bringing together Office 365, Windows 10, and Enterprise Mobility + Security (EMS).	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
E5 All of E3 plus the latest advanced threat protection, security, and collaboration tools.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F1 Purpose-built to connect frontline workers to the tools and resources needed to do their best work.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The Modern Workplace

Modern Desktop	Compliance	Security	Teamwork	Firstline Worker
Intelligent security Built-in advanced security, powered by cloud intelligence Enhanced productivity Cloud-connected and designed to drive success of Office and mobile Flexible management Cloud-based management of office administration, app/PCs, settings, and updates Simplified updates Streamlined provisioning and deployment tools make it easy to update and scale up to scale	Assess and manage risk Deliver ongoing risk assessment, actionable insights Protect & govern sensitive data Automatically classify, protect, and govern data Streamline & respond Respond to regulatory, legal or information requests quickly Broad set of compliance standards Provides comprehensive set of international and industry-specific compliance offerings	Identity and access management Protect user identities and control access to sensitive resources based on your risk level Information protection Protect documents and emails are automatically protected Threat protection Protect against advanced threats and secure quality when connected Security management Gain visibility and control over security tools	Collaboration Rich and secure toolset that brings together everything a team needs to be more productive Communication Online meeting solution in the cloud Leadership connection A single solution for connecting, communicating, and engaging with teams across the organization Learning & sharing Build communities where professional, cross-business knowledge is shared best practices, and discuss new releases	Footer culture & community Provide space for teams to connect and share Train & upskill employees Manage and deliver content and roles Digitize business process Modernize frontline tasks, activities, and workflows Deliver real-time expertise Close technology gaps and rethink productivity

Business Value Scenarios

To see how the Microsoft 365 Business Value Scenarios can impact your business, use the Value Discovery Workshop.

- Go to transform.microsoft.com
- Sign in with your LinkedIn account.
- Provide some details of your organization and your role in it.
- Identify areas of interest and your needs.
- Click **Create Workshop**.

	Office 365 ProPlus	Exchange Online	SharePoint Online	OneDrive for Business	Skype for Business Online	Microsoft Teams	Microsoft StaffHub	Project Online	EMS	Windows 10
Collaborate on documents in real time	■	■	■	■	■	■	■	■	■	■
Access collective knowledge	■	■	■	■	■	■	■	■	■	■
Empower users to transform business processes	■	■	■	■	■	■	■	■	■	■
Share the company culture	■	■	■	■	■	■	■	■	■	■
Change projects, tasks, and deadlines	■	■	■	■	■	■	■	■	■	■
Engage your frontline workers	■	■	■	■	■	■	■	■	■	■
Use intelligent assistance for design, writing, & content discovery	■	■	■	■	■	■	■	■	■	■
Discover insights, analyze your data	■	■	■	■	■	■	■	■	■	■
Streamline your work habits	■	■	■	■	■	■	■	■	■	■
Communicate with your team to stay informed	■	■	■	■	■	■	■	■	■	■
Communicate with partners, colleagues, and customers	■	■	■	■	■	■	■	■	■	■
Stream and share data	■	■	■	■	■	■	■	■	■	■
Work securely from anywhere	■	■	■	■	■	■	■	■	■	■
Manage access of critical apps, content, and mobility	■	■	■	■	■	■	■	■	■	■
Protect your information	■	■	■	■	■	■	■	■	■	■
Detect and protect against external threats	■	■	■	■	■	■	■	■	■	■
Protect your users and their accounts	■	■	■	■	■	■	■	■	■	■
Support your organization with enhanced privacy and compliance	■	■	■	■	■	■	■	■	■	■

Roadmap to adoption

Get the help you need to streamline:

- Design and planning.
- Implementation of an integrated and secure configuration.
- User adoption and the realization of business value in your organization.

1. Networking

2. Identity

3. Windows 10

4. Office 365 ProPlus

5. Mobile device management

6. Information protection

Security built into each phase

Exchange Online

SharePoint Online

OneDrive for Business

Microsoft Teams

Migration

Teams and SharePoint Online sites for highly regulated data

Workloads and scenarios

Self-guided at aka.ms/m365deploy >

Guided by FastTrack at microsoft.com/fasttrack/microsoft-365 >



Related Topics

- Windows 10 deployment scenarios
- Modern Desktop Deployment Center

What's new in Windows 10 deployment

6/18/2019 • 8 minutes to read • [Edit Online](#)

Applies to

- Windows 10

In this topic

This topic provides an overview of new solutions and online content related to deploying Windows 10 in your organization.

- For an all-up overview of new features in Windows 10, see [What's new in Windows 10](#).
- For a detailed list of changes to Windows 10 ITPro TechNet library content, see [Online content change history](#).

Recent additions to this page

[SetupDiag 1.4.1](#) is released.

The [Windows ADK for Windows 10, version 1903](#) is available.

New [Windows Autopilot](#) content is available.

[Windows 10 Subscription Activation](#) now supports Windows 10 Education.

The Modern Desktop Deployment Center

The [Modern Desktop Deployment Center](#) has launched with tons of content to help you with large-scale deployment of Windows 10 and Office 365 ProPlus.

Microsoft 365

Microsoft 365 is a new offering from Microsoft that combines

- Windows 10
- Office 365
- Enterprise Mobility and Security (EMS).

See [Deploy Windows 10 with Microsoft 365](#) for an overview, which now includes a link to download a nifty [M365 Enterprise poster](#).

Windows 10 servicing and support

- **Delivery Optimization:** Improved Peer Efficiency for enterprises and educational institutions with complex networks is enabled with of [new policies](#). This now supports Office 365 ProPlus updates, and Intune content, with System Center Configuration Manager content coming soon!
- **Automatic Restart Sign-on (ARSO):** Windows will automatically logon as the user and lock their device in order to complete the update, ensuring that when the user returns and unlocks the device, the update will be completed.
- **Windows Update for Business:** There will now be a single, common start date for phased deployments (no more SAC-T designation). In addition, there will a new notification and reboot scheduling experience for end users, the ability to enforce update installation and reboot deadlines, and the ability to provide end user control over reboots for a specific time period.

- **Update rollback improvements:** You can now automatically recover from startup failures by removing updates if the startup failure was introduced after the installation of recent driver or quality updates. When a device is unable to start up properly after the recent installation of Quality of driver updates, Windows will now automatically uninstall the updates to get the device back up and running normally.
- **Pause updates:** We have extended the ability to pause updates for both feature and monthly updates. This extension ability is for all editions of Windows 10, including Home. You can pause both feature and monthly updates for up to 35 days (seven days at a time, up to five times). Once the 35-day pause period is reached, you will need to update your device before pausing again.
- **Improved update notifications:** When there's an update requiring you to restart your device, you'll see a colored dot on the Power button in the Start menu and on the Windows icon in your taskbar.
- **Intelligent active hours:** To further enhance active hours, users will now have the option to let Windows Update intelligently adjust active hours based on their device-specific usage patterns. You must enable the intelligent active hours feature for the system to predict device-specific usage patterns.
- **Improved update orchestration to improve system responsiveness:** This feature will improve system performance by intelligently coordinating Windows updates and Microsoft Store updates, so they occur when users are away from their devices to minimize disruptions.

Microsoft previously announced that we are [extending support](#) for Windows 10 Enterprise and Windows 10 Education editions to 30 months from the version release date. This includes all past versions and future versions that are targeted for release in September (versions ending in 09, ex: 1809). Future releases that are targeted for release in March (versions ending in 03, ex: 1903) will continue to be supported for 18 months from their release date. All releases of Windows 10 Home, Windows 10 Pro, and Office 365 ProPlus will continue to be supported for 18 months (there is no change for these editions). These support policies are summarized in the table below.

Products	March targeted releases	September targeted releases
Windows 10 Enterprise		30 months (formerly 18 months)
Windows 10 Education		
Windows 10 Pro	18 months	18 months
Windows 10 Home		
Office 365 ProPlus		

Windows 10 Enterprise upgrade

Windows 10 version 1703 includes a Windows 10 Enterprise E3 and E5 benefit to Microsoft customers with Enterprise Agreements (EA) or Microsoft Products & Services Agreements (MPSA). These customers can now subscribe users to Windows 10 Enterprise E3 or E5 and activate their subscriptions on up to five devices. Virtual machines can also be activated. For more information, see [Windows 10 Enterprise Subscription Activation](#).

Windows 10 Enterprise E3 launched in the Cloud Solution Provider (CSP) channel on September 1, 2016. Previously, only organizations with a Microsoft Volume Licensing Agreement could deploy Windows 10 Enterprise to their users. With Windows 10 Enterprise E3 in CSP, small and medium-sized organizations can more easily take advantage of Windows 10 Enterprise features.

For more information, see [Windows 10 Enterprise E3 in CSP](#)

Deployment solutions and tools

Windows Autopilot

[Windows Autopilot](#) streamlines and automates the process of setting up and configuring new devices, with minimal interaction required from the end user. You can also use Windows Autopilot to reset, repurpose and recover devices.

The following Windows Autopilot features are available in Windows 10, version 1903 and later:

- [Windows Autopilot for white glove deployment](#) is new in Windows 10, version 1903. "White glove" deployment enables partners or IT staff to pre-provision devices so they are fully configured and business ready for your users.
- The Intune [enrollment status page](#) (ESP) now tracks Intune Management Extensions.
- [Cortana voiceover](#) and speech recognition during OOB is disabled by default for all Windows 10 Pro Education, and Enterprise SKUs.
- Windows Autopilot is self-updating during OOB. Starting with the Windows 10, version 1903 Autopilot functional and critical updates will begin downloading automatically during OOB.
- Windows Autopilot will set the [diagnostics data](#) level to Full on Windows 10 version 1903 and later during OOB.

Windows 10 Subscription Activation

Windows 10 Education support has been added to Windows 10 Subscription Activation.

With Windows 10, version 1903, you can step-up from Windows 10 Pro Education to the enterprise-grade edition for educational institutions – Windows 10 Education. For more information, see [Windows 10 Subscription Activation](#).

SetupDiag

[SetupDiag](#) is a standalone diagnostic tool that can be used to obtain details about why a Windows 10 upgrade was unsuccessful.

SetupDiag version 1.4.1 was released on 5/17/2019.

Upgrade Readiness

The Upgrade Readiness tool moved from public preview to general availability on March 2, 2017.

Upgrade Readiness helps you ensure that applications and drivers are ready for a Windows 10 upgrade. The solution provides up-to-date application and driver inventory, information about known issues, troubleshooting guidance, and per-device readiness and tracking details.

The development of Upgrade Readiness has been heavily influenced by input from the community the development of new features is ongoing. To begin using Upgrade Readiness, add it to an existing Operation Management Suite (OMS) workspace or sign up for a new OMS workspace with the Upgrade Readiness solution enabled.

For more information about Upgrade Readiness, see the following topics:

- [Windows Analytics blog](#)
- [Manage Windows upgrades with Upgrade Readiness](#)

Update Compliance

Update Compliance helps you to keep Windows 10 devices in your organization secure and up-to-date.

Update Compliance is a solution built using OMS Logs and Analytics that provides information about installation status of monthly quality and feature updates. Details are provided about the deployment progress of existing updates and the status of future updates. Information is also provided about devices that might need attention to resolve issues.

For more information about Update Compliance, see [Monitor Windows Updates with Update Compliance](#).

Device Health

Device Health is the newest Windows Analytics solution that complements the existing Upgrade Readiness and Update Compliance solutions by helping to identify device crashes and the cause. Device drivers that are causing

crashes are identified along with alternative drivers that might reduce the number of crashes. Windows Information Protection misconfigurations are also identified. For more information, see [Monitor the health of devices with Device Health](#)

MBR2GPT

MBR2GPT.EXE converts a disk from Master Boot Record (MBR) to GUID Partition Table (GPT) partition style without modifying or deleting data on the disk. Previously, it was necessary to image, then wipe and reload a disk to change from MBR format to GPT.

There are many benefits to converting the partition style of a disk to GPT, including the use of larger disk partitions, added data reliability, and faster boot and shutdown speeds. The GPT format also enables you to use the Unified Extensible Firmware Interface (UEFI) which replaces the Basic Input/Output System (BIOS) firmware interface. Security features of Windows 10 that require UEFI mode include: Secure Boot, Early Launch Anti-malware (ELAM) driver, Windows Trusted Boot, Measured Boot, Device Guard, Credential Guard, and BitLocker Network Unlock.

For more information, see [MBR2GPT.EXE](#).

Microsoft Deployment Toolkit (MDT)

MDT build 8456 (12/19/2018) is available, including support for Windows 10, version 1809, and Windows Server 2019.

For more information about MDT, see the [MDT resource page](#).

Windows Assessment and Deployment Kit (ADK)

The Windows Assessment and Deployment Kit (Windows ADK) contains tools that can be used by IT Pros to deploy Windows. See the following topics:

- [What's new in ADK kits and tools](#)
- [Windows ADK for Windows 10 scenarios for IT Pros](#)

Testing and validation guidance

Windows 10 deployment proof of concept (PoC)

The Windows 10 PoC guide enables you to test Windows 10 deployment in a virtual environment and become familiar with deployment tools such as MDT and Configuration Manager. The PoC guide provides step-by-step instructions for installing and using Hyper-V to create a virtual lab environment. The guide makes extensive use of Windows PowerShell to streamline each phase of the installation and setup.

For more information, see the following guides:

- [Step by step guide: Configure a test lab to deploy Windows 10](#)
- [Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit](#)
- [Deploy Windows 10 in a test lab using System Center Configuration Manager](#)

Troubleshooting guidance

[Resolve Windows 10 upgrade errors](#) was published in October of 2016 and will continue to be updated with new fixes. The topic provides a detailed explanation of the Windows 10 upgrade process and instructions on how to locate, interpret, and resolve specific errors that can be encountered during the upgrade process.

Online content change history

The following topics provide a change history for Windows 10 ITPro TechNet library content related to deploying and using Windows 10.

[Change history for Deploy Windows 10](#)

[Change history for Access Protection](#)

[Change history for Device Security](#)

[Change history for Threat Protection](#)

Related topics

[Overview of Windows as a service](#)

[Windows 10 deployment considerations](#)

[Windows 10 release information](#)

[Windows 10 Specifications & Systems Requirements](#)

[Windows 10 upgrade paths](#)

[Windows 10 deployment tools](#)

Windows 10 deployment scenarios

6/18/2019 • 11 minutes to read • [Edit Online](#)

Applies to

- Windows 10

To successfully deploy the Windows 10 operating system in your organization, it is important to understand the different ways that it can be deployed, especially now that there are new scenarios to consider. Choosing among these scenarios, and understanding the capabilities and limitations of each, is a key task.

The following table summarizes various Windows 10 deployment scenarios. The scenarios are each assigned to one of three categories.

- Modern deployment methods are recommended unless you have a specific need to use a different procedure. These methods are supported with existing tools such as Microsoft Deployment Toolkit (MDT) and System Center Configuration Manager. These methods are discussed in detail on the [Modern Desktop Deployment Center](#).
- Dynamic deployment methods enable you to configure applications and settings for specific use cases.
- Traditional deployment methods use existing tools to deploy operating system images.

Category	Scenario	Description	More information
Modern	Windows Autopilot	Customize the out-of-box-experience (OOBE) for your organization, and deploy a new system with apps and settings already configured.	Overview of Windows Autopilot
	In-place upgrade	Use Windows Setup to update your OS and migrate apps and settings. Rollback data is saved in Windows.old.	Perform an in-place upgrade to Windows 10 with MDT Perform an in-place upgrade to Windows 10 using Configuration Manager
Dynamic	Subscription Activation	Switch from Windows 10 Pro to Enterprise when a subscribed user signs in.	Windows 10 Subscription Activation
	AAD / MDM	The device is automatically joined to AAD and configured by MDM.	Azure Active Directory integration with MDM
	Provisioning packages	Using the Windows Imaging and Configuration Designer tool, create provisioning packages that can be applied to devices.	Configure devices without MDM

Traditional	Bare metal	Deploy a new device, or wipe an existing device and deploy with a fresh image.	Deploy a Windows 10 image using MDT Install a new version of Windows on a new computer with System Center Configuration Manager
	Refresh	Also called wipe and load. Redeploy a device by saving the user state, wiping the disk, then restoring the user state.	Refresh a Windows 7 computer with Windows 10 Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager
	Replace	Replace an existing device with a new one by saving the user state on the old device and then restoring it to the new device.	Replace a Windows 7 computer with a Windows 10 computer Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager

IMPORTANT

The Windows Autopilot and Subscription Activation scenarios require that the beginning OS be Windows 10 version 1703, or later.

Except for clean install scenarios such as traditional bare metal and Windows Autopilot, all the methods described can optionally migrate apps and settings to the new OS.

Modern deployment methods

Modern deployment methods embrace both traditional on-prem and cloud services to deliver a simple, streamlined, cost effective deployment experience.

Windows Autopilot

Windows Autopilot is a new suite of capabilities designed to simplify and modernize the deployment and management of new Windows 10 PCs. Windows Autopilot enables IT professionals to customize the Out of Box Experience (OOBE) for Windows 10 PCs and provide end users with a fully configured new Windows 10 device after just a few clicks. There are no images to deploy, no drivers to inject, and no infrastructure to manage. Users can go through the deployment process independently, without the need consult their IT administrator.

For more information about Windows Autopilot, see [Overview of Windows Autopilot](#) and [Modernizing Windows deployment with Windows Autopilot](#).

In-place upgrade

For existing computers running Windows 7, Windows 8, or Windows 8.1, the recommended path for organizations deploying Windows 10 leverages the Windows installation program (Setup.exe) to perform an in-place upgrade, which automatically preserves all data, settings, applications, and drivers from the existing operating system version. This requires the least IT effort, because there is no need for any complex deployment infrastructure.

Although consumer PCs will be upgraded using Windows Update, organizations want more control over the

process. This is accomplished by leveraging tools like System Center Configuration Manager or the Microsoft Deployment Toolkit to completely automate the upgrade process through simple task sequences.

The in-place upgrade process is designed to be extremely reliable, with the ability to automatically roll back to the previous operating system if any issues are encountered during the deployment process, without any IT staff involvement. Rolling back manually can also be done by leveraging the automatically-created recovery information (stored in the Windows.old folder), in case any issues are encountered after the upgrade is finished. The upgrade process is also typically faster than traditional deployments, because applications do not need to be reinstalled as part of the process.

Because existing applications are preserved through the process, the upgrade process uses the standard Windows installation media image (Install.wim); custom images are not needed and cannot be used because the upgrade process is unable to deal with conflicts between apps in the old and new operating system. (For example, Contoso Timecard 1.0 in Windows 7 and Contoso Timecard 3.0 in the Windows 10 image.)

Scenarios that support in-place upgrade with some additional procedures include changing from BIOS to UEFI boot mode and upgrade of devices that use non-Microsoft disk encryption software.

- **Legacy BIOS to UEFI booting:** To perform an in-place upgrade on a UEFI-capable system that currently boots using legacy BIOS, first perform the in-place upgrade to Windows 10, maintaining the legacy BIOS boot mode. Windows 10 does not require UEFI, so it will work fine to upgrade a system using legacy BIOS emulation. After the upgrade, if you wish to enable Windows 10 features that require UEFI (such as Secure Boot), you can convert the system disk to a format that supports UEFI boot using the [MBR2GPT](#) tool. Note: [UEFI specification](#) requires GPT disk layout. After the disk has been converted, you must also configure the firmware to boot in UEFI mode.
- **Non-Microsoft disk encryption software:** While devices encrypted with BitLocker can easily be upgraded, more work is necessary for non-Microsoft disk encryption tools. Some ISVs will provide instructions on how to integrate their software into the in-place upgrade process. Check with your ISV to see if they have instructions. The following articles provide details on how to provision encryption drivers for use during Windows Setup via the ReflectDrivers setting:
 - [Windows Setup Automation Overview](#)
 - [Windows Setup Command-Line Options](#)

There are some situations where you cannot use in-place upgrade; in these situations, you can use traditional deployment (wipe-and-load) instead. Examples of these situations include:

- Changing from Windows 7, Windows 8, or Windows 8.1 x86 to Windows 10 x64. The upgrade process cannot change from a 32-bit operating system to a 64-bit operating system, because of possible complications with installed applications and drivers.
- Windows To Go and Boot from VHD installations. The upgrade process is unable to upgrade these installations. Instead, new installations would need to be performed.
- Updating existing images. While it might be tempting to try to upgrade existing Windows 7, Windows 8, or Windows 8.1 images to Windows 10 by installing the old image, upgrading it, and then recapturing the new Windows 10 image, this is not supported – preparing an upgraded OS for imaging (using Sysprep.exe) is not supported and will not work when it detects the upgraded OS.
- Dual-boot and multi-boot systems. The upgrade process is designed for devices running a single OS; if using dual-boot or multi-boot systems with multiple operating systems (not leveraging virtual machines for the second and subsequent operating systems), additional care should be taken.

Dynamic provisioning

For new PCs, organizations have historically replaced the version of Windows included on the device with their own custom Windows image, because this was often faster and easier than leveraging the preinstalled version.

But this is an added expense due to the time and effort required. With the new dynamic provisioning capabilities and tools provided with Windows 10, it is now possible to avoid this.

The goal of dynamic provisioning is to take a new PC out of the box, turn it on, and transform it into a productive organization device, with minimal time and effort. The types of transformations that are available include:

Windows 10 Subscription Activation

Windows 10 Subscription Activation is a modern deployment method that enables you to change the SKU from Pro to Enterprise with no keys and no reboots. For more information about Subscription Activation, see [Windows 10 Subscription Activation](#).

Azure Active Directory (AAD) join with automatic mobile device management (MDM) enrollment

In this scenario, the organization member just needs to provide their work or school user ID and password; the device can then be automatically joined to Azure Active Directory and enrolled in a mobile device management (MDM) solution with no additional user interaction. Once done, the MDM solution can finish configuring the device as needed. For more information, see [Azure Active Directory integration with MDM](#).

Provisioning package configuration

Using the [Windows Imaging and Configuration Designer \(ICD\)](#), IT administrators can create a self-contained package that contains all of the configuration, settings, and apps that need to be applied to a machine. These packages can then be deployed to new PCs through a variety of means, typically by IT professionals. For more information, see [Configure devices without MDM](#).

These scenarios can be used to enable "choose your own device" (CYOD) programs where the organization's users can pick their own PC and not be restricted to a small list of approved or certified models (programs that are difficult to implement using traditional deployment scenarios).

While the initial Windows 10 release includes a variety of provisioning settings and deployment mechanisms, these will continue to be enhanced and extended based on feedback from organizations. As with all Windows features, organizations can submit suggestions for additional features through the Windows Feedback app or through their Microsoft Support contacts.

Traditional deployment:

New versions of Windows have typically been deployed by organizations using an image-based process built on top of tools provided in the [Windows Assessment and Deployment Kit](#), Windows Deployment Services, the [Deploy Windows 10 with the Microsoft Deployment Toolkit](#), and [System Center Configuration Manager](#).

With the release of Windows 10, all of these tools are being updated to fully support Windows 10. Although newer scenarios such as in-place upgrade and dynamic provisioning may reduce the need for traditional deployment capabilities in some organizations, these traditional methods remain important and will continue to be available to organizations that need them.

The traditional deployment scenario can be divided into different sub-scenarios. These are explained in detail in the following sections, but the following provides a brief summary:

- **New computer.** A bare-metal deployment of a new machine.
- **Computer refresh.** A reinstall of the same machine (with user-state migration and an optional full Windows Imaging (WIM) image backup).
- **Computer replace.** A replacement of the old machine with a new machine (with user-state migration and an optional full WIM image backup).

New computer

Also called a "bare metal" deployment. This scenario occurs when you have a blank machine you need to deploy, or an existing machine you want to wipe and redeploy without needing to preserve any existing data. The setup

starts from a boot media, using CD, USB, ISO, or Pre-Boot Execution Environment (PXE). You can also generate a full offline media that includes all the files needed for a client deployment, allowing you to deploy without having to connect to a central deployment share. The target can be a physical computer, a virtual machine, or a Virtual Hard Disk (VHD) running on a physical computer (boot from VHD).

The deployment process for the new machine scenario is as follows:

1. Start the setup from boot media (CD, USB, ISO, or PXE).
2. Wipe the hard disk clean and create new volume(s).
3. Install the operating system image.
4. Install other applications (as part of the task sequence).

After taking these steps, the computer is ready for use.

Computer refresh

A refresh is sometimes called wipe-and-load. The process is normally initiated in the running operating system. User data and settings are backed up and restored later as part of the deployment process. The target can be the same as for the new computer scenario.

The deployment process for the wipe-and-load scenario is as follows:

1. Start the setup on a running operating system.
2. Save the user state locally.
3. Wipe the hard disk clean (except for the folder containing the backup).
4. Install the operating system image.
5. Install other applications.
6. Restore the user state.

After taking these steps, the machine is ready for use.

Computer replace

A computer replace is similar to the refresh scenario. However, since we are replacing the machine, we divide this scenario into two main tasks: backup of the old client and bare-metal deployment of the new client. As with the refresh scenario, user data and settings are backed up and restored.

The deployment process for the replace scenario is as follows:

1. Save the user state (data and settings) on the server through a backup job on the running operating system.
2. Deploy the new computer as a bare-metal deployment.

Note

In some situations, you can use the replace scenario even if the target is the same machine. For example, you can use replace if you want to modify the disk layout from the master boot record (MBR) to the GUID partition table (GPT), which will allow you to take advantage of the Unified Extensible Firmware Interface (UEFI) functionality. You can also use replace if the disk needs to be repartitioned since user data needs to be transferred off the disk.

Related topics

- [Upgrade to Windows 10 with the Microsoft Deployment Toolkit](#)

- [Upgrade to Windows 10 with System Center Configuration Manager](#)
- [Deploy Windows 10 with System Center 2012 R2 Configuration Manager](#)
- [Deploy Windows 10 with the Microsoft Deployment Toolkit](#)
- [Windows setup technical reference](#)
- [Windows Imaging and Configuration Designer](#)
- [UEFI firmware](#)

Overview of Windows Autopilot

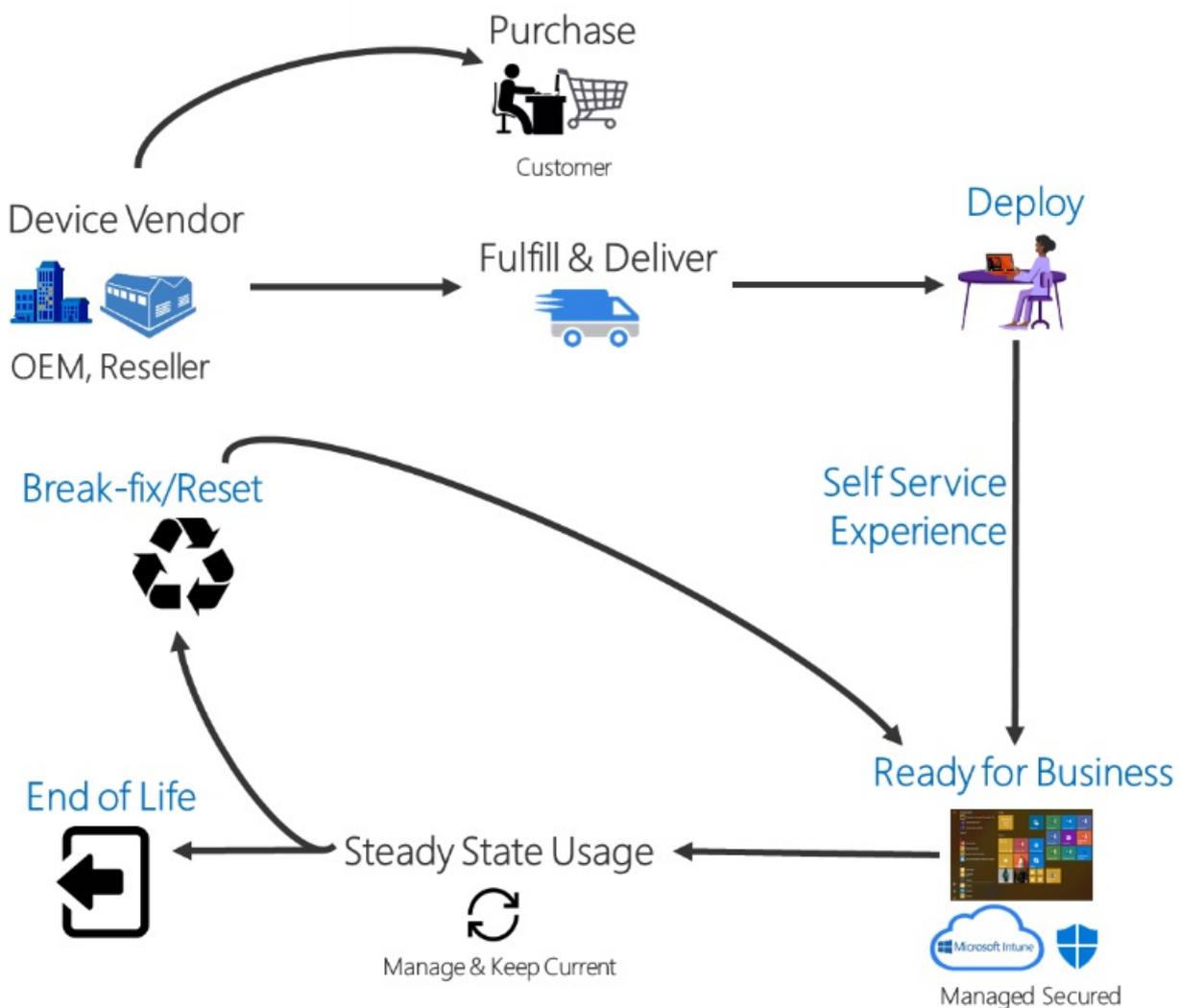
6/18/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Windows Autopilot is a collection of technologies used to set up and pre-configure new devices, getting them ready for productive use. You can also use Windows Autopilot to reset, repurpose and recover devices. This solution enables an IT department to achieve the above with little to no infrastructure to manage, with a process that's easy and simple.

Windows Autopilot is designed to simplify all parts of the lifecycle of Windows devices, for both IT and end users, from initial deployment through the eventual end of life. Leveraging cloud-based services, it can reduce the overall costs for deploying, managing, and retiring devices by reducing the amount of time that IT needs to spend on these processes and the amount of infrastructure that they need to maintain, while ensuring ease of use for all types of end users. See the following diagram:



When initially deploying new Windows devices, Windows Autopilot leverages the OEM-optimized version of Windows 10 that is preinstalled on the device, saving organizations the effort of having to maintain custom images and drivers for every model of device being used. Instead of re-imaging the device, your existing Windows 10 installation can be transformed into a "business-ready" state, applying settings and policies, installing apps, and

even changing the edition of Windows 10 being used (e.g. from Windows 10 Pro to Windows 10 Enterprise) to support advanced features.

Once deployed, Windows 10 devices can be managed by tools such as Microsoft Intune, Windows Update for Business, System Center Configuration Manager, and other similar tools. Windows Autopilot can also be used to re-purpose a device by leveraging Windows Autopilot Reset to quickly prepare a device for a new user, or in break/fix scenarios to enable a device to quickly be brought back to a business-ready state.

Windows Autopilot enables you to:

- Automatically join devices to Azure Active Directory (Azure AD) or Active Directory (via Hybrid Azure AD Join). See [Introduction to device management in Azure Active Directory](#) for more information about the differences between these two join options.
- Auto-enroll devices into MDM services, such as Microsoft Intune (*Requires an Azure AD Premium subscription*).
- Restrict the Administrator account creation.
- Create and auto-assign devices to configuration groups based on a device's profile.
- Customize OOB content specific to the organization.

Windows Autopilot walkthrough

The following video shows the process of setting up Windows Autopilot:

<https://www.youtube.com/embed/4K4hC5NchbE>

Benefits of Windows Autopilot

Traditionally, IT pros spend a lot of time building and customizing images that will later be deployed to devices. Windows Autopilot introduces a new approach.

From the user's perspective, it only takes a few simple operations to make their device ready to use.

From the IT pro's perspective, the only interaction required from the end user is to connect to a network and to verify their credentials. Everything beyond that is automated.

Requirements

Windows 10 version 1703 or higher is required to use Windows Autopilot. See [Windows Autopilot requirements](#) for detailed information on software, configuration, network, and licensing requirements.

Related topics

[Enroll Windows devices in Intune by using Windows Autopilot](#)

[Windows Autopilot scenarios and capabilities](#)

Windows 10 Subscription Activation

5/21/2019 • 13 minutes to read • [Edit Online](#)

Starting with Windows 10, version 1703 Windows 10 Pro supports the Subscription Activation feature, enabling users to “step-up” from Windows 10 Pro to **Windows 10 Enterprise** automatically if they are subscribed to Windows 10 Enterprise E3 or E5.

With Windows 10, version 1903 the Subscription Activation feature also supports the ability to step-up from Windows 10 Pro Education to the Enterprise grade edition for educational institutions – **Windows 10 Education**.

The Subscription Activation feature eliminates the need to manually deploy Windows 10 Enterprise or Education images on each target device, then later standing up on-prem key management services such as KMS or MAK based activation, entering GVLKs, and subsequently rebooting client devices.

Subscription Activation for Windows 10 Enterprise

With Windows 10, version 1703 both Windows 10 Enterprise E3 and Windows 10 Enterprise E5 are available as online services via subscription. Deploying [Windows 10 Enterprise](#) in your organization can now be accomplished with no keys and no reboots.

If you are running Windows 10, version 1703 or later:

- Devices with a current Windows 10 Pro license can be seamlessly upgraded to Windows 10 Enterprise.
- Product key-based Windows 10 Enterprise software licenses can be transitioned to Windows 10 Enterprise subscriptions.

Organizations that have an Enterprise agreement can also benefit from the new service, using traditional Active Directory-joined devices. In this scenario, the Active Directory user that signs in on their device must be synchronized with Azure AD using [Azure AD Connect Sync](#).

Subscription Activation for Windows 10 Education

Subscription Activation for Education works the same as the Enterprise version, but in order to use Subscription Activation for Education, you must have a device running Windows 10 Pro Education, version 1903 or later and an active subscription plan with a Windows 10 Enterprise license. For more information, see the [requirements](#) section.

In this article

- [Inherited Activation](#): Description of a new feature available in Windows 10, version 1803 and later.
- [The evolution of Windows 10 deployment](#): A short history of Windows deployment.
- [Requirements](#): Prerequisites to use the Windows 10 Subscription Activation model.
- [Benefits](#): Advantages of Windows 10 subscription-based licensing.
- [How it works](#): A summary of the subscription-based licensing option.
- [Virtual Desktop Access \(VDA\)](#): Enable Windows 10 Subscription Activation for VMs in the cloud.

For information on how to deploy Windows 10 Enterprise licenses, see [Deploy Windows 10 Enterprise licenses](#).

Inherited Activation

Inherited Activation is a new feature available in Windows 10, version 1803 that allows Windows 10 virtual

machines to inherit activation state from their Windows 10 host.

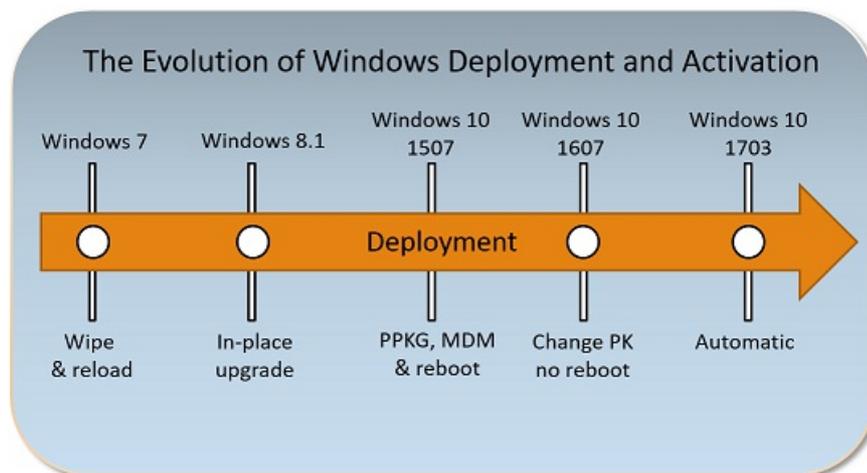
When a user with Windows 10 E3/E5 or A3/A5 license assigned creates a new Windows 10 virtual machine (VM) using a Windows 10 local host, the VM inherits the activation state from a host machine independent of whether user signs on with a local account or using an Azure Active Directory (AAD) account on a VM.

To support Inherited Activation, both the host computer and the VM must be running Windows 10, version 1803 or later.

The evolution of deployment

The original version of this section can be found at [Changing between Windows SKUs](#).

The following figure illustrates how deploying Windows 10 has evolved with each release. With this release, deployment is automatic.



- **Windows 7** required you to redeploy the operating system using a full wipe-and-load process if you wanted to change from Windows 7 Professional to Windows 10 Enterprise.
- **Windows 8.1** added support for a Windows 8.1 Pro to Windows 8.1 Enterprise in-place upgrade (considered a "repair upgrade" because the OS version was the same before and after). This was a lot easier than wipe-and-load, but it was still time-consuming.
- **Windows 10, version 1507** added the ability to install a new product key using a provisioning package or using MDM to change the SKU. This required a reboot, which would install the new OS components, and took several minutes to complete. However, it was a lot quicker than in-place upgrade.
- **Windows 10, version 1607** made a big leap forward. Now you can just change the product key and the SKU instantly changes from Windows 10 Pro to Windows 10 Enterprise. In addition to provisioning packages and MDM, you can just inject a key using SLMGR.VBS (which injects the key into WMI), so it became trivial to do this using a command line.
- **Windows 10, version 1703** made this "step-up" from Windows 10 Pro to Windows 10 Enterprise automatic for those that subscribed to Windows 10 Enterprise E3 or E5 via the CSP program.
- **Windows 10, version 1709** adds support for Windows 10 Subscription Activation, very similar to the CSP support but for large enterprises, enabling the use of Azure AD for assigning licenses to users. When those users sign in on an AD or Azure AD-joined machine, it automatically steps up from Windows 10 Pro to Windows 10 Enterprise.
- **Windows 10, version 1803** updates Windows 10 Subscription Activation to enable pulling activation keys directly from firmware for devices that support firmware-embedded keys. It is no longer necessary to run a script to perform the activation step on Windows 10 Pro prior to activating Enterprise. For virtual machines and hosts running Windows 10, version 1803 [Inherited Activation](#) is also enabled.
- **Windows 10, version 1903** updates Windows 10 Subscription Activation to enable step up from Windows 10

Pro Education to Windows 10 Education for those with a qualifying Windows 10 or Microsoft 365 subscription.

Requirements

Windows 10 Enterprise requirements

For Microsoft customers with Enterprise Agreements (EA) or Microsoft Products & Services Agreements (MPSA), you must have the following:

- Windows 10 (Pro or Enterprise) version 1703 or later installed on the devices to be upgraded.
- Azure Active Directory (Azure AD) available for identity management.
- Devices must be Azure AD-joined or Hybrid Azure AD joined. Workgroup-joined or Azure AD registered devices are not supported.

NOTE

An issue has been identified with Hybrid Azure AD joined devices that have enabled [multi-factor authentication](#) (MFA). If a user signs into a device using their Active Directory account and MFA is enabled, the device will not successfully upgrade to their Windows Enterprise subscription. To resolve this issue, the user must either sign in with an Azure Active Directory account, or you must disable MFA for this user during the 30-day polling period and renewal.

For Microsoft customers that do not have EA or MPSA, you can obtain Windows 10 Enterprise E3/E5 or A3/A5 through a cloud solution provider (CSP). Identity management and device requirements are the same when you use CSP to manage licenses, with the exception that Windows 10 Enterprise E3 is also available through CSP to devices running Windows 10, version 1607. For more information about obtaining Windows 10 Enterprise E3 through your CSP, see [Windows 10 Enterprise E3 in CSP](#).

If devices are running Windows 7 or Windows 8.1, see [New Windows 10 upgrade benefits for Windows Cloud Subscriptions in CSP](#)

Windows 10 Education requirements

1. Windows 10 Pro Education, version 1903 or later installed on the devices to be upgraded.
2. A device with a Windows 10 Pro Education digital license. You can confirm this information in Settings > Update & Security> Activation.
3. The Education tenant must have an active subscription to Microsoft 365 with a Windows 10 Enterprise license or a Windows 10 Enterprise or Education subscription.
4. Devices must be Azure AD-joined or Hybrid Azure AD joined. Workgroup-joined or Azure AD registered devices are not supported.

If Windows 10 Pro is converted to Windows 10 Pro Education [using benefits available in Store for Education](#), then the feature will not work. You will need to re-image the device using a Windows 10 Pro Education edition.

Benefits

With Windows 10 Enterprise or Windows 10 Education, businesses and institutions can benefit from enterprise-level security and control. Previously, only organizations with a Microsoft Volume Licensing Agreement could deploy Windows 10 Education or Windows 10 Enterprise to their users. Now, with Windows 10 Enterprise E3 or A3 and E5 or A5 being available as a true online service, it is available in select channels thus allowing all organizations to take advantage of enterprise-grade Windows 10 features. To compare Windows 10 editions and review pricing, see the following:

- [Compare Windows 10 editions](#)
- [Enterprise Mobility + Security Pricing Options](#)

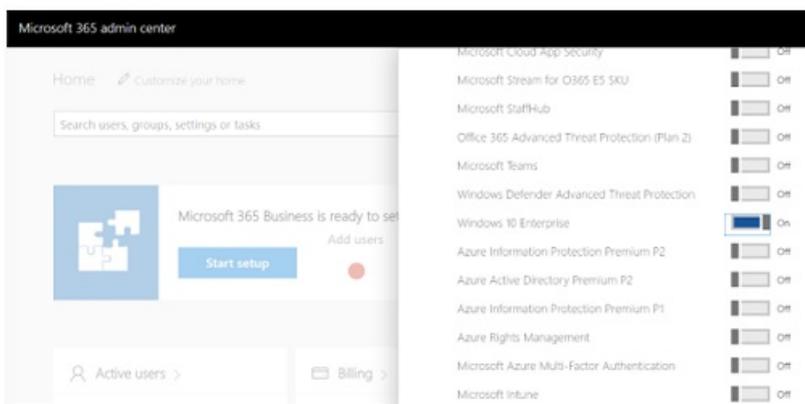
You can benefit by moving to Windows as an online service in the following ways:

1. Licenses for Windows 10 Enterprise and Education are checked based on Azure Active Directory (Azure AD) credentials, so now businesses have a systematic way to assign licenses to end users and groups in their organization.
2. User logon triggers a silent edition upgrade, with no reboot required
3. Support for mobile worker/BYOD activation; transition away from on-prem KMS and MAK keys.
4. Compliance support via seat assignment.
5. Licenses can be updated to different users dynamically, enabling you to optimize your licensing investment against changing needs.

How it works

The device is AAD joined from Settings > Accounts > Access work or school.

The IT administrator assigns Windows 10 Enterprise to a user. See the following figure.

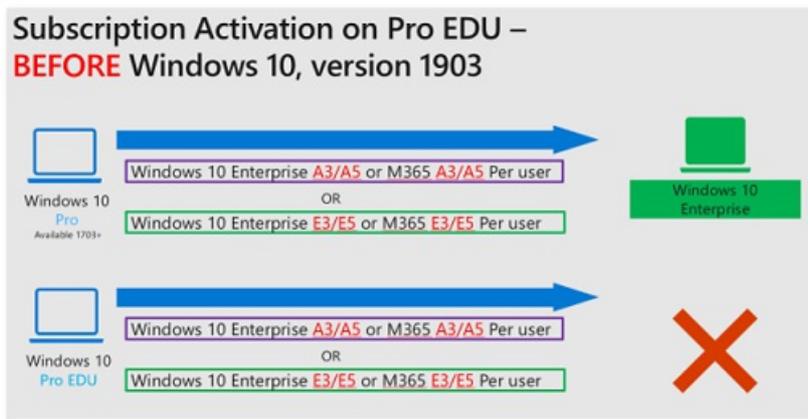


When a licensed user signs in to a device that meets requirements using their Azure AD credentials, the operating system steps up from Windows 10 Pro to Windows 10 Enterprise (or Windows 10 Pro Education to Windows 10 Education) and all the appropriate Windows 10 Enterprise/Education features are unlocked. When a user's subscription expires or is transferred to another user, the device reverts seamlessly to Windows 10 Pro / Windows 10 Pro Education edition, once current subscription validity expires.

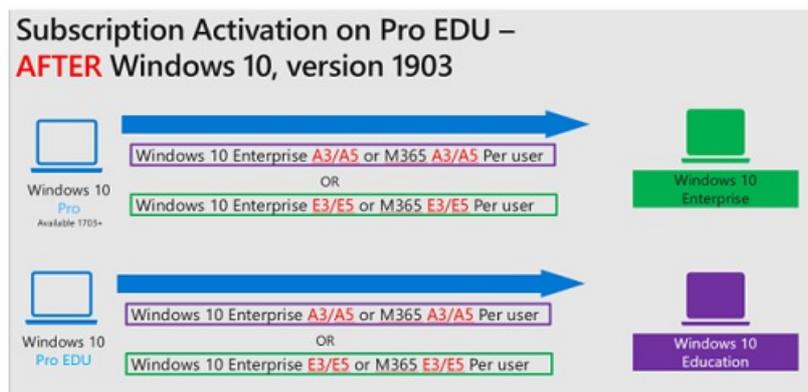
Devices running Windows 10 Pro, version 1703 or Windows 10 Pro Education, version 1903 or later can get Windows 10 Enterprise or Education Semi-Annual Channel on up to five devices for each user covered by the license. This benefit does not include Long Term Servicing Channel.

The following figures summarize how the Subscription Activation model works:

Before Windows 10, version 1903:



After Windows 10, version 1903:



Note:

1. A Windows 10 Pro Education device will only step up to Windows 10 Education edition when “Windows 10 Enterprise” license is assigned from M365 Admin center (as of May 2019).
2. A Windows 10 Pro device will only step up to Windows 10 Enterprise edition when “Windows 10 Enterprise” license is assigned from M365 Admin center (as of May 2019).

Scenarios

Scenario #1: You are using Windows 10, version 1803 or above, and just purchased Windows 10 Enterprise E3 or E5 subscriptions (or have had an E3 or E5 subscription for a while but haven’t yet deployed Windows 10 Enterprise).

All of your Windows 10 Pro devices will step-up to Windows 10 Enterprise, and devices that are already running Windows 10 Enterprise will migrate from KMS or MAK activated Enterprise edition to Subscription activated Enterprise edition when a Subscription Activation-enabled user signs in to the device.

Scenario #2: You are using Windows 10, version 1607, 1703, or 1709 with KMS for activation, and just purchased Windows 10 Enterprise E3 or E5 subscriptions (or have had an E3 or E5 subscription for a while but haven’t yet deployed Windows 10 Enterprise).

To change all of your Windows 10 Pro devices to Windows 10 Enterprise, run the following command on each computer:

```
csccript.exe c:\windows\system32\s1mgr.vbs /ipk NPPR9-FWDCX-D2C8J-H872K-2YT43
```

The command causes the OS to change to Windows 10 Enterprise and then seek out the KMS server to reactivate. This key comes from [Appendix A: KMS Client Setup Keys](#) in the Volume Activation guide. It is also possible to inject the Windows 10 Pro key from this article if you wish to step back down from Enterprise to Pro.

Scenario #3: Using Azure AD-joined devices or Active Directory-joined devices running Windows 10 1709 or

later, and with Azure AD synchronization configured, just follow the steps in [Deploy Windows 10 Enterprise licenses](#) to acquire a \$0 SKU and get a new Windows 10 Enterprise E3 or E5 license in Azure AD. Then, assign that license to all of your Azure AD users. These can be AD-synced accounts. The device will automatically change from Windows 10 Pro to Windows 10 Enterprise when that user signs in.

In summary, if you have a Windows 10 Enterprise E3 or E5 subscription, but are still running Windows 10 Pro, it's really simple (and quick) to move to Windows 10 Enterprise using one of the scenarios above.

If you're running Windows 7, it can be more work. A wipe-and-load approach works, but it is likely to be easier to upgrade from Windows 7 Pro directly to Windows 10 Enterprise. This is a supported path, and completes the move in one step. This method also works if you are running Windows 8.1 Pro.

Licenses

The following policies apply to acquisition and renewal of licenses on devices:

- Devices that have been upgraded will attempt to renew licenses about every 30 days, and must be connected to the Internet to successfully acquire or renew a license.
- If a device is disconnected from the Internet until its current subscription expires, the operating system will revert to Windows 10 Pro or Windows 10 Pro Education. As soon as the device is connected to the Internet again, the license will automatically renew.
- Up to five devices can be upgraded for each user license.
- If a device meets requirements and a licensed user signs in on that device, it will be upgraded.

Licenses can be reallocated from one user to another user, allowing you to optimize your licensing investment against changing needs.

When you have the required Azure AD subscription, group-based licensing is the preferred method to assign Enterprise E3 and E5 licenses to users. For more information, see [Group-based licensing basics in Azure AD](#).

Existing Enterprise deployments

If you are running Windows 10, version 1803 or later, Subscription Activation will automatically pull the firmware-embedded Windows 10 activation key and activate the underlying Pro License. The license will then step-up to Windows 10 Enterprise using Subscription Activation. This automatically migrates your devices from KMS or MAK activated Enterprise to Subscription activated Enterprise.

If you are using Windows 10, version 1607, 1703, or 1709 and have already deployed Windows 10 Enterprise, but you want to move away from depending on KMS servers and MAK keys for Windows client machines, you can seamlessly transition as long as the computer has been activated with a firmware-embedded Windows 10 Pro product key.

If the computer has never been activated with a Pro key, run the following script. Copy the text below into a .cmd file and run the file from an elevated command prompt:

```
@echo off
FOR /F "skip=1" %%A IN ('wmic path SoftwareLicensingService get OA3xOriginalProductKey') DO (
SET "ProductKey=%%A"
goto InstallKey
)

:InstallKey
IF [%ProductKey%]==[] (
echo No key present
) ELSE (
echo Installing %ProductKey%
changekey.exe /ProductKey %ProductKey%
)
)
```

Obtaining an Azure AD license

Enterprise Agreement/Software Assurance (EA/SA):

- Organizations with a traditional EA must order a \$0 SKU, process e-mails sent to the license administrator for the company, and assign licenses using Azure AD (ideally to groups using the new Azure AD Premium feature for group assignment). For more information, see [Enabling Subscription Activation with an existing EA](#).
- The license administrator can assign seats to Azure AD users with the same process that is used for O365.
- New EA/SA Windows Enterprise customers can acquire both an SA subscription and an associated \$0 cloud subscription.

Microsoft Products & Services Agreements (MPSA):

- Organizations with MPSA are automatically emailed the details of the new service. They must take steps to process the instructions.
- Existing MPSA customers will receive service activation emails that allow their customer administrator to assign users to the service.
- New MPSA customers who purchase the Software Subscription Windows Enterprise E3 and E5 will be enabled for both the traditional key-based and new subscriptions activation method.

Deploying licenses

See [Deploy Windows 10 Enterprise licenses](#).

Virtual Desktop Access (VDA)

Subscriptions to Windows 10 Enterprise are also available for virtualized clients. Windows 10 Enterprise E3 and E5 are available for Virtual Desktop Access (VDA) in Windows Azure or in another [qualified multitenant hoster](#).

Virtual machines (VMs) must be configured to enable Windows 10 Enterprise subscriptions for VDA. Active Directory-joined and Azure Active Directory-joined clients are supported. See [Enable VDA for Subscription Activation](#).

Related topics

[Connect domain-joined devices to Azure AD for Windows 10 experiences](#)

[Compare Windows 10 editions](#)

[Windows for business](#)

Windows 10 Enterprise E3 in CSP

6/18/2019 • 16 minutes to read • [Edit Online](#)

Windows 10 Enterprise E3 launched in the Cloud Solution Provider (CSP) channel on September 1, 2016. Windows 10 Enterprise E3 in CSP is a new offering that delivers, by subscription, exclusive features reserved for Windows 10 Enterprise edition. This offering is available through the Cloud Solution Provider (CSP) channel via the Partner Center as an online service. Windows 10 Enterprise E3 in CSP provides a flexible, per-user subscription for small- and medium-sized organizations (from one to hundreds of users). To take advantage of this offering, you must have the following:

- Windows 10 Pro, version 1607 (Windows 10 Anniversary Update) or later, installed and activated, on the devices to be upgraded
- Azure Active Directory (Azure AD) available for identity management

Starting with Windows 10, version 1607 (Windows 10 Anniversary Update), you can move from Windows 10 Pro to Windows 10 Enterprise more easily than ever before—no keys and no reboots. After one of your users enters the Azure AD credentials associated with a Windows 10 Enterprise E3 license, the operating system turns from Windows 10 Pro to Windows 10 Enterprise and all the appropriate Windows 10 Enterprise features are unlocked. When a subscription license expires or is transferred to another user, the Windows 10 Enterprise device seamlessly steps back down to Windows 10 Pro.

Previously, only organizations with a Microsoft Volume Licensing Agreement could deploy Windows 10 Enterprise to their users. Now, with Windows 10 Enterprise E3 in CSP, small- and medium-sized organizations can more easily take advantage of Windows 10 Enterprise features.

When you purchase Windows 10 Enterprise E3 via a partner, you get the following benefits:

- **Windows 10 Enterprise edition.** Devices currently running Windows 10 Pro, version 1607 can get Windows 10 Enterprise Current Branch (CB) or Current Branch for Business (CBB). This benefit does not include Long Term Service Branch (LTSB).
- **Support from one to hundreds of users.** Although the Windows 10 Enterprise E3 in CSP program does not have a limitation on the number of licenses an organization can have, the program is designed for small- and medium-sized organizations.
- **Deploy on up to five devices.** For each user covered by the license, you can deploy Windows 10 Enterprise edition on up to five devices.
- **Roll back to Windows 10 Pro at any time.** When a user's subscription expires or is transferred to another user, the Windows 10 Enterprise device reverts seamlessly to Windows 10 Pro edition (after a grace period of up to 90 days).
- **Monthly, per-user pricing model.** This makes Windows 10 Enterprise E3 affordable for any organization.
- **Move licenses between users.** Licenses can be quickly and easily reallocated from one user to another user, allowing you to optimize your licensing investment against changing needs.

How does the Windows 10 Enterprise E3 in CSP program compare with Microsoft Volume Licensing Agreements and Software Assurance?

- [Microsoft Volume Licensing](#) programs are broader in scope, providing organizations with access to licensing for all Microsoft products.
- [Software Assurance](#) provides organizations with the following categories of benefits:

- **Deployment and management.** These benefits include planning services, Microsoft Desktop Optimization (MDOP), Windows Virtual Desktop Access Rights, Windows-To-Go Rights, Windows Roaming Use Rights, Windows Thin PC, Windows RT Companion VDA Rights, and other benefits.
- **Training.** These benefits include training vouchers, online e-learning, and a home use program.
- **Support.** These benefits include 24x7 problem resolution support, backup capabilities for disaster recovery, System Center Global Service Monitor, and a passive secondary instance of SQL Server.
- **Specialized.** These benefits include step-up licensing availability (which enables you to migrate software from an earlier edition to a higher-level edition) and to spread license and Software Assurance payments across three equal, annual sums.

In addition, in Windows 10 Enterprise E3 in CSP, a partner can manage your licenses for you. With Software Assurance, you, the customer, manage your own licenses.

In summary, the Windows 10 Enterprise E3 in CSP program is an upgrade offering that provides small- and medium-sized organizations easier, more flexible access to the benefits of Windows 10 Enterprise edition, whereas Microsoft Volume Licensing programs and Software Assurance are broader in scope and provide benefits beyond access to Windows 10 Enterprise edition.

Compare Windows 10 Pro and Enterprise editions

Windows 10 Enterprise edition has a number of features that are unavailable in Windows 10 Pro. Table 1 lists the Windows 10 Enterprise features not found in Windows 10 Pro. Many of these features are security-related, whereas others enable finer-grained device management.

Table 1. Windows 10 Enterprise features not found in Windows 10 Pro

FEATURE	DESCRIPTION
Credential Guard	<p>This feature uses virtualization-based security to help protect security secrets (for example, NTLM password hashes, Kerberos Ticket Granting Tickets) so that only privileged system software can access them. This helps prevent Pass-the-Hash or Pass-the-Ticket attacks.</p> <p>Credential Guard has the following features:</p> <ul style="list-style-type: none"> • Hardware-level security. Credential Guard uses hardware platform security features (such as Secure Boot and virtualization) to help protect derived domain credentials and other secrets. • Virtualization-based security. Windows services that access derived domain credentials and other secrets run in a virtualized, protected environment that is isolated. • Improved protection against persistent threats. Credential Guard works with other technologies (e.g., Device Guard) to help provide further protection against attacks, no matter how persistent. • Improved manageability. Credential Guard can be managed through Group Policy, Windows Management Instrumentation (WMI), or Windows PowerShell. <p>For more information, see Protect derived domain credentials with Credential Guard.</p> <p><i>Credential Guard requires UEFI 2.3.1 or greater with Trusted Boot; Virtualization Extensions such as Intel VT-x, AMD-V, and SLAT must be enabled; x64 version of Windows; IOMMU, such as Intel VT-d, AMD-Vi; BIOS Lockdown; TPM 2.0 recommended for device health attestation (will use software if TPM 2.0 not present)</i></p>

FEATURE	DESCRIPTION
Device Guard	<p>This feature is a combination of hardware and software security features that allows only trusted applications to run on a device. Even if an attacker manages to get control of the Windows kernel, he or she will be much less likely to run executable code. Device Guard can use virtualization-based security (VBS) in Windows 10 Enterprise edition to isolate the Code Integrity service from the Windows kernel itself. With VBS, even if malware gains access to the kernel, the effects can be severely limited, because the hypervisor can prevent the malware from executing code.</p> <p>Device Guard does the following:</p> <ul style="list-style-type: none"> • Helps protect against malware • Helps protect the Windows system core from vulnerability and zero-day exploits • Allows only trusted apps to run <p>For more information, see Introduction to Device Guard.</p>
AppLocker management	<p>This feature helps IT pros determine which applications and files users can run on a device (also known as “whitelisting”). The applications and files that can be managed include executable files, scripts, Windows Installer files, dynamic-link libraries (DLLs), packaged apps, and packaged app installers.</p> <p>For more information, see AppLocker.</p>
Application Virtualization (App-V)	<p>This feature makes applications available to end users without installing the applications directly on users’ devices. App-V transforms applications into centrally managed services that are never installed and don’t conflict with other applications. This feature also helps ensure that applications are kept current with the latest security updates.</p> <p>For more information, see Getting Started with App-V for Windows 10.</p>
User Experience Virtualization (UE-V)	<p>With this feature, you can capture user-customized Windows and application settings and store them on a centrally managed network file share. When users log on, their personalized settings are applied to their work session, regardless of which device or virtual desktop infrastructure (VDI) sessions they log on to.</p> <p>UE-V provides the ability to do the following:</p> <ul style="list-style-type: none"> • Specify which application and Windows settings synchronize across user devices • Deliver the settings anytime and anywhere users work throughout the enterprise • Create custom templates for your third-party or line-of-business applications • Recover settings after hardware replacement or upgrade, or after re-imaging a virtual machine to its initial state <p>For more information, see User Experience Virtualization (UE-V) for Windows 10 overview.</p>

FEATURE	DESCRIPTION
Managed User Experience	<p>This feature helps customize and lock down a Windows device's user interface to restrict it to a specific task. For example, you can configure a device for a controlled scenario such as a kiosk or classroom device. The user experience would be automatically reset once a user signs off. You can also restrict access to services including Cortana or the Windows Store, and manage Start layout options, such as:</p> <ul style="list-style-type: none"> • Removing and preventing access to the Shut Down, Restart, Sleep, and Hibernate commands • Removing Log Off (the User tile) from the Start menu • Removing frequent programs from the Start menu • Removing the All Programs list from the Start menu • Preventing users from customizing their Start screen • Forcing Start menu to be either full-screen size or menu size • Preventing changes to Taskbar and Start menu settings

Deployment of Windows 10 Enterprise E3 licenses

See [Deploy Windows 10 Enterprise licenses](#).

Deploy Windows 10 Enterprise features

Now that you have Windows 10 Enterprise edition running on devices, how do you take advantage of the Enterprise edition features and capabilities? What are the next steps that need to be taken for each of the features discussed in [Table 1](#)?

The following sections provide you with the high-level tasks that need to be performed in your environment to help users take advantage of the Windows 10 Enterprise edition features.

Credential Guard*

You can implement Credential Guard on Windows 10 Enterprise devices by turning on Credential Guard on these devices. Credential Guard uses Windows 10 virtualization-based security features (Hyper-V features) that must be enabled on each device before you can turn on Credential Guard. You can turn on Credential Guard by using one of the following methods:

- **Automated.** You can automatically turn on Credential Guard for one or more devices by using Group Policy. The Group Policy settings automatically add the virtualization-based security features and configure the Credential Guard registry settings on managed devices.
- **Manual.** You can manually turn on Credential Guard by doing the following:
 - Add the virtualization-based security features by using Programs and Features or Deployment Image Servicing and Management (DISM).
 - Configure Credential Guard registry settings by using the Registry Editor or the [Device Guard and Credential Guard hardware readiness tool](#).

You can automate these manual steps by using a management tool such as System Center Configuration Manager.

For more information about implementing Credential Guard, see the following resources:

- [Protect derived domain credentials with Credential Guard](#)

- [PC OEM requirements for Device Guard and Credential Guard](#)
- [Device Guard and Credential Guard hardware readiness tool](#)

** Requires UEFI 2.3.1 or greater with Trusted Boot; Virtualization Extensions such as Intel VT-x, AMD-V, and SLAT must be enabled; x64 version of Windows; IOMMU, such as Intel VT-d, AMD-Vi; BIOS Lockdown; TPM 2.0 recommended for device health attestation (will use software if TPM 2.0 not present)*

Device Guard

Now that the devices have Windows 10 Enterprise, you can implement Device Guard on the Windows 10 Enterprise devices by performing the following steps:

1. **Optionally, create a signing certificate for code integrity policies.** As you deploy code integrity policies, you might need to sign catalog files or code integrity policies internally. To do this, you will either need a publicly issued code signing certificate (that you purchase) or an internal certificate authority (CA). If you choose to use an internal CA, you will need to create a code signing certificate.
2. **Create code integrity policies from “golden” computers.** When you have identified departments or roles that use distinctive or partly distinctive sets of hardware and software, you can set up “golden” computers containing that software and hardware. In this respect, creating and managing code integrity policies to align with the needs of roles or departments can be similar to managing corporate images. From each “golden” computer, you can create a code integrity policy and decide how to manage that policy. You can merge code integrity policies to create a broader policy or a master policy, or you can manage and deploy each policy individually.
3. **Audit the code integrity policy and capture information about applications that are outside the policy.** We recommend that you use “audit mode” to carefully test each code integrity policy before you enforce it. With audit mode, no application is blocked—the policy just logs an event whenever an application outside the policy is started. Later, you can expand the policy to allow these applications, as needed.
4. **Create a “catalog file” for unsigned line-of-business (LOB) applications.** Use the Package Inspector tool to create and sign a catalog file for your unsigned LOB applications. In later steps, you can merge the catalog file's signature into your code integrity policy so that applications in the catalog will be allowed by the policy.
5. **Capture needed policy information from the event log, and merge information into the existing policy as needed.** After a code integrity policy has been running for a time in audit mode, the event log will contain information about applications that are outside the policy. To expand the policy so that it allows for these applications, use Windows PowerShell commands to capture the needed policy information from the event log, and then merge that information into the existing policy. You can merge code integrity policies from other sources also, for flexibility in how you create your final code integrity policies.
6. **Deploy code integrity policies and catalog files.** After you confirm that you have completed all the preceding steps, you can begin deploying catalog files and taking code integrity policies out of audit mode. We strongly recommend that you begin this process with a test group of users. This provides a final quality-control validation before you deploy the catalog files and code integrity policies more broadly.
7. **Enable desired hardware security features.** Hardware-based security features—also called virtualization-based security (VBS) features—strengthen the protections offered by code integrity policies.

For more information about implementing Device Guard, see:

- [Planning and getting started on the Device Guard deployment process](#)
- [Device Guard deployment guide](#)

AppLocker management

You can manage AppLocker in Windows 10 Enterprise by using Group Policy. Group Policy requires that you have AD DS and that the Windows 10 Enterprise devices are joined to the your AD DS domain. You can create AppLocker rules by using Group Policy, and then target those rules to the appropriate devices.

For more information about AppLocker management by using Group Policy, see [AppLocker deployment guide](#).

App-V

App-V requires an App-V server infrastructure to support App-V clients. The primary App-V components that you must have are as follows:

- **App-V server.** The App-V server provides App-V management, virtualized app publishing, app streaming, and reporting services. Each of these services can be run on one server or can be run individually on multiple servers. For example, you could have multiple streaming servers. App-V clients contact App-V servers to determine which apps are published to the user or device, and then run the virtualized app from the server.
- **App-V sequencer.** The App-V sequencer is a typical client device that is used to sequence (capture) apps and prepare them for hosting from the App-V server. You install apps on the App-V sequencer, and the App-V sequencer software determines the files and registry settings that are changed during app installation. Then the sequencer captures these settings to create a virtualized app.
- **App-V client.** The App-V client must be enabled on any client device on which apps will be run from the App-V server. These will be the Windows 10 Enterprise E3 devices.

For more information about implementing the App-V server, App-V sequencer, and App-V client, see the following resources:

- [Getting Started with App-V for Windows 10](#)
- [Deploying the App-V server](#)
- [Deploying the App-V Sequencer and Configuring the Client](#)

UE-V

UE-V requires server- and client-side components that you you'll need to download, activate, and install. These components include:

- **UE-V service.** The UE-V service (when enabled on devices) monitors registered applications and Windows for any settings changes, then synchronizes those settings between devices.
- **Settings packages.** Settings packages created by the UE-V service store application settings and Windows settings. Settings packages are built, locally stored, and copied to the settings storage location.
- **Settings storage location.** This location is a standard network share that your users can access. The UE-V service verifies the location and creates a hidden system folder in which to store and retrieve user settings.
- **Settings location templates.** Settings location templates are XML files that UE-V uses to monitor and synchronize desktop application settings and Windows desktop settings between user computers. By default, some settings location templates are included in UE-V. You can also create, edit, or validate custom settings location templates by using the UE-V template generator. Settings location templates are not required for Windows applications.
- **Universal Windows applications list.** UE-V determines which Windows applications are enabled for settings synchronization using a managed list of applications. By default, this list includes most Windows applications.

For more information about deploying UE-V, see the following resources:

- [User Experience Virtualization \(UE-V\) for Windows 10 overview](#)
- [Get Started with UE-V](#)

- [Prepare a UE-V Deployment](#)

Managed User Experience

The Managed User Experience feature is a set of Windows 10 Enterprise edition features and corresponding settings that you can use to manage user experience. Table 2 describes the Managed User Experience settings (by category), which are only available in Windows 10 Enterprise edition. The management methods used to configure each feature depend on the feature. Some features are configured by using Group Policy, while others are configured by using Windows PowerShell, Deployment Image Servicing and Management (DISM), or other command-line tools. For the Group Policy settings, you must have AD DS with the Windows 10 Enterprise devices joined to your AD DS domain.

Table 2. Managed User Experience features

FEATURE	DESCRIPTION
Start layout customization	You can deploy a customized Start layout to users in a domain. No reimaging is required, and the Start layout can be updated simply by overwriting the .xml file that contains the layout. This enables you to customize Start layouts for different departments or organizations, with minimal management overhead. For more information on these settings, see Customize Windows 10 Start and taskbar with Group Policy .
Unbranded boot	You can suppress Windows elements that appear when Windows starts or resumes and can suppress the crash screen when Windows encounters an error from which it cannot recover. For more information on these settings, see Unbranded Boot .
Custom logon	You can use the Custom Logon feature to suppress Windows 10 UI elements that relate to the Welcome screen and shutdown screen. For example, you can suppress all elements of the Welcome screen UI and provide a custom logon UI. You can also suppress the Blocked Shutdown Resolver (BSDR) screen and automatically end applications while the OS waits for applications to close before a shutdown. For more information on these settings, see Custom Logon .
Shell launcher	Enables Assigned Access to run only a classic Windows app via Shell Launcher to replace the shell. For more information on these settings, see Shell Launcher .
Keyboard filter	You can use Keyboard Filter to suppress undesirable key presses or key combinations. Normally, users can use certain Windows key combinations like Ctrl+Alt+Delete or Ctrl+Shift+Tab to control a device by locking the screen or using Task Manager to close a running application. This is not desirable on devices intended for a dedicated purpose. For more information on these settings, see Keyboard Filter .

FEATURE	DESCRIPTION
Unified write filter	You can use Unified Write Filter (UWF) on your device to help protect your physical storage media, including most standard writable storage types that are supported by Windows, such as physical hard disks, solid-state drives, internal USB devices, external SATA devices, and so on. You can also use UWF to make read-only media appear to the OS as a writable volume. For more information on these settings, see Unified Write Filter .

Related topics

[Windows 10 Enterprise Subscription Activation](#)

[Connect domain-joined devices to Azure AD for Windows 10 experiences](#)

[Compare Windows 10 editions](#)

[Windows for business](#)

Configure VDA for Windows 10 Subscription Activation

6/19/2019 • 6 minutes to read • [Edit Online](#)

This document describes how to configure virtual machines (VMs) to enable [Windows 10 Subscription Activation](#) in a Windows Virtual Desktop Access (VDA) scenario. Windows VDA is a device or user-based licensing mechanism for managing access to virtual desktops.

Deployment instructions are provided for the following scenarios:

1. [Active Directory-joined VMs](#)
2. [Azure Active Directory-joined VMs](#)
3. [Azure Gallery VMs](#)

Requirements

- VMs must be running Windows 10 Pro, version 1703 (also known as the Creator's Update) or later.
- VMs must be Active Directory-joined or Azure Active Directory (AAD)-joined.
- VMs must be generation 1.
- VMs must be hosted by a [Qualified Multitenant Hoster](#) (QMTH).

Activation

Scenario 1

- The VM is running Windows 10, version 1803 or later.
- The VM is hosted in Azure or another [Qualified Multitenant Hoster](#) (QMTH).

When a user with VDA rights signs in to the VM using their AAD credentials, the VM is automatically stepped-up to Enterprise and activated. There is no need to perform Windows 10 Pro activation. This eliminates the need to maintain KMS or MAK in the qualifying cloud infrastructure.

Scenario 2

- The Hyper-V host and the VM are both running Windows 10, version 1803 or later.

[Inherited Activation](#) is enabled. All VMs created by a user with a Windows 10 E3 or E5 license are automatically activated independent of whether a user signs in with a local account or using an Azure Active Directory account.

Scenario 3

- The VM is running Windows 10, version 1703 or 1709, or the hoster is not an authorized [QMTH](#) partner.

In this scenario, the underlying Windows 10 Pro license must be activated prior to Subscription Activation of Windows 10 Enterprise. Activation is accomplished using a Windows 10 Pro Generic Volume License Key (GVLK) and a Volume License KMS activation server provided by the hoster. Alternatively, a KMS activation server on your corporate network can be used if you have configured a private connection, such as [ExpressRoute](#) or [VPN Gateway](#).

For examples of activation issues, see [Troubleshoot the user experience](#).

Active Directory-joined VMs

1. Use the following instructions to prepare the VM for Azure: [Prepare a Windows VHD or VHDX to upload to Azure](#)

2. (Optional) To disable network level authentication, type the following at an elevated command prompt:

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v  
UserAuthentication /t REG_DWORD /d 0 /f
```

3. At an elevated command prompt, type **sysdm.cpl** and press ENTER.

4. On the Remote tab, choose **Allow remote connections to this computer** and then click **Select Users**.

5. Click **Add**, type **Authenticated users**, and then click **OK** three times.

6. Follow the instructions to use sysprep at [Steps to generalize a VHD](#) and then start the VM again.

7. [Install Windows Configuration Designer](#).

8. Open Windows Configuration Designer and click **Provision desktop services**.

9. If you must activate Windows 10 Pro as described for [scenario 3](#), complete the following steps. Otherwise, skip to step 10.

a. Under **Name**, type **Desktop AD Enrollment Pro GVLK**, click **Finish**, and then on the **Set up device** page enter a device name.

- Note: You can use a different project name, but this name is also used with dism.exe in a subsequent step.

b. Under **Enter product key** type the Pro GVLK key: **W269N-WFGWX-YVC9B-4J6C9-T83GX**.

10. On the Set up network page, choose **Off**.

11. On the Account Management page, choose **Enroll into Active Directory** and then enter the account details.

- Note: This step is different for [Azure AD-joined VMs](#).

12. On the Add applications page, add applications if desired. This step is optional.

13. On the Add certificates page, add certificates if desired. This step is optional.

14. On the Finish page, click **Create**.

15. If you must activate Windows 10 Pro as described for [scenario 3](#), complete the following steps. Otherwise, skip to step 16.

a. In file explorer, double-click the VHD to mount the disk image. Determine the drive letter of the mounted image.

b. Type the following at an elevated command prompt. Replace the letter **G** with the drive letter of the mounted image, and enter the project name you used if it is different than the one suggested:

```
Dism.exe /Image=G:\ /Add-ProvisioningPackage /PackagePath: "Desktop AD Enrollment Pro GVLK.ppkg"
```

c. Right-click the mounted image in file explorer and click **Eject**.

16. See instructions at [Upload and create VM from generalized VHD](#) to log in to Azure, get your storage account details, upload the VHD, and create a managed image.

Azure Active Directory-joined VMs

IMPORTANT

Azure Active Directory (Azure AD) provisioning packages have a 180 day limit on bulk token usage. You will need to update the provisioning package and re-inject it into the image after 180 days. Existing virtual machines that are Azure AD-joined and deployed will not need to be recreated.

For Azure AD-joined VMs, follow the same instructions (above) as for [Active Directory-joined VMs](#) with the following exceptions:

- In step 9, during setup with Windows Configuration Designer, under **Name**, type a name for the project that indicates it is not for Active Directory joined VMs, such as **Desktop Bulk Enrollment Token Pro GVLK**.
- In step 11, during setup with Windows Configuration Designer, on the Account Management page, instead of enrolling in Active Directory, choose **Enroll in Azure AD**, click **Get Bulk Token**, sign in and add the bulk token using your organization's credentials.
- In step 15, sub-step 2, when entering the PackagePath, use the project name you entered in step 9 (ex: **Desktop Bulk Enrollment Token Pro GVLK.ppkg**)
- When attempting to access the VM using remote desktop, you will need to create a custom RDP settings file as described below in [Create custom RDP settings for Azure](#).

Azure Gallery VMs

1. (Optional) To disable network level authentication, type the following at an elevated command prompt:

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v  
UserAuthentication /t REG_DWORD /d 0 /f
```

2. At an elevated command prompt, type **sysdm.cpl** and press ENTER.
3. On the Remote tab, choose **Allow remote connections to this computer** and then click **Select Users**.
4. Click **Add**, type **Authenticated users**, and then click **OK** three times.
5. [Install Windows Configuration Designer](#).
6. Open Windows Configuration Designer and click **Provision desktop services**.
7. If you must activate Windows 10 Pro as described for [scenario 3](#), complete the following steps. Otherwise, skip to step 8.
 - a. Under **Name**, type **Desktop Bulk Enrollment Token Pro GVLK**, click **Finish**, and then on the **Set up device** page enter a device name.
 - b. Under **Enter product key** type the Pro GVLK key: **W269N-WFGWX-YVC9B-4J6C9-T83GX**.
8. Under **Name**, type **Desktop Bulk Enrollment**, click **Finish**, and then on the **Set up device** page enter a device name.
9. On the Set up network page, choose **Off**.
10. On the Account Management page, choose **Enroll in Azure AD**, click **Get Bulk Token**, sign in, and add the bulk token using your organizations credentials.
11. On the Add applications page, add applications if desired. This step is optional.
12. On the Add certificates page, add certificates if desired. This step is optional.

13. On the Finish page, click **Create**.
14. Copy the .ppkg file to the remote Virtual machine. Double click to initiate the provisioning package install. This will reboot the system.
 - When attempting to access the VM using remote desktop, you will need to create a custom RDP settings file as described [below](#).

Create custom RDP settings for Azure

To create custom RDP settings for Azure:

1. Open Remote Desktop Connection and enter the IP address or DNS name for the remote host.
2. Click **Show Options**, and then under Connection settings click **Save As** and save the RDP file to the location where you will use it.
3. Close the Remote Desktop Connection window and open Notepad.
4. Drag the RDP file into the Notepad window to edit it.
5. Enter or replace the line that specifies authentication level with the following two lines of text:

```
enablecredsspssupport:i:0  
authentication level:i:2
```

6. **enablecredsspssupport** and **authentication level** should each appear only once in the file.
7. Save your changes, and then use this custom RDP file with your Azure AD credentials to connect to the Azure VM.

Related topics

[Windows 10 Subscription Activation](#)

[Recommended settings for VDI desktops](#)

[Licensing the Windows Desktop for VDI Environments](#)

Deploy Windows 10 Enterprise licenses

6/18/2019 • 9 minutes to read • [Edit Online](#)

This topic describes how to deploy Windows 10 Enterprise E3 or E5 licenses with [Windows 10 Enterprise Subscription Activation](#) or [Windows 10 Enterprise E3 in CSP](#) and Azure Active Directory (Azure AD).

NOTE

- Windows 10 Enterprise Subscription Activation (EA or MPSA) requires Windows 10 Pro, version 1703 or later.
- Windows 10 Enterprise E3 in CSP requires Windows 10 Pro, version 1607 or later.
- Automatic, non-KMS activation requires Windows 10, version 1803 or later, on a device with a firmware-embedded activation key.

Firmware-embedded activation key

To determine if the computer has a firmware-embedded activation key, type the following command at an elevated Windows PowerShell prompt

```
(Get-WmiObject -query 'select * from SoftwareLicensingService').OA3xOriginalProductKey
```

If the device has a firmware-embedded activation key, it will be displayed in the output. If the output is blank, the device does not have a firmware embedded activation key. Most OEM-provided devices designed to run Windows 8 or later will have a firmware-embedded key.

Enabling Subscription Activation with an existing EA

If you are an EA customer with an existing Office 365 tenant, use the following steps to enable Windows 10 Subscription licenses on your existing tenant:

1. Work with your reseller to place an order for one \$0 SKU per user. There are two SKUs available, depending on their current Windows Enterprise SA license:
2. **AAA-51069** - Win10UsrOLSActv Alng MonthlySub Addon E3
3. **AAA-51068** - Win10UsrOLSActv Alng MonthlySub Addon E5
4. After placing an order, the OLS admin on the agreement will receive a service activation email, indicating their subscription licenses have been provisioned on the tenant.
5. The admin can now assign subscription licenses to users.

Use the following process if you need to update contact information and retrigger activation in order to resend the activation email:

1. Sign in to the [Microsoft Volume Licensing Service Center](#).
2. Click on **Subscriptions**.
3. Click on **Online Services Agreement List**.
4. Enter your agreement number, and then click **Search**.
5. Click the **Service Name**.
6. In the **Subscription Contact** section, click the name listed under **Last Name**.
7. Update the contact information, then click **Update Contact Details**. This will trigger a new email.

Also in this article:

- [Explore the upgrade experience](#): How to upgrade devices using the deployed licenses.
- [Troubleshoot the user experience](#): Examples of some license activation issues that can be encountered, and how to resolve them.

Active Directory synchronization with Azure AD

You probably have on-premises Active Directory Domain Services (AD DS) domains. Users will use their domain-based credentials to sign in to the AD DS domain. Before you start deploying Windows 10 Enterprise E3 or E5 licenses to users, you need to synchronize the identities in the on-premises AD DS domain with Azure AD.

You might ask why you need to synchronize these identities. The answer is so that users will have a *single identity* that they can use to access their on-premises apps and cloud services that use Azure AD (such as Windows 10 Enterprise E3 or E5). This means that users can use their existing credentials to sign in to Azure AD and access the cloud services that you provide and manage for them.

Figure 1 illustrates the integration between the on-premises AD DS domain with Azure AD. [Microsoft Azure Active Directory Connect](#) (Azure AD Connect) is responsible for synchronization of identities between the on-premises AD DS domain and Azure AD. Azure AD Connect is a service that you can install on-premises or in a virtual machine in Azure.

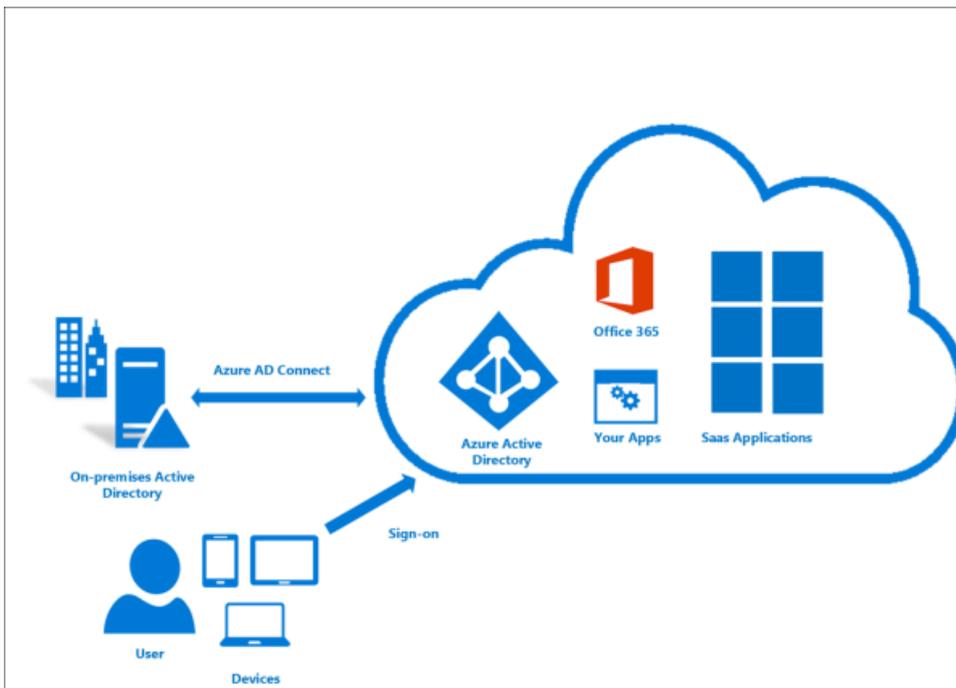


Figure 1. On-premises AD DS integrated with Azure AD

For more information about integrating on-premises AD DS domains with Azure AD, see the following resources:

- [Integrating your on-premises identities with Azure Active Directory](#)
- [Azure AD + Domain Join + Windows 10](#)

NOTE

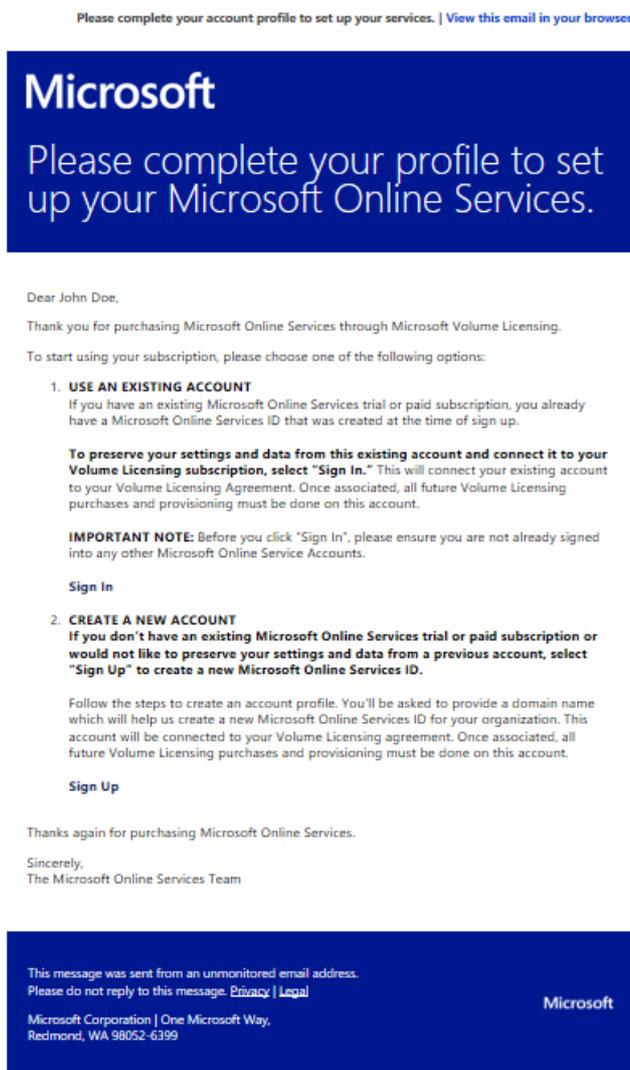
If you are implementing Azure AD, and you already have an on-premises domain, you don't need to integrate with Azure AD, since your main authentication method is your internal AD. If you want to manage all your infrastructure in the cloud, you can safely configure your domain controller remotely to integrate your computers with Azure AD, but you won't be able to apply fine controls using GPO. Azure AD is best suited for the global administration of devices when you don't have any on-premises servers.

Preparing for deployment: reviewing requirements

Devices must be running Windows 10 Pro, version 1703, and be Azure Active Directory joined, or hybrid domain joined with Azure AD Connect. Customers who are federated with Azure Active Directory are also eligible. For more information, see [Review requirements on devices](#), later in this topic.

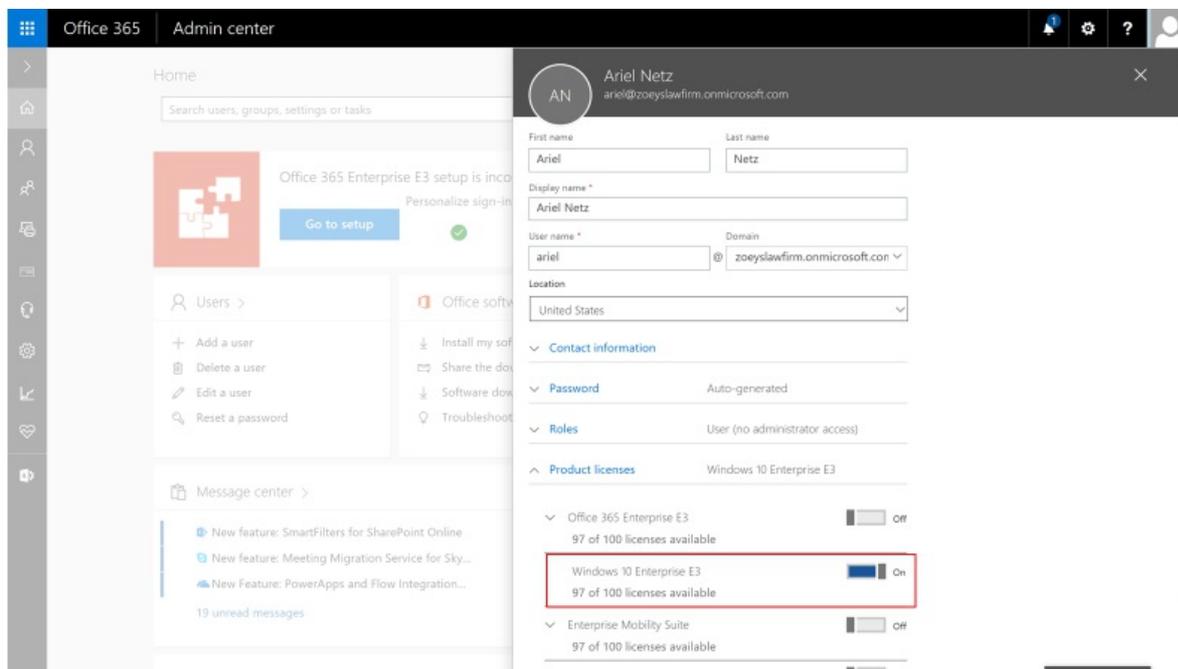
Assigning licenses to users

Upon acquisition of Windows 10 subscription has been completed (Windows 10 Business, E3 or E5), customers will receive an email that will provide guidance on how to use Windows as an online service:



The following methods are available to assign licenses:

1. When you have the required Azure AD subscription, [group-based licensing](#) is the preferred method to assign Enterprise E3 or E5 licenses to users.
2. You can sign in to [portal.office.com](#) and manually assign licenses:



3. You can assign licenses by uploading a spreadsheet.
4. A per-user [PowerShell scripted method](#) of assigning licenses is available.
5. Organizations can use synchronized [AD groups](#) to automatically assign licenses.

Explore the upgrade experience

Now that your subscription has been established and Windows 10 Enterprise E3 or E5 licenses have been assigned to users, the users are ready to upgrade their devices running Windows 10 Pro, (version 1703 or later) to Windows 10 Enterprise. What will the users experience? How will they upgrade their devices?

Step 1: Join Windows 10 Pro devices to Azure AD

Users can join a Windows 10 Pro device to Azure AD the first time they start the device (during setup), or they can join a device that they already use running Windows 10 Pro, version 1703.

To join a device to Azure AD the first time the device is started

1. During the initial setup, on the **Who owns this PC?** page, select **My organization**, and then click **Next**, as illustrated in **Figure 2**.

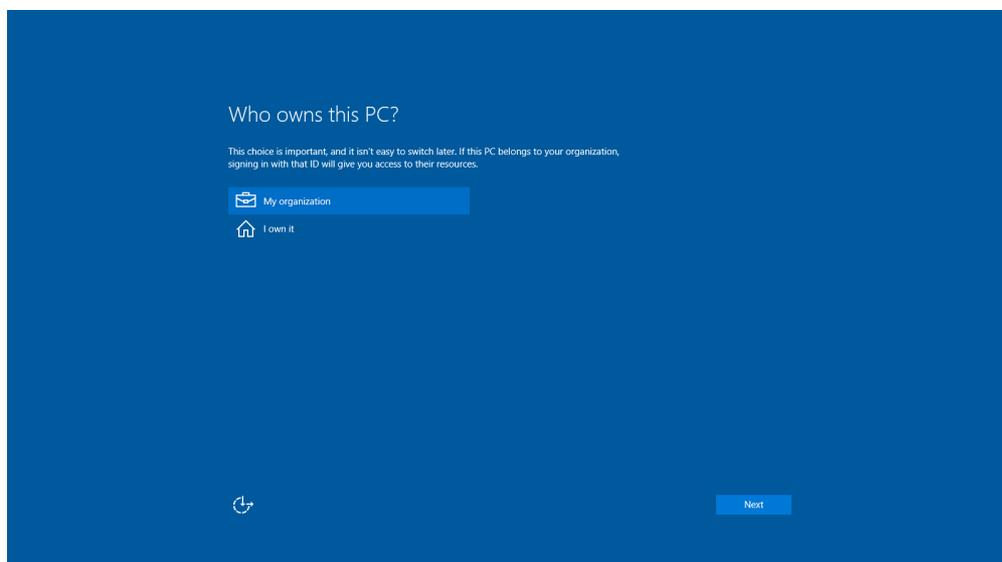


Figure 2. The “Who owns this PC?” page in initial Windows 10 setup

2. On the **Choose how you'll connect** page, select **Join Azure AD**, and then click **Next**, as illustrated in **Figure 3**.

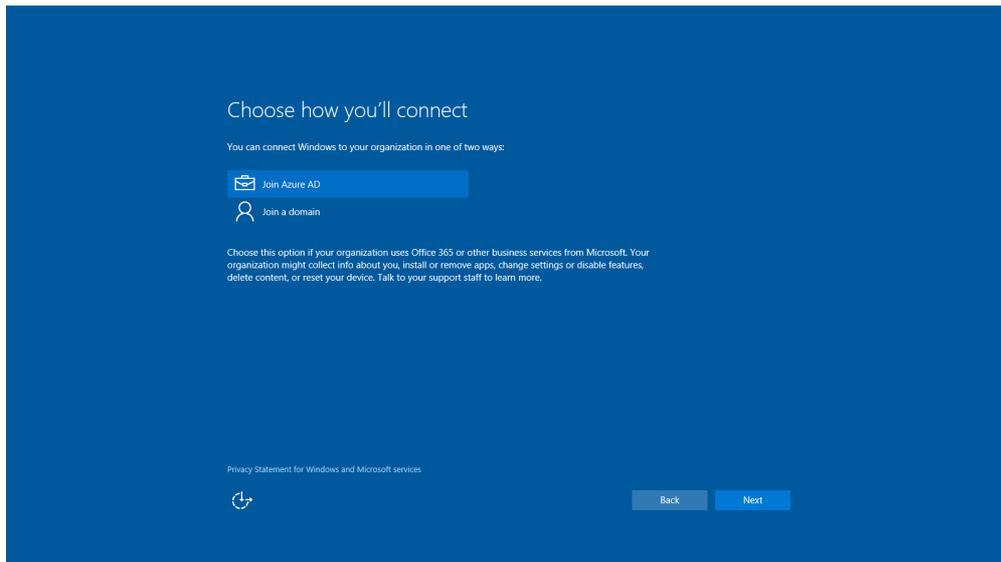


Figure 3. The "Choose how you'll connect" page in initial Windows 10 setup

3. On the **Let's get you signed in** page, enter the Azure AD credentials, and then click **Sign in**, as illustrated in **Figure 4**.

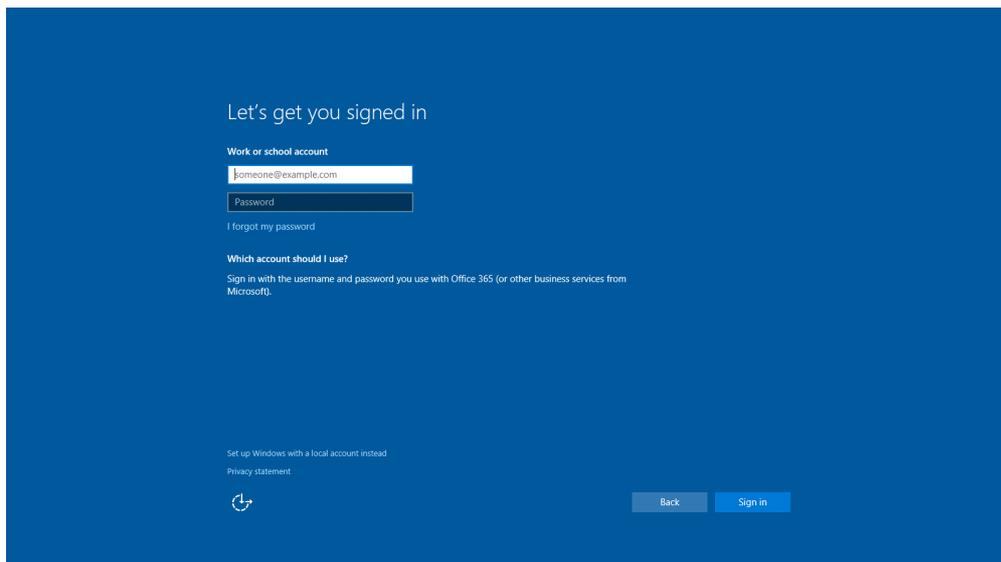


Figure 4. The "Let's get you signed in" page in initial Windows 10 setup

Now the device is Azure AD joined to the company's subscription.

To join a device to Azure AD when the device already has Windows 10 Pro, version 1703 installed and set up

IMPORTANT

Make sure that the user you're signing in with is **not** a BUILTIN/Administrator. That user cannot use the **+ Connect** button to join a work or school account.

1. Go to **Settings > Accounts > Access work or school**, as illustrated in **Figure 5**.

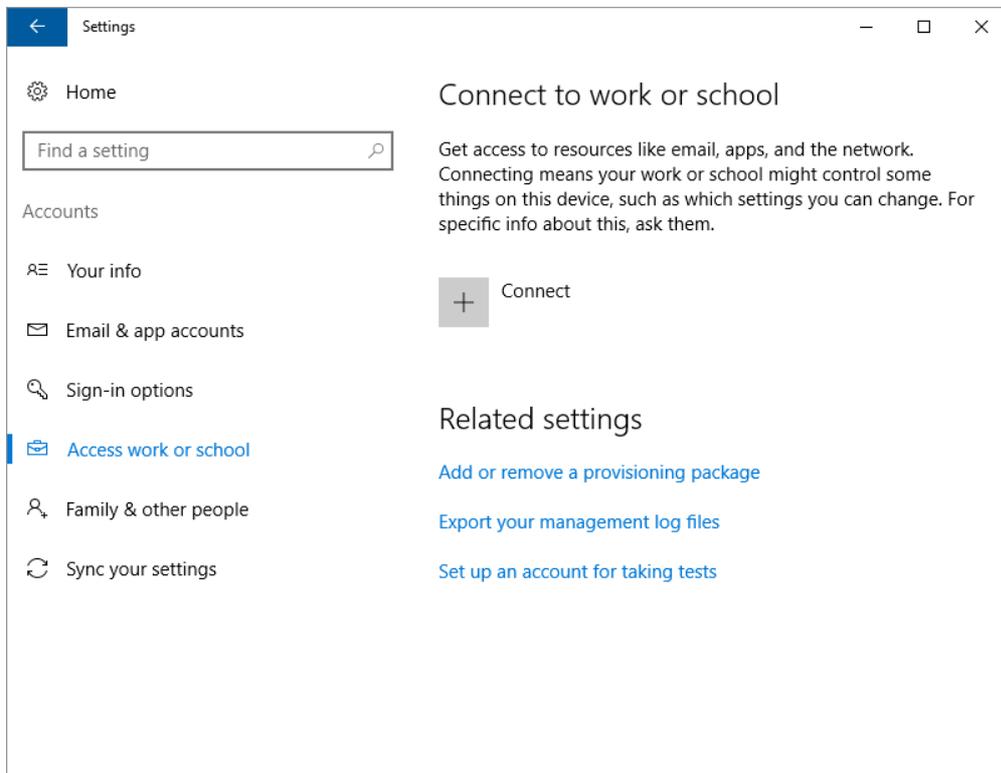


Figure 5. Connect to work or school configuration in Settings

2. In **Set up a work or school account**, click **Join this device to Azure Active Directory**, as illustrated in **Figure 6**.

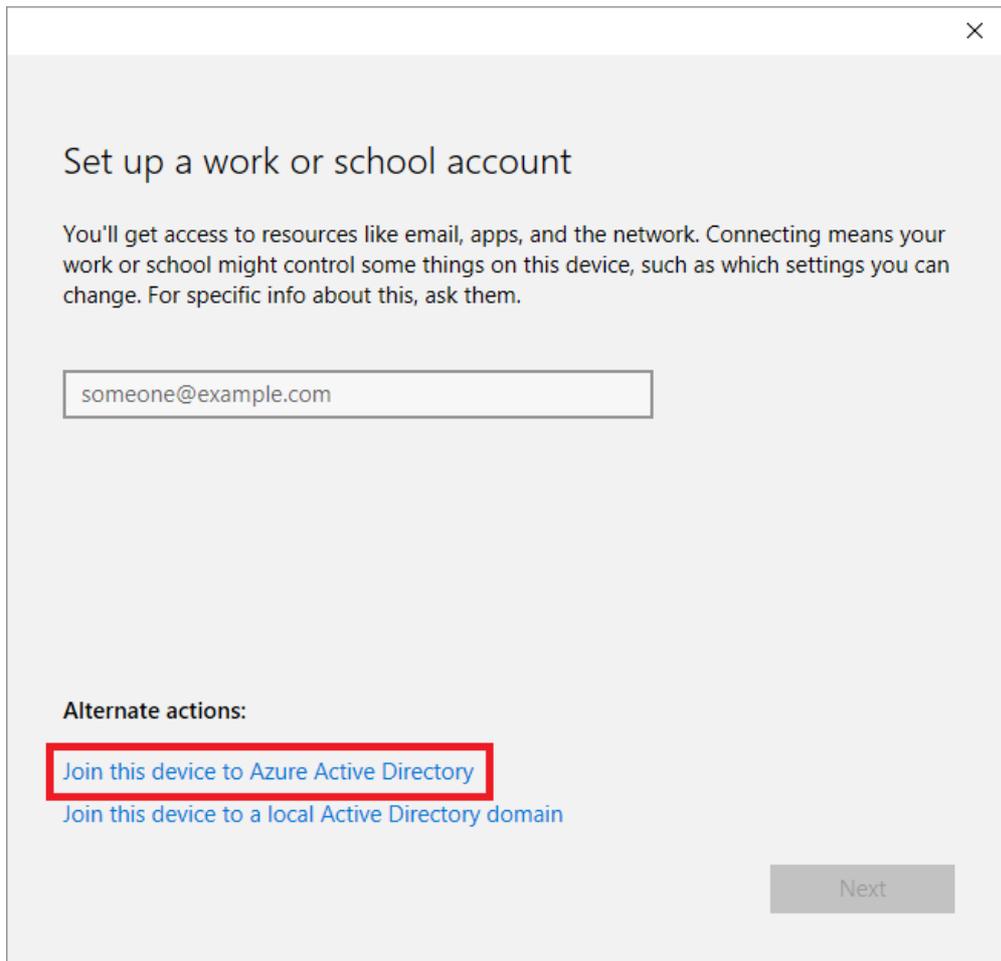


Figure 6. Set up a work or school account

3. On the **Let's get you signed in** page, enter the Azure AD credentials, and then click **Sign in**, as illustrated in **Figure 7**.

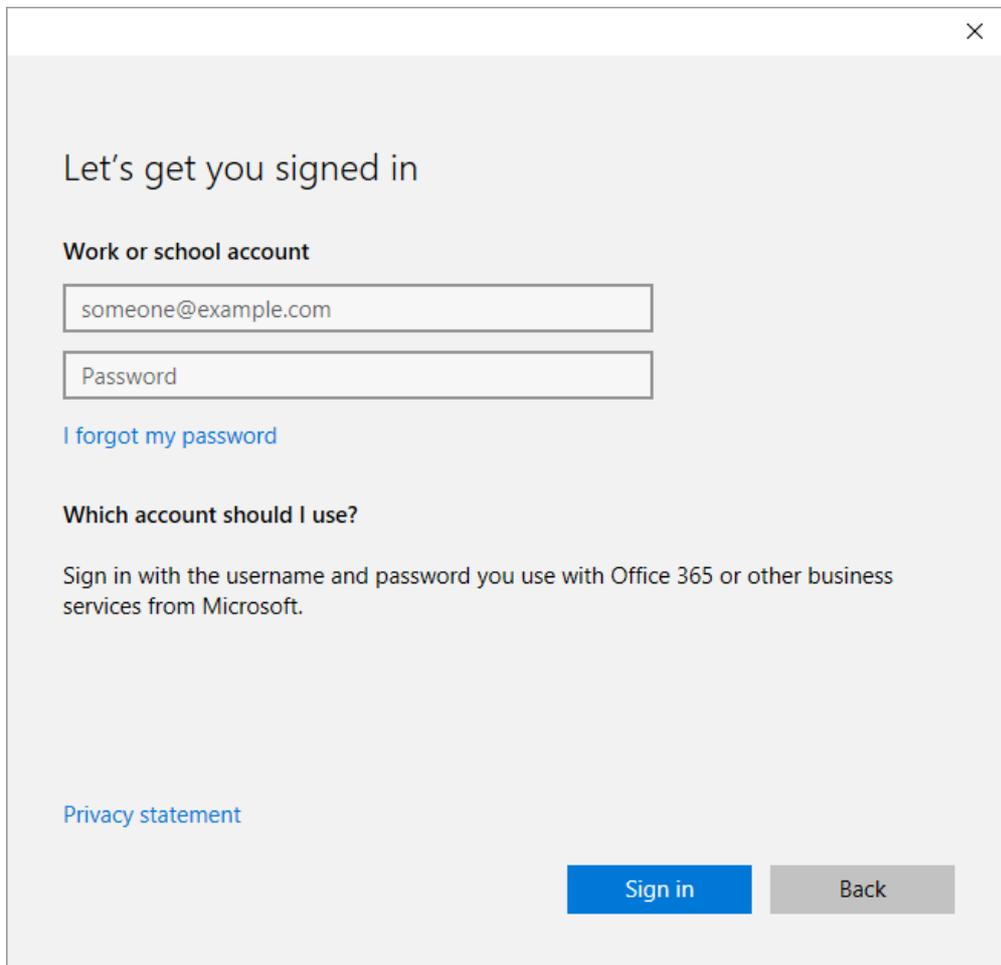


Figure 7. The “Let’s get you signed in” dialog box

Now the device is Azure AD joined to the company’s subscription.

Step 2: Pro edition activation

IMPORTANT

If your device is running Windows 10, version 1803 or later, this step is not needed. From Windows 10, version 1803, the device will automatically activate Windows 10 Enterprise using the firmware-embedded activation key. If the device is running Windows 10, version 1703 or 1709, then Windows 10 Pro must be successfully activated in **Settings > Update & Security > Activation**, as illustrated in **Figure 7a**.

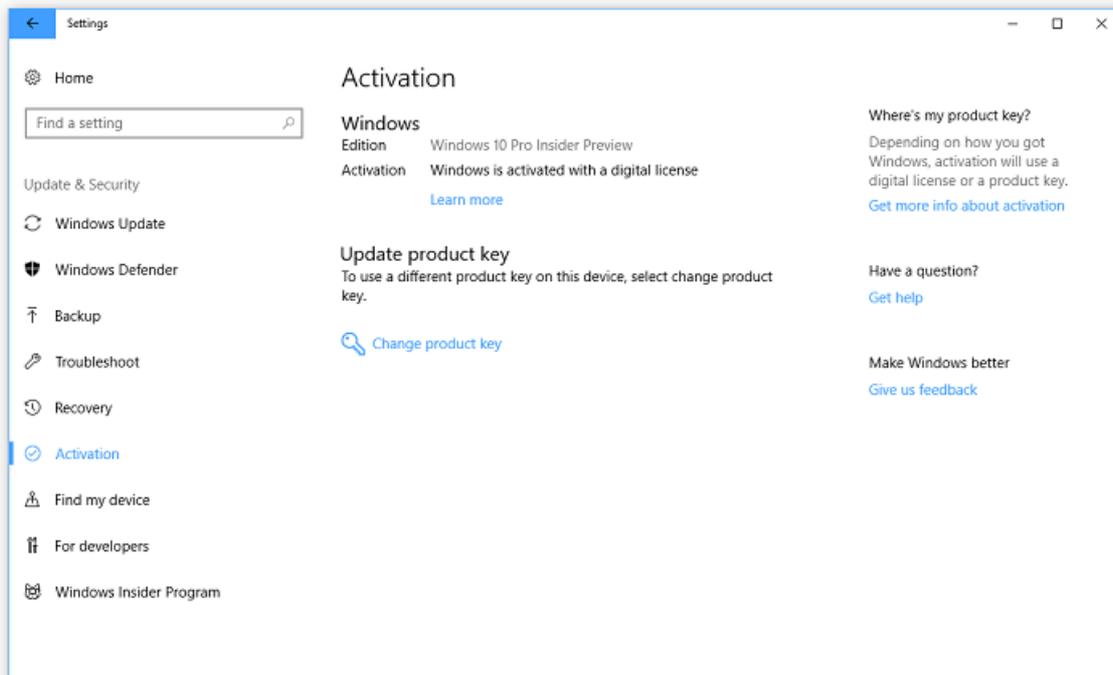


Figure 7a -

Windows 10 Pro activation in Settings

Windows 10 Pro activation is required before Enterprise E3 or E5 can be enabled (Windows 10, versions 1703 and 1709 only).

Step 3: Sign in using Azure AD account

Once the device is joined to your Azure AD subscription, the user will sign in by using his or her Azure AD account, as illustrated in **Figure 8**. The Windows 10 Enterprise E3 or E5 license associated with the user will enable Windows 10 Enterprise edition capabilities on the device.

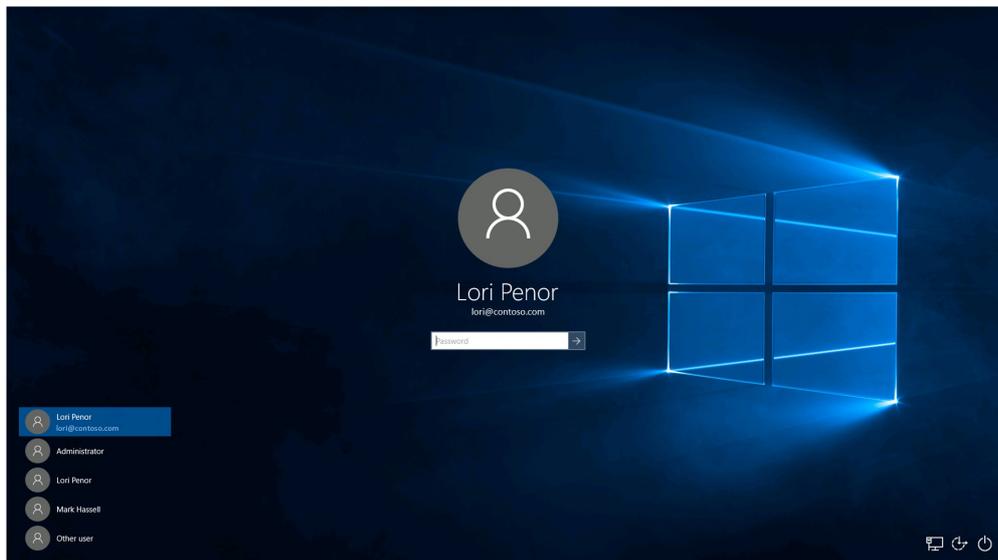


Figure 8. Sign in by using Azure AD account

Step 4: Verify that Enterprise edition is enabled

You can verify the Windows 10 Enterprise E3 or E5 subscription in **Settings > Update & Security > Activation**, as illustrated in **Figure 9**.

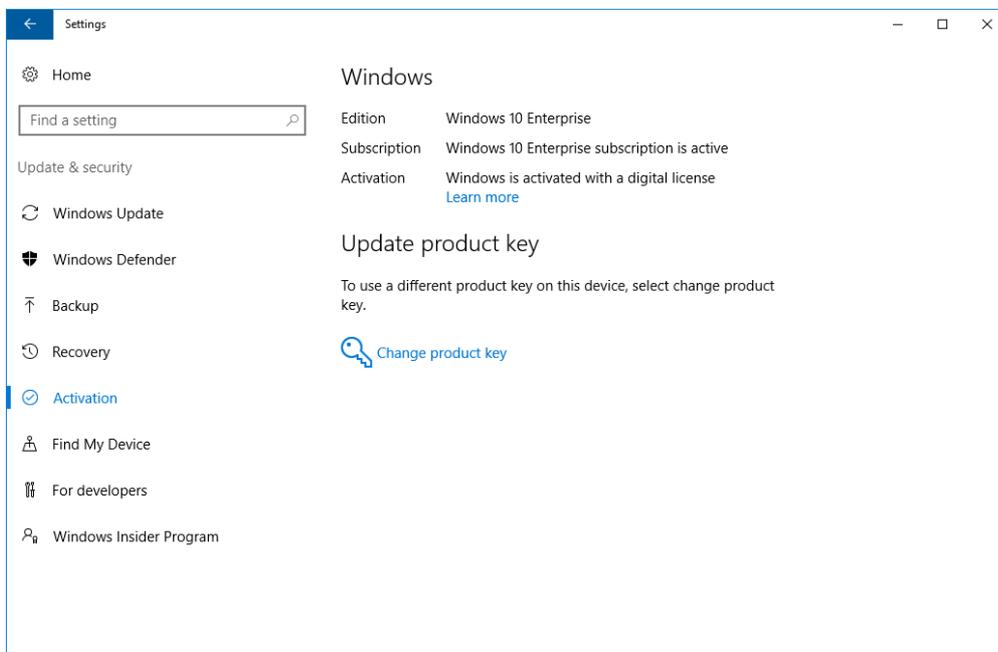


Figure 9 - Windows 10 Enterprise subscription in Settings

If there are any problems with the Windows 10 Enterprise E3 or E5 license or the activation of the license, the **Activation** panel will display the appropriate error message or status. You can use this information to help you diagnose the licensing and activation process.

NOTE

If you use `slmgr /dli` or `/dlv` commands to retrieve the activation information for the Windows 10 E3 or E5 license, the license information displayed will be the following: Name: Windows(R), Professional edition Description: Windows(R) Operating System, RETAIL channel Partial Product Key: 3V66T

Virtual Desktop Access (VDA)

Subscriptions to Windows 10 Enterprise are also available for virtualized clients. Windows 10 Enterprise E3 and E5 are available for Virtual Desktop Access (VDA) in Windows Azure or in another [qualified multitenant hoster](#).

Virtual machines (VMs) must be configured to enable Windows 10 Enterprise subscriptions for VDA. Active Directory-joined and Azure Active Directory-joined clients are supported. See [Enable VDA for Enterprise Subscription Activation](#).

Troubleshoot the user experience

In some instances, users may experience problems with the Windows 10 Enterprise E3 or E5 subscription. The most common problems that users may experience are as follows:

- The existing Windows 10 Pro, version 1703 or 1709 operating system is not activated. This problem does not apply to Windows 10, version 1803 or later.
- The Windows 10 Enterprise E3 or E5 subscription has lapsed or has been removed.

Use the following figures to help you troubleshoot when users experience these common problems:

- [Figure 9](#) (above) illustrates a device in a healthy state, where Windows 10 Pro is activated and the Windows 10 Enterprise subscription is active.
- [Figure 10](#) (below) illustrates a device on which Windows 10 Pro is not activated, but the Windows 10 Enterprise subscription is active.

- **Figure 11** (below) illustrates a device on which Windows 10 Pro is activated, but the Windows 10 Enterprise subscription is lapsed or removed.
- **Figure 12** (below) illustrates a device on which Windows 10 Pro license is not activated and the Windows 10 Enterprise subscription is lapsed or removed.

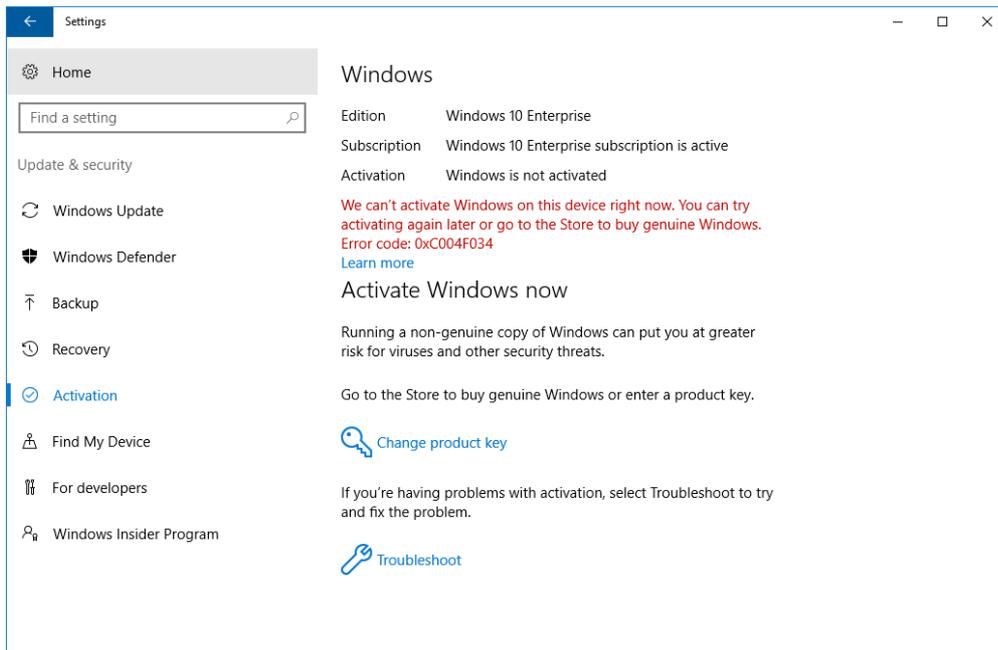


Figure 10 - Windows 10

Pro, version 1703 edition not activated in Settings

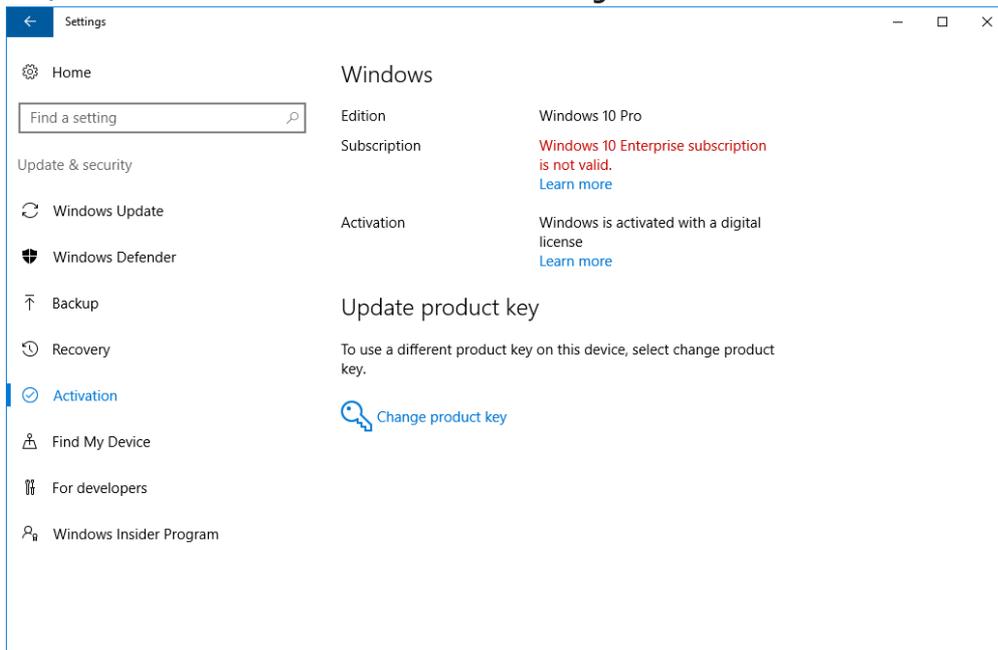


Figure 11 - Windows 10

Enterprise subscription lapsed or removed in Settings

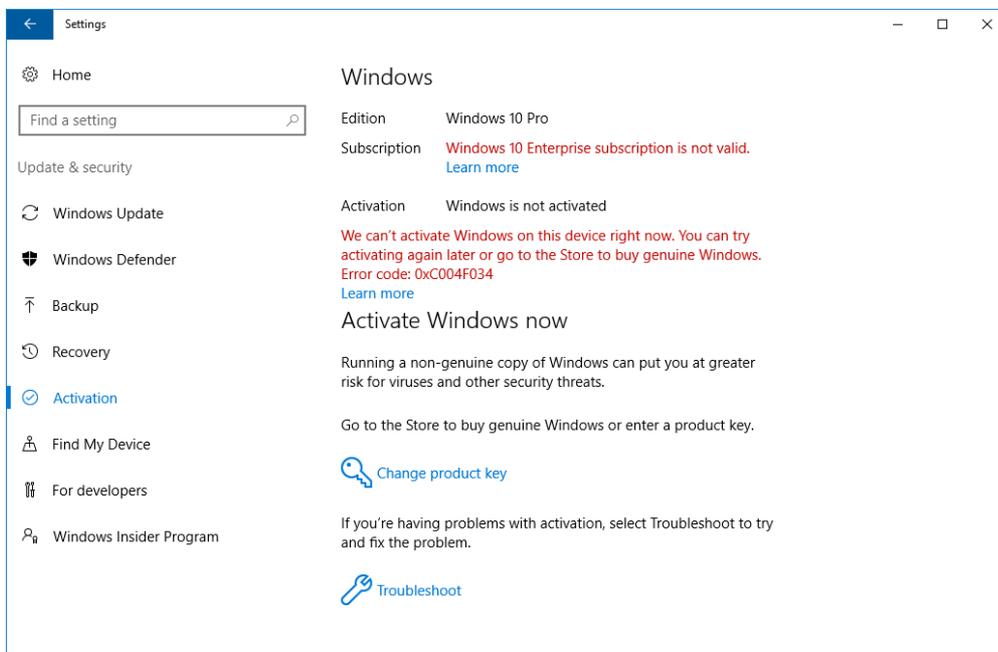


Figure 12 - Windows 10

Pro, version 1703 edition not activated and Windows 10 Enterprise subscription lapsed or removed in Settings

Review requirements on devices

Devices must be running Windows 10 Pro, version 1703, and be Azure Active Directory joined, or hybrid domain joined with Azure AD Connect. Customers who are federated with Azure Active Directory are also eligible. You can use the following procedures to review whether a particular device meets requirements.

To determine if a device is Azure Active Directory joined:

1. Open a command prompt and type **dsregcmd /status**.
2. Review the output under Device State. If the **AzureAdJoined** status is YES, the device is Azure Active Directory joined.

To determine the version of Windows 10:

- At a command prompt, type: **winver**

A popup window will display the Windows 10 version number and detailed OS build information.

If a device is running a previous version of Windows 10 Pro (for example, version 1511), it will not be upgraded to Windows 10 Enterprise when a user signs in, even if the user has been assigned a subscription in the CSP portal.

Resolve Windows 10 upgrade errors : Technical information for IT Pros

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

This article contains technical instructions for IT administrators. If you are not an IT administrator, try some of the [quick fixes](#) described in this article then contact [Microsoft Support](#) starting with the Virtual Agent. To talk to a person about your issue, click **Get started** to interact with the Virtual Agent, then enter "Talk to a person" two times. The Virtual Agent can also help you to resolve many Windows upgrade issues. Also see: [Get help with Windows 10 upgrade and installation errors](#) and [Submit Windows 10 upgrade errors using Feedback Hub](#).

This article contains a brief introduction to Windows 10 installation processes, and provides resolution procedures that IT administrators can use to resolve issues with Windows 10 upgrade.

The article was originally one page, but has been divided into sub-topics of different technical levels. Basic level provides common procedures that can resolve several types of upgrade errors. Advanced level requires some experience with detailed troubleshooting methods.

The following four levels are assigned:

Level 100: Basic

Level 200: Moderate

Level 300: Moderate advanced

Level 400: Advanced

In this guide

See the following topics in this article:

- [Quick fixes](#): \Level 100\ Steps you can take to eliminate many Windows upgrade errors.
- [SetupDiag](#): \Level 300\ SetupDiag is a new tool to help you isolate the root cause of an upgrade failure.
- [Troubleshooting upgrade errors](#): \Level 300\ General advice and techniques for troubleshooting Windows 10 upgrade errors, and an explanation of phases used during the upgrade process.
- [Windows Error Reporting](#): \Level 300\ How to use Event Viewer to review details about a Windows 10 upgrade.
- [Upgrade error codes](#): \Level 400\ The components of an error code are explained.
 - [Result codes](#): Information about result codes.
 - [Extend codes](#): Information about extend codes.
- [Log files](#): \Level 400\ A list and description of log files useful for troubleshooting.
 - [Log entry structure](#): The format of a log entry is described.
 - [Analyze log files](#): General procedures for log file analysis, and an example.
- [Resolution procedures](#): \Level 200\ Causes and mitigation procedures associated with specific error codes.
 - [0xC1900101](#): Information about the 0xC1900101 result code.
 - [0x800xxxxx](#): Information about result codes that start with 0x800.

- [Other result codes](#): Additional causes and mitigation procedures are provided for some result codes.
- [Other error codes](#): Additional causes and mitigation procedures are provided for some error codes.
- [Submit Windows 10 upgrade errors](#): \Level 100\ Submit upgrade errors to Microsoft for analysis.

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Quick fixes

6/14/2019 • 9 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 100 level topic (basic).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

The following list of fixes can resolve many Windows upgrade problems. You should try these steps before contacting Microsoft support, or attempting a more advanced analysis of a Windows upgrade failure. Also review information at [Windows 10 help](#).

The Microsoft Virtual Agent provided by [Microsoft Support](#) can help you to analyze and correct some Windows upgrade errors. **To talk to a person about your issue**, start the Virtual Agent (click **Get started**) and enter "Talk to a person" two times.

You might also wish to try a new tool available from Microsoft that helps to diagnose many Windows upgrade errors. For more information and to download this tool, see [SetupDiag](#). The topic is more advanced (300 level) because several advanced options are available for using the tool. However, you can now just download and then double-click the tool to run it. By default when you click Save, the tool is saved in your **Downloads** folder. Double-click the tool in the folder and wait until it finishes running (it might take a few minutes), then double-click the **SetupDiagResults.log** file and open it using Notepad to see the results of the analysis.

List of fixes

1. Remove nonessential external hardware, such as docks and USB devices. [More information](#).
2. Check the system drive for errors and attempt repairs. [More information](#).
3. Run the Windows Update troubleshooter. [More information](#).
4. Attempt to restore and repair system files. [More information](#).
5. Update Windows so that all available recommended updates are installed, and ensure the computer is rebooted if this is necessary to complete installation of an update. [More information](#).
6. Temporarily uninstall non-Microsoft antivirus software. [More information](#).
7. Uninstall all nonessential software. [More information](#).
8. Update firmware and drivers. [More information](#)
9. Ensure that "Download and install updates (recommended)" is accepted at the start of the upgrade process. [More information](#).
10. Verify at least 16 GB of free space is available to upgrade a 32-bit OS, or 20 GB for a 64-bit OS. [More information](#).

Step by step instructions

Remove external hardware

If the computer is portable and it is currently in a docking station, [undock the computer](#).

Unplug nonessential external hardware devices from the computer, such as:

- Headphones
- Joysticks
- Printers
- Plotters
- Projectors
- Scanners
- Speakers
- USB flash drives
- Portable hard drives
- Portable CD/DVD/Blu-ray drives
- Microphones
- Media card readers
- Cameras/Webcams
- Smart phones
- Secondary monitors, keyboards, mice

For more information about disconnecting external devices, see [Safely remove hardware in Windows 10](#)

Repair the system drive

The system drive is the drive that contains the [system partition](#). This is usually the **C:** drive.

To check and repair errors on the system drive:

1. Click **Start**.
2. Type **command**.
3. Right-click **Command Prompt** and then left-click **Run as administrator**.
4. If you are prompted by UAC, click **Yes**.
5. Type **chkdsk /F** and press ENTER.
6. When you are prompted to schedule a check the next time the system restarts, type **Y**.
7. See the following example

```
C:\WINDOWS\system32>chkdsk /F
The type of the file system is NTFS.
Cannot lock current drive.

Chkdsk cannot run because the volume is in use by another
process. Would you like to schedule this volume to be
checked the next time the system restarts? (Y/N) Y

This volume will be checked the next time the system restarts.
```

8. Restart the computer. The computer will pause before loading Windows and perform a repair of your hard drive.

Windows Update Troubleshooter

The Windows Update troubleshooter tool will automatically analyze and fix problems with Windows Update, such as a corrupted download. It will also tell you if there is a pending reboot that is preventing Windows from updating.

For Windows 7 and 8.1, the tool is [here](#).

For Windows 10, the tool is [here](#).

To run the tool, click the appropriate link above. Your web browser will prompt you to save or open the file. Select **open** and the tool will automatically start. The tool will walk you through analyzing and fixing some common problems.

You can also download the Windows Update Troubleshooter by starting the Microsoft [Virtual Agent](#), typing **update Windows**, selecting the version of Windows you are running, and then answering **Yes** when asked "Do you need help troubleshooting Windows Update?"

If any errors are displayed in the Windows Update Troubleshooter, use the Microsoft [Virtual Agent](#) to ask about these errors. The Virtual Agent will perform a search and provide a list of helpful links.

Repair system files

This fix is also described in detail at answers.microsoft.com.

To check and repair system files:

1. Click **Start**.
2. Type **command**.
3. Right-click **Command Prompt** and then left-click **Run as administrator**.
4. If you are prompted by UAC, click **Yes**.
5. Type **sfc /scannow** and press ENTER. See the following example:

```
C:\>sfc /scannow

Beginning system scan. This process will take some time.

Beginning verification phase of system scan.
Verification 100% complete.

Windows Resource Protection did not find any integrity violations.
```

6. If you are running Windows 8.1 or later, type **DISM.exe /Online /Cleanup-image /Restorehealth** and press ENTER (the DISM command options are not available for Windows 7). See the following example:

```
C:\>DISM.exe /Online /Cleanup-image /Restorehealth

Deployment Image Servicing and Management tool
Version: 10.0.16299.15

Image Version: 10.0.16299.309

[=====100.0%=====] The restore operation completed
successfully.
The operation completed successfully.
```

It may take several minutes for the command operations to be completed. For more information, see [Repair a Windows Image](#).

Update Windows

You should ensure that all important updates are installed before attempting to upgrade. This includes updates to hardware drivers on your computer.

The Microsoft [Virtual Agent](#) can walk you through the process of making sure that Windows is updated.

Start the [Virtual Agent](#) and then type "update windows."

Answer questions that the agent asks, and follow instructions to ensure that Windows is up to date. You can also run the [Windows Update Troubleshooter](#) described above.

Click **Start**, click power options, and then restart the computer.

Uninstall non-Microsoft antivirus software

Use Windows Defender for protection during the upgrade.

Verify compatibility information, and if desired re-install antivirus applications after the upgrade. If you plan to re-install the application after upgrading, be sure that you have the installation media and all required activation information before removing the program.

To remove the application, go to **Control Panel\Programs\Programs and Features** and click the antivirus application, then click Uninstall. Choose **Yes** when you are asked to confirm program removal.

For more information, see [Windows 7 - How to properly uninstall programs](#) or [Repair or remove programs in Windows 10](#).

Uninstall non-essential software

Outdated applications can cause problems with a Windows upgrade. Removing old or non-essential applications from the computer can therefore help.

If you plan to reinstall the application later, be sure that you have the installation media and all required activation information before removing it.

To remove programs, use the same steps as are provided [above](#) for uninstalling non-Microsoft antivirus software, but instead of removing the antivirus application repeat the steps for all your non-essential, unused, or out-of-date software.

Update firmware and drivers

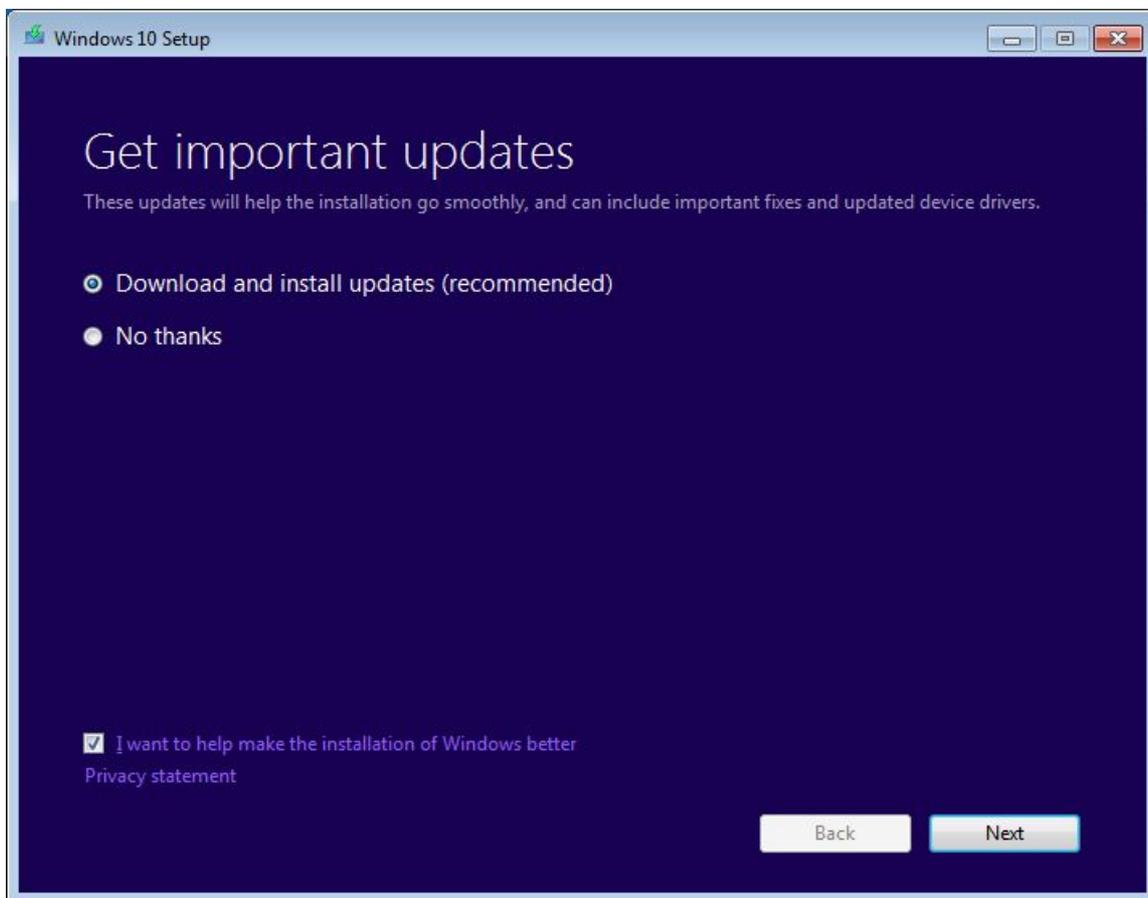
Updating firmware (such as the BIOS) and installing hardware drivers is a somewhat advanced task. Do not attempt to update BIOS if you aren't familiar with BIOS settings or are not sure how to restore the previous BIOS version if there are problems. Most BIOS updates are provided as a "flash" update. Your manufacturer might provide a tool to perform the update, or you might be required to enter the BIOS and update it manually. Be sure to save your working BIOS settings, since some updates can reset your configuration and make the computer fail to boot if (for example) a RAID configuration is changed.

Most BIOS and other hardware updates can be obtained from a website maintained by your computer manufacturer. For example, Microsoft Surface device drivers can be obtained at: [Download the latest firmware and drivers for Surface devices](#).

To obtain the proper firmware drivers, search for the most updated driver version provided by your computer manufacturer. Install these updates and reboot the computer after installation. Request assistance from the manufacturer if you have any questions.

Ensure that "Download and install updates" is selected

When you begin a Windows Update, the setup process will ask you to **Get important updates**. Answer **Yes** if the computer you are updating is connected to the Internet. See the following example:



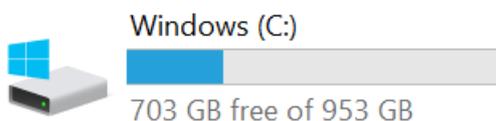
Verify disk space

You can see a list of requirements for Windows 10 at [Windows 10 Specifications & System Requirements](#). One of the requirements is that enough hard drive space be available for the installation to take place. At least 16 GB of free space must be available on the system drive to upgrade a 32-bit OS, or 20 GB for a 64-bit OS.

To view how much hard drive space is available on your computer, open [File Explorer](#). In Windows 7, this was called Windows Explorer.

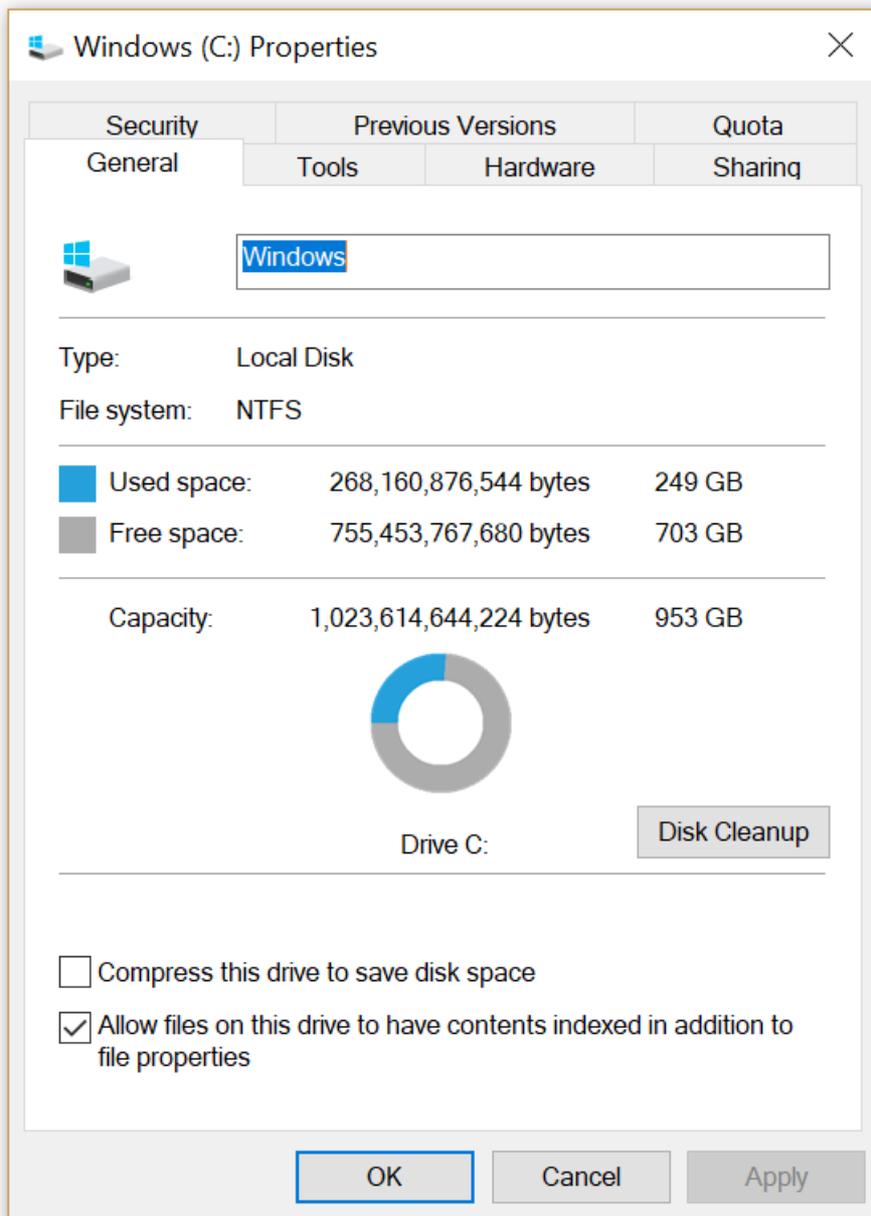
In File Explorer, click on **Computer** or **This PC** on the left, then look under **Hard Disk Drives** or under **Devices and drives**. If there are multiple drives listed, the system drive is the drive that includes a Microsoft Windows logo above the drive icon.

The amount of space available on the system drive will be displayed under the drive. See the following example:



In the previous example, there is 703 GB of available free space on the system drive (C:).

To free up additional space on the system drive, begin by running Disk Cleanup. You can access Disk Cleanup by right-clicking the hard drive icon and then clicking Properties. See the following example:



For instructions to run Disk Cleanup and other suggestions to free up hard drive space, see [Tips to free up drive space on your PC](#).

When you run Disk Cleanup and enable the option to Clean up system files, you can remove previous Windows installations which can free a large amount of space. You should only do this if you do not plan to restore the old OS version.

Open an elevated command prompt

It is no longer necessary to open an elevated command prompt to run the [SetupDiag](#) tool. However, this is still the optimal way to run the tool.

To launch an elevated command prompt, press the Windows key on your keyboard, type **cmd**, press Ctrl+Shift+Enter, and then Alt+C to confirm the elevation prompt. Screenshots and other steps to open an administrator (aka elevated) command prompt are [here](#).

Note: When you open an elevated command prompt, you will usually start in the **C:\WINDOWS\system32** directory. To run a program that you recently downloaded, you must change to the directory where the program is located. Alternatively, you can move or copy the program to a location on the computer that is automatically searched. These directories are listed in the [PATH variable](#).

If this is too complicated for you, then use File Explorer to create a new folder under C: with a short name such as "new" then copy or move the programs you want to run (like SetupDiag) to this folder using File Explorer. When you open an elevated command prompt, change to this directory by typing "cd c:\new" and now you can run the programs in that folder.

If you downloaded the SetupDiag.exe program to your computer, then copied it to the folder C:\new, and you opened an elevated command prompt then typed cd c:\new to change to this directory, you can just type setupdiag and press ENTER to run the program. This program will analyze the files on your computer to see why a Windows Upgrade failed and if the reason was a common one, it will report this reason. It will not fix the problem for you but knowing why the upgrade failed enables you to take steps to fix the problem.

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

SetupDiag

6/26/2019 • 21 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 300 level topic (moderate advanced).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

[↓ Download SetupDiag](#)

About SetupDiag

Current version of SetupDiag: 1.5.0.0

SetupDiag is a standalone diagnostic tool that can be used to obtain details about why a Windows 10 upgrade was unsuccessful.

SetupDiag works by examining Windows Setup log files. It attempts to parse these log files to determine the root cause of a failure to update or upgrade the computer to Windows 10. SetupDiag can be run on the computer that failed to update, or you can export logs from the computer to another location and run SetupDiag in offline mode.

To quickly use SetupDiag on your current computer:

1. Verify that your system meets the [requirements](#) described below. If needed, install the [.NET framework 4.6](#).
2. [Download SetupDiag](#).
3. If your web browser asks what to do with the file, choose **Save**. By default, the file will be saved to your **Downloads** folder. You can also save it to a different location if desired by using **Save As**.
4. When SetupDiag has finished downloading, open the folder where you downloaded the file. As mentioned above, by default this is your **Downloads** folder which is displayed in File Explorer under **Quick access** in the left navigation pane.
5. Double-click the **SetupDiag** file to run it. Click **Yes** if you are asked to approve running the program.
 - Double-clicking the file to run it will automatically close the command window when SetupDiag has completed its analysis. If you wish to keep this window open instead, and review the messages that you see, run the program by typing **SetupDiag** at the command prompt instead of double-clicking it. You will need to change directories to the location of SetupDiag to run it this way.
6. A command window will open while SetupDiag diagnoses your computer. Wait for this to finish.
7. When SetupDiag finishes, two files will be created in the same folder where you double-clicked SetupDiag. One is a configuration file, the other is a log file.
8. Use Notepad to open the log file: **SetupDiagResults.log**.
9. Review the information that is displayed. If a rule was matched this can tell you why the computer failed to upgrade, and potentially how to fix the problem. See the [Text log sample](#) below.

For instructions on how to run the tool in offline mode and with more advanced options, see the [Parameters](#) and [Examples](#) sections below.

The [Release notes](#) section at the bottom of this topic has information about recent updates to this tool.

Requirements

1. The destination OS must be Windows 10.
2. [.NET Framework 4.6](#) must be installed. If you are not sure what version of .NET is currently installed, see [How to: Determine Which .NET Framework Versions Are Installed](#). You can also use the following command-line query to display the installed v4 versions:

```
reg query "HKLM\SOFTWARE\Microsoft\Net Framework Setup\NDP\v4" /s
```

Parameters

PARAMETER	DESCRIPTION
<code>/?</code>	<ul style="list-style-type: none">• Displays interactive help
<code>/Output:<path to results file></code>	<ul style="list-style-type: none">• This optional parameter enables you to specify the output file for results. This is where you will find what SetupDiag was able to determine. Only text format output is supported. UNC paths will work, provided the context under which SetupDiag runs has access to the UNC path. If the path has a space in it, you must enclose the entire path in double quotes (see the example section below).• Default: If not specified, SetupDiag will create the file SetupDiagResults.log in the same directory where SetupDiag.exe is run.
<code>/LogsPath:<Path to logs></code>	<ul style="list-style-type: none">• This optional parameter tells SetupDiag.exe where to find the log files for an offline analysis. These log files can be in a flat folder format, or containing multiple subdirectories. SetupDiag will recursively search all child directories.
<code>/ZipLogs:<True False></code>	<ul style="list-style-type: none">• This optional parameter tells SetupDiag.exe to create a zip file containing the results and all the log files it parsed. The zip file is created in the same directory where SetupDiag.exe is run.• Default: If not specified, a value of 'true' is used.
<code>/Format:<xml json></code>	<ul style="list-style-type: none">• This optional parameter can be used to output log files in xml or JSON format. If this parameter is not specified, text format is used by default.
<code>/Scenario:[Recovery]</code>	<ul style="list-style-type: none">• This optional parameter instructs SetupDiag.exe to look for and process reset and recovery logs and ignore setup/upgrade logs.

PARAMETER	DESCRIPTION
/Verbose	<ul style="list-style-type: none"> This optional parameter will output much more data to a log file. By default, SetupDiag will only produce a log file entry for serious errors. Using /Verbose will cause SetupDiag to always produce an additional log file with debugging details. These details can be useful when reporting a problem with SetupDiag.
/NoTel	<ul style="list-style-type: none"> This optional parameter tells SetupDiag.exe not to send diagnostic telemetry to Microsoft.
/AddReg	<ul style="list-style-type: none"> This optional parameter instructs SetupDiag.exe to add failure information to the registry in offline mode. By default, SetupDiag will add failure information to the registry in online mode only. Registry data is added to the following location on the system where SetupDiag is run: HKLM\SYSTEM\Setup\MoSetup\Volatile\SetupDiag.

Note: The **/Mode** parameter is deprecated in version 1.4.0.0 of SetupDiag.

- In previous versions, this command was used with the LogsPath parameter to specify that SetupDiag should run in an offline manner to analyze a set of log files that were captured from a different computer. In version 1.4.0.0 when you specify /LogsPath then SetupDiag will automatically run in offline mode, therefore the /Mode parameter is not needed.

Examples:

In the following example, SetupDiag is run with default parameters (online mode, results file is SetupDiagResults.log in the same folder where SetupDiag is run).

```
SetupDiag.exe
```

In the following example, SetupDiag is run in online mode (this is the default). It will know where to look for logs on the current (failing) system, so there is no need to gather logs ahead of time. A custom location for results is specified.

```
SetupDiag.exe /Output:C:\SetupDiag\Results.log
```

The following example uses the /Output parameter to save results to a path name that contains a space:

```
SetupDiag /Output:"C:\Tools\SetupDiag\SetupDiag Results\Results.log"
```

The following example specifies that SetupDiag is to run in offline mode, and to process the log files found in **D:\Temp\Logs\LogSet1**.

```
SetupDiag.exe /Output:C:\SetupDiag\Results.log /LogsPath:D:\Temp\Logs\LogSet1
```

The following example sets recovery scenario in offline mode. In the example, SetupDiag will search for reset/recovery logs in the specified LogsPath location and output the results to the directory specified by the /Output parameter.

```
SetupDiag.exe /Output:C:\SetupDiag\RecoveryResults.log /LogsPath:D:\Temp\Cabs\PBR_Log /Scenario:Recovery
```

The following example sets recovery scenario in online mode. In the example, SetupDiag will search for reset/recovery logs on the current system and output results in XML format.

```
SetupDiag.exe /Scenario:Recovery /Format:xml
```

Log files

[Windows Setup Log Files and Event Logs](#) has information about where logs are created during Windows Setup. For offline processing, you should run SetupDiag against the contents of the entire folder. For example, depending on when the upgrade failed, copy one of the following folders to your offline location:

```
\$Windows.~bt\sources\panther  
\$Windows.~bt\Sources\Rollback  
\Windows\Panther  
\Windows\Panther\NewOS
```

If you copy the parent folder and all sub-folders, SetupDiag will automatically search for log files in all subdirectories.

Setup bug check analysis

When Microsoft Windows encounters a condition that compromises safe system operation, the system halts. This condition is called a bug check. It is also commonly referred to as a system crash, a kernel error, a Stop error, or BSOD. Typically a hardware device, hardware driver, or related software causes this error.

If crash dumps [are enabled](#) on the system, a crash dump file is created. If the bug check occurs during an upgrade, Windows Setup will extract a minidump (setupmem.dmp) file. SetupDiag can also debug these setup related minidumps.

To debug a setup related bug check, you must:

- Specify the **/LogsPath** parameter. You cannot debug memory dumps in online mode.
- Gather the setup memory dump file (setupmem.dmp) from the failing system.
 - Setupmem.dmp will be created in either **%SystemDrive%\\$Windows.~bt\Sources\Rollback**, or in **%WinDir%\Panther\NewOS\Rollback** depending on when the bug check occurs.
- Install the [Windows Debugging Tools](#) on the computer that runs SetupDiag.

In the following example, the **setupmem.dmp** file is copied to the **D:\Dump** directory and the Windows Debugging Tools are installed prior to running SetupDiag:

```
SetupDiag.exe /Output:C:\SetupDiag\Dumpdebug.log /LogsPath:D:\Dump
```

Known issues

1. Some rules can take a long time to process if the log files involved are large.
2. If the failing computer is opted into the Insider program and getting regular pre-release updates, or an update is already pending on the computer when SetupDiag is run, it can encounter problems trying to open these log files. This will likely cause a failure to determine a root cause. In this case, try gathering the log files and running SetupDiag in offline mode.

Sample output

The following is an example where SetupDiag is run in offline mode.

```
D:\SetupDiag>SetupDiag.exe /output:c:\setupdiag\result.xml /logspath:D:\Tests\Logs\f55be736-beed-4b9b-aedf-
c133536c946e /format:xml

SetupDiag v1.5.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

Searching for setup logs...
Found d:\tests\Logs\f55be736-beed-4b9b-aedf-c133536c946e\setupact_6.log with update date 6/12/2019 2:44:20 PM
to be the correct setup log.
Found d:\tests\Logs\f55be736-beed-4b9b-aedf-c133536c946e\setupact_1.log with update date 6/12/2019 2:45:19 PM
to be the correct rollback log.

Gathering baseline information from setup logs...

SetupDiag: processing rule: CompatScanOnly.
...No match.

...

SetupDiag: processing rule: DISMImageSessionFailure.
..
Error: SetupDiag reports DISM provider failure.
Last Phase: Safe OS
Last Operation: Apply Optional Component status
Message = Failed to get the IDismImage instance from the image session
Function: CDISMManager::CloseImageSession
Error: 0x800706ba
Recommend you re-download the update source files, reboot and try the update again.

SetupDiag found 1 matching issue.

SetupDiag results were logged to: c:\setupdiag\results.xml
Logs ZipFile created at: c:\setupdiag\Logs_14.zip
```

Rules

When searching log files, SetupDiag uses a set of rules to match known issues. These rules are contained in the rules.xml file which is extracted when SetupDiag is run. The rules.xml file might be updated as new versions of SetupDiag are made available. See [Release notes](#) for more information.

Each rule name and its associated unique rule identifier are listed with a description of the known upgrade-blocking issue. In the rule descriptions, the term "down-level" refers to the first phase of the upgrade process, which runs under the starting OS.

1. CompatScanOnly - FFDAFD37-DB75-498A-A893-472D49A1311D
 - This rule indicates that setup.exe was called with a specific command line parameter that indicated setup was to do a compat scan only, not an upgrade.
2. BitLockerHardblock - C30152E2-938E-44B8-915B-D1181BA635AE
 - This is a block when the target OS does not support BitLocker, yet the host OS has BitLocker enabled.
3. VHDHardblock - D9ED1B82-4ED8-4DFD-8EC0-BE69048978CC
 - This block happens when the host OS is booted to a VHD image. Upgrade is not supported when the host OS is booted from a VHD image.
4. PortableWorkspaceHardblock - 5B0D3AB4-212A-4CE4-BDB9-37CA404BB280
 - This indicates that the host OS is booted from a Windows To-Go device (USB key). Upgrade is not

supported in the Windows To-Go environment.

5. AuditModeHardblock - A03BD71B-487B-4ACA-83A0-735B0F3F1A90
 - This block indicates that the host OS is currently booted into Audit Mode, a special mode for modifying the Windows state. Upgrade is not supported from this state.
6. SafeModeHardblock - 404D9523-B7A8-4203-90AF-5FBB05B6579B
 - This block indicates that the host OS is booted to Safe Mode, where upgrade is not supported.
7. InsufficientSystemPartitionDiskSpaceHardblock - 3789FBF8-E177-437D-B1E3-D38B4C4269D1
 - This block is encountered when setup determines the system partition (where the boot loader files are stored) does not have enough space to be serviced with the newer boot files required during the upgrade process.
8. CompatBlockedApplicationAutoUninstall – BEBA5BC6-6150-413E-8ACE-5E1EC8D34DD5
 - This rule indicates there is an application that needs to be uninstalled before setup can continue.
9. CompatBlockedApplicationDismissable - EA52620B-E6A0-4BBC-882E-0686605736D9
 - When running setup in /quiet mode, there are dismissible application messages that turn into blocks unless the command line also specifies “/compat /ignore warning”. This rule indicates setup was executed in /quiet mode but there is an application dismissible block message that have prevented setup from continuing.
10. CompatBlockedApplicationManualUninstall - 9E912E5F-25A5-4FC0-BEC1-CA0EA5432FF4
 - This rule indicates that an application without an Add/Remove Programs entry, is present on the system and blocking setup from continuing. This typically requires manual removal of the files associated with this application to continue.
11. HardblockDeviceOrDriver - ED3AEFA1-F3E2-4F33-8A21-184ADF215B1B
 - This indicates a device driver that is loaded on the host OS is not compatible with the newer OS version and needs to be removed prior to the upgrade.
12. HardblockMismatchedLanguage - 60BA8449-CF23-4D92-A108-D6FCEFB95B45
 - This rule indicates the host OS and the target OS language editions do not match.
13. HardblockFlightSigning - 598F2802-3E7F-4697-BD18-7A6371C8B2F8
 - This rule indicates the target OS is a pre-release, Windows Insider build, and the target machine has Secure Boot enabled. This will block the pre-release signed build from booting if installed on the machine.
14. DiskSpaceBlockInDownLevel - 6080AFAC-892E-4903-94EA-7A17E69E549E
 - This failure indicates the system ran out of disk space during the down-level operations of upgrade.
15. DiskSpaceFailure - 981DCBA5-B8D0-4BA7-A8AB-4030F7A10191
 - This failure indicates the system drive ran out of available disk space at some point after the first reboot into the upgrade.
16. DeviceInstallHang - 37BB1C3A-4D79-40E8-A556-FDA126D40BC6
 - This failure rule indicates the system hung or bug checked during the device installation phase of upgrade.
17. DebugSetupMemoryDump - C7C63D8A-C5F6-4255-8031-74597773C3C6
 - This offline only rule indicates a bug check occurred during setup. If the debugger tools are available on the system, SetupDiag will debug the memory dump and provide details.
18. DebugSetupCrash - CEEBA202-6F04-4BC3-84B8-7B99AED924B1
 - This offline only rule indicates that setup itself encountered a failure that resulted in a process memory dump. If the debugger tools are installed on the system, SetupDiag will debug the memory dump and give further details.
19. DebugMemoryDump - 505ED489-329A-43F5-B467-FCAAF6A1264C
 - This offline only rule is for any memory.dmp file that resulted during the setup/upgrade operation. If the debugger tools are installed on the system, SetupDiag will debug the memory dump and give further

details.

20. BootFailureDetected - 4FB446C2-D4EC-40B4-97E2-67EB19D1CFB7
 - This rule indicates a boot failure occurred during a specific phase of the update. The rule will indicate the failure code and phase for diagnostic purposes.
21. FindDebugInfoFromRollbackLog - 9600EB68-1120-4A87-9FE9-3A4A70ACFC37
 - This rule will determine and give details when a bug check occurs during the setup/upgrade process that resulted in a memory dump, but without the requirement of the debugger package being on the executing machine.
22. AdvancedInstallerFailed - 77D36C96-32BE-42A2-BB9C-AAFFE64FCADC
 - Finds fatal advanced installer operations that cause setup failures.
23. FindMigApplyUnitFailure - A4232E11-4043-4A37-9BF4-5901C46FD781
 - Detects a migration unit failure that caused the update to fail. This rule will output the name of the migration plug-in as well as the error code it produced for diagnostic purposes.
24. FindMigGatherUnitFailure - D04C064B-CD77-4E64-96D6-D26F30B4EE29
 - Detects a migration gather unit failure that caused the update to fail. This rule will output the name of the gather unit/plug-in as well as the error code it produced for diagnostic purposes.
25. CriticalSafeOSDUFailure - 73566DF2-CA26-4073-B34C-C9BC70DBF043
 - This rule indicates a failure occurred while updating the SafeOS image with a critical dynamic update. It will indicate the phase and error code that occurred while attempting to update the SafeOS image for diagnostic purposes.
26. UserProfileCreationFailureDuringOnlineApply - 678117CE-F6A9-40C5-BC9F-A22575C78B14
 - Indicates there was a critical failure while creating or modifying a User Profile during the online apply phase of the update. It will indicate the operation and error code associated with the failure for diagnostic purposes.
27. WimMountFailure - BE6DF2F1-19A6-48C6-AEF8-D3B0CE3D4549
 - This rule indicates the update failed to mount a wim file. It will show the name of the wim file as well as the error message and error code associated with the failure for diagnostic purposes.
28. FindSuccessfulUpgrade - 8A0824C8-A56D-4C55-95A0-22751AB62F3E
 - Determines if the given setup was a success or not based off the logs.
29. FindSetupHostReportedFailure - 6253C04F-2E4E-4F7A-B88E-95A69702F7EC
 - Gives information about failures surfaced early in the upgrade process by setuphost.exe
30. FindDownlevelFailure - 716334B7-F46A-4BAA-94F2-3E31BC9EFA55
 - Gives failure information surfaced by SetupPlatform, later in the down-level phase.
31. FindAbruptDownlevelFailure - 55882B1A-DA3E-408A-9076-23B22A0472BD
 - Gives last operation failure information when the system fails in the down-level, but the log just ends abruptly.
32. FindSetupPlatformFailedOperationInfo - 307A0133-F06B-4B75-AEA8-116C3B53C2D1
 - Gives last phase and error information when SetupPlatform indicates a critical failure. This rule will indicate the operation and error associated with the failure for diagnostic purposes.
33. FindRollbackFailure - 3A43C9B5-05B3-4F7C-A955-88F991BB5A48
 - Gives last operation, failure phase and error information when a rollback occurs.
34. AdvancedInstallerGenericFailure – 4019550D-4CAA-45B0-A222-349C48E86F71
 - A rule to match AdvancedInstaller read/write failures in a generic sense. Will output the executable being called as well as the error code and exit code reported.
35. OptionalComponentFailedToGetOCsFromPackage – D012E2A2-99D8-4A8C-BBB2-088B92083D78 (NOTE: This rule replaces the OptionalComponentInstallFailure rule present in v1.10.)
 - This matches a specific Optional Component failure when attempting to enumerate components in a package. Will output the package name and error code.

36. OptionalComponentOpenPackageFailed – 22952520-EC89-4FBD-94E0-B67DF88347F6
 - Matches a specific Optional Component failure when attempting to open an OC package. Will output the package name and error code.
37. OptionalComponentInitCBSSessionFailed – 63340812-9252-45F3-A0F2-B2A4CA5E9317
 - Matches a specific failure where the advanced installer service or components aren't operating or started on the system. Will output the error code.
38. UserProfileCreationFailureDuringFinalize – C6677BA6-2E53-4A88-B528-336D15ED1A64
 - Matches a specific User Profile creation error during the finalize phase of setup. Will output the failure code.
39. WimApplyExtractFailure – 746879E9-C9C5-488C-8D4B-0C811FF3A9A8
 - Matches a wim apply failure during wim extraction phases of setup. Will output the extension, path and error code.
40. UpdateAgentExpanderFailure – 66E496B3-7D19-47FA-B19B-4040B9FD17E2
 - Matches DPX expander failures in the down-level phase of update from WU. Will output the package name, function, expression and error code.
41. FindFatalPluginFailure – E48E3F1C-26F6-4AFB-859B-BF637DA49636
 - Matches any plug-in failure that setupplatform decides is fatal to setup. Will output the plugin name, operation and error code.
42. AdvancedInstallerFailed - 77D36C96-32BE-42A2-BB9C-AAFFE64FCADC
 - Indicates critical failure in the AdvancedInstaller while running an installer package, includes the .exe being called, the phase, mode, component and error codes.
43. MigrationAbortedDueToPluginFailure - D07A24F6-5B25-474E-B516-A730085940C9
 - Indicates a critical failure in a migration plugin that causes setup to abort the migration. Will provide the setup operation, plug-in name, plug-in action and error code.
44. DISMAddPackageFailed - 6196FF5B-E69E-4117-9EC6-9C1EAB20A3B9
 - Indicates a critical failure during a DISM add package operation. Will specify the Package Name, DISM error and add package error code.
45. PlugInComplianceBlock - D912150B-1302-4860-91B5-527907D08960
 - Detects all compat blocks from Server compliance plug-ins. Outputs the block information and remediation.
46. AdvancedInstallerGenericFailure - 4019550D-4CAA-45B0-A222-349C48E86F71
 - Triggers on advanced installer failures in a generic sense, outputting the application called, phase, mode, component and error code.
47. FindMigGatherApplyFailure - A9964E6C-A2A8-45FF-B6B5-25E0BD71428E
 - Shows errors when the migration Engine fails out on a gather or apply operation. Indicates the Migration Object (file or registry path), the Migration
48. OptionalComponentFailedToGetOCsFromPackage - D012E2A2-99D8-4A8C-BBB2-088B92083D78
 - Indicates the optional component (OC) migration operation failed to enumerate optional components from an OC Package. Outputs the package name and error code.
49. OptionalComponentOpenPackageFailed - 22952520-EC89-4FBD-94E0-B67DF88347F6
 - Indicates the optional component migration operation failed to open an optional component Package. Outputs the package name and error code.
50. OptionalComponentInitCBSSessionFailed - 63340812-9252-45F3-A0F2-B2A4CA5E9317
 - Indicates corruption in the servicing stack on the down-level system. Outputs the error code encountered while trying to initialize the servicing component on the existing OS.
51. DISMproviderFailure - D76EF86F-B3F8-433F-9EBF-B4411F8141F4
 - Triggers when a DISM provider (plug-in) fails in a critical operation. Outputs the file (plug-in name), function called + error code, and error message from the provider.

52. SysPrepLaunchModuleFailure - 7905655C-F295-45F7-8873-81D6F9149BFD
 - Indicates a sysPrep plug-in has failed in a critical operation. Indicates the plug-in name, operation name and error code.
53. UserProvidedDriverInjectionFailure - 2247C48A-7EE3-4037-AFAB-95B92DE1D980
 - A driver provided to setup (via command line input) has failed in some way. Outputs the driver install function and error code.
54. PlugInComplianceBlock - D912150B-1302-4860-91B5-527907D08960
 - These are for server upgrades only, will output the compliance block and remediation required.
55. PreReleaseWimMountDriverFound - 31EC76CC-27EC-4ADC-9869-66AABEDB56F0
 - Captures failures due to having an unrecognized wimmount.sys driver registered on the system.
56. WinSetupBootFilterFailure - C073BFC8-5810-4E19-B53B-4280B79E096C
 - Detects failures in the kernel mode file operations.
57. WimMountDriverIssue - 565B60DD-5403-4797-AE3E-BC5CB972FBAE
 - Detects failures in WimMount.sys registration on the system.
58. DISMImageSessionFailure - 61B7886B-10CD-4C98-A299-B987CB24A11C
 - Captures failure information when DISM fails to start an image session successfully.
59. FindEarlyDownlevelError - A4CE4FC9-5E10-4BB1-8ECE-3B29EB9D7C52
 - Detects failures in down-level phase before setup platform is invoked.
60. FindSPFatalError - A4028172-1B09-48F8-AD3B-86CDD7D55852
 - Captures failure information when setup platform encounters a fatal error.

Release notes

06/19/2019 - SetupDiag v1.5.0.0 is released with 60 rules, as a standalone tool available from the Download Center.

- All date and time outputs are updated to localized format per user request.
- Added setup Operation and Phase information to /verbose log.
- Added last Setup Operation and last Setup Phase information to most rules where it make sense (see new output below).
- Performance improvement in searching setupact.logs to determine correct log to parse.
- Added SetupDiag version number to text report (xml and json always had it).
- Added "no match" reports for xml and json per user request.
- Formatted Json output for easy readability.
- Performance improvements when searching for setup logs; this should be much faster now.
- Added 7 new rules: PlugInComplianceBlock, PreReleaseWimMountDriverFound, WinSetupBootFilterFailure, WimMountDriverIssue, DISMImageSessionFailure, FindEarlyDownlevelError, and FindSPFatalError. See the [Rules](#) section above for more information.
- Diagnostic information is now output to the registry at **HKLM\SYSTEM\Setup\MoSetup\Volatile\SetupDiag**
 - The **/AddReg** command was added to toggle registry output. This setting is off by default for offline mode, and on by default for online mode. The command has no effect for online mode and enables registry output for offline mode.
 - This registry key is deleted as soon as SetupDiag is run a second time, and replaced with current data, so it's always up to date.
 - This registry key also gets deleted when a new update instance is invoked.
 - For an example, see [Sample registry key](#).

05/17/2019 - SetupDiag v1.4.1.0 is released with 53 rules, as a standalone tool available from the Download

Center.

- This release adds the ability to find and diagnose reset and recovery failures (Push Button Reset).

12/18/2018 - SetupDiag v1.4.0.0 is released with 53 rules, as a standalone tool available from the Download Center.

- This release includes major improvements in rule processing performance: ~3x faster rule processing performance!
 - The FindDownlevelFailure rule is up to 10x faster.
- New rules have been added to analyze failures upgrading to Windows 10 version 1809.
- A new help link is available for resolving servicing stack failures on the down-level OS when the rule match indicates this type of failure.
- Removed the need to specify /Mode parameter. Now if you specify /LogsPath, it automatically assumes offline mode.
- Some functional and output improvements were made for several rules.

07/16/2018 - SetupDiag v1.3.1 is released with 44 rules, as a standalone tool available from the Download Center.

- This release fixes a problem that can occur when running SetupDiag in online mode on a computer that produces a setupmem.dmp file, but does not have debugger binaries installed.

07/10/2018 - SetupDiag v1.30 is released with 44 rules, as a standalone tool available from the Download Center.

- Bug fix for an over-matched plug-in rule. The rule will now correctly match only critical (setup failure) plug-in issues.
- New feature: Ability to output logs in JSON and XML format.
 - Use "/Format:xml" or "/Format:json" command line parameters to specify the new output format. See [sample logs](#) at the bottom of this topic.
 - If the "/Format:xml" or "/Format:json" parameter is omitted, the log output format will default to text.
- New Feature: Where possible, specific instructions are now provided in rule output to repair the identified error. For example, instructions are provided to remediate known blocking issues such as uninstalling an incompatible app or freeing up space on the system drive.
- 3 new rules added: AdvancedInstallerFailed, MigrationAbortedDueToPluginFailure, DISMAddPackageFailed.

05/30/2018 - SetupDiag v1.20 is released with 41 rules, as a standalone tool available from the Download Center.

- Fixed a bug in device install failure detection in online mode.
- Changed SetupDiag to work without an instance of setupact.log. Previously, SetupDiag required at least one setupact.log to operate. This change enables the tool to analyze update failures that occur prior to calling SetupHost.
- Telemetry is refactored to only send the rule name and GUID (or "NoRuleMatched" if no rule is matched) and the Setup360 ReportId. This change assures data privacy during rule processing.

05/02/2018 - SetupDiag v1.10 is released with 34 rules, as a standalone tool available from the Download Center.

- A performance enhancement has been added to result in faster rule processing.
- Rules output now includes links to support articles, if applicable.
- SetupDiag now provides the path and name of files that it is processing.
- You can now run SetupDiag by simply clicking on it and then examining the output log file.
- An output log file is now always created, whether or not a rule was matched.

03/30/2018 - SetupDiag v1.00 is released with 26 rules, as a standalone tool available from the Download Center.

Sample logs

Text log sample

Matching Profile found: OptionalComponentOpenPackageFailed - 22952520-EC89-4FBD-94E0-B67DF88347F6

System Information:

Machine Name = Offline
Manufacturer = MSI
Model = MS-7998
HostOSArchitecture = x64
FirmwareType = PCAT
BiosReleaseDate = 20160727000000.000000+000
BiosVendor = BIOS Date: 07/27/16 10:01:46 Ver: V1.70
BiosVersion = 1.70
HostOSVersion = 10.0.15063
HostOSBuildString = 15063.0.amd64fre.rs2_release.170317-1834
TargetOSBuildString = 10.0.16299.15 (rs3_release.170928-1534)
HostOSLanguageId = 2057
HostOSEdition = Core
RegisteredAV = Windows Defender,
FilterDrivers = WdFilter,wcifs,WIMMount,luafv,Wof,FileInfo,
UpgradeStartTime = 3/21/2018 9:47:16 PM
UpgradeEndTime = 3/21/2018 10:02:40 PM
UpgradeElapsedTime = 00:15:24
ReportId = dd4db176-4e3f-4451-aef6-22cf46de8bde

Error: SetupDiag reports Optional Component installation failed to open OC Package. Package Name: Foundation, Error: 0x8007001F

Recommend you check the "Windows Modules Installer" service (Trusted Installer) is started on the system and set to automatic start, reboot and try the update again. Optionally, you can check the status of optional components on the system (search for Windows Features), uninstall any unneeded optional components, reboot and try the update again.

Error: SetupDiag reports down-level failure, Operation: Finalize, Error: 0x8007001F - 0x50015

Refer to <https://docs.microsoft.com/windows/deployment/upgrade/upgrade-error-codes> for error information.

XML log sample

```

<?xml version="1.0" encoding="utf-16"?>
<SetupDiag xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://docs.microsoft.com/windows/deployment/upgrade/setupdiag">
  <Version>1.5.0.0</Version>
  <ProfileName>FindSPFatalError</ProfileName>
  <ProfileGuid>A4028172-1B09-48F8-AD3B-86CDD7D55852</ProfileGuid>
  <SystemInfo>
    <MachineName>Offline</MachineName>
    <Manufacturer>Gigabyte Technology Co., Ltd.</Manufacturer>
    <Model>X470 AORUS ULTRA GAMING</Model>
    <HostOSArchitecture>1033</HostOSArchitecture>
    <FirmwareType>UEFI</FirmwareType>
    <BiosReleaseDate>20180808000000.000000+000</BiosReleaseDate>
    <BiosVendor>F3</BiosVendor>
    <BiosVersion />
    <HostOSVersion>10.0.18908</HostOSVersion>
    <HostOSBuildString>18908.1000.amd64fre.rs_prerelease.190524-1658</HostOSBuildString>
    <TargetOSBuildString>10.0.18912.1001 (rs_prerelease.190601-1739)</TargetOSBuildString>
    <HostOSLanguageId />
    <HostOSEdition>Professional</HostOSEdition>
    <RegisteredAV>Windows Defender</RegisteredAV>
    <FilterDrivers />
    <UpgradeStartTime>2019-06-06T21:19:10</UpgradeStartTime>
    <UpgradeElapsedTime />
    <UpgradeEndTime>2019-06-06T22:21:49</UpgradeEndTime>
    <RollbackStartTime>0001-01-01T00:00:00</RollbackStartTime>
    <RollbackEndTime>0001-01-01T00:00:00</RollbackEndTime>
    <RollbackElapsedTime />
    <FinalizeStartTime>0001-01-01T00:00:00</FinalizeStartTime>
    <PostOOBESuccessTime>0001-01-01T00:00:00</PostOOBESuccessTime>
    <TotalOfflineTime />
    <CommercialId>Offline</CommercialId>
    <CV>MgUweCzk90KdwUiZ</CV>
    <SetupReportId>F21F8FB6-00FD-4349-84FB-2AC75F389E73</SetupReportId>
    <ReportId>F21F8FB6-00FD-4349-84FB-2AC75F389E73</ReportId>
  </SystemInfo>
  <LogErrorLine>2019-06-06 21:47:11, Error SP Error converting install time 5/2/2019 to
structure[gle=0x00000057]</LogErrorLine>
  <FailureData>
    Error: SetupDiag reports Fatal Error.
    Last Setup Phase = Downlevel
    Last Setup Operation: Gather data, scope: EVERYTHING
    Error: 0x00000057</FailureData>
    <FailureData>LogEntry: 2019-06-06 21:47:11, Error SP Error converting install time
5/2/2019 to structure[gle=0x00000057]</FailureData>
    <FailureData>LogEntry: 2019-06-06 21:47:11, Error SP Error converting install time
5/2/2019 to structure[gle=0x00000057]</FailureData>
    <FailureData>
    Refer to "https://docs.microsoft.com/windows/desktop/Debug/system-error-codes" for error information.
  </FailureData>
  <FailureDetails>Err = 0x00000057, LastOperation = Gather data, scope: EVERYTHING, LastPhase =
Downlevel</FailureDetails>
</SetupDiag>

```

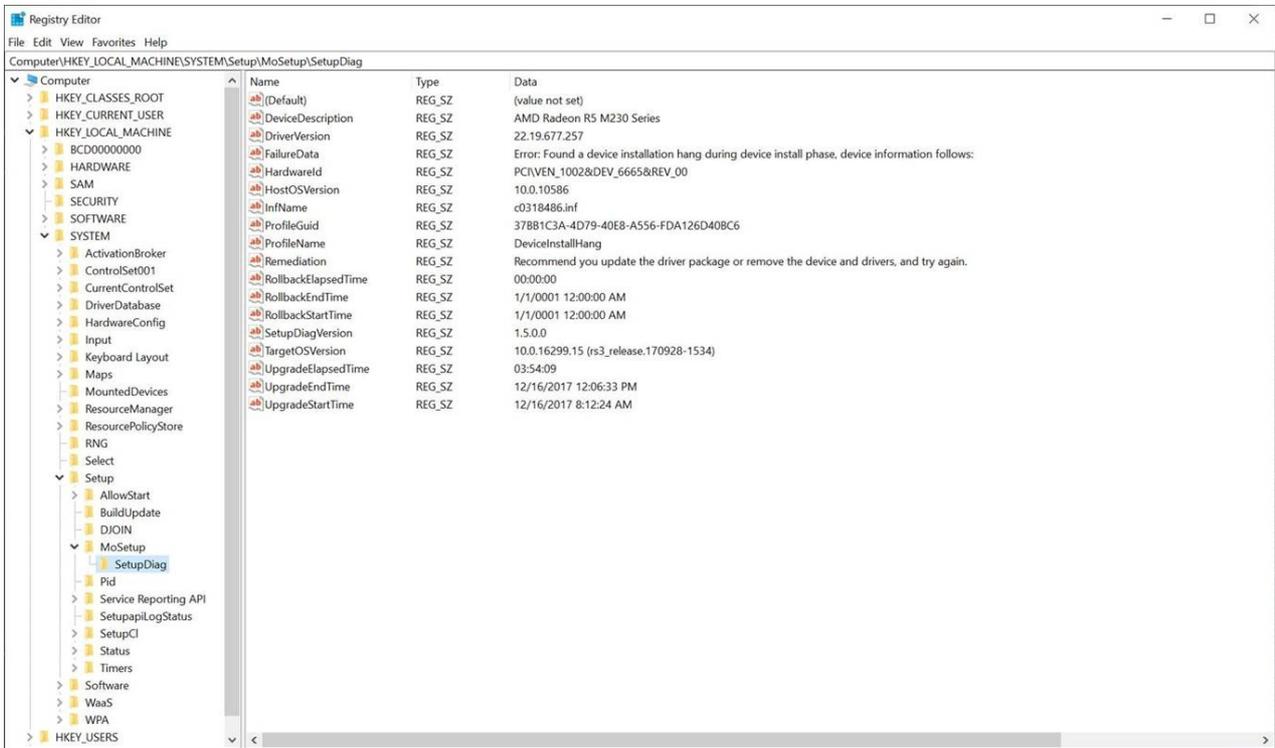
JSON log sample

```

{
  "Version": "1.5.0.0",
  "ProfileName": "FindSPFatalError",
  "ProfileGuid": "A4028172-1B09-48F8-AD3B-86CDD7D55852",
  "SystemInfo": {
    "BiosReleaseDate": "20180808000000.000000+000",
    "BiosVendor": "F3",
    "BiosVersion": "F3",
    "CV": "MgUweCZk90KdwUiz",
    "CommercialId": "Offline",
    "FilterDrivers": "",
    "FinalizeStartTime": "\\Date(-62135568000000-0800)\\",
    "FirmwareType": "UEFI",
    "HostOSArchitecture": "x64",
    "HostOSBuildString": "18908.1000.amd64fre.rs_prerelease.190524-1658",
    "HostOSEdition": "Professional",
    "HostOSLanguageId": "",
    "HostOSVersion": "",
    "MachineName": "Offline",
    "Manufacturer": "Gigabyte Technology Co., Ltd.",
    "Model": "X470 AORUS ULTRA GAMING",
    "PostOOBESuccessTime": "\\Date(-62135568000000-0800)\\",
    "RegisteredAV": "Windows Defender",
    "ReportId": "F21F8FB6-00FD-4349-84FB-2AC75F389E73",
    "RollbackElapsedTime": "PT0S",
    "RollbackEndTime": "\\Date(-62135568000000-0800)\\",
    "RollbackStartTime": "\\Date(-62135568000000-0800)\\",
    "SetupReportId": "F21F8FB6-00FD-4349-84FB-2AC75F389E73",
    "TargetOSArchitecture": null,
    "TargetOSBuildString": "10.0.18912.1001 (rs_prerelease.190601-1739)",
    "TotalOfflineTime": "PT0S",
    "UpgradeElapsedTime": "PT1H2M39S",
    "UpgradeEndTime": "\\Date(1559884909000-0700)\\",
    "UpgradeStartTime": "\\Date(1559881150000-0700)\\",
  },
  "LogErrorLine": "2019-06-06 21:47:11, Error SP Error converting install time 5\\2\\2019
to structure[
  gle=0x00000057
]",
  "FailureData": [
    "\u000aError: SetupDiag reports Fatal Error.\u000aLast Setup Phase = Downlevel\u000aLast Setup
Operation: Gather data, scope: EVERYTHING\u000aError: 0x00000057",
    "LogEntry: 2019-06-06 21:47:11, Error SP Error converting install time 5\\2\\2019
to structure[
  gle=0x00000057
]",
    "LogEntry: 2019-06-06 21:47:11, Error SP Error converting install time 5\\2\\2019
to structure[
  gle=0x00000057
]",
    "\u000aRefer to \"https://docs.microsoft.com/en-us/windows/desktop/Debug/system-error-codes\"
for error information."
  ],
  "FailureDetails": "Err = 0x00000057, LastOperation = Gather data, scope: EVERYTHING, LastPhase =
Downlevel",
  "DeviceDriverInfo": null,
  "Remediation": [
  ],
  "SetupPhaseInfo": null,
  "SetupOperationInfo": null
}

```

Sample registry key



Related topics

[Resolve Windows 10 upgrade errors: Technical information for IT Pros](#)

Troubleshooting upgrade errors

6/14/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 300 level topic (moderately advanced).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

If a Windows 10 upgrade is not successful, it can be very helpful to understand *when* an error occurred in the upgrade process.

Briefly, the upgrade process consists of four phases: **Downlevel**, **SafeOS**, **First boot**, and **Second boot**. The computer will reboot once between each phase. Note: Progress is tracked in the registry during the upgrade process using the following key: **HKLM\System\Setup\mosetup\volatile\SetupProgress**. This key is volatile and only present during the upgrade process; it contains a binary value in the range 0-100.

These phases are explained in greater detail [below](#). First, let's summarize the actions performed during each phase because this affects the type of errors that can be encountered.

1. **Downlevel phase:** Because this phase runs on the source OS, upgrade errors are not typically seen. If you do encounter an error, ensure the source OS is stable. Also ensure the Windows setup source and the destination drive are accessible.
2. **SafeOS phase:** Errors most commonly occur during this phase due to hardware issues, firmware issues, or non-microsoft disk encryption software.

Since the computer is booted into Windows PE during the SafeOS phase, a useful troubleshooting technique is to boot into [Windows PE](#) using installation media. You can use the [media creation tool](#) to create bootable media, or you can use tools such as the [Windows ADK](#), and then boot your device from this media to test for hardware and firmware compatibility issues.

TIP

If you attempt to use the media creation tool with a USB drive and this fails with error 0x80004005 - 0xa001a, this is because the USB drive is using GPT partition style. The tool requires that you use MBR partition style. You can use the DISKPART command to convert the USB drive from GPT to MBR. For more information, see [Change a GUID Partition Table Disk into a Master Boot Record Disk](#).

Do not proceed with the Windows 10 installation after booting from this media. This method can only be used to perform a clean install which will not migrate any of your apps and settings, and you will be required re-enter your Windows 10 license information.

If the computer does not successfully boot into Windows PE using the media that you created, this is likely due to a hardware or firmware issue. Check with your hardware manufacturer and apply any recommended BIOS and firmware updates. If you are still unable to boot to installation media after applying updates, disconnect or replace legacy hardware.

If the computer successfully boots into Windows PE, but you are not able to browse the system drive on the

computer, it is possible that non-Microsoft disk encryption software is blocking your ability to perform a Windows 10 upgrade. Update or temporarily remove the disk encryption.

3. **First boot phase:** Boot failures in this phase are relatively rare, and almost exclusively caused by device drivers. Disconnect all peripheral devices except for the mouse, keyboard, and display. Obtain and install updated device drivers, then retry the upgrade.
4. **Second boot phase:** In this phase, the system is running under the target OS with new drivers. Boot failures are most commonly due to anti-virus software or filter drivers. Disconnect all peripheral devices except for the mouse, keyboard, and display. Obtain and install updated device drivers, temporarily uninstall anti-virus software, then retry the upgrade.

If the general troubleshooting techniques described above or the [quick fixes](#) detailed below do not resolve your issue, you can attempt to analyze [log files](#) and interpret [upgrade error codes](#). You can also [Submit Windows 10 upgrade errors using Feedback Hub](#) so that Microsoft can diagnose your issue.

The Windows 10 upgrade process

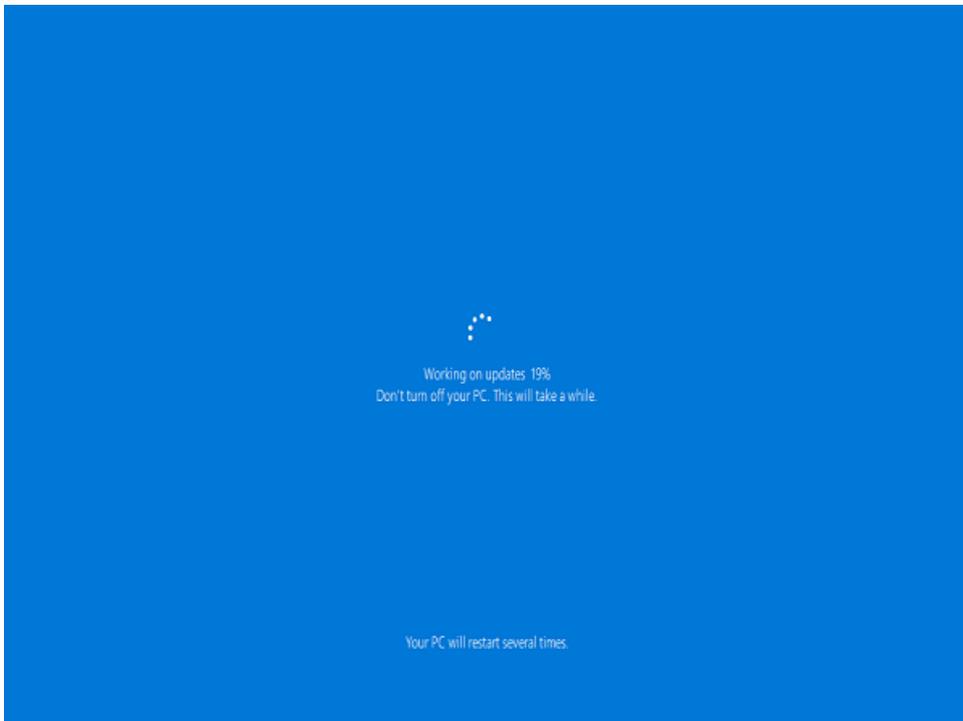
The **Windows Setup** application is used to upgrade a computer to Windows 10, or to perform a clean installation. Windows Setup starts and restarts the computer, gathers information, copies files, and creates or adjusts configuration settings.

When performing an operating system upgrade, Windows Setup uses phases described below. A reboot occurs between each of the phases. After the first reboot, the user interface will remain the same until the upgrade is completed. Percent progress is displayed and will advance as you move through each phase, reaching 100% at the end of the second boot phase.

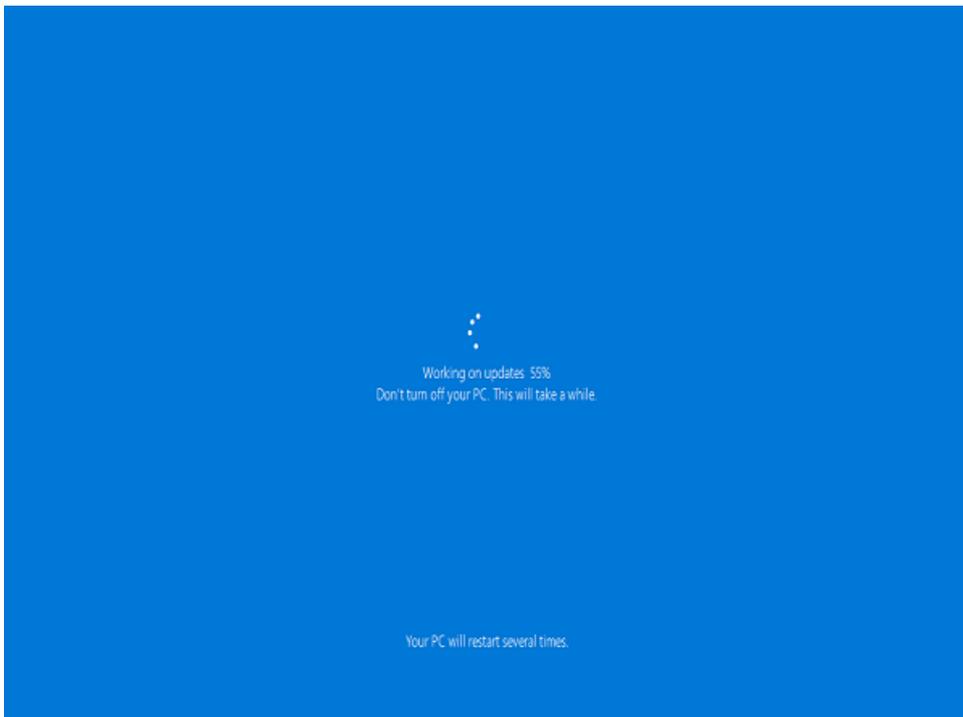
1. **Downlevel phase:** The downlevel phase is run within the previous operating system. Windows files are copied and installation components are gathered.



2. **Safe OS phase:** A recovery partition is configured, Windows files are expanded, and updates are installed. An OS rollback is prepared if needed. Example error codes: 0x2000C, 0x20017.

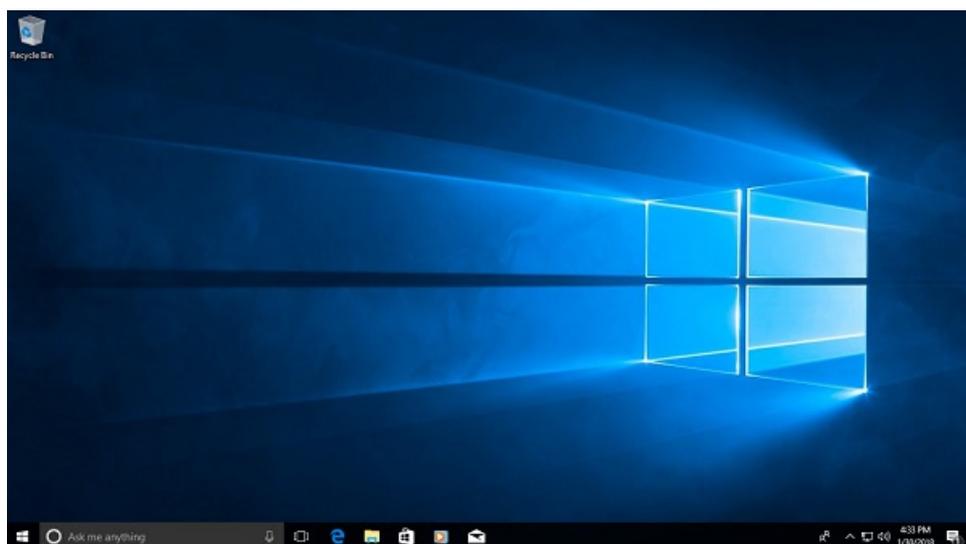
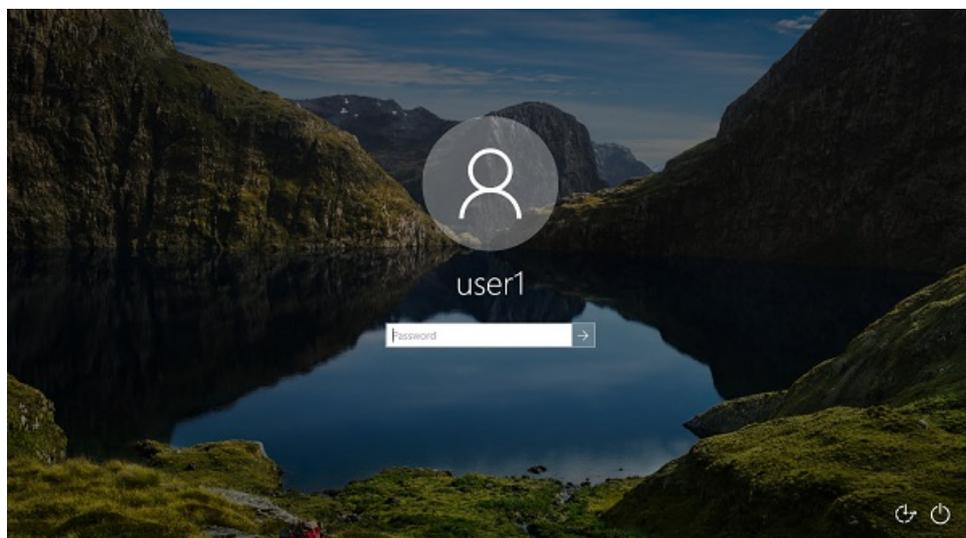
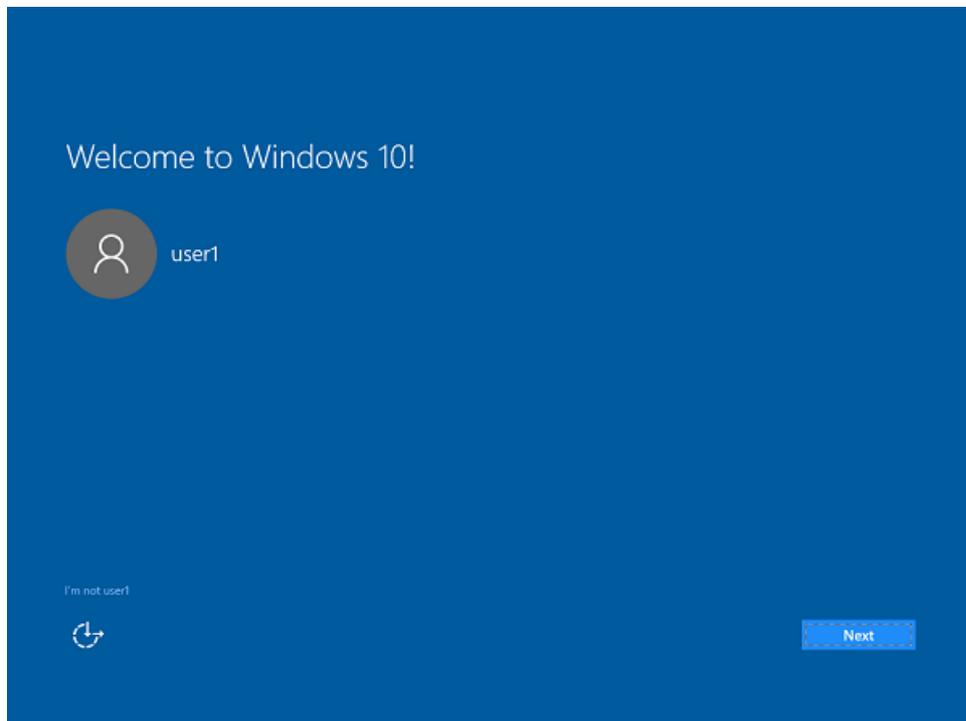


3. **First boot phase:** Initial settings are applied. Example error codes: 0x30018, 0x3000D.



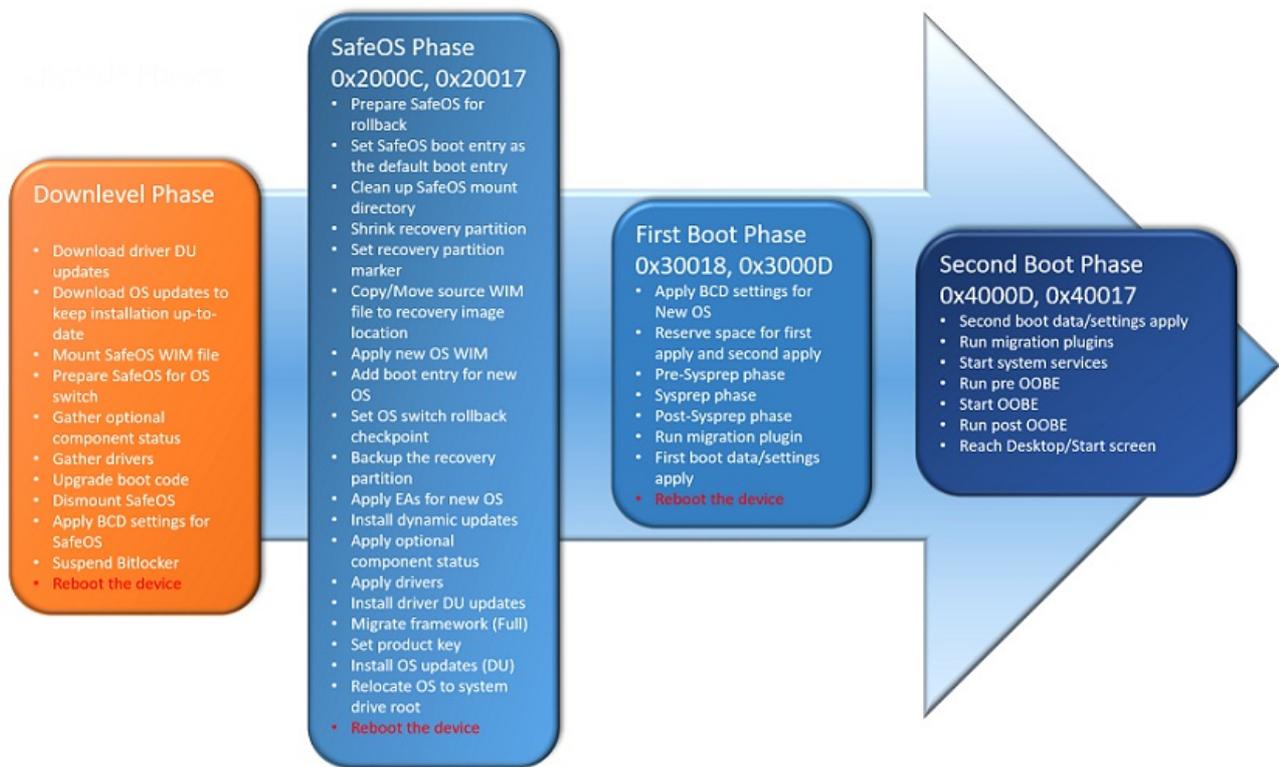
4. **Second boot phase:** Final settings are applied. This is also called the **OOBE boot phase**. Example error codes: 0x4000D, 0x40017.

At the end of the second boot phase, the **Welcome to Windows 10** screen is displayed, preferences are configured, and the Windows 10 sign-in prompt is displayed.



5. **Uninstall phase:** This phase occurs if upgrade is unsuccessful (image not shown). Example error codes: 0x50000, 0x50015.

Figure 1: Phases of a successful Windows 10 upgrade (uninstall is not shown):



DU = Driver/device updates.

OOBE = Out of box experience.

WIM = Windows image (Microsoft)

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Windows Error Reporting

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 300 level topic (moderately advanced).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

When Windows Setup fails, the result and extend code are recorded as an informational event in the Application log by Windows Error Reporting as event 1001. The event name is **WinSetupDiag02**. You can use Event Viewer to review this event, or you can use Windows PowerShell.

To use Windows PowerShell, type the following commands from an elevated Windows PowerShell prompt:

IMPORTANT

The following source will be available only if you have updated from a previous version of Windows 10 to a new version. If you installed the current version and have not updated, the source named **WinSetupDiag02** will be unavailable.

```
$events = Get-WinEvent -FilterHashtable @{LogName="Application";ID="1001";Data="WinSetupDiag02"}
$event = [xml]$events[0].ToXml()
$event.Event.EventData.Data
```

To use Event Viewer:

1. Open Event Viewer and navigate to **Windows Logs\Application**.
2. Click **Find**, and then search for **winsetupdiag02**.
3. Double-click the event that is highlighted.

Note: For legacy operating systems, the Event Name was WinSetupDiag01.

Ten parameters are listed in the event:

PARAMETERS

P1: The Setup Scenario (1=Media,5=WindowsUpdate,7=Media Creation Tool)

P2: Setup Mode (x=default,1=Downlevel,5=Rollback)

P3: New OS Architecture (x=default,0=X86,9=AMD64)

P4: Install Result (x=default,0=Success,1=Failure,2=Cancel,3=Blocked)

P5: Result Error Code (Ex: 0xc1900101)

P6: Extend Error Code (Ex: 0x20017)

PARAMETERS

P7: Source OS build (Ex: 9600)

P8: Source OS branch (not typically available)

P9: New OS build (Ex: 16299)

P10: New OS branch (Ex: rs3_release)

The event will also contain links to log files that can be used to perform a detailed diagnosis of the error. An example of this event from a successful upgrade is shown below.

Event Properties - Event 1001, Windows Error Reporting

General Details

Fault bucket, type 0
Event Name: WinSetupDiag02
Response: Not available
Cab Id: 0

Problem signature:
P1: 1
P2: 4
P3: 9
P4: 0
P5: 0x0
P6: 0x0
P7: 7601
P8: win7sp1_rtm
P9: 16299
P10: rs3_release

Attached files:
[\\?\C:\Windows\Panther\SetupAct.log](#)
[\\?\C:\Windows\Panther\diagerr.xml](#)
[\\?\C:\Windows\inf\setupapi.setup.log](#)
[\\?\C:\Windows\inf\setupapi.dev.log](#)
[\\?\C:\Windows\inf\setupapi.offline.log](#)
[\\?\C:\ProgramData\Microsoft\Windows\WER\Temp\WER8A0F.tmp.WERInternalMetadata.xml](#)
[\\?\C:\ProgramData\Microsoft\Windows\WER\Temp\WER8BD4.tmp.csv](#)
[\\?\C:\ProgramData\Microsoft\Windows\WER\Temp\WER8C04.tmp.txt](#)

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\NonCritical_1_d2dd2fd6e01459e92a498934e219ae428bafdcaa_00000000_cab_08928c51

Analysis symbol:
Rechecking for solution: 0
Report Id: d53897e2-49f6-4399-b9c2-17adb556149f
Report Status: 4
Hashed bucket:

Log Name:	Application	Logged:	1/31/2018 11:39:55 AM
Source:	Windows Error Reporting	Task Category:	None
Event ID:	1001	Keywords:	Classic
Level:	Information	Computer:	user1-PC
User:	N/A		
OpCode:			
More Information:	Event Log Online Help		

Copy Close

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Upgrade error codes

6/14/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 400 level topic (advanced).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

If the upgrade process is not successful, Windows Setup will return two codes:

1. **A result code:** The result code corresponds to a specific Win32 or NTSTATUS error.
2. **An extend code:** The extend code contains information about both the *phase* in which an error occurred, and the *operation* that was being performed when the error occurred.

For example, a result code of **0xC1900101** with an extend code of **0x4000D** will be returned as: **0xC1900101 - 0x4000D**.

Note: If only a result code is returned, this can be because a tool is being used that was not able to capture the extend code. For example, if you are using the [Windows 10 Upgrade Assistant](#) then only a result code might be returned.

TIP

If you are unable to locate the result and extend error codes, you can attempt to find these codes using Event Viewer. For more information, see [Windows Error Reporting](#).

Result codes

A result code of **0xC1900101** is generic and indicates that a rollback occurred. In most cases, the cause is a driver compatibility issue.

To troubleshoot a failed upgrade that has returned a result code of 0xC1900101, analyze the extend code to determine the Windows Setup phase, and see the [Resolution procedures](#) section later in this article.

The following set of result codes are associated with [Windows Setup](#) compatibility warnings:

RESULT CODE	MESSAGE	DESCRIPTION
0xC1900210	MOSETUP_E_COMPAT_SCANONLY	Setup did not find any compat issue
0xC1900208	MOSETUP_E_COMPAT_INSTALLREQ_BLOCK	Setup found an actionable compat issue, such as an incompatible app
0xC1900204	MOSETUP_E_COMPAT_MIGCHOICE_BLOCK	The migration choice selected is not available (ex: Enterprise to Home)

RESULT CODE	MESSAGE	DESCRIPTION
0xC1900200	MOSETUP_E_COMPAT_SYSREQ_BLOCK	The computer is not eligible for Windows 10
0xC190020E	MOSETUP_E_INSTALLDISKSPACE_BLOCK	The computer does not have enough free space to install

A list of modern setup (mosetup) errors with descriptions in the range is available in the [Resolution procedures](#) topic in this article.

Other result codes can be matched to the specific type of error encountered. To match a result code to an error:

1. Identify the error code type as either Win32 or NTSTATUS using the first hexadecimal digit:
 - 8** = Win32 error code (ex: 0x**8**0070070)
 - C** = NTSTATUS value (ex: 0x**C**1900107)
2. Write down the last 4 digits of the error code (ex: 0x8007**0070** = 0070). These digits are the actual error code type as defined in the [HRESULT](#) or the [NTSTATUS](#) structure. Other digits in the code identify things such as the device type that produced the error.
3. Based on the type of error code determined in the first step (Win32 or NTSTATUS), match the 4 digits derived from the second step to either a Win32 error code or NTSTATUS value using the following links:
 - [Win32 error code](#)
 - [NTSTATUS value](#)

Examples:

- 0x80070070
 - Based on the "8" this is a Win32 error code
 - The last four digits are 0070, so look up 0x00000070 in the [Win32 error code](#) table
 - The error is: **ERROR_DISK_FULL**
- 0xC1900107
 - Based on the "C" this is an NTSTATUS error code
 - The last four digits are 0107, so look up 0x00000107 in the [NTSTATUS value](#) table
 - The error is: **STATUS_SOME_NOT_MAPPED**

Some result codes are self-explanatory, whereas others are more generic and require further analysis. In the examples shown above, ERROR_DISK_FULL indicates that the hard drive is full and additional room is needed to complete Windows upgrade. The message STATUS_SOME_NOT_MAPPED is more ambiguous, and means that an action is pending. In this case, the action pending is often the cleanup operation from a previous installation attempt, which can be resolved with a system reboot.

Extend codes

Important: Extend codes reflect the current Windows 10 upgrade process, and might change in future releases of Windows 10. The codes discussed in this section apply to Windows 10 version 1607, also known as the Anniversary Update.

Extend codes can be matched to the phase and operation when an error occurred. To match an extend code to the phase and operation:

1. Use the first digit to identify the phase (ex: 0x4000D = 4).
2. Use the last two digits to identify the operation (ex: 0x4000D = 0D).
3. Match the phase and operation to values in the tables provided below.

The following tables provide the corresponding phase and operation for values of an extend code:

Extend code: phase	
Hex	Phase
0	SP_EXECUTION_UNKNOWN
1	SP_EXECUTION_DOWNLEVEL
2	SP_EXECUTION_SAFE_OS
3	SP_EXECUTION_FIRST_BOOT
4	SP_EXECUTION_OOBE_BOOT
5	SP_EXECUTION_UNINSTALL

Extend code: operation			
Hex	Operation	Hex	Operation
0	SP_EXECUTION_OP_UNKNOWN	10	SP_EXECUTION_OP_ADD_DRIVER
1	SP_EXECUTION_OP_COPY_PAYLOAD	11	SP_EXECUTION_OP_ENABLE_FEATURE
2	SP_EXECUTION_OP_DOWNLOAD_UPDATES	12	SP_EXECUTION_OP_DISABLE_FEATURE
3	SP_EXECUTION_OP_INSTALL_UPDATES	13	SP_EXECUTION_OP_REGISTER_ASYNC_PROCESS
4	SP_EXECUTION_OP_INSTALL_RECOVERY_ENVIRONMENT	14	SP_EXECUTION_OP_REGISTER_SYNC_PROCESS
5	SP_EXECUTION_OP_INSTALL_RECOVERY_IMAGE	15	SP_EXECUTION_OP_CREATE_FILE
6	SP_EXECUTION_OP_REPLICATE_OPC	16	SP_EXECUTION_OP_CREATE_REGISTRY
7	SP_EXECUTION_OP_INSTALL_DRIVERS	17	SP_EXECUTION_OP_BOOT
8	SP_EXECUTION_OP_PREPARE_SAFE_OS	18	SP_EXECUTION_OP_SYSPREP
9	SP_EXECUTION_OP_PREPARE_ROLLBACK	19	SP_EXECUTION_OP_OOBE
A	SP_EXECUTION_OP_PREPARE_FIRST_BOOT	1A	SP_EXECUTION_OP_BEGIN_FIRST_BOOT
B	SP_EXECUTION_OP_PREPARE_OOBE_BOOT	1B	SP_EXECUTION_OP_END_FIRST_BOOT
C	SP_EXECUTION_OP_APPLY_IMAGE	1C	SP_EXECUTION_OP_BEGIN_OOBE_BOOT
D	SP_EXECUTION_OP_MIGRATE_DATA	1D	SP_EXECUTION_OP_END_OOBE_BOOT
E	SP_EXECUTION_OP_SET_PRODUCT_KEY	1E	SP_EXECUTION_OP_PRE_OOBE
F	SP_EXECUTION_OP_ADD_UNATTEND	1F	SP_EXECUTION_OP_POST_OOBE
		20	SP_EXECUTION_OP_ADD_PROVISIONING_PACKAGE

For example: An extend code of **0x4000D**, represents a problem during phase 4 (**0x4**) with data migration (**000D**).

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Log files

6/26/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 400 level topic (advanced).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

Several log files are created during each phase of the upgrade process. These log files are essential for troubleshooting upgrade problems. By default, the folders that contain these log files are hidden on the upgrade target computer. To view the log files, configure Windows Explorer to view hidden items, or use a tool to automatically gather these logs. The most useful log is **setupact.log**. The log files are located in a different folder depending on the Windows Setup phase. Recall that you can determine the phase from the extend code.

Note: Also see the [Windows Error Reporting](#) section in this document for help locating error codes and log files.

The following table describes some log files and how to use them for troubleshooting purposes:

Log file	Phase: Location	Description	When to use
setupact.log	Down-Level: \$Windows.~BT\Sources\Panther	Contains information about setup actions during the downlevel phase.	All down-level failures and starting point for rollback investigations. This is the most important log for diagnosing setup issues.
	OOBE: \$Windows.~BT\Sources\Panther\UnattendGC	Contains information about actions during the OOBE phase.	Investigating rollbacks that failed during OOBE phase and operations – 0x4001C, 0x4001D, 0x4001E, 0x4001F.
	Rollback: \$Windows.~BT\Sources\Rollback	Contains information about actions during rollback.	Investigating generic rollbacks - 0xC1900101.
	Pre-initialization (prior to downlevel): Windows	Contains information about initializing setup.	If setup fails to launch.
	Post-upgrade (after OOBE): Windows\Panther	Contains information about setup actions during the installation.	Investigate post-upgrade related issues.

setuperr.log	Same as setupact.log	Contains information about setup errors during the installation.	Review all errors encountered during the installation phase.
miglog.xml	Post-upgrade (after OOBE): Windows\Panther	Contains information about what was migrated during the installation.	Identify post upgrade data migration issues.
BlueBox.log	Down-Level: Windows\Logs\Mosetup	Contains information communication between setup.exe and Windows Update.	Use during WSUS and WU down-level failures or for 0xC1900107.
Supplemental rollback logs: Setupmem.dmp setupapi.dev.log Event logs (*.evtx)	\$Windows.~\BT\Sources\Rollback	Additional logs collected during rollback.	Setupmem.dmp: If OS bugchecks during upgrade, setup will attempt to extract a mini-dump. Setupapi: Device install issues - 0x30018 Event logs: Generic rollbacks (0xC1900101) or unexpected reboots.

Log entry structure

A setupact.log or setuperr.log entry (files are located at C:\Windows) includes the following elements:

1. **The date and time** - 2016-09-08 09:20:05.
2. **The log level** - Info, Warning, Error, Fatal Error.
3. **The logging component** - CONX, MOUPG, PANTHR, SP, IBSLIB, MIG, DISM, CSI, CBS.
 - The logging components SP (setup platform), MIG (migration engine), and CONX (compatibility information) are particularly useful for troubleshooting Windows Setup errors.
4. **The message** - Operation completed successfully.

See the following example:

DATE/TIME	LOG LEVEL	COMPONENT	MESSAGE
2016-09-08 09:23:50,	Warning	MIG	Could not replace object C:\Users\name\Cookies. Target Object cannot be removed.

Analyze log files

The following instructions are meant for IT professionals. Also see the [Upgrade error codes](#) section in this guide to familiarize yourself with [result codes](#) and [extend codes](#).

To analyze Windows Setup log files:

1. Determine the Windows Setup error code. This code should be returned by Windows Setup if it is not successful with the upgrade process.
2. Based on the [extend code](#) portion of the error code, determine the type and location of a [log files](#) to investigate.
3. Open the log file in a text editor, such as notepad.

4. Using the [result code](#) portion of the Windows Setup error code, search for the result code in the file and find the last occurrence of the code. Alternatively search for the "abort" and abandoning" text strings described in step 7 below.
5. To find the last occurrence of the result code:
 - a. Scroll to the bottom of the file and click after the last character.
 - b. Click **Edit**.
 - c. Click **Find**.
 - d. Type the result code.
 - e. Under **Direction** select **Up**.
 - f. Click **Find Next**.
6. When you have located the last occurrence of the result code, scroll up a few lines from this location in the file and review the processes that failed just prior to generating the result code.
7. Search for the following important text strings:
 - **Shell application requested abort**
 - **Abandoning apply due to error for object**
8. Decode Win32 errors that appear in this section.
9. Write down the timestamp for the observed errors in this section.
10. Search other log files for additional information matching these timestamps or errors.

For example, assume that the error code for an error is 0x8007042B - 0x2000D. Searching for "8007042B" reveals the following content from the setuperr.log file:

Some lines in the text below are shortened to enhance readability. The date and time at the start of each line (ex: 2016-10-05 15:27:08) is shortened to minutes and seconds, and the certificate file name which is a long text string is shortened to just "CN."

setuperr.log content:

```
27:08, Error      SP      Error READ, 0x00000570 while gathering/applying object: File, C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]. Will return
0[gLe=0x00000570]
27:08, Error      MIG     Error 1392 while gathering object C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]. Shell application requested abort!
[gLe=0x00000570]
27:08, Error      Gather failed. Last error: 0x00000000
27:08, Error      SP      SPDoFrameworkGather: Gather operation failed. Error: 0x0000002C
27:09, Error      SP      CMigrateFramework: Gather framework failed. Status: 44
27:09, Error      SP      Operation failed: Migrate framework (Full). Error: 0x8007042B[gLe=0x00000b7]
27:09, Error      SP      Operation execution failed: 13. hr = 0x8007042B[gLe=0x00000b7]
27:09, Error      SP      CSetupPlatformPrivate::Execute: Execution of operations queue failed, abandoning. Error: 0x8007042B[gLe=0x00000b7]
```

The first line indicates there was an error **0x00000570** with the file **C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]** (shown below):

```
27:08, Error      SP      Error READ, 0x00000570 while gathering/applying object: File, C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]. Will return
0[gLe=0x00000570]
```

The error 0x00000570 is a [Win32 error code](#) corresponding to: ERROR_FILE_CORRUPT: The file or directory is corrupted and unreadable.

Therefore, Windows Setup failed because it was not able to migrate the corrupt file **C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18[CN]**. This file is a local system certificate and can be safely deleted. Searching the setupact.log file for additional details, the phrase "Shell application requested abort" is found in a location with the same timestamp as the lines in setuperr.log. This confirms our suspicion that this file is the cause of the upgrade failure:

setupact.log content:

```
27:00, Info          Gather started at 10/5/2016 23:27:00
27:00, Info [0x000489] MIG  Setting system object filter context (System)
27:00, Info [0x0003e5] MIG  Not unmapping HKCU\Software\Classes; it is not mapped
27:00, Info [0x0003e5] MIG  Not unmapping HKCU; it is not mapped
27:00, Info          SP      ExecuteProgress: Elapsed events:1 of 4, Percent: 12
27:00, Info [0x0002c6] MIG  Processing GATHER for migration unit: <System>\UpgradeFramework (CMXEAgent)
27:00, Error        SP      Error READ, 0x00000570 while gathering/applying object: File, C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]. Will return
0[gLe=0x00000570]
27:00, Error        MIG    Error 1392 while gathering object C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18 [CN]. Shell application requested abort!
[gLe=0x00000570]
27:00, Info          SP      ExecuteProgress: Elapsed events:2 of 4, Percent: 25
27:00, Info          SP      ExecuteProgress: Elapsed events:3 of 4, Percent: 37
27:00, Info [0x000489] MIG  Setting system object filter context (System)
27:00, Info [0x0003e5] MIG  Not unmapping HKCU\Software\Classes; it is not mapped
27:00, Info [0x0003e5] MIG  Not unmapping HKCU; it is not mapped
27:00, Info          MIG    COutOfProcPluginFactory::FreeSurrogateHost: Shutdown in progress.
27:00, Info          MIG    COutOfProcPluginFactory::LaunchSurrogateHost::CommandLine: -shortened-
27:00, Info          MIG    COutOfProcPluginFactory::LaunchSurrogateHost: Successfully launched host and got control object.
27:00, Error        Gather failed. Last error: 0x00000000
27:00, Info          Gather ended at 10/5/2016 23:27:08 with result 44
27:00, Info          Leaving MigGather method
27:00, Error        SPDOFrameworkGather: Gather operation failed. Error: 0x0000002C
```

This analysis indicates that the Windows upgrade error can be resolved by deleting the C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18[CN] file. Note: In this example, the full, unshortened file name is C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18\be8228fb2d3cb6c6b0ccd9ad51b320b4_a43d512c-69f2-42de-ae9-7a88fabdaa3f.

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Resolution procedures

6/14/2019 • 17 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 200 level topic (moderate).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

0xC1900101

A frequently observed result code is 0xC1900101. This result code can be thrown at any stage of the upgrade process, with the exception of the downlevel phase. 0xC1900101 is a generic rollback code, and usually indicates that an incompatible driver is present. The incompatible driver can cause blue screens, system hangs, and unexpected reboots. Analysis of supplemental log files is often helpful, such as:

- The minidump file: `$Windows.~bt\Sources\Rollback\setupmem.dmp`,
- Event logs: `$Windows.~bt\Sources\Rollback*.evtx`
- The device install log: `$Windows.~bt\Sources\Rollback\setupapi\setupapi.dev.log`

The device install log is particularly helpful if rollback occurs during the sysprep operation (extend code 0x30018). To resolve a rollback due to driver conflicts, try running setup using a minimal set of drivers and startup programs by performing a [clean boot](#) before initiating the upgrade process.

See the following general troubleshooting procedures associated with a result code of 0xC1900101:

<p>Code 0xC1900101 - 0x20004</p> <p>Cause Windows Setup encountered an error during the SAFE_OS with the INSTALL_RECOVERY_ENVIRONMENT operation This is generally caused by out-of-date drivers.</p>	<p>Mitigation Uninstall antivirus applications. Remove all unused SATA devices. Remove all unused devices and drivers. Update drivers and BIOS.</p>
<p>Code 0xC1900101 - 0x2000c</p> <p>Cause Windows Setup encountered an unspecified error during Wim apply in the WinPE phase. This is generally caused by out-of-date drivers.</p>	<p>Mitigation Disconnect all peripheral devices that are connected to the system, except for the mouse, keyboard and display. Contact your hardware vendor to obtain updated device drivers. Ensure that "Download and install updates (recommended)" is accepted at the start of the upgrade process.</p>

<p>Code 0xC1900101 - 0x20017</p> <p>Cause A driver has caused an illegal operation. Windows was not able to migrate the driver, resulting in a rollback of the operating system. This is a SafeOS boot failure, typically caused by drivers or non-Microsoft disk encryption software.</p>	<p>Mitigation Ensure that all that drivers are updated. Open the Setuperr.log and Setupact.log files in the %windir%\Panther directory, and then locate the problem drivers. For more information, see Understanding Failures and Log Files. Update or uninstall the problem drivers.</p>
<p>Code 0xC1900101 - 0x30018</p> <p>Cause A device driver has stopped responding to setup.exe during the upgrade process.</p>	<p>Mitigation Disconnect all peripheral devices that are connected to the system, except for the mouse, keyboard and display. Contact your hardware vendor to obtain updated device drivers. Ensure that "Download and install updates (recommended)" is accepted at the start of the upgrade process.</p>
<p>Code 0xC1900101 - 0x3000D</p> <p>Cause Installation failed during the FIRST_BOOT phase while attempting the MIGRATE_DATA operation. This can occur due to a problem with a display driver.</p>	<p>Mitigation Disconnect all peripheral devices that are connected to the system, except for the mouse, keyboard and display. Update or uninstall the display driver.</p>
<p>Code 0xC1900101 - 0x4000D</p> <p>Cause A rollback occurred due to a driver configuration issue. Installation failed during the second boot phase while attempting the MIGRATE_DATA operation. This can occur due to incompatible drivers.</p>	<p>Mitigation</p> <p>Check supplemental rollback logs for a setupmem.dmp file, or event logs for any unexpected reboots or errors. Review the rollback log and determine the stop code. The rollback log is located in the %\$SystemRoot%\Sources\Panther folder. An example analysis is shown below. This example is not representative of all cases:</p> <pre>Info SP Crash 0x0000007E detected Info SP Module name : Info SP Bugcheck parameter 1 : 0xFFFFFFFFC0000005 Info SP Bugcheck parameter 2 : 0xFFFFF8015BC0036A Info SP Bugcheck parameter 3 : 0xFFFFD000E5D23728 Info SP Bugcheck parameter 4 : 0xFFFFD000E5D22F40 Info SP Cannot recover the system. Info SP Rollback: Showing splash window with restoring text: Restoring your previous version of Windows.</pre> <p>Typically, there is a dump file for the crash to analyze. If you are not equipped to debug the dump, then attempt the following basic troubleshooting procedures:</p> <ol style="list-style-type: none"> 1. Make sure you have enough disk space. 2. If a driver is identified in the bug check message, disable the driver or check with the manufacturer for driver updates. 3. Try changing video adapters. 4. Check with your hardware vendor for any BIOS updates. 5. Disable BIOS memory options such as caching or shadowing.

<p>Code 0xC1900101 - 0x40017</p> <p>Cause Windows 10 upgrade failed after the second reboot. This is usually caused by a faulty driver. For example: antivirus filter drivers or encryption drivers.</p>	<p>Mitigation Clean boot into Windows, and then attempt the upgrade to Windows 10. For more information, see How to perform a clean boot in Windows.</p> <p>Ensure you select the option to "Download and install updates (recommended)."</p>
--	--

0x800xxxxx

Result codes starting with the digits 0x800 are also important to understand. These error codes indicate general operating system errors, and are not unique to the Windows upgrade process. Examples include timeouts, devices not functioning, and a process stopping unexpectedly.

See the following general troubleshooting procedures associated with a result code of 0x800xxxxx:

<p>Code 80040005 - 0x20007</p> <p>Cause An unspecified error occurred with a driver during the SafeOS phase.</p>	<p>Mitigation This error has more than one possible cause. Attempt quick fixes, and if not successful, analyze log files in order to determine the problem and solution.</p>
<p>Code 0x80073BC3 - 0x20009 0x8007002 - 0x20009 0x80073B92 - 0x20009</p> <p>Cause The requested system device cannot be found, there is a sharing violation, or there are multiple devices matching the identification criteria.</p>	<p>Mitigation These errors occur during partition analysis and validation, and can be caused by the presence of multiple system partitions. For example, if you installed a new system drive but left the previous system drive connected, this can cause a conflict. To resolve the errors, disconnect or temporarily disable drives that contain the unused system partition. You can reconnect the drive after the upgrade has completed. Alternatively, you can delete the unused system partition.</p>
<p>Code 800704B8 - 0x3001A</p> <p>Cause An extended error has occurred during the first boot phase.</p>	<p>Mitigation Disable or uninstall non-Microsoft antivirus applications, disconnect all unnecessary devices, and perform a clean boot.</p>

<p>Code 8007042B - 0x4000D</p> <p>Cause The installation failed during the second boot phase while attempting the MIGRATE_DATA operation. This issue can occur due to file system, application, or driver issues.</p>	<p>Mitigation Analyze log files in order to determine the file, application, or driver that is not able to be migrated. Disconnect, update, remove, or replace the device or object.</p>
<p>Code 8007001F - 0x3000D</p> <p>Cause The installation failed in the FIRST_BOOT phase with an error during MIGRATE_DATA operation.</p>	<p>Mitigation Analyze log files in order to determine the files or registry entries that are blocking data migration.</p> <p>This error can be due to a problem with user profiles. It can occur due to corrupt registry entries under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList or invalid files in the \Users directory.</p> <p>Note: If a previous upgrade did not complete, invalid profiles might exist in the Windows.old\Users directory.</p> <p>To repair this error, ensure that deleted accounts are not still present in the Windows registry and that files under the \Users directory are valid. Delete the invalid files or user profiles that are causing this error. The specific files and profiles that are causing the error will be recorded in the Windows setup log files.</p>
<p>Code 8007001F - 0x4000D</p> <p>Cause General failure, a device attached to the system is not functioning.</p>	<p>Mitigation Analyze log files in order to determine the device that is not functioning properly. Disconnect, update, or replace the device.</p>
<p>Code 8007042B - 0x4001E</p> <p>Cause The installation failed during the second boot phase while attempting the PRE_OOBE operation.</p>	<p>Mitigation This error has more than one possible cause. Attempt quick fixes, and if not successful, analyze log files in order to determine the problem and solution.</p>

Other result codes

Error code	Cause	Mitigation
0xC1800118	WSUS has downloaded content that it cannot use due to a missing decryption key.	See Steps to resolve error 0xC1800118 for information.

0xC1900200	Setup.exe has detected that the machine does not meet the minimum system requirements.	Ensure the system you are trying to upgrade meets the minimum system requirements. See Windows 10 specifications for information.
0x80090011	A device driver error occurred during user data migration.	Contact your hardware vendor and get all the device drivers updated. It is recommended to have an active internet connection during upgrade process. Ensure that "Download and install updates (recommended)" is accepted at the start of the upgrade process.
0xC7700112	Failure to complete writing data to the system drive, possibly due to write access failure on the hard disk.	This issue is resolved in the latest version of Upgrade Assistant. Ensure that "Download and install updates (recommended)" is accepted at the start of the upgrade process.
0x80190001	An unexpected error was encountered while attempting to download files required for upgrade.	To resolve this issue, download and run the media creation tool. See Download windows 10 .
0x80246007	The update was not downloaded successfully.	Attempt other methods of upgrading the operating system. Download and run the media creation tool. See Download windows 10 . Attempt to upgrade using .ISO or USB. Note: Windows 10 Enterprise isn't available in the media creation tool. For more information, go to the Volume Licensing Service Center .
0x80244018	Your machine is connected through a proxy server.	Make sure Automatically Detect Settings is selected in internet options. (Control Panel > Internet Options > Connections > LAN Settings).
0xC1900201	The system did not pass the minimum requirements to install the update.	Contact the hardware vendor to get the latest updates.
0x80240017	The upgrade is unavailable for this edition of Windows.	Administrative policies enforced by your organization might be preventing the upgrade. Contact your IT administrator.
0x80070020	The existing process cannot access the file because it is being used by another process.	Use the MSCONFIG tool to perform a clean boot on the machine and then try to perform the update again. For more information, see How to perform a clean boot in Windows .
0x80070522	The user doesn't have required privilege or credentials to upgrade.	Ensure that you have signed in as a local administrator or have local administrator privileges.

0xC1900107	A cleanup operation from a previous installation attempt is still pending and a system reboot is required in order to continue the upgrade.	Reboot the device and run setup again. If restarting device does not resolve the issue, then use the Disk Cleanup utility and cleanup the temporary as well as the System files. For more information, see Disk cleanup in Windows 10 .
0xC1900209	The user has chosen to cancel because the system does not pass the compatibility scan to install the update. Setup.exe will report this error when it can upgrade the machine with user data but cannot migrate installed applications.	<p>Incompatible software is blocking the upgrade process. Uninstall the application and try the upgrade again. See Windows 10 Pre-Upgrade Validation using SETUP.EXE for more information.</p> <p>You can also download the Windows Assessment and Deployment Kit (ADK) for Windows 10 and install Application Compatibility Tools.</p>
0x8007002	This error is specific to upgrades using System Center Configuration Manager 2012 R2 SP1 CU3 (5.00.8238.1403)	<p>Analyze the SMSTS.log and verify that the upgrade is failing on "Apply Operating system" Phase: Error 80072efe DownloadFileWithRanges() failed. 80072efe. ApplyOperatingSystem (0x0760)</p> <p>The error 80072efe means that the connection with the server was terminated abnormally.</p> <p>To resolve this issue, try the OS Deployment test on a client in same VLAN as the Configuration Manager server. Check the network configuration for random client-server connection issues happening on the remote VLAN.</p>
0x80240FFF	Occurs when update synchronization fails. It can occur when you are using Windows Server Update Services on its own or when it is integrated with System Center Configuration Manager. If you enable update synchronization before you install hotfix 3095113 , WSUS doesn't recognize the Upgrades classification and instead treats the upgrade like a regular update.	<p>You can prevent this by installing hotfix 3095113 before you enable update synchronization. However, if you have already run into this problem, do the following:</p> <ol style="list-style-type: none"> 1. Disable the Upgrades classification. 2. Install hotfix 3095113. 3. Delete previously synched updates. 4. Enable the Upgrades classification. 5. Perform a full synch. <p>For detailed information on how to run these steps check out How to delete upgrades in WSUS.</p>

0x8007007E	Occurs when update synchronization fails because you do not have hotfix 3095113 installed before you enable update synchronization. Specifically, the CopyToCache operation fails on clients that have already downloaded the upgrade because Windows Server Update Services has bad metadata related to the upgrade. It can occur when you are using standalone Windows Server Update Services or when WSUS is integrated with System Center Configuration Manager.	Use the following steps to repair Windows Server Update Services. You must run these steps on each WSUS server that synched metadata before you installed the hotfix. <ol style="list-style-type: none"> 1. Stop the Windows Update service. Sign in as a user with administrative privileges, and then do the following: <ol style="list-style-type: none"> a. Open Administrative Tools from the Control Panel. b. Double-click Services. c. Find the Windows Update service, right-click it, and then click Stop. If prompted, enter your credentials. 2. Delete all files and folders under c:\Windows\SoftwareDistribution\DataStore. 3. Restart the Windows Update service.
------------	--	--

Other error codes

Error Codes	Cause	Mitigation
0x80070003- 0x20007	This is a failure during SafeOS phase driver installation.	Verify device drivers on the computer, and analyze log files to determine the problem driver.
0x8007025D - 0x2000C	This error occurs if the ISO file's metadata is corrupt.	"Re-download the ISO/Media and re-attempt the upgrade. Alternatively, re-create installation media the Media Creation Tool .
0x80070490 - 0x20007	An incompatible device driver is present.	Verify device drivers on the computer, and analyze log files to determine the problem driver.
0xC1900101 - 0x2000c	An unspecified error occurred in the SafeOS phase during WIM apply. This can be caused by an outdated driver or disk corruption.	Run checkdisk to repair the file system. For more information, see the quick fixes section in this guide. Update drivers on the computer, and select "Download and install updates (recommended)" during the upgrade process. Disconnect devices other than the mouse, keyboard and display.
0xC1900200 - 0x20008	The computer doesn't meet the minimum requirements to download or upgrade to Windows 10.	See Windows 10 Specifications and verify the computer meets minimum requirements. Review logs for compatibility information .

0x80070004 - 0x3000D	This is a problem with data migration during the first boot phase. There are multiple possible causes.	Analyze log files to determine the issue.
0xC1900101 - 0x4001E	Installation failed in the SECOND_BOOT phase with an error during PRE_OOBE operation.	This is a generic error that occurs during the OOBE phase of setup. See the 0xC1900101 section of this guide and review general troubleshooting procedures described in that section.
0x80070005 - 0x4000D	The installation failed in the SECOND_BOOT phase with an error in during MIGRATE_DATA operation. This error indicates that access was denied while attempting to migrate data.	Analyze log files to determine the data point that is reporting access denied.
0x80070004 - 0x50012	Windows Setup failed to open a file.	Analyze log files to determine the data point that is reporting access problems.
0xC190020e 0x80070070 - 0x50011 0x80070070 - 0x50012 0x80070070 - 0x60000	These errors indicate the computer does not have enough free space available to install the upgrade.	<p>To upgrade a computer to Windows 10, it requires 16 GB of free hard drive space for a 32-bit OS, and 20 GB for a 64-bit OS. If there is not enough space, attempt to free up drive space before proceeding with the upgrade.</p> <p>Note: If your device allows it, you can use an external USB drive for the upgrade process. Windows setup will back up the previous version of Windows to a USB external drive. The external drive must be at least 8GB (16GB is recommended). The external drive should be formatted using NTFS. Drives that are formatted in FAT32 may run into errors due to FAT32 file size limitations. USB drives are preferred over SD cards because drivers for SD cards are not migrated if the device does not support Connected Standby.</p>

Modern setup errors

Also see the following sequential list of modern setup (mosetup) error codes with a brief description of the cause.

RESULT CODE	MESSAGE	DESCRIPTION
0XC1900100	MOSETUP_E_VERSION_MISMATCH	An unexpected version of Setup Platform binaries was encountered. Please verify the package contents.
0XC1900101	MOSETUP_E_SETUP_PLATFORM	The Setup Platform has encountered an unspecified error.
0XC1900102	MOSETUP_E_SHUTDOWN_BLOCK	Unable to create or destroy the shutdown block message.

RESULT CODE	MESSAGE	DESCRIPTION
0XC1900103	MOSETUP_E_COMPAT_TIMEOUT	The compatibility issues were not resolved within the required time limit.
0XC1900104	MOSETUP_E_PROCESS_TIMEOUT	The installation process did not complete within the required time limit.
0XC1900105	MOSETUP_E_TEST_MODE	The installation process is being used in a test environment.
0XC1900106	MOSETUP_E_TERMINATE_PROCESS	The installation process was terminated.
0XC1900107	MOSETUP_E_CLEANUP_PENDING	A cleanup operation from a previous installation attempt is still pending. A system reboot is required.
0XC1900108	MOSETUP_E_REPORTING	An error has occurred and the result value must be consolidated for telemetry purposes.
0XC1900109	MOSETUP_E_COMPAT_TERMINATE	The installation process was terminated during the actionable compatibility phase.
0XC190010a	MOSETUP_E_UNKNOWN_CMD_LINE	The installation process was launched with an unknown command line argument.
0XC190010b	MOSETUP_E_INSTALL_IMAGE_NOT_FOUND	The installation image was not found.
0XC190010c	MOSETUP_E_AUTOMATION_INVALID	The provided automation information was invalid.
0XC190010d	MOSETUP_E_INVALID_CMD_LINE	The installation process was launched with an invalid command line argument.
0XC190010e	MOSETUP_E_EULA_ACCEPT_REQUIRED	The installation process requires that the user accept the license agreement.
0XC1900110	MOSETUP_E_EULA_CANCEL	The user has chosen to cancel for license agreement.
0XC1900111	MOSETUP_E_ADVERTISE_CANCEL	The user has chosen to cancel for advertisement.
0XC1900112	MOSETUP_E_TARGET_DRIVE_NOT_FOUND	Could not find a target drive letter.
0XC1900113	MOSETUP_E_EULA_DECLINED	The user has declined the license terms.
0XC190011e	MOSETUP_E_FLIGHTING_BVT	The installation process has been halted for testing purposes.
0XC190011f	MOSETUP_E_PROCESS_CRASHED	The installation process crashed.

RESULT CODE	MESSAGE	DESCRIPTION
0XC1900120	MOSETUP_E_EULA_TIMEOUT	The user has not accepted Eula within the required time limit.
0XC1900121	MOSETUP_E_ADVERTISE_TIMEOUT	The user has not accepted Advertisement within the required time limit.
0XC1900122	MOSETUP_E_DOWNLOADDISKSPACE_TIMEOUT	The download disk space issues were not resolved within the required time limit.
0XC1900123	MOSETUP_E_INSTALLDISKSPACE_TIMEOUT	The install disk space issues were not resolved within the required time limit.
0XC1900124	MOSETUP_E_COMPAT_SYSREQ_TIMEOUT	The minimum requirements compatibility issues were not resolved within the required time limit.
0XC1900125	MOSETUP_E_COMPAT_DOWNLOADREQ_TIMEOUT	The compatibility issues for download were not resolved within the required time limit.
0XC1900126	MOSETUP_E_GATHER_OS_STATE_SIGNATURE	The GatherOsState executable has invalid signature.
0XC1900127	MOSETUP_E_UNINSTALL_ALLOWED_ABORT	The user has chosen to abort Setup to keep Uninstall option active.
0XC1900128	MOSETUP_E_MISSING_TASK	The install cannot continue because a required task is missing.
0XC1900129	MOSETUP_E_UPDATEREQUESTED	A more up-to-date version of setup will be launched to continue installation.
0XC190012f	MOSETUP_E_FINALIZE_ALREADY_REQUESTED	The install cannot continue because a finalize operation was already requested.
0XC1900130	MOSETUP_E_INSTALL_HASH_MISSING	The install cannot continue because the instance hash was not found.
0XC1900131	MOSETUP_E_INSTALL_HASH_MISMATCH	The install cannot continue because the instance hash does not match.
0XC19001df	MOSETUP_E_DISK_FULL	The install cannot continue because the system is out of disk space.
0XC19001e0	MOSETUP_E_GATHER_OS_STATE_FAILED	The GatherOsState executable has failed to execute.
0XC19001e1	MOSETUP_E_PROCESS_SUSPENDED	The installation process was suspended.
0XC19001e2	MOSETUP_E_PREINSTALL_SCRIPT_FAILED	A preinstall script failed to execute or returned an error.

RESULT CODE	MESSAGE	DESCRIPTION
0XC19001e3	MOSETUP_E_PRECOMMIT_SCRIPT_FAILED	A precommit script failed to execute or returned an error.
0XC19001e4	MOSETUP_E_FAILURE_SCRIPT_FAILED	A failure script failed to execute or returned an error.
0XC19001e5	MOSETUP_E_SCRIPT_TIMEOUT	A script exceeded the timeout limit.
0XC1900200	MOSETUP_E_COMPAT_SYSREQ_BLOCK	The system does not pass the minimum requirements to install the update.
0XC1900201	MOSETUP_E_COMPAT_SYSREQ_CANCEL	The user has chosen to cancel because the system does not pass the minimum requirements to install the update.
0XC1900202	MOSETUP_E_COMPAT_DOWNLOADREQ_BLOCK	The system does not pass the minimum requirements to download the update.
0XC1900203	MOSETUP_E_COMPAT_DOWNLOADREQ_CANCEL	The user has chosen to cancel because the system does not pass the minimum requirements to download the update.
0XC1900204	MOSETUP_E_COMPAT_MIGCHOICE_BLOCK	The system does not pass the requirements for desired migration choice.
0XC1900205	MOSETUP_E_COMPAT_MIGCHOICE_CANCEL	The user has chosen to cancel because the system does not pass the requirements for desired migration choice.
0XC1900206	MOSETUP_E_COMPAT_DEVICEREQ_BLOCK	The system does not pass the device scan to install the update.
0XC1900207	MOSETUP_E_COMPAT_DEVICEREQ_CANCEL	The user has chosen to cancel because the system does not pass the device scan to install the update.
0XC1900208	MOSETUP_E_COMPAT_INSTALLREQ_BLOCK	The system does not pass the compat scan to install the update.
0XC1900209	MOSETUP_E_COMPAT_INSTALLREQ_CANCEL	The user has chosen to cancel because the system does not pass the compat scan to install the update.
0XC190020a	MOSETUP_E_COMPAT_RECOVERYREQ_BLOCK	The system does not pass the minimum requirements to recover Windows.
0XC190020b	MOSETUP_E_COMPAT_RECOVERYREQ_CANCEL	The user has chosen to cancel because the system does not pass the minimum requirements to recover Windows.
0XC190020c	MOSETUP_E_DOWNLOADDISKSPACE_BLOCK	The system does not pass the disk space requirements to download the payload.

RESULT CODE	MESSAGE	DESCRIPTION
0XC190020d	MOSETUP_E_DOWNLOADDISKSPACE_CANCEL	The user has chosen to cancel as the device does not have enough disk space to download.
0XC190020e	MOSETUP_E_INSTALLDISKSPACE_BLOCK	The system does not pass the disk space requirements to install the payload.
0XC190020f	MOSETUP_E_INSTALLDISKSPACE_CANCEL	The user has chosen to cancel as the device does not have enough disk space to install.
0XC1900210	MOSETUP_E_COMPAT_SCANONLY	The user has used the setup.exe command line to do scanonly, not to install the OS.
0XC1900211	MOSETUP_E_DOWNLOAD_UNPACK_DISKSPACE_BLOCK	The system does not pass the disk space requirements to download and unpack media.
0XC1900212	MOSETUP_E_DOWNLOAD_UNPACK_DISKSPACE_MULTIARCH_BLOCK	The system does not pass the disk space requirements to download and unpack multi-architecture media.
0XC1900213	MOSETUP_E_NO_OFFER_FOUND	There was no offer found that matches the required criteria.
0XC1900214	MOSETUP_E_UNSUPPORTED_VERSION	This version of the tool is not supported.
0XC1900215	MOSETUP_E_NO_MATCHING_INSTALL_IMAGE	Could not find an install image for this system.
0XC1900216	MOSETUP_E_ROLLBACK_PENDING	Found pending OS rollback operation.
0XC1900220	MOSETUP_E_COMPAT_REPORT_NOT_DISPLAYED	The compatibility report cannot be displayed due to a missing system component.
0XC1900400	MOSETUP_E_UA_VERSION_MISMATCH	An unexpected version of Update Agent client was encountered.
0XC1900401	MOSETUP_E_UA_NO_PACKAGES_TO_DOWNLOAD	No packages to be downloaded.
0XC1900402	MOSETUP_E_UA_UPDATE_CANNOT_BE_MERGED	No packages to be downloaded.
0XC1900403	MOSETUP_E_UA_CORRUPT_PAYLOAD_FILES	Payload files were corrupt.
0XC1900404	MOSETUP_E_UA_BOX_NOT_FOUND	The installation executable was not found.

RESULT CODE	MESSAGE	DESCRIPTION
0XC1900405	MOSETUP_E_UA_BOX_CRASHED	The installation process terminated unexpectedly.

Related topics

[Windows 10 FAQ for IT professionals](#)

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

[Fix Windows Update errors by using the DISM or System Update Readiness tool](#)

Submit Windows 10 upgrade errors using Feedback Hub

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

NOTE

This is a 100 level topic (basic).

See [Resolve Windows 10 upgrade errors](#) for a full list of topics in this article.

In this topic

This topic describes how to submit problems with a Windows 10 upgrade to Microsoft using the Windows 10 Feedback Hub.

About the Feedback Hub

The Feedback Hub app lets you tell Microsoft about any problems you run in to while using Windows 10 and send suggestions to help us improve your Windows experience. Previously, you could only use the Feedback Hub if you were in the Windows Insider Program. Now anyone can use this tool. You can download the Feedback Hub app from the Microsoft Store [here](#).

The Feedback Hub requires Windows 10 or Windows 10 mobile. If you are having problems upgrading from an older version of Windows to Windows 10, you can use the Feedback Hub to submit this information, but you must collect the log files from the legacy operating system and then attach these files to your feedback using a device that is running Windows 10. If you are upgrading to Windows 10 from a previous version of Windows 10, the Feedback Hub will collect log files automatically.

Submit feedback

To submit feedback about a failed Windows 10 upgrade, click the following link: [Feedback Hub](#)

The Feedback Hub will open.

- Under **Tell us about it**, and then under **Summarize your issue**, type **Upgrade failing**.
- Under **Give us more detail**, provide additional information about the failed upgrade, such as:
 - When did the failure occur?
 - Were there any reboots?
 - How many times did the system reboot?
 - How did the upgrade fail?
 - Were any error codes visible?
 - Did the computer fail to a blue screen?
 - Did the computer automatically roll back or did it hang, requiring you to power cycle it before it rolled back?
- Additional details

- What type of security software is installed?
- Is the computer up to date with latest drivers and firmware?
- Are there any external devices connected?
- If you used the link above, the category and subcategory will be automatically selected. If it is not selected, choose **Install and Update** and **Windows Installation**.

You can attach a screenshot or file if desired. This is optional, but can be extremely helpful when diagnosing your upgrade issue. The location of these files is described here: [Windows Setup log files and event logs](#).

Click **Submit** to send your feedback.

See the following example:

Feedback Hub

New feedback

What kind of feedback is it?

Suggestion Problem

Tell us about it

Summarize your issue

Upgrade failing

15/150

Give us more detail (optional)

The computer rebooted twice
There is a hang after the second reboot, requiring a power cycle
The only anti-virus software installed is Windows Defender
The computer has updated BIOS and firmware

267/1000

Will be visible to Microsoft internal users, and partners if needed. Will post as Greg Lindsay.

- Make your title concise and descriptive. This will help others find and upvote your feedback.
- Send one thought per feedback.
- Information about your device, operating system, and applications are automatically included in each reported feedback.

Select a category and subcategory

[Show category suggestions](#)

Install and Update Windows installation

Attachments (optional)

Send attached files and diagnostics to Microsoft along with my feedback. [Learn more](#)

Attach a screenshot Attach a file Recreate my problem

Submit Cancel

After you click Submit, that's all you need to do. Microsoft will receive your feedback and begin analyzing the issue. You can check on your feedback periodically to see what solutions have been provided.

Link to your feedback

After your feedback is submitted, you can email or post links to it by opening the Feedback Hub, clicking My feedback at the top, clicking the feedback item you submitted, clicking **Share**, then copying the short link that is displayed.

Short Link [Copy link](#) [Share](#)

Related topics

[Windows 10 release information](#)

Deploy Windows 10

6/18/2019 • 2 minutes to read • [Edit Online](#)

Windows 10 upgrade options are discussed and information is provided about planning, testing, and managing your production deployment. Procedures are provided to help you with a new deployment of the Windows 10 operating system, or to upgrade from a previous version of Windows to Windows 10. The following sections and topics are available.

TOPIC	DESCRIPTION
Overview of Windows Autopilot	This topic provides an overview of Windows Autopilot deployment, a new zero-touch method for deploying Windows 10 in the enterprise.
Windows 10 upgrade paths	This topic provides information about support for upgrading directly to Windows 10 from a previous operating system.
Windows 10 edition upgrade	This topic provides information about support for upgrading from one edition of Windows 10 to another.
Windows 10 volume license media	This topic provides information about updates to volume licensing media in the current version of Windows 10.
Manage Windows upgrades with Upgrade Readiness	With Upgrade Readiness, enterprises now have the tools to plan and manage the upgrade process end to end, allowing them to adopt new Windows releases more quickly. With Windows diagnostic data enabled, Upgrade Readiness collects system, application, and driver data for analysis. We then identify compatibility issues that can block an upgrade and suggest fixes when they are known to Microsoft. The Upgrade Readiness workflow steps you through the discovery and rationalization process until you have a list of computers that are ready to be upgraded.
Windows 10 deployment test lab	This guide contains instructions to configure a proof of concept (PoC) environment requiring a minimum amount of resources. The guide makes extensive use of Windows PowerShell and Hyper-V. Subsequent companion guides contain steps to deploy Windows 10 using the PoC environment. After completing this guide, additional guides are provided to deploy Windows 10 in the test lab using Microsoft Deployment Toolkit or System Center Configuration Manager .
Plan for Windows 10 deployment	This section describes Windows 10 deployment considerations and provides information to assist in Windows 10 deployment planning.
Deploy Windows 10 with the Microsoft Deployment Toolkit	This guide will walk you through the process of deploying Windows 10 in an enterprise environment using the Microsoft Deployment Toolkit (MDT).

TOPIC	DESCRIPTION
Deploy Windows 10 with System Center 2012 R2 Configuration Manager	If you have Microsoft System Center 2012 R2 Configuration Manager in your environment, you will most likely want to use it to deploy Windows 10. This topic will show you how to set up Configuration Manager for operating system deployment and how to integrate Configuration Manager with the Microsoft Deployment Toolkit (MDT) or.
Windows 10 deployment tools	Learn about available tools to deploy Windows 10, such as the Windows ADK, DISM, USMT, WDS, MDT, Windows PE and more.
How to install fonts that are missing after upgrading to Windows 10	Windows 10 introduced changes to the fonts that are included in the image by default. Learn how to install additional fonts from Optional features after you install Windows 10 or upgrade from a previous version.

Related topics

[Modern Desktop Deployment Center](#)

Overview of Windows Autopilot

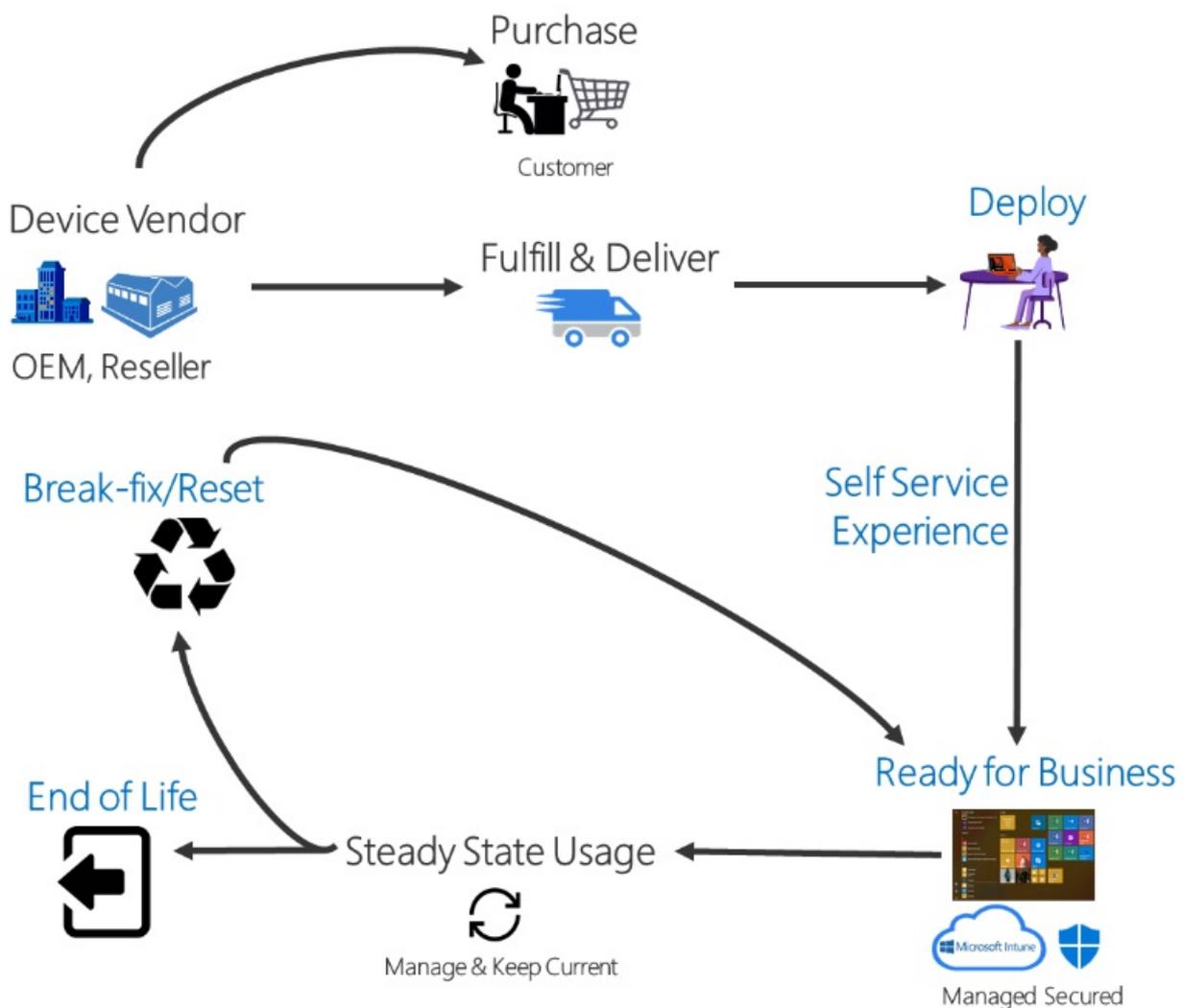
6/18/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Windows Autopilot is a collection of technologies used to set up and pre-configure new devices, getting them ready for productive use. You can also use Windows Autopilot to reset, repurpose and recover devices. This solution enables an IT department to achieve the above with little to no infrastructure to manage, with a process that's easy and simple.

Windows Autopilot is designed to simplify all parts of the lifecycle of Windows devices, for both IT and end users, from initial deployment through the eventual end of life. Leveraging cloud-based services, it can reduce the overall costs for deploying, managing, and retiring devices by reducing the amount of time that IT needs to spend on these processes and the amount of infrastructure that they need to maintain, while ensuring ease of use for all types of end users. See the following diagram:



When initially deploying new Windows devices, Windows Autopilot leverages the OEM-optimized version of Windows 10 that is preinstalled on the device, saving organizations the effort of having to maintain custom images and drivers for every model of device being used. Instead of re-imaging the device, your existing Windows 10 installation can be transformed into a "business-ready" state, applying settings and policies, installing apps, and

even changing the edition of Windows 10 being used (e.g. from Windows 10 Pro to Windows 10 Enterprise) to support advanced features.

Once deployed, Windows 10 devices can be managed by tools such as Microsoft Intune, Windows Update for Business, System Center Configuration Manager, and other similar tools. Windows Autopilot can also be used to re-purpose a device by leveraging Windows Autopilot Reset to quickly prepare a device for a new user, or in break/fix scenarios to enable a device to quickly be brought back to a business-ready state.

Windows Autopilot enables you to:

- Automatically join devices to Azure Active Directory (Azure AD) or Active Directory (via Hybrid Azure AD Join). See [Introduction to device management in Azure Active Directory](#) for more information about the differences between these two join options.
- Auto-enroll devices into MDM services, such as Microsoft Intune (*Requires an Azure AD Premium subscription*).
- Restrict the Administrator account creation.
- Create and auto-assign devices to configuration groups based on a device's profile.
- Customize OOB content specific to the organization.

Windows Autopilot walkthrough

The following video shows the process of setting up Windows Autopilot:

<https://www.youtube.com/embed/4K4hC5NchbE>

Benefits of Windows Autopilot

Traditionally, IT pros spend a lot of time building and customizing images that will later be deployed to devices. Windows Autopilot introduces a new approach.

From the user's perspective, it only takes a few simple operations to make their device ready to use.

From the IT pro's perspective, the only interaction required from the end user is to connect to a network and to verify their credentials. Everything beyond that is automated.

Requirements

Windows 10 version 1703 or higher is required to use Windows Autopilot. See [Windows Autopilot requirements](#) for detailed information on software, configuration, network, and licensing requirements.

Related topics

[Enroll Windows devices in Intune by using Windows Autopilot](#)
[Windows Autopilot scenarios and capabilities](#)

	Mobile Enterpri se								D	✓
--	--------------------------	--	--	--	--	--	--	--	---	---

Related Topics

[Windows 10 deployment scenarios](#)

[Windows upgrade and migration considerations](#)

[Windows 10 edition upgrade](#)

Windows 10 edition upgrade

6/14/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

With Windows 10, you can quickly upgrade from one edition of Windows 10 to another, provided the upgrade path is supported. For information on what edition of Windows 10 is right for you, see [Compare Windows 10 Editions](#). For a comprehensive list of all possible upgrade paths to Windows 10, see [Windows 10 upgrade paths](#). Downgrading the edition of Windows is discussed in the [License expiration](#) section on this page.

For a list of operating systems that qualify for the Windows 10 Pro Upgrade or Windows 10 Enterprise Upgrade through Microsoft Volume Licensing, see [Windows 10 Qualifying Operating Systems](#).

The following table shows the methods and paths available to change the edition of Windows 10 that is running on your computer. **Note:** The reboot requirement for upgrading from Pro to Enterprise was removed in version 1607.

Note: Although it isn't displayed yet in the table, edition upgrade is also possible using [edition upgrade policy](#) in System Center Configuration Manager.

 (X) = not supported

 (green checkmark) = supported, reboot required

 (blue checkmark) = supported, no reboot required

EDITION UPGRADE	USING MOBILE DEVICE MANAGEMENT (MDM)	USING A PROVISIONING PACKAGE	USING A COMMAND-LINE TOOL	USING MICROSOFT STORE FOR BUSINESS OR PC	ENTERING A PRODUCT KEY MANUALLY	PURCHASING A LICENSE FROM THE MICROSOFT STORE
Home > Pro						
Home > Pro for Workstations						
Home > Pro Education						
Home > Education						
Pro > Pro for Workstations				 (MSfB)		
Pro > Pro Education				 (MSfB)		

EDITION UPGRADE	USING MOBILE DEVICE MANAGEMENT (MDM)	USING A PROVISIONING PACKAGE	USING A COMMAND- LINE TOOL	USING MICROSOFT STORE FOR BUSINESS OR PC	ENTERING A PRODUCT KEY MANUALLY	PURCHASING A LICENSE FROM THE MICROSOFT STORE
Pro > Education	✓	✓	✓	✓ (MSfB)	✓	✗
Pro > Enterprise	✓	✓	✓	✓ (1703 - PC) (1709 - MSfB)	✓	✗
Pro for Workstations > Pro Education	✓	✓	✓	✓ (MSfB)	✓	✗
Pro for Workstations > Education	✓	✓	✓	✓ (MSfB)	✓	✗
Pro for Workstations > Enterprise	✓	✓	✓	✓ (1703 - PC) (1709 - MSfB)	✓	✗
Pro Education > Education	✓	✓	✓	✓ (MSfB)	✓	✗
Enterprise > Education	✓	✓	✓	✓ (MSfB)	✓	✗
Mobile > Mobile Enterprise	✓	✓	✗	✗	✗	✗

NOTE

- For information about upgrade paths in Windows 10 in S mode (for Pro or Education), check out [Windows 10 Pro/Enterprise in S mode](#)
- Each desktop edition in the table also has an N and KN SKU. These editions have had media-related functionality removed. Devices with N or KN SKUs installed can be upgraded to corresponding N or KN SKUs using the same methods.
- Due to [naming changes](#) the term LTSB might still be displayed in some products. This name will change to LTSC with subsequent feature updates.

Upgrade using mobile device management (MDM)

- To upgrade desktop editions of Windows 10 using MDM, you'll need to enter the product key for the upgraded edition in the **UpgradeEditionWithProductKey** policy setting of the **WindowsLicensing** CSP. For more info, see [WindowsLicensing CSP](#).
- To upgrade mobile editions of Windows 10 using MDM, you'll need to enter the product key for the upgraded edition in the **UpgradeEditionWithLicense** policy setting of the **WindowsLicensing** CSP. For

more info, see [WindowsLicensing CSP](#).

Upgrade using a provisioning package

Use Windows Configuration Designer to create a provisioning package to upgrade a desktop edition or mobile edition of Windows 10. To get started, [install Windows Configuration Designer from the Microsoft Store](#).

- To create a provisioning package for upgrading desktop editions of Windows 10, go to **Runtime settings > EditionUpgrade > UpgradeEditionWithProductKey** in the **Available customizations** panel in Windows ICD and enter the product key for the upgraded edition.
- To create a provisioning package for upgrading mobile editions of Windows 10, go to **Runtime settings > EditionUpgrade > UpgradeEditionWithLicense** in the **Available customizations** panel in Windows ICD and enter the product key for the upgraded edition.

For more info about Windows Configuration Designer, see these topics:

- [Create a provisioning package for Windows 10](#)
- [Apply a provisioning package](#)

Upgrade using a command-line tool

You can run the changepk.exe command-line tool to upgrade devices to a supported edition of Windows 10:

```
changepk.exe /ProductKey <enter your new product key here>
```

You can also upgrade using slmgr.vbs and a [KMS client setup key](#). For example, the following command will upgrade to Windows 10 Enterprise.

```
Cscript.exe c:\windows\system32\slmgr.vbs /ipk NPPR9-FWDCX-D2C8J-H872K-2YT43
```

Upgrade by manually entering a product key

If you are upgrading only a few devices, you may want to enter a product key for the upgraded edition manually.

To manually enter a product key

1. From either the Start menu or the Start screen, type 'Activation' and click on the Activation shortcut.
2. Click **Change product key**.
3. Enter your product key.
4. Follow the on-screen instructions.

Upgrade by purchasing a license from the Microsoft Store

If you do not have a product key, you can upgrade your edition of Windows 10 through the Microsoft Store.

To upgrade through the Microsoft Store

1. From either the **Start** menu or the **Start** screen, type 'Activation' and click on the Activation shortcut.
2. Click **Go to Store**.
3. Follow the on-screen instructions.

Note

If you are a Windows 10 Home N or Windows 10 Home KN user and have trouble finding your applicable upgrade in the Microsoft Store, click [here](#).

License expiration

Volume license customers whose license has expired will need to change the edition of Windows 10 to an edition with an active license. Switching to a downgraded edition of Windows 10 is possible using the same methods that were used to perform an edition upgrade. If the downgrade path is supported, then your apps and settings can be migrated from the current edition. If a path is not supported, then a clean install is required.

Downgrading from any edition of Windows 10 to Windows 7, 8, or 8.1 by entering a different product key is not supported. You also cannot downgrade from a later version to an earlier version of the same edition (Ex: Windows 10 Pro 1709 to 1703) unless the rollback process is used. This topic does not discuss version downgrades.

Note: If you are using [Windows 10 Enterprise Subscription Activation](#) and a license expires, devices will automatically revert to the original edition when the grace period expires.

Scenario example

Downgrading from Enterprise

- Original edition: **Professional OEM**
- Upgrade edition: **Enterprise**
- Valid downgrade paths: **Pro, Pro for Workstations, Pro Education, Education**

You can move directly from Enterprise to any valid destination edition. In this example, downgrading to Pro for Workstations, Pro Education, or Education requires an additional activation key to supersede the firmware-embedded Pro key. In all cases, you must comply with [Microsoft License Terms](#). If you are a volume license customer, refer to the [Microsoft Volume Licensing Reference Guide](#).

Supported Windows 10 downgrade paths

✓ = Supported downgrade path

S = Supported; Not considered a downgrade or an upgrade

[blank] = Not supported or not a downgrade

Destination edition		Home	Pro	Pro for Workstations	Pro Education	Education	Enterprise LTSC	Enterprise
Starting edition	Home							
	Pro							
	Pro for Workstations							
	Pro Education							
	Education		✓	✓	✓			S
	Enterprise LTSC							

Enterprise		✓	✓	✓	S		
------------	--	---	---	---	---	--	--

Windows 10 LTSC/LTSB: Due to [naming changes](#), product versions that display Windows 10 LTSB will be replaced with Windows 10 LTSC in subsequent feature updates. The term LTSC is used here to refer to all long term servicing versions.

Windows N/KN: Windows "N" and "KN" SKUs follow the same rules shown above.

Some slightly more complex scenarios are not represented by the table above. For example, you can perform an upgrade from Pro to Pro for Workstation on a computer with an embedded Pro key using a Pro for Workstation license key, and then later downgrade this computer back to Pro with the firmware-embedded key. The downgrade is allowed but only because the pre-installed OS is Pro.

Related topics

[Windows 10 upgrade paths](#)

[Windows 10 volume license media](#)

[Windows 10 Subscription Activation](#)

Windows 10 volume license media

6/18/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10

With each release of Windows 10, volume license media is made available on the [Volume Licensing Service Center](#) (VLSC) and other relevant channels such as Windows Update for Business, Windows Server Update Services (WSUS), and Visual Studio Subscriptions. This topic provides a description of volume license media, and describes some of the changes that have been implemented with the current release of Windows 10.

Windows 10 media

To download Windows 10 installation media from the VLSC, use the product search filter to find "Windows 10." A list of products will be displayed. The page then allows you to use your search results to download products, view keys, and view product and key descriptions.

When you select a product, for example "Windows 10 Enterprise" or "Windows 10 Education", you can then choose the specific release by clicking **Download** and choosing the **Download Method**, **Language**, and **Operating system Type** (bitness).

If you do not see a Windows 10 release available in the list of downloads, verify the [release date](#).

In Windows 10, version 1709 the packaging of volume licensing media and upgrade packages is different than it has been for previous releases. Instead of having separate media and packages for Windows 10 Pro (volume licensing version), Windows 10 Enterprise, and Windows 10 Education, all three are bundled together. The following section explains this change.

Windows 10, version 1709

Windows 10, version 1709 is available starting on 10/17/2017 in all relevant distribution channels. Note: An updated [Windows ADK for Windows 10](#) is also available.

For ISOs that you download from the VLSC or Visual Studio Subscriptions, you can still search for the individual Windows editions. However, each of these editions (Windows 10 Pro, Windows 10 Enterprise, and Windows 10 Education) will point to the same ISO file, so you only need to download the ISO once. A single Windows image (WIM) file is included in the ISO that contains all the volume licensing images:

Image Name	Index
Windows 10 Education	1
Windows 10 Education N	2
Windows 10 Enterprise	3
Windows 10 Enterprise N	4
Windows 10 Pro	5
Windows 10 Pro N	6

When using the contents of these ISOs with tools such as the Microsoft Deployment Toolkit or System Center Configuration Manager, make sure you select the appropriate image index in any task sequences that you create or

update.

For packages published to Windows Server Update Services (WSUS), you'll also notice the change because, instead of having separate packages for each Windows edition, there will be just one package:

TITLE	CLASSIFICATION	DESCRIPTION
Feature update to Windows 10, version 1709, <language>	Upgrades	Package to upgrade Windows 10 Pro (VL), Windows 10 Enterprise, or Windows 10 Education to version 1709
Windows 7 and 8.1 upgrade to Windows 10, version 1709, <language>	Upgrades	Package to upgrade Windows 7 Professional (VL), Windows 7 Enterprise, Windows 8.1 Professional (VL), or Windows 8.1 Enterprise to Windows 10 1709

When you approve one of these packages, it applies to all of the editions.

This Semi-Annual Channel release of Windows 10 continues the Windows as a service methodology. For more information about implementing Windows as a service in your organization in order to stay up to date with Windows, see [Update Windows 10 in the enterprise](#).

Language packs

- **Windows 10 versions 1507 and 1511:** you can select **Windows 10 Enterprise Language Pack**, click **Download** and then select **English** and **64-bit** to see these downloads.
- **Windows 10 1607 and later:** you must select **Multilanguage** from the drop-down list of languages.

See the following example for Windows 10, version 1709:

Files	OS	Size	Format ?	Download
Windows 10 Language Packs (Released Jul '16) 32/64 Bit MultiLanguage	32/64 bit	3962 MB	ISO	↓
Windows 10 Language Packs (Released Mar '17) 32/64 Bit MultiLanguage	32/64 bit	4346 MB	ISO	↓
Windows 10 Language Packs (Released Sept'17) 32/64 Bit MultiLanguage	32/64 bit	7099 MB	ISO	↓

Features on demand

[Features on demand](#) can be downloaded by searching for "**Windows 10 Enterprise Features on Demand**" and then following the same download process that is described above.

Features on demand is a method for adding features to your Windows 10 image that aren't included in the base operating system image.

Related topics

[Microsoft Volume Licensing Service Center \(VLSC\) User Guide](#)

[Volume Activation for Windows 10](#)

[Plan for volume activation](#)

[VLSC downloads FAQ](#)

[Download and burn an ISO file on the volume licensing site \(VLSC\)](#)

Windows 10 in S mode - What is it?

6/18/2019 • 2 minutes to read • [Edit Online](#)

S mode is an evolution of the S SKU introduced with Windows 10 April 2018 Update. It's a configuration that's available on all Windows Editions when enabled at the time of manufacturing. The edition of Windows can be upgraded at any time as shown below. However, the switch from S mode is a onetime switch and can only be undone by a wipe and reload of the OS.

Configuration & Features
Non-Store applications
Domain Join on premise
Azure AD domain join
Windows Store Apps (incl. Win32 Centennial Apps)
OneDrive setup and sync automatically (req MSA)
Microsoft default apps set for default files
Windows Update for Business
Windows Store for Business
Mobile Device Management (MDM)
Bitlocker
Enterprise state roaming with Azure AD
Shared PC Configuration

Other
Microsoft Edge/IE search default: Bing and designated regional search providers
Switch to non S mode (through Windows Store)

*not exhaustive, illustrative to outline configuration differences

Home S	Home
	●
●	●
●	Configurable
●	Configurable
Limited	Limited

Pro S	Pro
	●
	●
●	●
●	●
●	Configurable
●	Configurable
●	●
●	●
●	●
●	●
●	●
●	●
●	●

Home S	Home
●	Configurable
●	Switch →

Pro S	Pro
●	Configurable
●	Switch →

S mode Configuration

- The Windows you know
- The best of the cloud *and* full featured apps
- Designed for Modern devices
- Users can switch to non S mode

S mode key features

Microsoft-verified security

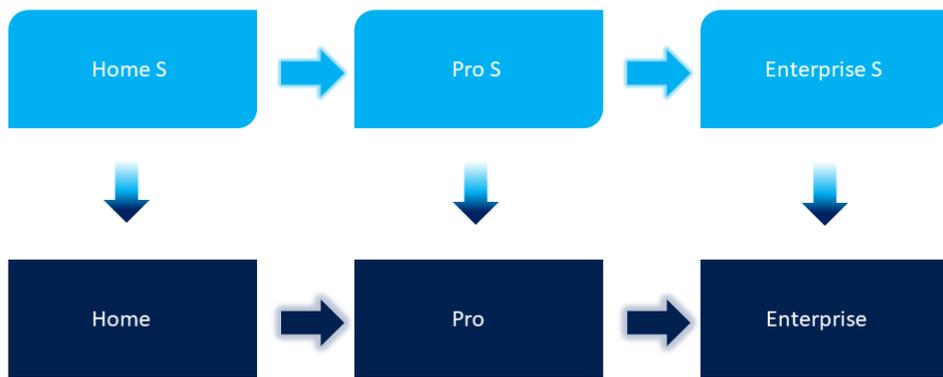
With Windows 10 in S mode, you'll find your favorite applications, such as Office, Evernote, and Spotify in the Microsoft Store where they're Microsoft-verified for security. You can also feel secure when you're online. Microsoft Edge, your default browser, gives you protection against phishing and socially engineered malware.

Performance that lasts

Start-ups are quick, and S mode is built to keep them that way. With Microsoft Edge as your browser, your online experience is fast and secure. Plus, you'll enjoy a smooth, responsive experience, whether you're streaming HD video, opening apps, or being productive on the go.

Choice and flexibility

Save your files to your favorite cloud, like OneDrive or Dropbox, and access them from any device you choose. Browse the Microsoft Store for thousands of apps, and if you don't find exactly what you want, you can easily [switch out of S mode](#) to Windows 10 Home, Pro, or Enterprise editions at any time and search the web for more choices, as shown below.



Deployment

Windows 10 in S mode is built for [modern management](#) which means using [Windows Autopilot](#). Windows Autopilot lets you deploy the device directly to a user without IT having to touch the physical device. Instead of manually deploying a custom image, Windows Autopilot will start with a generic PC that can only be used to join the company domain; policies are then deployed automatically through mobile device management to customize the device to the user and the desired environment. Devices are shipped in S mode; you can either keep them in S mode or use Windows Autopilot to switch the device out of S mode during the first run process or later using mobile device management, if desired.

Keep line of business apps functioning with Desktop Bridge

Worried about your line of business apps not working in S mode? [Desktop Bridge](#) enables you to convert your line of business apps to a packaged app with UWP manifest. After testing and validating you can distribute the app through the Microsoft Store, making it ideal for Windows 10 in S mode.

Repackage Win32 apps into the MSIX format

The [MSIX Packaging Tool](#), available from the Microsoft Store, enables you to repackage existing Win32 applications to the MSIX format. You can run your desktop installers through this tool interactively and obtain an MSIX package that you can install on your device and upload to the Microsoft Store. This is another way to get your apps ready to run on Windows 10 in S mode.

Related links

- [Consumer applications for S mode](#)
- [S mode devices](#)
- [Windows Defender Application Control deployment guide](#)
- [Windows Defender Advanced Threat Protection](#)

Switch to Windows 10 Pro or Enterprise from S mode

6/18/2019 • 4 minutes to read • [Edit Online](#)

We recommend staying in S mode. However, in some limited scenarios, you might need to switch to Windows 10 Pro, Home, or Enterprise (not in S mode). You can switch devices running Windows 10, version 1709 or later.

A number of other transformations are possible depending on which version and edition of Windows 10 you are starting with. Depending on the details, you might *switch* between S mode and the ordinary version or *convert* between different editions while staying in or out of S mode. The following quick reference table summarizes all of the switches or conversions that are supported by various means:

IF A DEVICE IS RUNNING THIS VERSION OF WINDOWS 10	AND THIS EDITION OF WINDOWS 10	THEN YOU CAN SWITCH OR CONVERT IT TO THIS EDITION OF WINDOWS 10 BY THESE METHODS:		
		Store for Education (switch/convert all devices in your tenant)	Microsoft Store (switch/convert one device at a time)	Intune (switch/convert any number of devices selected by admin)
Windows 10, version 1709	Pro in S mode	Pro EDU	Pro	Not by this method
	Pro	Pro EDU	Not by any method	Not by any method
	Home	Not by any method	Not by any method	Not by any method
Windows 10, version 1803	Pro in S mode	Pro EDU in S mode	Pro	Not by this method
	Pro	Pro EDU	Not by any method	Not by any method
	Home in S mode	Not by any method	Home	Not by this method
	Home	Not by any method	Not by any method	Not by any method
Windows 10, version 1809	Pro in S mode	Pro EDU in S mode	Pro	Pro
	Pro	Pro EDU	Not by any method	Not by any method
	Home in S mode	Not by any method	Home	Home
	Home	Not by any method	Not by any method	Not by any method

Use the following information to switch to Windows 10 Pro through the Microsoft Store.

IMPORTANT

While it's free to switch to Windows 10 Pro, it's not reversible. The only way to rollback this kind of switch is through a [bare-metal recovery \(BMR\)](#) reset. This restores a Windows device to the factory state, even if the user needs to replace the hard drive or completely wipe the drive clean. If a device is switched out of S mode via the Microsoft Store, it will remain out of S mode even after the device is reset.

Switch one device through the Microsoft Store

Use the following information to switch to Windows 10 Pro through the Microsoft Store or by navigating to **Settings** and then **Activation** on the device.

Note these differences affecting switching modes in various releases of Windows 10:

- In Windows 10, version 1709, you can switch devices one at a time from Windows 10 Pro in S mode to Windows 10 Pro by using the Microsoft Store or **Settings**. No other switches are possible.
- In Windows 10, version 1803, you can switch devices running any S mode edition to the equivalent non-S mode edition one at a time by using the Microsoft Store or **Settings**.
- Windows 10, version 1809, you can switch devices running any S mode edition to the equivalent non-S mode edition one at a time by using the Microsoft Store, **Settings**, or you can switch multiple devices in bulk by using Intune. You can also block users from switching devices themselves.

1. Sign into the Microsoft Store using your Microsoft account.
2. Search for "S mode".
3. In the offer, select **Buy**, **Get**, or **Learn more**.

You'll be prompted to save your files before the switch starts. Follow the prompts to switch to Windows 10 Pro.

Switch one or more devices by using Microsoft Intune

Starting with Windows 10, version 1809, if you need to switch multiple devices in your environment from Windows 10 Pro in S mode to Windows 10 Pro, you can use Microsoft Intune or any other supported mobile device management software. You can configure devices to switch out of S mode during OOBE or post-OOBE - this gives you flexibility to manage Windows 10 in S mode devices at any point during the device lifecycle.

1. Start Microsoft Intune.
2. Navigate to **Device configuration > Profiles > Windows 10 and later > Edition upgrade and mode switch**.
3. Follow the instructions to complete the switch.

Block users from switching

You can control which devices or users can use the Microsoft Store to switch out of S mode in Windows 10. To set this, go to **Device configuration > Profiles > Windows 10 and later > Edition upgrade and mode switch in Microsoft Intune**, and then choose **Keep in S mode**.

S mode management with CSPs

In addition to using Microsoft Intune or another modern device management tool to manage S mode, you can also use the [WindowsLicensing](#) configuration service provider (CSP). In Windows 10, version 1809, we added S mode functionality that lets you switch devices, block devices from switching, and check the status (whether a device is in S mode).

Related topics

[FAQs](#)

[Compare Windows 10 editions](#)

[Windows 10 Pro Education](#)

[Introduction to Microsoft Intune in the Azure portal](#)

Step by step guide: Configure a test lab to deploy Windows 10

6/18/2019 • 47 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This guide contains instructions to configure a proof of concept (PoC) environment requiring a minimum amount of resources. The guide makes extensive use of Windows PowerShell and Hyper-V. Subsequent companion guides contain steps to deploy Windows 10 using the PoC environment. After completing this guide, see the following Windows 10 PoC deployment guides:

- [Step by step: Deploy Windows 10 in a test lab using MDT](#)
- [Step by step: Deploy Windows 10 in a test lab using System Center Configuration Manager](#)

The PoC deployment guides are intended to provide a demonstration of Windows 10 deployment tools and processes for IT professionals that are not familiar with these tools, and those that are interested in setting up a proof of concept environment. The instructions in this guide should not be used in a production setting, and are not meant to replace the instructions found in production deployment guidance.

Approximately 3 hours are required to configure the PoC environment. You will need a Hyper-V capable computer running Windows 8.1 or later with at least 16GB of RAM. Detailed [requirements](#) are provided below. You will also need to have a [Microsoft account](#) to use for downloading evaluation software.

Windows PowerShell commands are provided to set up the PoC environment quickly. You do not need to be an expert in Windows PowerShell to complete the steps in the guide, however you are required to customize some commands to your environment.

Instructions to "type" Windows PowerShell commands provided in this guide can be followed literally by typing the commands, but the preferred method is to copy and paste these commands.

A Windows PowerShell window can be used to run all commands in this guide. However, when commands are specified for a command prompt, you must either type CMD at the Windows PowerShell prompt to enter the command prompt, or preface the command with "cmd /c", or if desired you can escape special characters in the command using the back-tick character (`). In most cases, the simplest thing is to type cmd and enter a command prompt, type the necessary commands, then type "exit" to return to Windows PowerShell.

Hyper-V is installed, configured and used extensively in this guide. If you are not familiar with Hyper-V, review the [terminology](#) used in this guide before starting.

In this guide

This guide contains instructions for three general procedures: Install Hyper-V, configure Hyper-V, and configure VMs. If you already have a computer running Hyper-V, you can use this computer and skip the first procedure. In this case, your virtual switch settings must be modified to match those used in this guide, or the steps in this guide can be modified to use your existing Hyper-V settings.

After completing the instructions in this guide, you will have a PoC environment that enables you to test Windows 10 deployment procedures by following instructions in companion guides that are written to use the PoC environment. Links are provided to download trial versions of Windows Server 2012, Windows 10 Enterprise,

and all deployment tools necessary to complete the lab.

Topics and procedures in this guide are summarized in the following table. An estimate of the time required to complete each procedure is also provided. Time required to complete procedures will vary depending on the resources available to the Hyper-V host and assigned to VMs, such as processor speed, memory allocation, disk speed, and network speed.

Topic	Description	Time
Hardware and software requirements	Prerequisites to complete this guide.	Informational
Lab setup	A description and diagram of the PoC environment.	Informational
Configure the PoC environment	Parent topic for procedures.	Informational
Verify support and install Hyper-V	Verify that installation of Hyper-V is supported, and install the Hyper-V server role.	10 minutes
Download VHD and ISO files	Download evaluation versions of Windows Server 2012 R2 and Windows 10 and prepare these files to be used on the Hyper-V host.	30 minutes
Convert PC to VM	Convert a physical computer on your network to a VM hosted in Hyper-V.	30 minutes
Resize VHD	Increase the storage capacity for one of the Windows Server VMs.	5 minutes
Configure Hyper-V	Create virtual switches, determine available RAM for virtual machines, and add virtual machines.	15 minutes
Configure service and user accounts	Start virtual machines and configure all services and settings.	60 minutes
Configure VMs	Start virtual machines and configure all services and settings.	60 minutes
Appendix A: Verify the configuration	Verify and troubleshoot network connectivity and services in the PoC environment.	30 minutes
Appendix B: Terminology in this guide	Terms used in this guide.	Informational

Hardware and software requirements

One computer that meets the hardware and software specifications below is required to complete the guide; A second computer is recommended to validate the upgrade process.

- **Computer 1:** the computer you will use to run Hyper-V and host virtual machines. This computer should have 16 GB or more of installed RAM and a multi-core processor.
- **Computer 2:** a client computer from your corporate network. It is shadow-copied to create a VM that can be added to the PoC environment, enabling you to test a mirror image of a computer on your network. If you do not have a computer to use for this simulation, you can download an evaluation VHD and use it to represent this computer. Subsequent guides use this computer to simulate Windows 10 replace and refresh scenarios, so the VM is required even if you cannot create this VM using computer 2.

Hardware requirements are displayed below:

	Computer 1 (required)	Computer 2 (recommended)
Role	Hyper-V host	Client computer

Description	This computer will run Hyper-V, the Hyper-V management tools, and the Hyper-V Windows PowerShell module.	This computer is a Windows 7 or Windows 8/8.1 client on your corporate network that will be converted to a VM to demonstrate the upgrade process.
OS	Windows 8.1/10 or Windows Server 2012/2012 R2/2016*	Windows 7 or a later
Edition	Enterprise, Professional, or Education	Any
Architecture	64-bit	Any <i>Note: Retaining applications and settings requires that architecture (32 or 64-bit) is the same before and after the upgrade.</i>
RAM	8 GB RAM (16 GB recommended) to test Windows 10 deployment with MDT. 16 GB RAM to test Windows 10 deployment with System Center Configuration Manager.	Any
Disk	200 GB available hard disk space, any format.	Any size, MBR formatted.
CPU	SLAT-Capable CPU	Any
Network	Internet connection	Any

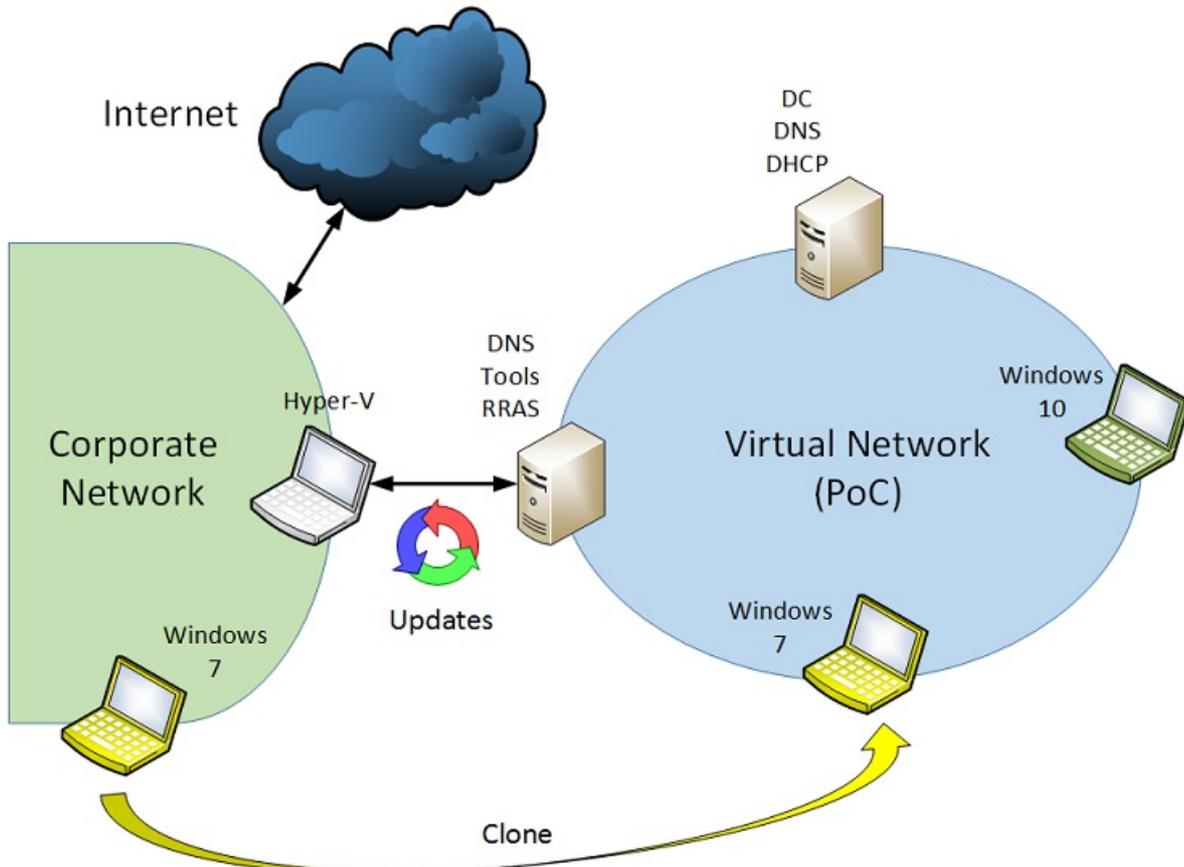
*

The Hyper-V server role can also be installed on a computer running Windows Server 2008 R2. However, the Windows PowerShell module for Hyper-V is not available on Windows Server 2008 R2, therefore you cannot use many of the steps provided in this guide to configure Hyper-V. To manage Hyper-V on Windows Server 2008 R2, you can use Hyper-V WMI, or you can use the Hyper-V Manager console. Providing all steps in this guide as Hyper-V WMI or as 2008 R2 Hyper-V Manager procedures is beyond the scope of the guide.

The Hyper-V role cannot be installed on Windows 7 or earlier versions of Windows.

Lab setup

The lab architecture is summarized in the following diagram:



- Computer 1 is configured to host four VMs on a private, PoC network.
 - Two VMs are running Windows Server 2012 R2 with required network services and tools installed.
 - Two VMs are client systems: One VM is intended to mirror a host on your corporate network (computer 2) and one VM is running Windows 10 Enterprise to demonstrate the hardware replacement scenario.

If you have an existing Hyper-V host, you can use this host and skip the Hyper-V installation section in this guide.

The two Windows Server VMs can be combined into a single VM to conserve RAM and disk space if required. However, instructions in this guide assume two server systems are used. Using two servers enables Active Directory Domain Services and DHCP to be installed on a server that is not directly connected to the corporate network. This mitigates the risk of clients on the corporate network receiving DHCP leases from the PoC network (i.e. "rogue" DHCP), and limits NETBIOS service broadcasts.

Configure the PoC environment

Hint: Before you begin, ensure that Windows PowerShell is pinned to the taskbar for easy access. If the Hyper-V host is running Windows Server then Windows PowerShell is automatically pinned to the taskbar. To pin Windows PowerShell to the taskbar on Windows 8.1 or Windows 10: Click **Start**, type **power**, right click **Windows PowerShell**, and then click **Pin to taskbar**. After Windows PowerShell is pinned to the taskbar, you can open an elevated Windows PowerShell prompt by right-clicking the icon on the taskbar and then clicking **Run as Administrator**.

Procedures in this section

[Verify support and install Hyper-V](#)

[Download VHD and ISO files](#)

[Convert PC to VM](#)

[Resize VHD](#)

[Configure Hyper-V](#)

[Configure VMs](#)

Verify support and install Hyper-V

Starting with Windows 8, the host computer's microprocessor must support second level address translation (SLAT) to install Hyper-V. See [Hyper-V: List of SLAT-Capable CPUs for Hosts](#) for more information.

1. To verify your computer supports SLAT, open an administrator command prompt, type **systeminfo**, press ENTER, and review the section displayed at the bottom of the output, next to Hyper-V Requirements. See the following example:

```
C:\>systeminfo

...
Hyper-V Requirements:      VM Monitor Mode Extensions: Yes
                           Virtualization Enabled In Firmware: Yes
                           Second Level Address Translation: Yes
                           Data Execution Prevention Available: Yes
```

In this example, the computer supports SLAT and Hyper-V.

If one or more requirements are evaluated as **No** then the computer does not support installing Hyper-V. However, if only the virtualization setting is incompatible, you might be able to enable virtualization in the BIOS and change the **Virtualization Enabled In Firmware** setting from **No** to **Yes**. The location of this

setting will depend on the manufacturer and BIOS version, but is typically found associated with the BIOS security settings.

You can also identify Hyper-V support using [tools](#) provided by the processor manufacturer, the [msinfo32](#) tool, or you can download the [coreinfo](#) utility and run it, as shown in the following example:

```
C:\>coreinfo -v

Coreinfo v3.31 - Dump information on system CPU and memory topology
Copyright (C) 2008-2014 Mark Russinovich
Sysinternals - www.sysinternals.com

Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
Intel64 Family 6 Model 42 Stepping 7, GenuineIntel
Microcode signature: 0000001B
HYPERVISOR      -      Hypervisor is present
VMX             *      Supports Intel hardware-assisted virtualization
EPT            *      Supports Intel extended page tables (SLAT)
```

Note: A 64-bit operating system is required to run Hyper-V.

2. The Hyper-V feature is not installed by default. To install it, open an elevated Windows PowerShell window and type the following command:

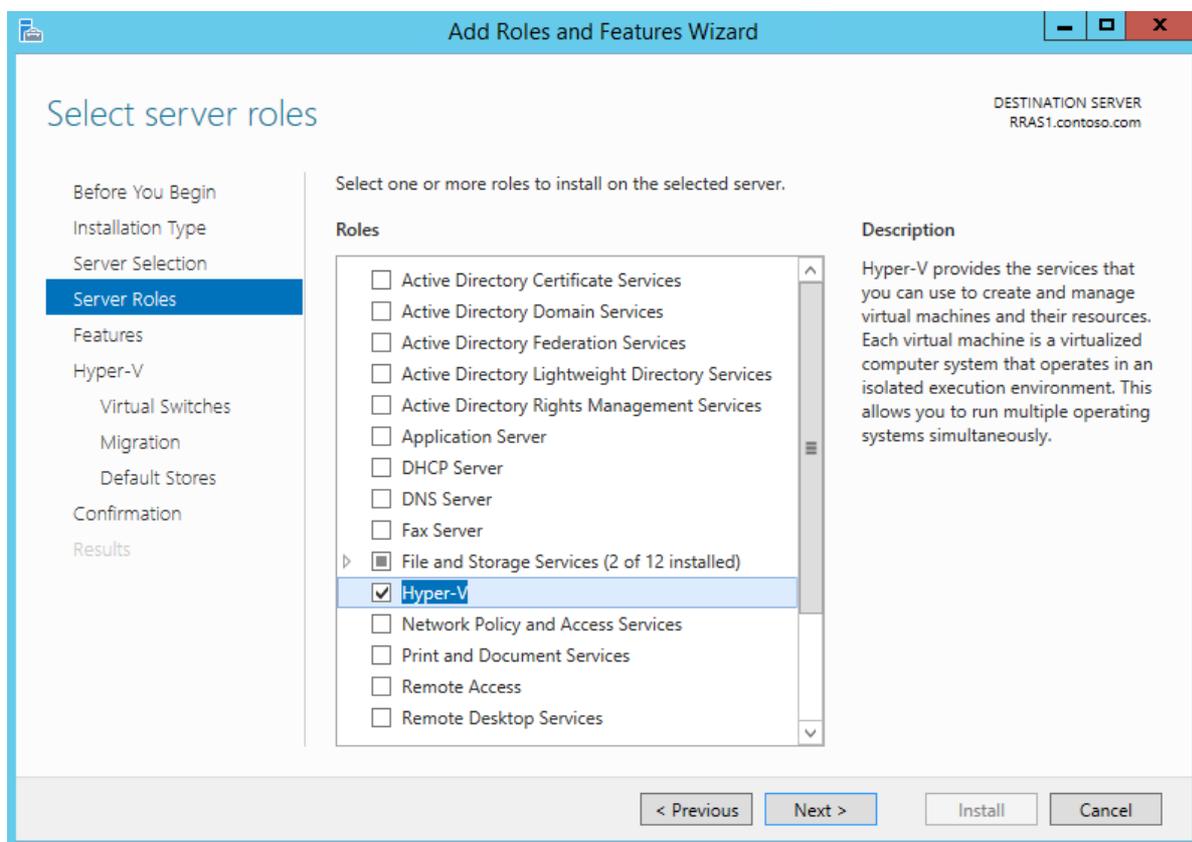
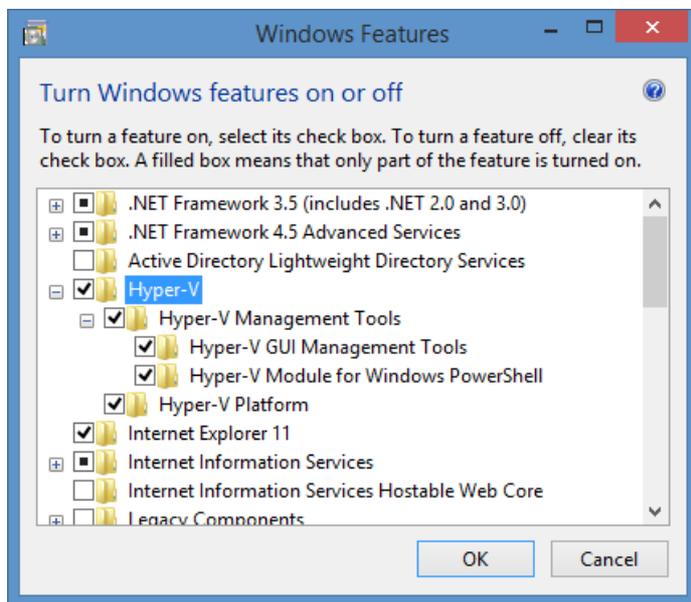
```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

This command works on all operating systems that support Hyper-V, but on Windows Server operating systems you must type an additional command to add the Hyper-V Windows PowerShell module and the Hyper-V Manager console. This command will also install Hyper-V if it isn't already installed, so if desired you can just type the following command on Windows Server 2012 or 2016 instead of using the Enable-WindowsOptionalFeature command:

```
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools
```

When you are prompted to restart the computer, choose **Yes**. The computer might restart more than once. After installation is complete, you can open Hyper-V Manager by typing **virtmgmt.msc** at an elevated command prompt.

Alternatively, you can install Hyper-V using the Control Panel in Windows under **Turn Windows features on or off** for a client operating system, or using Server Manager's **Add Roles and Features Wizard** on a server operating system, as shown below:



If you choose to install Hyper-V using Server Manager, accept all default selections. Also be sure to install both items under **Role Administration Tools\Hyper-V Management Tools**.

Download VHD and ISO files

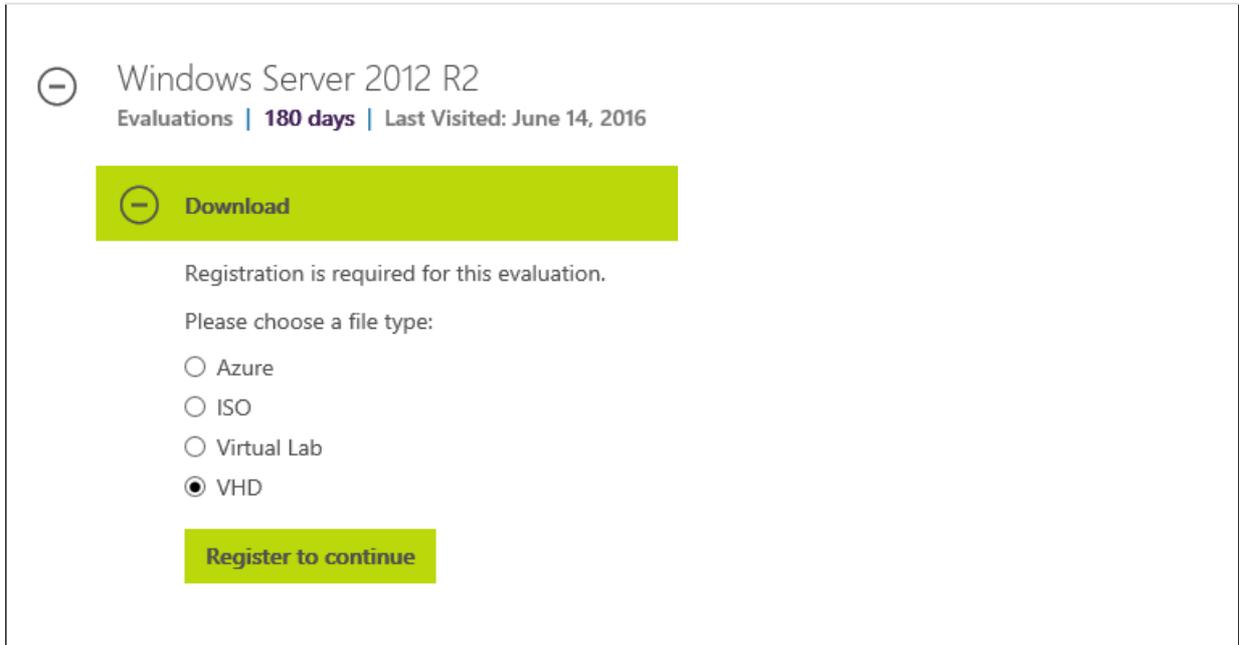
When you have completed installation of Hyper-V on the host computer, begin configuration of Hyper-V by downloading VHD and ISO files to the Hyper-V host. These files will be used to create the VMs used in the lab. Before you can download VHD and ISO files, you will need to register and sign in to the [TechNet Evaluation Center](#) using your Microsoft account.

1. Create a directory on your Hyper-V host named **C:\VHD** and download a single [Windows Server 2012 R2 VHD](#) from the TechNet Evaluation Center to the **C:\VHD** directory.

Important: This guide assumes that VHDs are stored in the **C:\VHD** directory on the Hyper-V host. If you use a different directory to store VHDs, you must adjust steps in this guide appropriately.

After completing registration you will be able to download the 7.47 GB Windows Server 2012 R2

evaluation VHD. An example of the download offering is shown below.



Windows Server 2012 R2
Evaluations | 180 days | Last Visited: June 14, 2016

Download

Registration is required for this evaluation.

Please choose a file type:

Azure

ISO

Virtual Lab

VHD

Register to continue

2. Download the file to the **C:\VHD** directory. When the download is complete, rename the VHD file that you downloaded to **2012R2-poc-1.vhd**. This is done to make the filename simple to recognize and type.
3. Copy the VHD to a second file also in the **C:\VHD** directory and name this VHD **2012R2-poc-2.vhd**.
4. Download the [Windows 10 Enterprise ISO](#) from the TechNet Evaluation Center to the **C:\VHD** directory on your Hyper-V host.

During registration, you must specify the type, version, and language of installation media to download. In this example, a Windows 10 Enterprise, 64 bit, English ISO is chosen. You can choose a different version if desired. **Note: The evaluation version of Windows 10 does not support in-place upgrade.**

5. Rename the ISO file that you downloaded to **w10-enterprise.iso**. Again, this is done so that the filename is simple to type and recognize. After completing registration you will be able to download the 3.63 GB Windows 10 Enterprise evaluation ISO.

After completing these steps, you will have three files in the **C:\VHD** directory: **2012R2-poc-1.vhd**, **2012R2-poc-2.vhd**, **w10-enterprise.iso**.

The following displays the procedures described in this section, both before and after downloading files:

```
C:>mkdir VHD
C:>cd VHD
C:\VHD>ren 9600*.vhd 2012R2-poc-1.vhd
C:\VHD>copy 2012R2-poc-1.vhd 2012R2-poc-2.vhd
    1 file(s) copied.
C:\VHD ren *.iso w10-enterprise.iso
C:\VHD>dir /B
2012R2-poc-1.vhd
2012R2-poc-2.vhd
w10-enterprise.iso
```

Convert PC to VM

Important: Do not attempt to use the VM resulting from the following procedure as a reference image. Also, to avoid conflicts with existing clients, do not start the VM outside the PoC network.

If you do not have a PC available to convert to VM, perform the following steps to download an evaluation VM:

1. Open the [Download virtual machines](#) page.
2. Under **Virtual machine**, choose **IE11 on Win7**.
3. Under **Select platform** choose **HyperV (Windows)**.
4. Click **Download .zip**. The download is 3.31 GB.
5. Extract the zip file. Three directories are created.
6. Open the **Virtual Hard Disks** directory and then copy **IE11 - Win7.vhd** to the **C:\VHD** directory.
7. Rename **IE11 - Win7.vhd** to **w7.vhd** (do not rename the file to w7.vhdx).
8. In step 5 of the [Configure Hyper-V](#) section, replace the VHD file name **w7.vhdx** with **w7.vhd**.

If you have a PC available to convert to VM (computer 2):

1. Sign in on computer 2 using an account with Administrator privileges.

Important: the account used in this step must have local administrator privileges. You can use a local computer account, or a domain account with administrative rights if domain policy allows the use of cached credentials. After converting the computer to a VM, you must be able to sign in on this VM with administrator rights while the VM is disconnected from the corporate network.

2. [Determine the VM generation and partition type](#) that is required.
3. Based on the VM generation and partition type, perform one of the following procedures: [Prepare a generation 1 VM](#), [Prepare a generation 2 VM](#), or [prepare a generation 1 VM from a GPT disk](#).

Determine the VM generation and partition type

When creating a VM in Hyper-V, you must specify either generation 1 or generation 2. The following table describes requirements for these two types of VMs.

	Architecture	Operating system	Partition style
Generation 1	32-bit or 64-bit	Windows 7 or later	MBR
Generation 2	64-bit	Windows 8 or later	MBR or GPT

If the PC is running a 32-bit OS or the OS is Windows 7, it must be converted to a generation 1 VM. Otherwise, it can be converted to a generation 2 VM.

- To determine the OS and architecture of a PC, type **systeminfo** at a command prompt and review the output next to **OS Name** and **System Type**.
- To determine the partition style, open a Windows PowerShell prompt on the PC and type the following command:

```
Get-WmiObject -Class Win32_DiskPartition | Select-Object -Property SystemName,Caption,Type
```

If the **Type** column does not indicate GPT, then the disk partition format is MBR ("Installable File System" = MBR). In the following example, the disk is GPT:

```
PS C:> Get-WmiObject -Class Win32_DiskPartition | Select-Object -Property SystemName,Caption,Type

SystemName          Caption                Type
-----
USER-PC1            Disk #0, Partition #0 GPT: System
USER-PC1            Disk #0, Partition #1 GPT: Basic Data
```

On a computer running Windows 8 or later, you can also type **Get-Disk** at a Windows PowerShell prompt to discover the partition style. The default output of this cmdlet displays the partition style for all attached disks. Both

commands are displayed below. In this example, the client computer is running Windows 8.1 and uses a GPT style partition format:

```
PS C:> Get-WmiObject -Class Win32_DiskPartition | Select-Object -Property SystemName,Caption,Type

SystemName          Caption              Type
-----
PC-X1              Disk #0, Partition #0  GPT: Unknown
PC-X1              Disk #0, Partition #1  GPT: System
PC-X1              Disk #0, Partition #2  GPT: Basic Data
PC-X1              Disk #0, Partition #3  GPT: Basic Data
PC-X1              Disk #0, Partition #4  GPT: Basic Data

PS C:> Get-Disk

Number Friendly Name          OperationalStatus      Total Size Partition
-----
0      INTEL SSDSCMMW240A3L        Online                223.57 GB GPT
```

Choosing a VM generation

The following table displays the Hyper-V VM generation to choose based on the OS, architecture, and partition style. Links to procedures to create the corresponding VMs are included.

OS	Partition style	Architecture	VM generation	Procedure
Windows 7	MBR	32	1	Prepare a generation 1 VM
		64	1	Prepare a generation 1 VM
	GPT	32	N/A	N/A
		64	1	Prepare a generation 1 VM from a GPT disk
Windows 8 or later	MBR	32	1	Prepare a generation 1 VM
		64	1, 2	Prepare a generation 1 VM
	GPT	32	1	Prepare a generation 1 VM from a GPT disk
		64	2	Prepare a generation 2 VM

Notes:

- If the PC is running Windows 7, it can only be converted and hosted in Hyper-V as a generation 1 VM. This Hyper-V requirement means that if the Windows 7 PC is also using a GPT partition style, the OS disk can be shadow copied, but a new system partition must be created. In this case, see [Prepare a generation 1 VM from a GPT disk](#).
- If the PC is running Windows 8 or later and uses the GPT partition style, you can capture the disk image and create a generation 2 VM. To do this, you must temporarily mount the EFI system partition which is accomplished using the **mountvol** command. In this case, see [Prepare a generation 2 VM](#).
- If the PC is using an MBR partition style, you can convert the disk to VHD and use it to create a generation 1 VM. If you use the Disk2VHD tool described in this guide, it is not necessary to mount the MBR system partition, but it is still necessary to capture it. In this case, see [Prepare a generation 1 VM](#).

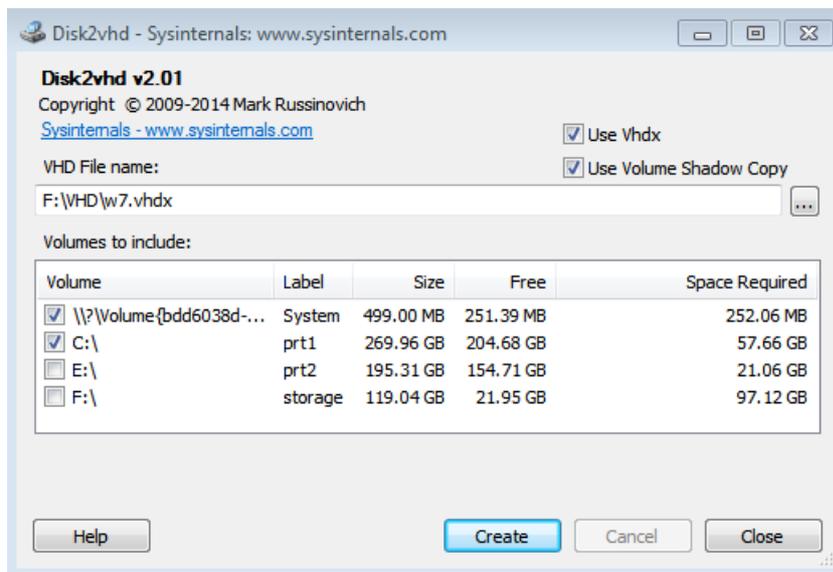
Prepare a generation 1 VM

1. Download the [Disk2vhd utility](#), extract the .zip file and copy **disk2vhd.exe** to a flash drive or other location

that is accessible from the computer you wish to convert.

You might experience timeouts if you attempt to run Disk2vhd from a network share, or specify a network share for the destination. To avoid timeouts, use local, portable media such as a USB drive.

2. On the computer you wish to convert, double-click the disk2vhd utility to start the graphical user interface.
3. Select the checkboxes next to the **C:** and the **system reserved** (BIOS/MBR) volumes. The system volume is not assigned a drive letter, but will be displayed in the Disk2VHD tool with a volume label similar to **\? \Volume{**. See the following example. **Important:** You must include the system volume in order to create a bootable VHD. If this volume is not displayed in the disk2vhd tool, then the computer is likely to be using the GPT partition style. For more information, see [Determine VM generation](#).
4. Specify a location to save the resulting VHD or VHDX file (F:\VHD\w7.vhdx in the following example) and click **Create**. See the following example:



Disk2vhd can save VHDs to local hard drives, even if they are the same as the volumes being converted. Performance is better however when the VHD is saved on a disk different than those being converted, such as a flash drive.

5. When the Disk2vhd utility has completed converting the source computer to a VHD, copy the VHDX file (w7.vhdx) to your Hyper-V host in the C:\VHD directory. There should now be four files in this directory:

```
C:\vhd>dir /B
2012R2-poc-1.vhd
2012R2-poc-2.vhd
w10-enterprise.iso
w7.VHDX
```

Prepare a generation 2 VM

1. Download the [Disk2vhd utility](#), extract the .zip file and copy **disk2vhd.exe** to a flash drive or other location that is accessible from the computer you wish to convert.

You might experience timeouts if you attempt to run Disk2vhd from a network share, or specify a network share for the destination. To avoid timeouts, use local, portable media such as a USB drive.

2. On the computer you wish to convert, open an elevated command prompt and type the following

command:

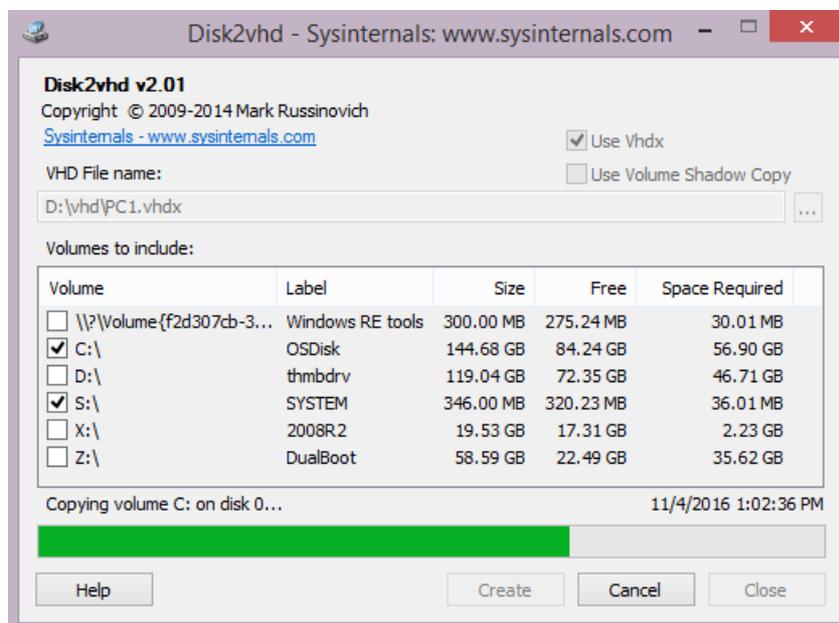
```
mountvol s: /s
```

This command temporarily assigns a drive letter of S to the system volume and mounts it. If the letter S is already assigned to a different volume on the computer, then choose one that is available (ex: mountvol z: /s).

3. On the computer you wish to convert, double-click the disk2vhd utility to start the graphical user interface.
4. Select the checkboxes next to the **C:** and the **S:** volumes, and clear the **Use Volume Shadow Copy checkbox**. Volume shadow copy will not work if the EFI system partition is selected.

Important: You must include the EFI system partition in order to create a bootable VHD. The Windows RE tools partition (shown below) is not required, but it can also be converted if desired.

5. Specify a location to save the resulting VHD or VHDX file (F:\VHD\PC1.vhdx in the following example) and click **Create**. See the following example:



Disk2vhd can save VHDs to local hard drives, even if they are the same as the volumes being converted. Performance is better however when the VHD is saved on a disk different than those being converted, such as a flash drive.

6. When the Disk2vhd utility has completed converting the source computer to a VHD, copy the VHDX file (PC1.vhdx) to your Hyper-V host in the C:\VHD directory. There should now be four files in this directory:

```
C:\vhd>dir /B
2012R2-poc-1.vhd
2012R2-poc-2.vhd
w10-enterprise.iso
PC1.VHDX
```

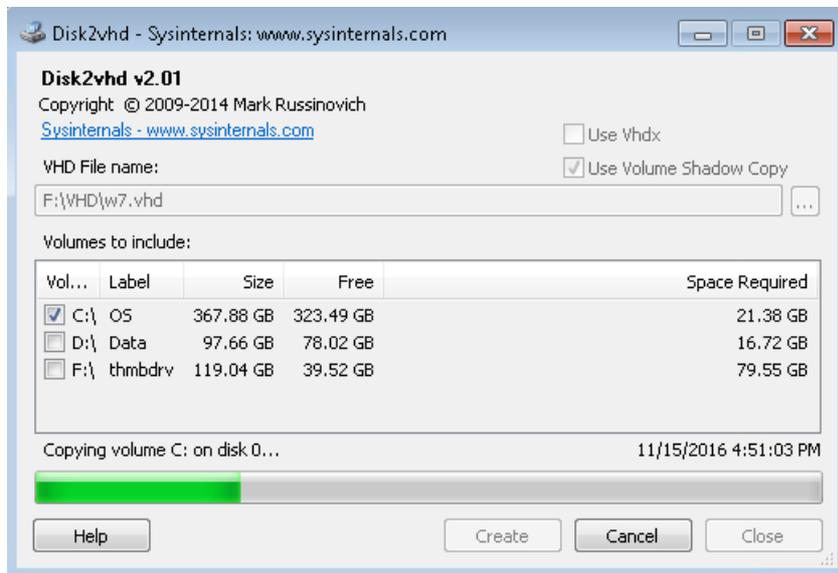
Prepare a generation 1 VM from a GPT disk

1. Download the [Disk2vhd utility](#), extract the .zip file and copy **disk2vhd.exe** to a flash drive or other location that is accessible from the computer you wish to convert.

You might experience timeouts if you attempt to run Disk2vhd from a network share, or specify a

network share for the destination. To avoid timeouts, use local, portable media such as a USB drive.

2. On the computer you wish to convert, double-click the disk2vhd utility to start the graphical user interface.
3. Select the checkbox next to the **C:** volume and clear the checkbox next to **Use Vhdx**. Note: the system volume is not copied in this scenario, it will be added later.
4. Specify a location to save the resulting VHD file (F:\VHD\w7.vhd in the following example) and click **Create**. See the following example:



Disk2vhd can save VHDs to local hard drives, even if they are the same as the volumes being converted. Performance is better however when the VHD is saved on a disk different than those being converted, such as a flash drive.

5. When the Disk2vhd utility has completed converting the source computer to a VHD, copy the VHD file (w7.vhd) to your Hyper-V host in the C:\VHD directory. There should now be four files in this directory:

```
C:\vhd>dir /B
2012R2-poc-1.vhd
2012R2-poc-2.vhd
w10-enterprise.iso
w7.VHD
```

In its current state, the w7.VHD file is not bootable. The VHD will be used to create a bootable VM later in the [Configure Hyper-V](#) section.

Resize VHD

Enhanced session mode

Important: Before proceeding, verify that you can take advantage of [enhanced session mode](#) when completing instructions in this guide. Enhanced session mode enables you to copy and paste the commands from the Hyper-V host to VMs, between VMs, and between RDP sessions. After copying some text, you can paste into a Windows PowerShell window by simply right-clicking. Before right-clicking, do not left click other locations as this can empty the clipboard. You can also copy and paste [files](#) directly from one computer to another by right-clicking and selecting copy on one computer, then right-clicking and selecting paste on another computer.

To ensure that enhanced session mode is enabled on the Hyper-V host, type the following command at an

elevated Windows PowerShell prompt on the Hyper-V host:

```
Set-VMhost -EnableEnhancedSessionMode $TRUE
```

If enhanced session mode was not previously enabled, close any existing virtual machine connections and re-open them to enable access to enhanced session mode. As mentioned previously: instructions to "type" commands provided in this guide can be typed, but the preferred method is to copy and paste these commands. Most of the commands to this point in the guide have been brief, but many commands in sections below are longer and more complex.

The second Windows Server 2012 R2 VHD needs to be expanded in size from 40GB to 100GB to support installing imaging tools and storing OS images.

1. To add available space for the partition, type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Resize-VHD -Path c:\VHD\2012R2-poc-2.vhd -SizeBytes 100GB
$x = (Mount-VHD -Path c:\VHD\2012R2-poc-2.vhd -passthru | Get-Disk | Get-Partition | Get-Volume).DriveLetter
Resize-Partition -DriveLetter $x -Size (Get-PartitionSupportedSize -DriveLetter $x).SizeMax
```

2. Verify that the mounted VHD drive is resized to 100 GB, and then dismount the drive:

```
Get-Volume -DriveLetter $x
Dismount-VHD -Path c:\VHD\2012R2-poc-2.vhd
```

Configure Hyper-V

1. Open an elevated Windows PowerShell window and type the following command to create two virtual switches named "poc-internal" and "poc-external":

If the Hyper-V host already has an external virtual switch bound to a physical NIC, do not attempt to add a second external virtual switch. Attempting to add a second external switch will result in an error indicating that the NIC is **already bound to the Microsoft Virtual Switch protocol**. In this case, choose one of the following options:

- A) Remove the existing external virtual switch, then add the poc-external switch
- B) Rename the existing external switch to "poc-external"
- C) Replace each instance of "poc-external" used in this guide with the name of your existing external virtual switch

If you choose B) or C), then do not run the second command below.

```
New-VMSwitch -Name poc-internal -SwitchType Internal -Notes "PoC Network"
New-VMSwitch -Name poc-external -NetAdapterName (Get-NetAdapter |?{$_.Status -eq "Up" -and !$_.Virtual}).Name -Notes "PoC External"
```

Note: The second command above will temporarily interrupt network connectivity on the Hyper-V host.

Since an external virtual switch is associated to a physical network adapter on the Hyper-V host, this adapter must be specified when adding the virtual switch. The previous commands automate this by filtering for active non-virtual ethernet adapters using the Get-NetAdapter cmdlet (`$.Status -eq "Up" -and !$_.Virtual`). If your Hyper-V host is dual-homed with multiple active ethernet adapters, this

automation will not work, and the second command above will fail. In this case, you must edit the command used to add the "poc-external" virtual switch by inserting the appropriate NetAdapterName. The NetAdapterName value corresponds to the name of the network interface you wish to use. For example, if the network interface you use on the Hyper-V host to connect to the Internet is named "Ethernet 2" then type the following command to create an external virtual switch: `New-VMSwitch -Name poc-external -NetAdapterName "Ethernet 2" -Notes "PoC External"`

2. At the elevated Windows PowerShell prompt, type the following command to determine the megabytes of RAM that are currently available on the Hyper-V host:

```
(Get-VMHostNumaNode).MemoryAvailable
```

This command will display the megabytes of RAM available for VMs. On a Hyper-V host computer with 16 GB of physical RAM installed, 10,000 MB of RAM or greater should be available if the computer is not also running other applications. On a computer with 8 GB of physical RAM installed, at least 4000 MB should be available. If the computer has less RAM available than this, try closing applications to free up more memory.

3. Determine the available memory for VMs by dividing the available RAM by 4. For example:

```
(Get-VMHostNumaNode).MemoryAvailable/4  
2775.5
```

In this example, VMs can use a maximum of 2700 MB of RAM each, to run four VMs simultaneously.

4. At the elevated Windows PowerShell prompt, type the following command to create two new VMs. Other VMs will be added later.

Important: Replace the value of 2700MB for \$maxRAM in the first command below with the RAM value that you calculated in the previous step.

```
$maxRAM = 2700MB  
New-VM -Name "DC1" -VHDPATH c:\vhd\2012R2-poc-1.vhd -SwitchName poc-internal  
Set-VMemory -VMName "DC1" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes  
$maxRAM -Buffer 20  
Enable-VMIntegrationService -Name "Guest Service Interface" -VMName DC1  
New-VM -Name "SRV1" -VHDPATH c:\vhd\2012R2-poc-2.vhd -SwitchName poc-internal  
Add-VMNetworkAdapter -VMName "SRV1" -SwitchName "poc-external"  
Set-VMemory -VMName "SRV1" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes  
$maxRAM -Buffer 80  
Enable-VMIntegrationService -Name "Guest Service Interface" -VMName SRV1
```

Note: The RAM values assigned to VMs in this step are not permanent, and can be easily increased or decreased later if needed to address performance issues.

5. Using the same elevated Windows PowerShell prompt that was used in the previous step, type one of the following sets of commands, depending on the type of VM that was prepared in the [Determine VM generation](#) section, either generation 1, generation 2, or generation 1 with GPT.

To create a generation 1 VM (using c:\vhd\w7.vhdx):

```
New-VM -Name "PC1" -VHDPath c:\vhd\w7.vhdx -SwitchName poc-internal
Set-VMMemory -VMName "PC1" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes
$maxRAM -Buffer 20
Enable-VMIntegrationService -Name "Guest Service Interface" -VMName PC1
```

To create a generation 2 VM (using c:\vhd\PC1.vhdx):

```
New-VM -Name "PC1" -Generation 2 -VHDPath c:\vhd\PC1.vhdx -SwitchName poc-internal
Set-VMMemory -VMName "PC1" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes
$maxRAM -Buffer 20
Enable-VMIntegrationService -Name "Guest Service Interface" -VMName PC1
```

To create a generation 1 VM from a GPT disk (using c:\vhd\w7.vhd):

Note: The following procedure is more complex because it includes steps to convert the OS partition from GPT to MBR format. Steps are included to create a temporary VHD and attach it to the VM, the OS image is saved to this drive, the OS drive is then reformatted to MBR, the OS image restored, and the temporary drive is removed.

First, type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host to create a temporary VHD that will be used to save the OS image. Do not forget to include a pipe (|) at the end of the first five commands:

```
New-VHD -Path c:\vhd\d.vhd -SizeBytes 1TB |
Mount-VHD -Passthru |
Get-Disk -Number {$_.DiskNumber} |
Initialize-Disk -PartitionStyle MBR -PassThru |
New-Partition -UseMaximumSize |
Format-Volume -Confirm:$false -FileSystem NTFS -force
Dismount-VHD -Path c:\vhd\d.vhd
```

Next, create the PC1 VM with two attached VHDs, and boot to DVD (\$maxram must be defined previously using the same Windows PowerShell prompt):

```
New-VM -Name "PC1" -VHDPath c:\vhd\w7.vhd -SwitchName poc-internal
Add-VMHardDiskDrive -VMName PC1 -Path c:\vhd\d.vhd
Set-VMVDvdDrive -VMName PC1 -Path c:\vhd\w10-enterprise.iso
Set-VMMemory -VMName "PC1" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes
$maxRAM -Buffer 20
Enable-VMIntegrationService -Name "Guest Service Interface" -VMName PC1
Start-VM PC1
vmconnect localhost PC1
```

The VM will automatically boot into Windows Setup. In the PC1 window:

- a. Click **Next**.
- b. Click **Repair your computer**.
- c. Click **Troubleshoot**.
- d. Click **Command Prompt**.

- e. Type the following command to save an image of the OS drive:

```
dism /Capture-Image /ImageFile:D:\c.wim /CaptureDir:C:\ /Name:Drive-C
```

- f. Wait for the OS image to complete saving, and then type the following commands to convert the C: drive to MBR:

```
diskpart
select disk 0
clean
convert MBR
create partition primary size=100
format fs=ntfs quick
active
create partition primary
format fs=ntfs quick label=OS
assign letter=c
exit
```

- g. Type the following commands to restore the OS image and boot files:

```
dism /Apply-Image /ImageFile:D:\c.wim /Index:1 /ApplyDir:C:\
bcdboot c:\windows
exit
```

- h. Click **Continue** and verify the VM boots successfully (do not boot from DVD).
- i. Click **Ctrl+Alt+Del**, and then in the bottom right corner, click **Shut down**.
- j. Type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host to remove the temporary disks and drives from PC1:

```
Remove-VMHardDiskDrive -VMName PC1 -ControllerType IDE -ControllerNumber 0 -
ControllerLocation 1
Set-VMdvdDrive -VMName PC1 -Path $null
```

Configure VMs

1. At an elevated Windows PowerShell prompt on the Hyper-V host, start the first Windows Server VM and connect to it by typing the following commands:

```
Start-VM DC1
vmconnect localhost DC1
```

2. Click **Next** to accept the default settings, read the license terms and click **I accept**, provide an administrator password of **pass@word1**, and click **Finish**.
3. Click **Ctrl+Alt+Del** in the upper left corner of the virtual machine connection window, and then sign in to DC1 using the Administrator account.
4. Right-click **Start**, point to **Shut down or sign out**, and click **Sign out**. The VM connection will reset and a new connection dialog box will appear enabling you to choose a custom display configuration. Select a desktop size, click **Connect** and sign in again with the local Administrator account. Note: Signing in this way ensures that **enhanced session mode** is enabled. It is only necessary to do this the first time you sign in to a new VM.

5. If DC1 is configured as described in this guide, it will currently be assigned an APIPA address, have a randomly generated hostname, and a single network adapter named "Ethernet." Open an elevated Windows PowerShell prompt on DC1 and type or paste the following commands to provide a new hostname and configure a static IP address and gateway:

```
Rename-Computer DC1
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 192.168.0.1 -PrefixLength 24 -
DefaultGateway 192.168.0.2
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses
192.168.0.1,192.168.0.2
```

The default gateway at 192.168.0.2 will be configured later in this guide.

Note: A list of available tasks for an app will be populated the first time you run it on the taskbar. Because these tasks aren't available until the App has been run, you will not see the **Run as Administrator** task until you have left-clicked Windows PowerShell for the first time. In this newly created VM, you will need to left-click Windows PowerShell one time, and then you can right-click and choose Run as Administrator to open an elevated Windows PowerShell prompt.

6. Install the Active Directory Domain Services role by typing the following command at an elevated Windows PowerShell prompt:

```
Install-WindowsFeature -Name AD-Domain-Services -IncludeAllSubFeature -
IncludeManagementTools
```

7. Before promoting DC1 to a Domain Controller, you must reboot so that the name change in step 3 above takes effect. To restart the computer, type the following command at an elevated Windows PowerShell prompt:

```
Restart-Computer
```

8. When DC1 has rebooted, sign in again and open an elevated Windows PowerShell prompt. Now you can promote the server to be a domain controller. The directory services restore mode password must be entered as a secure string. Type the following commands at the elevated Windows PowerShell prompt:

```
$pass = "pass@word1" | ConvertTo-SecureString -AsPlainText -Force
Install-ADDSForest -DomainName contoso.com -InstallDns -SafeModeAdministratorPassword $pass
-Force
```

Ignore any warnings that are displayed. The computer will automatically reboot upon completion.

9. When the reboot has completed, reconnect to DC1, sign in using the CONTOSO\Administrator account, open an elevated Windows PowerShell prompt, and use the following commands to add a reverse lookup zone for the PoC network, add the DHCP Server role, authorize DHCP in Active Directory, and suppress the post-DHCP-install alert:

```
Add-DnsServerPrimaryZone -NetworkID "192.168.0.0/24" -ReplicationScope Forest
Add-WindowsFeature -Name DHCP -IncludeManagementTools
netsh dhcp add securitygroups
Restart-Service DHCPServer
Add-DhcpServerInDC dc1.contoso.com 192.168.0.1
Set-ItemProperty -Path
registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ServerManager\Roles\12 -Name
ConfigurationState -Value 2
```

10. Next, add a DHCP scope and set option values:

```
Add-DhcpServerv4Scope -Name "PoC Scope" -StartRange 192.168.0.100 -EndRange 192.168.0.199 -
SubnetMask 255.255.255.0 -Description "Windows 10 PoC" -State Active
Set-DhcpServerv4OptionValue -ScopeId 192.168.0.0 -DnsDomain contoso.com -Router 192.168.0.2
-DnsServer 192.168.0.1,192.168.0.2 -Force
```

The -Force option is necessary when adding scope options to skip validation of 192.168.0.2 as a DNS server because we have not configured it yet. The scope should immediately begin issuing leases on the PoC network. The first DHCP lease that will be issued is to vEthernet interface on the Hyper-V host, which is a member of the internal network. You can verify this by using the command: `Get-DhcpServerv4Lease -Scopeld 192.168.0.0`.

11. The DNS server role will also be installed on the member server, SRV1, at 192.168.0.2 so that we can forward DNS queries from DC1 to SRV1 to resolve Internet names without having to configure a forwarder outside the PoC network. Since the IP address of SRV1 already exists on DC1's network adapter, it will be automatically added during the DCPROMO process. To verify this server-level DNS forwarder on DC1, type the following command at an elevated Windows PowerShell prompt on DC1:

```
Get-DnsServerForwarder
```

The following output should be displayed:

```
UseRootHint      : True
Timeout(s)       : 3
EnableReordering : True
IPAddress        : 192.168.0.2
ReorderedIPAddress : 192.168.0.2
```

If this output is not displayed, you can use the following command to add SRV1 as a forwarder:

```
Add-DnsServerForwarder -IPAddress 192.168.0.2
```

Configure service and user accounts

Windows 10 deployment with MDT and System Center Configuration Manager requires specific accounts to perform some actions. Service accounts will be created to use for these tasks. A user account is also added in the contoso.com domain that can be used for testing purposes. In the test lab environment, passwords are set to never expire.

To keep this test lab relatively simple, we will not create a custom OU structure and set permissions. Required permissions are enabled by adding accounts to the Domain Admins group. To configure these settings in a production environment, see [Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

On DC1, open an elevated Windows PowerShell prompt and type the following commands:

```
New-ADUser -Name User1 -UserPrincipalName user1 -Description "User account" -AccountPassword (ConvertTo-SecureString "pass@word1" -AsPlainText -Force) -ChangePasswordAtLogon $false -Enabled $true
New-ADUser -Name MDT_BA -UserPrincipalName MDT_BA -Description "MDT Build Account" -AccountPassword (ConvertTo-SecureString "pass@word1" -AsPlainText -Force) -ChangePasswordAtLogon $false -Enabled $true
New-ADUser -Name CM_JD -UserPrincipalName CM_JD -Description "Configuration Manager Join Domain Account" -AccountPassword (ConvertTo-SecureString "pass@word1" -AsPlainText -Force) -ChangePasswordAtLogon $false -Enabled $true
New-ADUser -Name CM_NAA -UserPrincipalName CM_NAA -Description "Configuration Manager Network Access Account" -AccountPassword (ConvertTo-SecureString "pass@word1" -AsPlainText -Force) -ChangePasswordAtLogon $false -Enabled $true
Add-ADGroupMember "Domain Admins" MDT_BA,CM_JD,CM_NAA
Set-ADUser -Identity user1 -PasswordNeverExpires $true
Set-ADUser -Identity administrator -PasswordNeverExpires $true
Set-ADUser -Identity MDT_BA -PasswordNeverExpires $true
Set-ADUser -Identity CM_JD -PasswordNeverExpires $true
Set-ADUser -Identity CM_NAA -PasswordNeverExpires $true
```

12. Minimize the DC1 VM window but **do not stop** the VM.

Next, the client VM will be started and joined to the contoso.com domain. This is done before adding a gateway to the PoC network so that there is no danger of duplicate DNS registrations for the physical client and its cloned VM in the corporate domain.

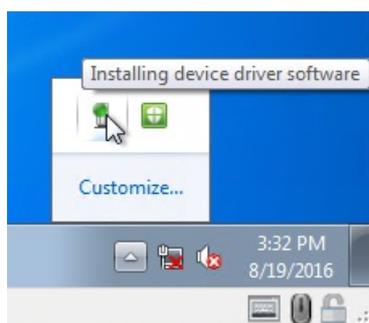
13. If the PC1 VM is not started yet, using an elevated Windows PowerShell prompt on the Hyper-V host, start the client VM (PC1), and connect to it:

```
Start-VM PC1
vmconnect localhost PC1
```

14. Sign in to PC1 using an account that has local administrator rights.

PC1 will be disconnected from its current domain, so you cannot use a domain account to sign on unless these credentials are cached and the use of cached credentials is permitted by Group Policy. If cached credentials are available and permitted, you can use these credentials to sign in. Otherwise, use an existing local administrator account.

15. After signing in, the operating system detects that it is running in a new environment. New drivers will be automatically installed, including the network adapter driver. The network adapter driver must be updated before you can proceed, so that you will be able to join the contoso.com domain. Depending on the resources allocated to PC1, installing the network adapter driver might take a few minutes. You can monitor device driver installation by clicking **Show hidden icons** in the notification area.



If the client was configured with a static address, you must change this to a dynamic one so that it can

obtain a DHCP lease.

- When the new network adapter driver has completed installation, you will receive an alert to set a network location for the contoso.com network. Select **Work network** and then click **Close**. When you receive an alert that a restart is required, click **Restart Later**.
- Open an elevated Windows PowerShell prompt on PC1 and verify that the client VM has received a DHCP lease and can communicate with the contoso.com domain controller.

To open Windows PowerShell on Windows 7, click **Start**, and search for "**power**." Right-click **Windows PowerShell** and then click **Pin to Taskbar** so that it is simpler to use Windows Powershell during this lab. Click **Windows PowerShell** on the taskbar, and then type **ipconfig** at the prompt to see the client's current IP address. Also type **ping dc1.contoso.com** and **nltest /dsgetdc:contoso.com** to verify that it can reach the domain controller. See the following examples of a successful network connection:

```
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 3:
    Connection-specific DNS Suffix  . : contoso.com
    Link-local IPv6 Address . . . . . : fe80::64c2:4d2a:7403:6e02%18
    Ipv4 Address. . . . . : 192.168.0.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.2

ping dc1.contoso.com

Pinging dc1.contoso.com [192.168.0.1] with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128

nltest /dsgetdc:contoso.com
    DC: \\DC1
    Address: \\192.168.0.1
    Dom Guid: fdbd0643-d664-411b-aea0-fe343d7670a8
    Dom Name: CONTOSO
    Forest Name: contoso.com
    Dc Site Name: Default-First-Site-Name
    Our Site Name: Default-First-Site-Name
    Flags: PDC GC DS LDAP KDC TIMESERV WRITABLE DNS_FOREST CLOSE_SITE FULL_SECRET WS 0xc000
```

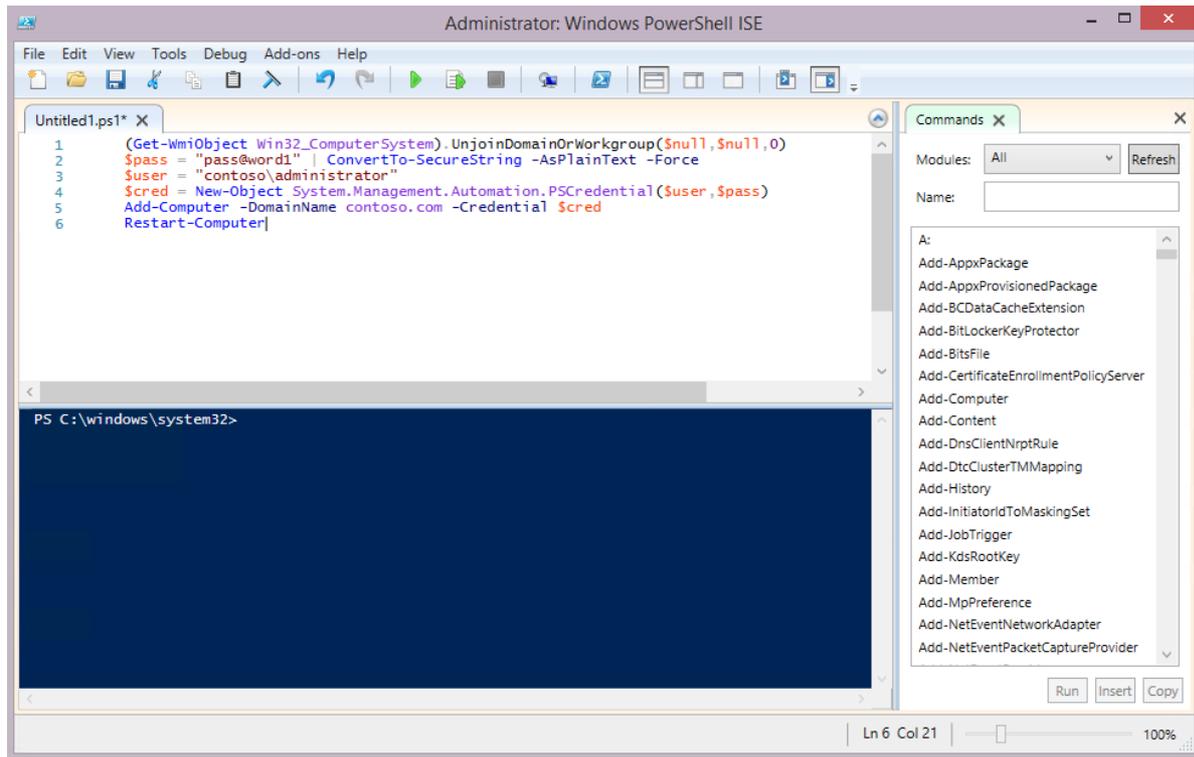
If PC1 is running Windows 7, enhanced session mode might not be available, which means that you cannot copy and paste commands from the Hyper-V host to a Windows PowerShell prompt on PC1. However, it is possible to use integration services to copy a file from the Hyper-V host to a VM. The next procedure demonstrates this. If the Copy-VMFile command fails, then type the commands below at an elevated Windows PowerShell prompt on PC1 instead of saving them to a script to run remotely. If PC1 is running Windows 8 or a later operating system, you can use enhanced session mode to copy and paste these commands instead of typing them.

- Minimize the PC1 window and switch to the Hyper-V host computer. Open an elevated Windows PowerShell ISE window on the Hyper-V host (right-click Windows PowerShell and then click **Run ISE as Administrator**) and type the following commands in the (upper) script editor pane:

```
(Get-WmiObject Win32_ComputerSystem).UnjoinDomainOrWorkgroup($null,$null,0)
$pass = "pass@word1" | ConvertTo-SecureString -AsPlainText -Force
$user = "contoso\administrator"
$cred = New-Object System.Management.Automation.PSCredential($user,$pass)
Add-Computer -DomainName contoso.com -Credential $cred
Restart-Computer
```

If you do not see the script pane, click **View** and verify **Show Script Pane Top** is enabled. Click **File** and then click **New**.

See the following example:



19. Click **File**, click **Save As**, and save the commands as **c:\VHD\pc1.ps1** on the Hyper-V host.
20. In the (lower) terminal input window, type the following commands to enable Guest Service Interface on PC1 and then use this service to copy the script to PC1:

```
Enable-VMIntegrationService -VMName PC1 -Name "Guest Service Interface"
Copy-VMFile "PC1" -SourcePath "C:\VHD\pc1.ps1" -DestinationPath "C:\pc1.ps1" -
CreateFullPath -FileSource Host
```

In order for this command to work properly, PC1 must be running the vmicguestinterface (Hyper-V Guest Service Interface) service. If this service is not enabled in this step, then the copy-VMFile command will fail. In this case, you can try updating integration services on the VM by mounting the Hyper-V Integration Services Setup (vmguest.iso), which is located in C:\Windows\System32 on Windows Server 2012 and 2012 R2 operating systems that are running the Hyper-V role service.

If the copy-vmfile command does not work and you cannot properly enable or upgrade integration services on PC1, then create the file c:\pc1.ps1 on the VM by typing the commands into this file manually. The copy-vmfile command is only used in this procedure as a demonstration of automation methods that can be used in a Hyper-V environment when enhanced session mode is not available. After typing the script file manually, be sure to save the file as a Windows PowerShell script file with the .ps1 extension and

not as a text (.txt) file.

21. On PC1, type the following commands at an elevated Windows PowerShell prompt:

```
Get-Content c:\pc1.ps1 | powershell.exe -noprofile -
```

The commands in this script might take a few moments to complete. If an error is displayed, check that you typed the command correctly, paying close attention to spaces. PC1 is removed from its domain in this step while not connected to the corporate network so as to ensure the computer object in the corporate domain is unaffected. PC1 is also not renamed to "PC1" in system properties so that it maintains some of its mirrored identity. However, if desired you can also rename the computer.

22. Upon completion of the script, PC1 will automatically restart. When it has restarted, sign in to the contoso.com domain using the **Switch User** option, with the **user1** account you created in step 11 of this section.

Important: The settings that will be used later to migrate user data specifically select only accounts that belong to the CONTOSO domain. However, this can be changed to migrate all user accounts, or only other specified accounts. If you wish to test migration of user data and settings with accounts other than those in the CONTOSO domain, you must specify these accounts or domains when you configure the value of **ScanStateArgs** in the MDT test lab guide. This value is specifically called out when you get to that step. If you wish to only migrate CONTOSO accounts, then you can log in with the user1 account or the administrator account at this time and modify some of the files and settings for later use in migration testing.

23. Minimize the PC1 window but do not turn it off while the second Windows Server 2012 R2 VM (SRV1) is configured. This verifies that the Hyper-V host has enough resources to run all VMs simultaneously. Next, SRV1 will be started, joined to the contoso.com domain, and configured with RRAS and DNS services.

24. On the Hyper-V host computer, at an elevated Windows PowerShell prompt, type the following commands:

```
Start-VM SRV1  
vmconnect localhost SRV1
```

25. Accept the default settings, read license terms and accept them, provide an administrator password of **pass@word1**, and click **Finish**. When you are prompted about finding PCs, devices, and content on the network, click **Yes**.
26. Sign in to SRV1 using the local administrator account. In the same way that was done on DC1, sign out of SRV1 and then sign in again to enable enhanced session mode. This will enable you to copy and paste Windows PowerShell commands from the Hyper-V host to the VM.
27. Open an elevated Windows PowerShell prompt on SRV1 and type the following commands:

```
Rename-Computer SRV1  
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 192.168.0.2 -PrefixLength 24  
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses 192.168.0.1,192.168.0.2  
Restart-Computer
```

IMPORTANT

Verify that you are configuring the correct interface in this step. The commands in this step assume that the poc-internal interface on SRV1 is named "Ethernet." If you are unsure how to check the interface, see step #30 below for instructions and tips on how to verify and modify the interface name.

28. Wait for the computer to restart, sign in again, then type the following commands at an elevated Windows PowerShell prompt:

```
$pass = "pass@word1" | ConvertTo-SecureString -AsPlainText -Force
$user = "contoso\administrator"
$cred = New-Object System.Management.Automation.PSCredential($user,$pass)
Add-Computer -DomainName contoso.com -Credential $cred
Restart-Computer
```

29. Sign in to the contoso.com domain on SRV1 using the domain administrator account (enter contoso\administrator as the user), open an elevated Windows PowerShell prompt, and type the following commands:

```
Install-WindowsFeature -Name DNS -IncludeManagementTools
Install-WindowsFeature -Name WDS -IncludeManagementTools
Install-WindowsFeature -Name Routing -IncludeManagementTools
```

30. Before configuring the routing service that was just installed, verify that network interfaces were added to SRV1 in the right order, resulting in an interface alias of "Ethernet" for the private interface, and an interface alias of "Ethernet 2" for the public interface. Also verify that the external interface has a valid external DHCP IP address lease.

To view a list of interfaces, associated interface aliases, and IP addresses on SRV1, type the following Windows PowerShell command. Example output of the command is also shown below:

```
Get-NetAdapter | ? status -eq 'up' | Get-NetIPAddress -AddressFamily IPv4 | ft IPAddress,
InterfaceAlias
```

IPAddress	InterfaceAlias
-----	-----
10.137.130.118	Ethernet 2
192.168.0.2	Ethernet

In this example, the poc-internal network interface at 192.168.0.2 is associated with the "Ethernet" interface and the Internet-facing poc-external interface is associated with the "Ethernet 2" interface. If your interfaces are different, you must adjust the commands provided in the next step appropriately to configure routing services. Also note that if the "Ethernet 2" interface has an IP address in the 192.168.0.100-105 range then it likely is getting a DHCP lease from DC1 instead of your corporate network. If this is the case, you can try removing and re-adding the second network interface from the SRV1 VM through its Hyper-V settings.

TIP

Sometimes a computer will have hidden, disconnected interfaces that prevent you from naming a network adapter. When you attempt to rename an adapter, you will receive an error that the adapter name already exists. These disconnected devices can be viewed in device manager by clicking **View** and then clicking **Show hidden devices**. The disconnected device can then be uninstalled, enabling you to reuse the adapter name.

31. To configure SRV1 with routing capability for the PoC network, type or paste the following commands at

an elevated Windows PowerShell prompt on SRV1:

```
Install-RemoteAccess -VpnType Vpn
cmd /c netsh routing ip nat install
cmd /c netsh routing ip nat add interface name="Ethernet 2" mode=FULL
cmd /c netsh routing ip nat add interface name="Ethernet" mode=PRIVATE
cmd /c netsh routing ip nat add interface name="Internal" mode=PRIVATE
```

32. The DNS service on SRV1 also needs to resolve hosts in the contoso.com domain. This can be accomplished with a conditional forwarder. Open an elevated Windows PowerShell prompt on SRV1 and type the following command:

```
Add-DnsServerConditionalForwarderZone -Name contoso.com -MasterServers 192.168.0.1
```

33. In most cases, this completes configuration of the PoC network. However, if your corporate network has a firewall that filters queries from local DNS servers, you will also need to configure a server-level DNS forwarder on SRV1 to resolve Internet names. To test whether or not DNS is working without this forwarder, try to reach a name on the Internet from DC1 or PC1, which are only using DNS services on the PoC network. You can test DNS with the ping command, for example:

```
ping www.microsoft.com
```

If you see "Ping request could not find host www.microsoft.com" on PC1 and DC1, but not on SRV1, then you will need to configure a server-level DNS forwarder on SRV1. To do this, open an elevated Windows PowerShell prompt on SRV1 and type the following command.

Note: This command also assumes that "Ethernet 2" is the external-facing network adapter on SRV1. If the external adapter has a different name, replace "Ethernet 2" in the command below with that name:

```
Add-DnsServerForwarder -IPAddress (Get-DnsClientServerAddress -InterfaceAlias "Ethernet 2").ServerAddresses
```

34. If DNS and routing are both working correctly, you will see the following on DC1 and PC1 (the IP address might be different, but that is OK):

```
PS C:\> ping www.microsoft.com

Pinging e2847.dspb.akamaiedge.net [23.222.146.170] with 32 bytes of data:
Reply from 23.222.146.170: bytes=32 time=3ms TTL=51
Reply from 23.222.146.170: bytes=32 time=2ms TTL=51
Reply from 23.222.146.170: bytes=32 time=2ms TTL=51
Reply from 23.222.146.170: bytes=32 time=1ms TTL=51

Ping statistics for 23.222.146.170:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms
```

35. Verify that all three VMs can reach each other, and the Internet. See [Appendix A: Verify the configuration](#) for more information.
36. Lastly, because the client computer has different hardware after copying it to a VM, its Windows activation will be invalidated and you might receive a message that you must activate Windows in 3 days. To extend this period to 30 days, type the following commands at an elevated Windows PowerShell prompt on PC1:

```
runas /noprofile /env /user:administrator@contoso.com "cmd /c slmgr -rearm"  
Restart-Computer
```

This completes configuration of the starting PoC environment. Additional services and tools are installed in subsequent guides.

Appendix A: Verify the configuration

Use the following procedures to verify that the PoC environment is configured properly and working as expected.

1. On DC1, open an elevated Windows PowerShell prompt and type the following commands:

```
Get-Service NTDS,DNS,DHCP  
DCDiag -a  
Get-DnsServerResourceRecord -ZoneName contoso.com -RRType A  
Get-DnsServerForwarder  
Resolve-DnsName -Server dc1.contoso.com -Name www.microsoft.com  
Get-DhcpServerInDC  
Get-DhcpServerv4Statistics  
ipconfig /all
```

Get-Service displays a status of "Running" for all three services.

DCDiag displays "passed test" for all tests.

Get-DnsServerResourceRecord displays the correct DNS address records for DC1, SRV1, and the computername of PC1. Additional address records for the zone apex (@), DomainDnsZones, and ForestDnsZones will also be registered.

Get-DnsServerForwarder displays a single forwarder of 192.168.0.2.

Resolve-DnsName displays public IP address results for www.microsoft.com.

Get-DhcpServerInDC displays 192.168.0.1, dc1.contoso.com.

Get-DhcpServerv4Statistics displays 1 scope with 2 addresses in use (these belong to PC1 and the Hyper-V host).

ipconfig displays a primary DNS suffix and suffix search list of contoso.com, IP address of 192.168.0.1, subnet mask of 255.255.255.0, default gateway of 192.168.0.2, and DNS server addresses of 192.168.0.1 and 192.168.0.2.

2. On SRV1, open an elevated Windows PowerShell prompt and type the following commands:

```
Get-Service DNS,RemoteAccess  
Get-DnsServerForwarder  
Resolve-DnsName -Server dc1.contoso.com -Name www.microsoft.com  
ipconfig /all  
netsh int ipv4 show address
```

Get-Service displays a status of "Running" for both services.

Get-DnsServerForwarder either displays no forwarders, or displays a list of forwarders you are required to use so that SRV1 can resolve Internet names.

Resolve-DnsName displays public IP address results for www.microsoft.com.

ipconfig displays a primary DNS suffix of contoso.com. The suffix search list contains contoso.com and your corporate domain. Two ethernet adapters are shown: Ethernet adapter "Ethernet" has an IP addresses of 192.168.0.2, subnet mask of 255.255.255.0, no default gateway, and DNS server addresses of 192.168.0.1 and 192.168.0.2. Ethernet adapter "Ethernet 2" has an IP address, subnet mask, and default gateway configured by DHCP on your corporate network.

netsh displays three interfaces on the computer: interface "Ethernet 2" with DHCP enabled = Yes and IP address assigned by your corporate network, interface "Ethernet" with DHCP enabled = No and IP address of 192.168.0.2, and interface "Loopback Pseudo-Interface 1" with IP address of 127.0.0.1.

3. On PC1, open an elevated Windows PowerShell prompt and type the following commands:

```
whoami
hostname
nslookup www.microsoft.com
ping -n 1 dc1.contoso.com
tracert www.microsoft.com
```

whoami displays the current user context, for example in an elevated Windows PowerShell prompt, contoso\administrator is displayed.

hostname displays the name of the local computer, for example W7PC-001.

nslookup displays the DNS server used for the query, and the results of the query. For example, server dc1.contoso.com, address 192.168.0.1, Name e2847.dspb.akamaiedge.net.

ping displays if the source can resolve the target name, and whether or not the target responds to ICMP. If it cannot be resolved, "..could not find host" will be displayed and if the target is found and also responds to ICMP, you will see "Reply from" and the IP address of the target.

tracert displays the path to reach the destination, for example srv1.contoso.com [192.168.0.2] followed by a list of hosts and IP addresses corresponding to subsequent routing nodes between the source and the destination.

Appendix B: Terminology used in this guide

Term	Definition
GPT	GUID partition table (GPT) is an updated hard-disk formatting scheme that enables the use of newer hardware. GPT is one of the partition formats that can be chosen when first initializing a hard drive, prior to creating and formatting partitions.
Hyper-V	Hyper-V is a server role introduced with Windows Server 2008 that lets you create a virtualized computing environment. Hyper-V can also be installed as a Windows feature on Windows client operating systems, starting with Windows 8.
Hyper-V host	The computer where Hyper-V is installed.
Hyper-V Manager	The user-interface console used to view and configure Hyper-V.
MBR	Master Boot Record (MBR) is a legacy hard-disk formatting scheme that limits support for newer hardware. MBR is one of the partition formats that can be chosen when first initializing a hard drive, prior to creating and formatting partitions. MBR is in the process of being replaced by the GPT partition format.
Proof of concept (PoC)	Confirmation that a process or idea works as intended. A PoC is carried out in a test environment to learn about and verify a process.
Shadow copy	A copy or "snapshot" of a computer at a point in time, created by the Volume Shadow Copy Service (VSS), typically for backup purposes.
Virtual machine (VM)	A VM is a virtual computer with its own operating system, running on the Hyper-V host.
Virtual switch	A virtual network connection used to connect VMs to each other and to physical network adapters on the Hyper-V host.

VM snapshot

A point in time image of a VM that includes its disk, memory and device state. It can be used to return a virtual machine to a former state corresponding to the time the snapshot was taken.

Related Topics

[Windows 10 deployment scenarios](#)

Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit

6/18/2019 • 21 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Important: This guide leverages the proof of concept (PoC) environment configured using procedures in the following guide:

- [Step by step guide: Configure a test lab to deploy Windows 10](#)

Please complete all steps in the prerequisite guide before starting this guide. This guide requires about 5 hours to complete, but can require less time or more time depending on the speed of the Hyper-V host. After completing the current guide, also see the companion guide:

- [Deploy Windows 10 in a test lab using System Center Configuration Manager](#)

The PoC environment is a virtual network running on Hyper-V with three virtual machines (VMs):

- **DC1:** A contoso.com domain controller, DNS server, and DHCP server.
- **SRV1:** A dual-homed contoso.com domain member server, DNS server, and default gateway providing NAT service for the PoC network.
- **PC1:** A contoso.com member computer running Windows 7, Windows 8, or Windows 8.1 that has been shadow-copied from a physical computer on your corporate network.

This guide uses the Hyper-V server role. If you do not complete all steps in a single session, consider using [checkpoints](#) and [saved states](#) to pause, resume, or restart your work.

In this guide

This guide provides instructions to install and configure the Microsoft Deployment Toolkit (MDT) to deploy a Windows 10 image.

Topics and procedures in this guide are summarized in the following table. An estimate of the time required to complete each procedure is also provided. Time required to complete procedures will vary depending on the resources available to the Hyper-V host and assigned to VMs, such as processor speed, memory allocation, disk speed, and network speed.

Topic	Description	Time
About MDT	A high-level overview of the Microsoft Deployment Toolkit (MDT).	Informational
Install MDT	Download and install MDT.	40 minutes
Create a deployment share and reference image	A reference image is created to serve as the template for deploying new images.	90 minutes
Deploy a Windows 10 image using MDT	The reference image is deployed in the PoC environment.	60 minutes

Refresh a computer with Windows 10	Export user data from an existing client computer, wipe the computer, install a new operating system, and then restore user data and settings.	60 minutes
Replace a computer with Windows 10	Back up an existing client computer, then restore this backup to a new computer.	60 minutes
Troubleshooting logs, events, and utilities	Log locations and troubleshooting hints.	Informational

About MDT

MDT performs deployments by using the Lite Touch Installation (LTI), Zero Touch Installation (ZTI), and User-Driven Installation (UDI) deployment methods.

- LTI is the deployment method used in the current guide, requiring only MDT and performed with a minimum amount of user interaction.
- ZTI is fully automated, requiring no user interaction and is performed using MDT and System Center Configuration Manager. After completing the steps in the current guide, see [Step by step: Deploy Windows 10 in a test lab using System Center Configuration Manager](#) to use the ZTI deployment method in the PoC environment.
- UDI requires manual intervention to respond to installation prompts such as machine name, password and language settings. UDI requires MDT and System Center Configuration Manager.

Install MDT

1. On SRV1, temporarily disable IE Enhanced Security Configuration for Administrators by typing the following commands at an elevated Windows PowerShell prompt:

```
$AdminKey = "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}"
Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 0
Stop-Process -Name Explorer
```

2. Download and install the 64-bit version of [Microsoft Deployment Toolkit \(MDT\)](#) on SRV1 using the default options. As of the writing of this guide, the latest version of MDT was 8443.
3. Download and install the latest [Windows Assessment and Deployment Kit \(ADK\)](#) on SRV1 using the default installation settings. The current version is the ADK for Windows 10, version 1703. Installation might require several minutes to acquire all components.
4. If desired, re-enable IE Enhanced Security Configuration:

```
Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 1
Stop-Process -Name Explorer
```

Create a deployment share and reference image

A reference image serves as the foundation for Windows 10 devices in your organization.

1. In [Step by step guide: Configure a test lab to deploy Windows 10](#), the Windows 10 Enterprise .iso file was saved to the c:\VHD directory as **c:\VHD\w10-enterprise.iso**. The first step in creating a deployment share is to mount this file on SRV1. To mount the Windows 10 Enterprise DVD on SRV1, open an elevated Windows PowerShell prompt on the Hyper-V host computer and type the following command:

```
Set-VMDvdDrive -VMName SRV1 -Path c:\VHD\w10-enterprise.iso
```

2. On SRV1, verify that the Windows Enterprise installation DVD is mounted as drive letter D.
3. The Windows 10 Enterprise installation files will be used to create a deployment share on SRV1 using the MDT deployment workbench. To open the deployment workbench, click **Start**, type **deployment**, and then click **Deployment Workbench**.
4. To enable quick access to the application, right-click **Deployment Workbench** on the taskbar and then click **Pin this program to the taskbar**.
5. In the Deployment Workbench console, right-click **Deployment Shares** and select **New Deployment Share**.
6. Use the following settings for the New Deployment Share Wizard:
 - Deployment share path: **C:\MDTBuildLab**
 - Share name: **MDTBuildLab\$**
 - Deployment share description: **MDT build lab**
 - Options: click **Next** to accept the default
 - Summary: click **Next**
 - Progress: settings will be applied
 - Confirmation: click **Finish**
7. Expand the **Deployment Shares** node, and then expand **MDT build lab**.
8. Right-click the **Operating Systems** node, and then click **New Folder**. Name the new folder **Windows 10**. Complete the wizard using default values and click **Finish**.
9. Right-click the **Windows 10** folder created in the previous step, and then click **Import Operating System**.
10. Use the following settings for the Import Operating System Wizard:
 - OS Type: **Full set of source files**
 - Source: **D:**
 - Destination: **W10Ent_x64**
 - Summary: click **Next**
 - Progress: wait for files to be copied
 - Confirmation: click **Finish**

For purposes of this test lab, we will only add the prerequisite .NET Framework feature. Commercial applications (ex: Microsoft Office) will not be added to the deployment share. For information about adding applications, see the [Add applications](#) section of the [Create a Windows 10 reference image](#) topic in the TechNet library.

11. The next step is to create a task sequence to reference the operating system that was imported. To create a task sequence, right-click the **Task Sequences** node and then click **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - Task sequence ID: **REFW10X64-001**
 - Task sequence name: **Windows 10 Enterprise x64 Default Image**
 - Task sequence comments: **Reference Build**
 - Template: **Standard Client Task Sequence**
 - Select OS: click **Windows 10 Enterprise Evaluation in W10Ent_x64 install.wim**

- Specify Product Key: **Do not specify a product key at this time**
 - Full Name: **Contoso**
 - Organization: **Contoso**
 - Internet Explorer home page: <http://www.contoso.com>
 - Admin Password: **Do not specify an Administrator password at this time**
 - Summary: click **Next**
 - Confirmation: click **Finish**
12. Edit the task sequence to add the Microsoft NET Framework 3.5, which is required by many applications. To edit the task sequence, double-click **Windows 10 Enterprise x64 Default Image** that was created in the previous step.
 13. Click the **Task Sequence** tab. Under **State Restore** click **Tatto** to highlight it, then click **Add** and choose **New Group**.
 14. On the Properties tab of the group that was created in the previous step, change the Name from **New Group** to **Custom Tasks (Pre-Windows Update)** and then click **Apply**. Click another location in the window to see the name change.
 15. Click the **Custom Tasks (Pre-Windows Update)** group again, click **Add**, point to **Roles**, and then click **Install Roles and Features**.
 16. Under **Select the roles and features that should be installed**, select **.NET Framework 3.5 (includes .NET 2.0 and 3.0)** and then click **Apply**.
 17. Enable Windows Update in the task sequence by clicking the **Windows Update (Post-Application Installation)** step, clicking the **Options** tab, and clearing the **Disable this step** checkbox.

Note: Since we are not installing applications in this test lab, there is no need to enable the Windows Update Pre-Application Installation step. However, you should enable this step if you are also installing applications.

18. Click **OK** to complete editing the task sequence.
19. The next step is to configure the MDT deployment share rules. To configure rules in the Deployment Workbench, right-click **MDT build lab (C:\MDTBuildLab)** and click **Properties**, and then click the **Rules** tab.
20. Replace the default rules with the following text:

```

[Settings]
Priority=Default

[Default]
_SMSTSORGNAME=Contoso
UserDataLocation=NONE
DoCapture=YES
OSInstall=Y
AdminPassword=pass@word1
TimeZoneName=Pacific Standard Time
OSDComputername=#Left("PC-%SerialNumber%",7)#
JoinWorkgroup=WORKGROUP
HideShell=YES
FinishAction=SHUTDOWN
DoNotCreateExtraPartition=YES
ApplyGPOPack=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=YES
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=YES
SkipBitLocker=YES
SkipSummary=YES
SkipRoles=YES
SkipCapture=NO
SkipFinalSummary=NO

```

21. Click **Apply** and then click **Edit Bootstrap.ini**. Replace the contents of the Bootstrap.ini file with the following text, and save the file:

```

[Settings]
Priority=Default

[Default]
DeployRoot=\\SRV1\MDTBuildLab$
UserDomain=CONTOSO
UserID=MDT_BA
UserPassword=pass@word1
SkipBDDWelcome=YES

```

22. Click **OK** to complete the configuration of the deployment share.
23. Right-click **MDT build lab (C:\MDTBuildLab)** and then click **Update Deployment Share**.
24. Accept all default values in the Update Deployment Share Wizard by clicking **Next** twice. The update process will take 5 to 10 minutes. When it has completed, click **Finish**.
25. Copy **c:\MDTBuildLab\Boot\LiteTouchPE_x86.iso** on SRV1 to the **c:\VHD** directory on the Hyper-V host computer. Note that in MDT, the x86 boot image can deploy both x86 and x64 operating systems, except on computers based on Unified Extensible Firmware Interface (UEFI).

Hint: To copy the file, right-click the **LiteTouchPE_x86.iso** file and click **Copy** on SRV1, then open the **c:\VHD** folder on the Hyper-V host, right-click inside the folder and click **Paste**.

26. Open a Windows PowerShell prompt on the Hyper-V host computer and type the following commands:

```
New-VM REFW10X64-001 -SwitchName poc-internal -NewVHDPath "c:\VHD\REFW10X64-001.vhdx" -NewVHDSizeBytes 60GB
Set-VMMemory REFW10X64-001 -DynamicMemoryEnabled $true -MinimumBytes 1024MB -MaximumBytes 1024MB -Buffer 20
Set-VMdvdDrive REFW10X64-001 -Path c:\VHD\LiteTouchPE_x86.iso
Start-VM REFW10X64-001
vmconnect localhost REFW10X64-001
```

The VM will require a few minutes to prepare devices and boot from the LiteTouchPE_x86.iso file.

27. In the Windows Deployment Wizard, select **Windows 10 Enterprise x64 Default Image**, and then click **Next**.
28. Accept the default values on the Capture Image page, and click **Next**. Operating system installation will complete after 5 to 10 minutes, and then the VM will reboot automatically. Allow the system to boot normally (do not press a key). The process is fully automated.

Additional system restarts will occur to complete updating and preparing the operating system. Setup will complete the following procedures:

- Install the Windows 10 Enterprise operating system.
- Install added applications, roles, and features.
- Update the operating system using Windows Update (or WSUS if optionally specified).
- Stage Windows PE on the local disk.
- Run System Preparation (Sysprep) and reboot into Windows PE.
- Capture the installation to a Windows Imaging (WIM) file.
- Turn off the virtual machine.

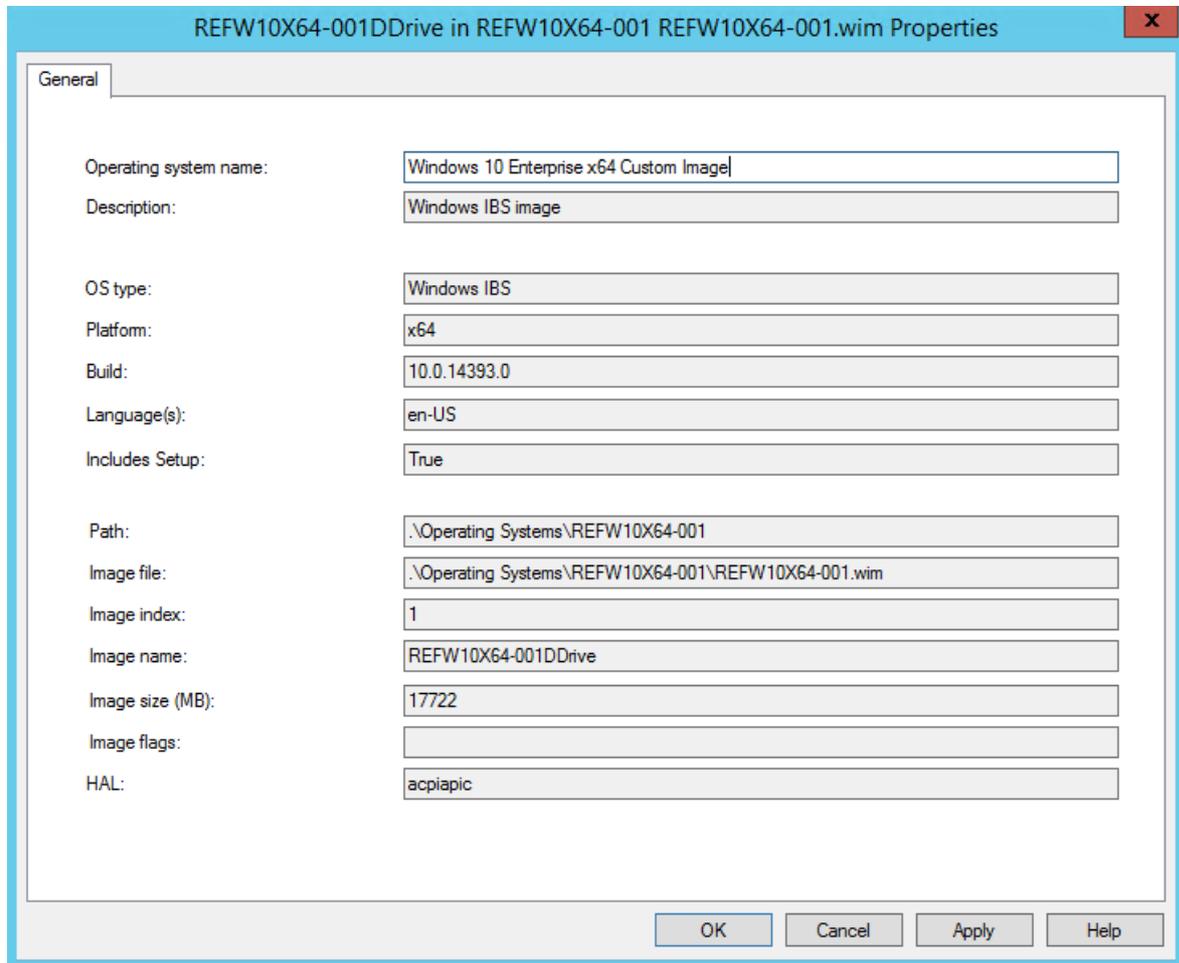
This step requires from 30 minutes to 2 hours, depending on the speed of the Hyper-V host. After some time, you will have a Windows 10 Enterprise x64 image that is fully patched and has run through Sysprep. The image is located in the C:\MDTBuildLab\Captures folder on your deployment server (SRV1). The file name is **REFW10X64-001.wim**.

Deploy a Windows 10 image using MDT

This procedure will demonstrate how to deploy the reference image to the PoC environment using MDT.

1. On SRV1, open the MDT Deployment Workbench console, right-click **Deployment Shares**, and then click **New Deployment Share**. Use the following values in the New Deployment Share Wizard:
 - **Deployment share path:** C:\MDTProd
 - **Share name:** MDTProd\$
 - **Deployment share description:** MDT Production
 - **Options:** accept the default
2. Click **Next**, verify the new deployment share was added successfully, then click **Finish**.
3. In the Deployment Workbench console, expand the MDT Production deployment share, right-click **Operating Systems**, and then click **New Folder**. Name the new folder **Windows 10** and complete the wizard using default values.
4. Right-click the **Windows 10** folder created in the previous step, and then click **Import Operating System**.
5. On the **OS Type** page, choose **Custom image file** and then click **Next**.
6. On the Image page, browse to the **C:\MDTBuildLab\Captures\REFW10X64-001.wim** file created in the previous procedure, click **Open**, and then click **Next**.
7. On the Setup page, select **Copy Windows 7, Windows Server 2008 R2, or later setup files from the specified path**.

8. Under **Setup source directory**, browse to **C:\MDTBuildLab\Operating Systems\W10Ent_x64** click **OK** and then click **Next**.
9. On the Destination page, accept the default Destination directory name of **REFW10X64-001**, click **Next** twice, wait for the import process to complete, and then click **Finish**.
10. In the **Operating Systems > Windows 10** node, double-click the operating system that was added to view its properties. Change the operating system name to **Windows 10 Enterprise x64 Custom Image** and then click **OK**. See the following example:



Create the deployment task sequence

1. Using the Deployment Workbench, right-click **Task Sequences** under the **MDT Production** node, click **New Folder** and create a folder with the name: **Windows 10**.
2. Right-click the **Windows 10** folder created in the previous step, and then click **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - Task sequence ID: W10-X64-001
 - Task sequence name: Windows 10 Enterprise x64 Custom Image
 - Task sequence comments: Production Image
 - Select Template: Standard Client Task Sequence
 - Select OS: Windows 10 Enterprise x64 Custom Image
 - Specify Product Key: Do not specify a product key at this time
 - Full Name: Contoso
 - Organization: Contoso
 - Internet Explorer home page: <http://www.contoso.com>
 - Admin Password: pass@word1

Configure the MDT production deployment share

1. On SRV1, open an elevated Windows PowerShell prompt and type the following commands:

```
copy-item "C:\Program Files\Microsoft Deployment Toolkit\Templates\Bootstrap.ini"  
C:\MDTProd\Control\Bootstrap.ini -Force  
copy-item "C:\Program Files\Microsoft Deployment Toolkit\Templates\CustomSettings.ini"  
C:\MDTProd\Control\CustomSettings.ini -Force
```

2. In the Deployment Workbench console on SRV1, right-click the **MDT Production** deployment share and then click **Properties**.
3. Click the **Rules** tab and replace the rules with the following text (don't click OK yet):

```
[Settings]  
Priority=Default  
  
[Default]  
_SMSTSORGNAME=Contoso  
OSInstall=YES  
UserDataLocation=AUTO  
TimeZoneName=Pacific Standard Time  
OSDComputername=#Left("PC-%SerialNumber%",7)#  
AdminPassword=pass@word1  
JoinDomain=contoso.com  
DomainAdmin=administrator  
DomainAdminDomain=CONTOSO  
DomainAdminPassword=pass@word1  
ScanStateArgs=/ue:* \* /ui:CONTOSO \*  
USMTMigFiles001=MigApp.xml  
USMTMigFiles002=MigUser.xml  
HideShell=YES  
ApplyGPOPack=NO  
SkipAppsOnUpgrade=NO  
SkipAdminPassword=YES  
SkipProductKey=YES  
SkipComputerName=YES  
SkipDomainMembership=YES  
SkipUserData=YES  
SkipLocaleSelection=YES  
SkipTaskSequence=NO  
SkipTimeZone=YES  
SkipApplications=NO  
SkipBitLocker=YES  
SkipSummary=YES  
SkipCapture=YES  
SkipFinalSummary=NO  
EventService=http://SRV1:9800
```

Note: The contents of the Rules tab are added to c:\MDTProd\Control\CustomSettings.ini.

In this example a **MachineObjectOU** entry is not provided. Normally this entry describes the specific OU where new client computer objects are created in Active Directory. However, for the purposes of this test lab clients are added to the default computers OU, which requires that this parameter be unspecified.

If desired, edit the follow line to include or exclude other users when migrating settings. Currently, the command is set to user exclude (ue) all users except for CONTOSO users specified by the user include option (ui):

```
ScanStateArgs=/ue:*\* /ui:CONTOSO\*
```

For example, to migrate **all** users on the computer, replace this line with the following:

```
ScanStateArgs=/all
```

For more information, see [ScanState Syntax](#).

4. Click **Edit Bootstap.ini** and replace text in the file with the following text:

```
[Settings]
Priority=Default

[Default]
DeployRoot=\\SRV1\MDTProd$
UserDomain=CONTOSO
UserID=MDT_BA
UserPassword=pass@word1
SkipBDDWelcome=YES
```

5. Click **OK** when finished.

Update the deployment share

1. Right-click the **MDT Production** deployment share and then click **Update Deployment Share**.
2. Use the default options for the Update Deployment Share Wizard. The update process requires 5 to 10 minutes to complete.
3. Click **Finish** when the update is complete.

Enable deployment monitoring

1. In the Deployment Workbench console, right-click **MDT Production** and then click **Properties**.
2. On the **Monitoring** tab, select the **Enable monitoring for this deployment share** checkbox, and then click **OK**.
3. Verify the monitoring service is working as expected by opening the following link on SRV1 in Internet Explorer: <http://localhost:9800/MDTMonitorEvent/>. If you do not see "**You have created a service**" at the top of the page, see [Troubleshooting MDT 2012 Monitoring](#).
4. Close Internet Explorer.

Configure Windows Deployment Services

1. Initialize Windows Deployment Services (WDS) by typing the following command at an elevated Windows PowerShell prompt on SRV1:

```
WDSUTIL /Verbose /Progress /Initialize-Server /Server:SRV1 /RemInst:"C:\RemoteInstall"
WDSUTIL /Set-Server /AnswerClients:All
```

2. Click **Start**, type **Windows Deployment**, and then click **Windows Deployment Services**.
3. In the Windows Deployment Services console, expand **Servers**, expand **SRV1.contoso.com**, right-click **Boot Images**, and then click **Add Boot Image**.
4. Browse to the **C:\MDTProd\Boot\LiteTouchPE_x64.wim** file, click **Open**, click **Next**, and accept the defaults in the Add Image Wizard. Click **Finish** to complete adding a boot image.

Deploy the client image

1. Before using WDS to deploy a client image, you must temporarily disable the external network adapter on SRV1. This is just an artifact of the lab environment. In a typical deployment environment WDS would not be installed on the default gateway.

Note: Do not disable the *internal* network interface. To quickly view IP addresses and interface names configured on the VM, type **Get-NetIPAddress | ft interfacealias, ipaddress**

Assuming the external interface is named "Ethernet 2", to disable the *external* interface on SRV1, open a Windows PowerShell prompt on SRV1 and type the following command:

```
Disable-NetAdapter "Ethernet 2" -Confirm:$false
```

Wait until the disable-netadapter command completes before proceeding.

2. Next, switch to the Hyper-V host and open an elevated Windows PowerShell prompt. Create a generation 2 VM on the Hyper-V host that will load its OS using PXE. To create this VM, type the following commands at an elevated Windows PowerShell prompt:

```
New-VM -Name "PC2" -NewVHDPATH "c:\vhd\pc2.vhdx" -NewVHDSIZEBYTES 60GB -SwitchName poc-internal -  
BootDevice NetworkAdapter -Generation 2  
Set-VMemory -VMName "PC2" -DynamicMemoryEnabled $true -MinimumBytes 720MB -MaximumBytes 2048MB -Buffer  
20
```

Dynamic memory is configured on the VM to conserve resources. However, this can cause memory allocation to be reduced past what is required to install an operating system. If this happens, reset the VM and begin the OS installation task sequence immediately. This ensures the VM memory allocation is not decreased too much while it is idle.

3. Start the new VM and connect to it:

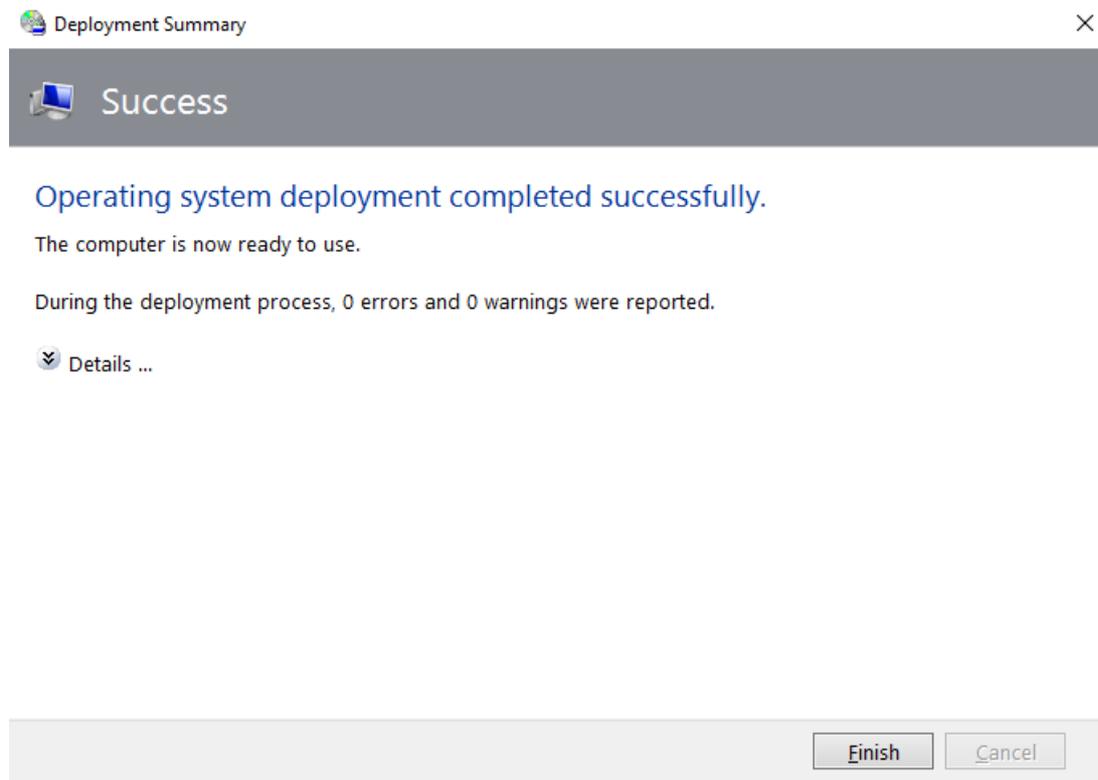
```
Start-VM PC2  
vmconnect localhost PC2
```

4. When prompted, hit ENTER to start the network boot process.
5. In the Windows Deployment Wizard, choose the **Windows 10 Enterprise x64 Custom Image** and then click **Next**.
6. After MDT lite touch installation has started, be sure to re-enable the external network adapter on SRV1. This is needed so the client can use Windows Update after operating system installation is complete. To re-enable the external network interface, open an elevated Windows PowerShell prompt on SRV1 and type the following command:

```
Enable-NetAdapter "Ethernet 2"
```

7. On SRV1, in the Deployment Workbench console, click on **Monitoring** and view the status of installation. Right-click **Monitoring** and click **Refresh** if no data is displayed.
8. OS installation requires about 10 minutes. When the installation is complete, the system will reboot automatically, configure devices, and install updates, requiring another 10-20 minutes. When the new client

computer is finished updating, click **Finish**. You will be automatically signed in to the local computer as administrator.



This completes the demonstration of how to deploy a reference image to the network. To conserve resources, turn off the PC2 VM before starting the next section.

Refresh a computer with Windows 10

This section will demonstrate how to export user data from an existing client computer, wipe the computer, install a new operating system, and then restore user data and settings. The scenario will use PC1, a computer that was cloned from a physical device to a VM, as described in [Step by step guide: Deploy Windows 10 in a test lab](#).

1. If the PC1 VM is not already running, then start and connect to it:

```
Start-VM PC1
vmconnect localhost PC1
```

2. Switch back to the Hyper-V host and create a checkpoint for the PC1 VM so that it can easily be reverted to its current state for troubleshooting purposes and to perform additional scenarios. Checkpoints are also known as snapshots. To create a checkpoint for the PC1 VM, type the following command at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Checkpoint-VM -Name PC1 -SnapshotName BeginState
```

3. Sign on to PC1 using the CONTOSO\Administrator account.

Specify **contoso\administrator** as the user name to ensure you do not sign on using the local administrator account. You must sign in with this account so that you have access to the deployment share.

4. Open an elevated command prompt on PC1 and type the following:

```
cscript \\SRV1\MDTProd$\Scripts\Litetouch.vbs
```

Note: Litetouch.vbs must be able to create the C:\MININT directory on the local computer.

5. Choose the **Windows 10 Enterprise x64 Custom Image** and then click **Next**.
6. Choose **Do not back up the existing computer** and click **Next**.

Note: The USMT will still back up the computer.

7. Lite Touch Installation will perform the following actions:

- Back up user settings and data using USMT.
- Install the Windows 10 Enterprise X64 operating system.
- Update the operating system via Windows Update.
- Restore user settings and data using USMT.

You can review the progress of installation on SRV1 by clicking on the **Monitoring** node in the deployment workbench. When OS installation is complete, the computer will restart, set up devices, and configure settings.

8. Sign in with the CONTOSO\Administrator account and verify that all CONTOSO domain user accounts and data have been migrated to the new operating system, or other user accounts as specified [previously](#).
9. Create another checkpoint for the PC1 VM so that you can review results of the computer refresh later. To create a checkpoint, type the following command at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Checkpoint-VM -Name PC1 -SnapshotName RefreshState
```

10. Restore the PC1 VM to its previous state in preparation for the replace procedure. To restore a checkpoint, type the following command at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Restore-VMSnapshot -VMName PC1 -Name BeginState -Confirm:$false  
Start-VM PC1  
vmconnect localhost PC1
```

11. Sign in to PC1 using the contoso\administrator account.

Replace a computer with Windows 10

At a high level, the computer replace process consists of:

- A special replace task sequence that runs the USMT backup and an optional full Windows Imaging (WIM) backup.
- A standard OS deployment on a new computer. At the end of the deployment, the USMT backup from the old computer is restored.

Create a backup-only task sequence

1. On SRV1, in the deployment workbench console, right-click the MDT Production deployment share, click **Properties**, click the **Rules** tab, and change the line **SkipUserData=YES** to **SkipUserData=NO**.
2. Click **OK**, right-click **MDT Production**, click **Update Deployment Share** and accept the default options in the wizard to update the share.

3. Type the following commands at an elevated Windows PowerShell prompt on SRV1:

```
New-Item -Path C:\MigData -ItemType directory
New-SmbShare -Name MigData$ -Path C:\MigData -ChangeAccess EVERYONE
icacls C:\MigData /grant '"contoso\administrator":(OI)(CI)(M)'
```

4. On SRV1 in the deployment workbench, under **MDT Production**, right-click the **Task Sequences** node, and click **New Folder**.

5. Name the new folder **Other**, and complete the wizard using default options.

6. Right-click the **Other** folder and then click **New Task Sequence**. Use the following values in the wizard:

- **Task sequence ID:** REPLACE-001
- **Task sequence name:** Backup Only Task Sequence
- **Task sequence comments:** Run USMT to back up user data and settings
- **Template:** Standard Client Replace Task Sequence (note: this is not the default template)

7. Accept defaults for the rest of the wizard and then click **Finish**. The replace task sequence will skip OS selection and settings.

8. Open the new task sequence that was created and review it. Note the type of capture and backup tasks that are present. Click **OK** when you are finished reviewing the task sequence.

Run the backup-only task sequence

1. If you are not already signed on to PC1 as **contoso\administrator**, sign in using this account. To verify the currently signed in account, type the following command at an elevated command prompt:

```
whoami
```

2. To ensure a clean environment before running the backup task sequence, type the following at an elevated Windows PowerShell prompt on PC1:

```
Remove-Item c:\minint -recurse
Remove-Item c:\_SMSTaskSequence -recurse
Restart-Computer
```

3. Sign in to PC1 using the contoso\administrator account, and then type the following at an elevated command prompt:

```
cscript \\SRV1\MDTProd$\Scripts\Litetouch.vbs
```

4. Complete the deployment wizard using the following:

- **Task Sequence:** Backup Only Task Sequence
- **User Data:** Specify a location: **\\SRV1\MigData\$PC1**
- **Computer Backup:** Do not back up the existing computer.

5. While the task sequence is running on PC1, open the deployment workbench console on SRV1 and click the **Monitoring* node. Press F5 to refresh the console, and view the status of current tasks.

6. On PC1, verify that **The user state capture was completed successfully** is displayed, and click **Finish** when the capture is complete.

7. On SRV1, verify that the file **USMT.MIG** was created in the **C:\MigData\PC1\USMT** directory. See the following example:

```
PS C:\> dir C:\MigData\PC1\USMT

Directory: C:\MigData\PC1\USMT

Mode                LastWriteTime         Length Name
----                -
-a---             9/6/2016  11:34 AM     14248685 USMT.MIG
```

Deploy PC3

8. On the Hyper-V host, type the following commands at an elevated Windows PowerShell prompt:

```
New-VM -Name "PC3" -NewVHDPATH "c:\vhd\pc3.vhdx" -NewVHDSIZEBYTES 60GB -SwitchName poc-internal -
BootDevice NetworkAdapter -Generation 2
Set-VMemory -VMName "PC3" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes 2048MB -Buffer
20
```

9. Temporarily disable the external network adapter on SRV1 again, so that we can successfully boot PC3 from WDS. To disable the adapter, type the following command at an elevated Windows PowerShell prompt on SRV1:

```
Disable-NetAdapter "Ethernet 2" -Confirm:$false
```

As mentioned previously, ensure that you disable the **external** network adapter, and wait for the command to complete before proceeding.

10. Start and connect to PC3 by typing the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Start-VM PC3
vmconnect localhost PC3
```

11. When prompted, press ENTER for network boot.
12. On PC3, use the following settings for the Windows Deployment Wizard:
- **Task Sequence:** Windows 10 Enterprise x64 Custom Image
 - **Move Data and Settings:** Do not move user data and settings
 - **User Data (Restore):** Specify a location: **\\SRV1\MigData\$PC1**
13. When OS installation has started on PC1, re-enable the external network adapter on SRV1 by typing the following command on SRV1:

```
Enable-NetAdapter "Ethernet 2"
```

14. Setup will install the Windows 10 Enterprise operating system, update via Windows Update, and restore the user settings and data from PC1.
15. When PC3 has completed installing the OS, sign in to PC3 using the contoso\administrator account. When the PC completes updating, click **Finish**.
16. Verify that settings have been migrated from PC1. This completes demonstration of the replace procedure.
17. Shut down PC3 in preparation for the [next](#) procedure.

Troubleshooting logs, events, and utilities

Deployment logs are available on the client computer in the following locations:

- Before the image is applied: X:\MININT\SMSOSD\OSDLOGS
- After the system drive has been formatted: C:\MININT\SMSOSD\OSDLOGS
- After deployment: %WINDIR%\TEMP\DeploymentLogs

You can review WDS events in Event Viewer at: **Applications and Services Logs > Microsoft > Windows > Deployment-Services-Diagnostics**. By default, only the **Admin** and **Operational** logs are enabled. To enable other logs, right-click the log and then click **Enable Log**.

Tools for viewing log files, and to assist with troubleshooting are available in the [System Center 2012 R2 Configuration Manager Toolkit](#)

Also see [Resolve Windows 10 upgrade errors](#) for detailed troubleshooting information.

Related Topics

[Microsoft Deployment Toolkit](#)

[Prepare for deployment with MDT](#)

Deploy Windows 10 in a test lab using System Center Configuration Manager

6/26/2019 • 44 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Important: This guide leverages the proof of concept (PoC) environment, and some settings that are configured in the following guides:

- [Step by step guide: Deploy Windows 10 in a test lab](#)
- [Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit](#)

Please complete all steps in these guides before attempting the procedures in this guide. If you wish to skip the Windows 10 deployment procedures in the MDT guide and move directly to this guide, you must at least install MDT and the Windows ADK before performing procedures in this guide. All steps in the first guide are required before attempting the procedures in this guide.

The PoC environment is a virtual network running on Hyper-V with three virtual machines (VMs):

- **DC1:** A contoso.com domain controller, DNS server, and DHCP server.
- **SRV1:** A dual-homed contoso.com domain member server, DNS server, and default gateway providing NAT service for the PoC network.
- **PC1:** A contoso.com member computer running Windows 7, Windows 8, or Windows 8.1 that has been cloned from a physical computer on your corporate network for testing purposes. This guide leverages the Hyper-V server role to perform procedures. If you do not complete all steps in a single session, consider using [checkpoints](#) and [saved states](#) to pause, resume, or restart your work.

Multiple features and services are installed on SRV1 in this guide. This is not a typical installation, and is only done to set up a lab environment with a bare minimum of resources. However, if less than 4 GB of RAM is allocated to SRV1 in the Hyper-V console, some procedures will be extremely slow to complete. If resources are limited on the Hyper-V host, consider reducing RAM allocation on DC1 and PC1, and then increasing the RAM allocation on SRV1. You can adjust RAM allocation for a VM by right-clicking the VM in the Hyper-V Manager console, clicking **Settings**, clicking **Memory**, and modifying the value next to **Maximum RAM**.

In this guide

This guide provides end-to-end instructions to install and configure System Center Configuration Manager, and use it to deploy a Windows 10 image. Depending on the speed of your Hyper-V host, the procedures in this guide will require 6-10 hours to complete.

Topics and procedures in this guide are summarized in the following table. An estimate of the time required to complete each procedure is also provided. Time required to complete procedures will vary depending on the resources available to the Hyper-V host and assigned to VMs, such as processor speed, memory allocation, disk speed, and network speed.

Topic	Description	Time
-------	-------------	------

Install prerequisites	Install prerequisite Windows Server roles and features, download, install and configure SQL Server, configure firewall rules, and install the Windows ADK.	60 minutes
Install System Center Configuration Manager	Download System Center Configuration Manager, configure prerequisites, and install the package.	45 minutes
Download MDOP and install DaRT	Download the Microsoft Desktop Optimization Pack 2015 and install DaRT 10.	15 minutes
Prepare for Zero Touch installation	Prerequisite procedures to support Zero Touch installation.	60 minutes
Create a boot image for Configuration Manager	Use the MDT wizard to create the boot image in Configuration Manager.	20 minutes
Create a Windows 10 reference image	This procedure can be skipped if it was done previously, otherwise instructions are provided to create a reference image.	0-60 minutes
Add a Windows 10 operating system image	Add a Windows 10 operating system image and distribute it.	10 minutes
Create a task sequence	Create a Configuration Manager task sequence with MDT integration using the MDT wizard	15 minutes
Finalize the operating system configuration	Enable monitoring, configure rules, and distribute content.	30 minutes
Deploy Windows 10 using PXE and Configuration Manager	Deploy Windows 10 using Configuration Manager deployment packages and task sequences.	60 minutes
Replace a client with Windows 10 using Configuration Manager	Replace a client computer with Windows 10 using Configuration Manager.	90 minutes
Refresh a client with Windows 10 using Configuration Manager	Use a task sequence to refresh a client with Windows 10 using Configuration Manager and MDT	90 minutes

Install prerequisites

1. Before installing System Center Configuration Manager, we must install prerequisite services and features. Type the following command at an elevated Windows PowerShell prompt on SRV1:

```
Install-WindowsFeature Web-Windows-Auth,Web-ISAPI-Ext,Web-Metabase,Web-WMI,BITS,RDC,NET-Framework-Features,Web-Asp-Net,Web-Asp-Net45,NET-HTTP-Activation,NET-Non-HTTP-Activ
```

If the request to add features fails, retry the installation by typing the command again.

2. Download [SQL Server 2014 SP2](#) from the Microsoft Evaluation Center as an .ISO file on the Hyper-V host computer. Save the file to the **C:\VHD** directory.
3. When you have downloaded the file **SQLServer2014SP2-FullSlipstream-x64-ENU.iso** and placed it in the C:\VHD directory, type the following command at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Set-VMDvdDrive -VMName SRV1 -Path c:\VHD\SQLServer2014SP2-FullSlipstream-x64-ENU.iso
```

This command mounts the .ISO file to drive D on SRV1.

4. Type the following command at an elevated Windows PowerShell prompt on SRV1 to install SQL Server:

```
D:\setup.exe /q /ACTION=Install /ERRORREPORTING="False"
/FEATURES=SQLENGINE,RS,IS,SSMS,TOOLS,ADV_SSMS,CONN /INSTANCENAME=MSSQLSERVER /INSTANCEDIR="C:\Program
Files\Microsoft SQL Server" /SQLSVCACCOUNT="NT AUTHORITY\System"
/SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS" /SQLSVCSTARTUPTYPE=Automatic /AGTSVCACCOUNT="NT
AUTHORITY\SYSTEM" /AGTSVCSTARTUPTYPE=Automatic /RSSVCACCOUNT="NT AUTHORITY\System"
/RSSVCSTARTUPTYPE=Automatic /ISSVCACCOUNT="NT AUTHORITY\System" /ISSVCSTARTUPTYPE=Disabled
/ASCOLLATION="Latin1_General_CI_AS" /SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS" /TCPENABLED="1"
/NPENABLED="1" /IAcceptSQLServerLicenseTerms
```

Installation will take several minutes. When installation is complete, the following output will be displayed:

```
Microsoft (R) SQL Server 2014 12.00.5000.00
Copyright (c) Microsoft Corporation. All rights reserved.

Microsoft (R) .NET Framework CasPol 2.0.50727.7905
Copyright (c) Microsoft Corporation. All rights reserved.

Success
Microsoft (R) .NET Framework CasPol 2.0.50727.7905
Copyright (c) Microsoft Corporation. All rights reserved.

Success
One or more affected files have operations pending.
You should restart your computer to complete this process.
PS C:\>
```

5. Type the following commands at an elevated Windows PowerShell prompt on SRV1:

```
New-NetFirewallRule -DisplayName "SQL Server" -Direction Inbound -Protocol TCP -LocalPort 1433 -Action
allow
New-NetFirewallRule -DisplayName "SQL Admin Connection" -Direction Inbound -Protocol TCP -LocalPort
1434 -Action allow
New-NetFirewallRule -DisplayName "SQL Database Management" -Direction Inbound -Protocol UDP -LocalPort
1434 -Action allow
New-NetFirewallRule -DisplayName "SQL Service Broker" -Direction Inbound -Protocol TCP -LocalPort 4022
-Action allow
New-NetFirewallRule -DisplayName "SQL Debugger/RPC" -Direction Inbound -Protocol TCP -LocalPort 135 -
Action allow
```

6. Download and install the latest [Windows Assessment and Deployment Kit \(ADK\)](#) on SRV1 using the default installation settings. The current version is the ADK for Windows 10, version 1703. Installation might require several minutes to acquire all components.

Install System Center Configuration Manager

1. On SRV1, temporarily disable IE Enhanced Security Configuration for Administrators by typing the following commands at an elevated Windows PowerShell prompt:

```
$AdminKey = "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A7-37EF-4b3f-8CFC-
4F3A74704073}"
Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 0
Stop-Process -Name Explorer
```

2. Download [System Center Configuration Manager and Endpoint Protection](#) on SRV1 (download the executable file anywhere on SRV1), double-click the file, enter **C:\configmgr** for **Unzip to folder**, and click **Unzip**. The C:\configmgr directory will be automatically created. Click **OK** and then close the **WinZip Self-Extractor** dialog box when finished.

- Before starting the installation, verify that WMI is working on SRV1. See the following examples. Verify that **Running** is displayed under **Status** and **True** is displayed next to **TcpTestSucceeded**:

```
Get-Service Winmgmt

Status   Name           DisplayName
-----   -
Running  Winmgmt        Windows Management Instrumentation

Test-NetConnection -ComputerName 192.168.0.2 -Port 135 -InformationLevel Detailed

ComputerName           : 192.168.0.2
RemoteAddress           : 192.168.0.2
RemotePort              : 135
AllNameResolutionResults :
MatchingIPsecRules      :
NetworkIsolationContext : Internet
InterfaceAlias          : Ethernet
SourceAddress           : 192.168.0.2
NetRoute (NextHop)     : 0.0.0.0
PingSucceeded           : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded        : True
```

You can also verify WMI using the WMI console by typing **wmimgmt.msc**, right-clicking **WMI Control (Local)** in the console tree, and then clicking **Properties**.

If the WMI service is not started, attempt to start it or reboot the computer. If WMI is running but errors are present, see [WMI Diagnostics](#) for troubleshooting information.

- To extend the Active Directory schema, type the following command at an elevated Windows PowerShell prompt:

```
cmd /c C:\configmgr\SMSSETUP\BIN\X64\extadsch.exe
```

- Temporarily switch to the DC1 VM, and type the following command at an elevated command prompt on DC1:

```
adsiedit.msc
```

- Right-click **ADSI Edit**, click **Connect to**, select **Default (Domain or server that you logged in to)** under **Computer** and then click **OK**.
- Expand **Default naming context > DC=contoso,DC=com**, and then in the console tree right-click **CN=System**, point to **New**, and then click **Object**.
- Click **container** and then click **Next**.
- Next to **Value**, type **System Management**, click **Next**, and then click **Finish**.
- Right-click **CN=system Management** and then click **Properties**.
- On the **Security** tab, click **Add**, click **Object Types**, select **Computers**, and click **OK**.
- Under **Enter the object names to select**, type **SRV1** and click **OK**.
- The **SRV1** computer account will be highlighted, select **Allow** next to **Full control**.
- Click **Advanced**, click **SRV1 (CONTOSO\SRV1\$)** and click **Edit**.

- Next to **Applies to**, choose **This object and all descendant objects**, and then click **OK** three times.
- Close the ADSI Edit console and switch back to SRV1.
- To start Configuration Manager installation, type the following command at an elevated Windows PowerShell prompt on SRV1:

```
cmd /c C:\configmgr\SMSSETUP\BIN\X64\Setup.exe
```

- Provide the following in the System Center Configuration Manager Setup Wizard:
 - **Before You Begin:** Read the text and click *Next*.
 - **Getting Started:** Choose **Install a Configuration Manager primary site** and select the **Use typical installation options for a stand-alone primary site** checkbox.
 - Click **Yes** in response to the popup window.
 - **Product Key:** Choose **Install the evaluation edition of this Product**.
 - **Microsoft Software License Terms:** Read the terms and then select the **I accept these license terms** checkbox.
 - **Prerequisite Licenses:** Review license terms and select all three checkboxes on the page.
 - **Prerequisite Downloads:** Choose **Download required files** and enter **c:\windows\temp** next to **Path**.
 - **Site and Installation Settings:** Site code: **PS1**, Site name: **Contoso**.
 - use default settings for all other options
 - **Usage Data:** Read the text and click **Next**.
 - **Service Connection Point Setup:** Accept the default settings (SRV1.contoso.com is automatically added under Select a server to use).
 - **Settings Summary:** Review settings and click **Next**.
 - **Prerequisite Check:** No failures should be listed. Ignore any warnings and click **Begin Install**.

There should be at most three warnings present: WSUS on site server, configuration for SQL Server memory usage, and SQL Server process memory allocation. These warnings can safely be ignored in this test environment.

Depending on the speed of the Hyper-V host and resources allocated to SRV1, installation can require approximately one hour. Click **Close** when installation is complete.

- If desired, re-enable IE Enhanced Security Configuration at this time on SRV1:

```
Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 1  
Stop-Process -Name Explorer
```

Download MDOP and install DaRT

IMPORTANT

This step requires an MSDN subscription or volume licence agreement. For more information, see [Ready for Windows 10: MDOP 2015 and more tools are now available](#). If your organization qualifies and does not already have an MSDN subscription, you can obtain a [free MSDN subscription with BizSpark](#).

- Download the [Microsoft Desktop Optimization Pack 2015](#) to the Hyper-V host using an MSDN subscription. Download the .ISO file

(mu_microsoft_desktop_optimization_pack_2015_x86_x64_dvd_5975282.iso, 2.79 GB) to the C:\VHD directory on the Hyper-V host.

2. Type the following command at an elevated Windows PowerShell prompt on the Hyper-V host to mount the MDOP file on SRV1:

```
Set-VMdvdDrive -VMName SRV1 -Path  
c:\VHD\mu_microsoft_desktop_optimization_pack_2015_x86_x64_dvd_5975282.iso
```

3. Type the following command at an elevated Windows PowerShell prompt on SRV1:

```
cmd /c "D:\DaRT\DaRT 10\Installers\en-us\x64\MSDaRT100.msi"
```

4. Install DaRT 10 using default settings.

5. Type the following commands at an elevated Windows PowerShell prompt on SRV1:

```
Copy-Item "C:\Program Files\Microsoft DaRT\v10\Toolsx64.cab" -Destination "C:\Program Files\Microsoft  
Deployment Toolkit\Templates\Distribution\Tools\x64"  
Copy-Item "C:\Program Files\Microsoft DaRT\v10\Toolsx86.cab" -Destination "C:\Program Files\Microsoft  
Deployment Toolkit\Templates\Distribution\Tools\x86"
```

Prepare for Zero Touch installation

This section contains several procedures to support Zero Touch installation with System Center Configuration Manager.

Create a folder structure

1. Type the following commands at a Windows PowerShell prompt on SRV1:

```
New-Item -ItemType Directory -Path "C:\Sources\OSD\Boot"  
New-Item -ItemType Directory -Path "C:\Sources\OSD\OS"  
New-Item -ItemType Directory -Path "C:\Sources\OSD\Settings"  
New-Item -ItemType Directory -Path "C:\Sources\OSD\Branding"  
New-Item -ItemType Directory -Path "C:\Sources\OSD\MDT"  
New-Item -ItemType Directory -Path "C:\Logs"  
New-SmbShare -Name Sources$ -Path C:\Sources -ChangeAccess EVERYONE  
New-SmbShare -Name Logs$ -Path C:\Logs -ChangeAccess EVERYONE
```

Enable MDT ConfigMgr integration

1. On SRV1, click **Start**, type **configmgr**, and then click **Configure ConfigMgr Integration**.
2. Type **PS1** next to **Site code**, and then click **Next**.
3. Verify **The process completed successfully** is displayed, and then click **Finish**.

Configure client settings

1. On SRV1, click **Start**, type **configuration manager**, right-click **Configuration Manager Console**, and then click **Pin to Taskbar**.
2. Click **Desktop**, and then launch the Configuration Manager console from the taskbar.
3. If the console notifies you that an update is available, click **OK**. It is not necessary to install updates to complete this lab.
4. In the console tree, open the **Administration** workspace (in the lower left corner) and click **Client Settings**.
5. In the display pane, double-click **Default Client Settings**.
6. Click **Computer Agent**, next to **Organization name displayed in Software Center** type **Contoso**, and

then click **OK**.

Configure the network access account

1. In the Administration workspace, expand **Site Configuration** and click **Sites**.
2. On the **Home** ribbon at the top of the console window, click **Configure Site Components** and then click **Software Distribution**.
3. On the **Network Access Account** tab, choose **Specify the account that accesses network locations**.
4. Click the yellow starburst and then click **New Account**.
5. Click **Browse** and then under **Enter the object name to select**, type **CM_NAA** and click **OK**.
6. Next to **Password** and **Confirm Password**, type **pass@word1**, and then click **OK** twice.

Configure a boundary group

1. In the Administration workspace, expand **Hierarchy Configuration**, right-click **Boundaries** and then click **Create Boundary**.
2. Next to **Description**, type **PS1**, next to **Type** choose **Active Directory Site**, and then click **Browse**.
3. Choose **Default-First-Site-Name** and then click **OK** twice.
4. In the Administration workspace, right-click **Boundary Groups** and then click **Create Boundary Group**.
5. Next to **Name**, type **PS1 Site Assignment and Content Location**, click **Add**, select the **Default-First-Site-Name** boundary and then click **OK**.
6. On the **References** tab in the **Create Boundary Group** window select the **Use this boundary group for site assignment** checkbox.
7. Click **Add**, select the **\\SRV1.contoso.com** checkbox, and then click **OK** twice.

Add the state migration point role

1. In the Administration workspace, expand **Site Configuration**, click **Sites**, and then in on the **Home** ribbon at the top of the console click **Add Site System Roles**.
2. In the Add site System Roles Wizard, click **Next** twice and then on the Specify roles for this server page, select the **State migration point** checkbox.
3. Click **Next**, click the yellow starburst, type **C:\MigData** for the **Storage folder**, and click **OK**.
4. Click **Next**, and then verify under **Boundary groups** that **PS1 Site Assignment and Content Location** is displayed.
5. Click **Next** twice and then click **Close**.

Enable PXE on the distribution point

IMPORTANT

Before enabling PXE in Configuration Manager, ensure that any previous installation of WDS does not cause conflicts. Configuration Manager will automatically configure the WDS service to manage PXE requests. To disable a previous installation, if it exists, type the following commands at an elevated Windows PowerShell prompt on SRV1:

```
WDSUTIL /Set-Server /AnswerClients:None
```

1. Determine the MAC address of the internal network adapter on SRV1. To determine this, type the following command at an elevated Windows PowerShell prompt on SRV1:

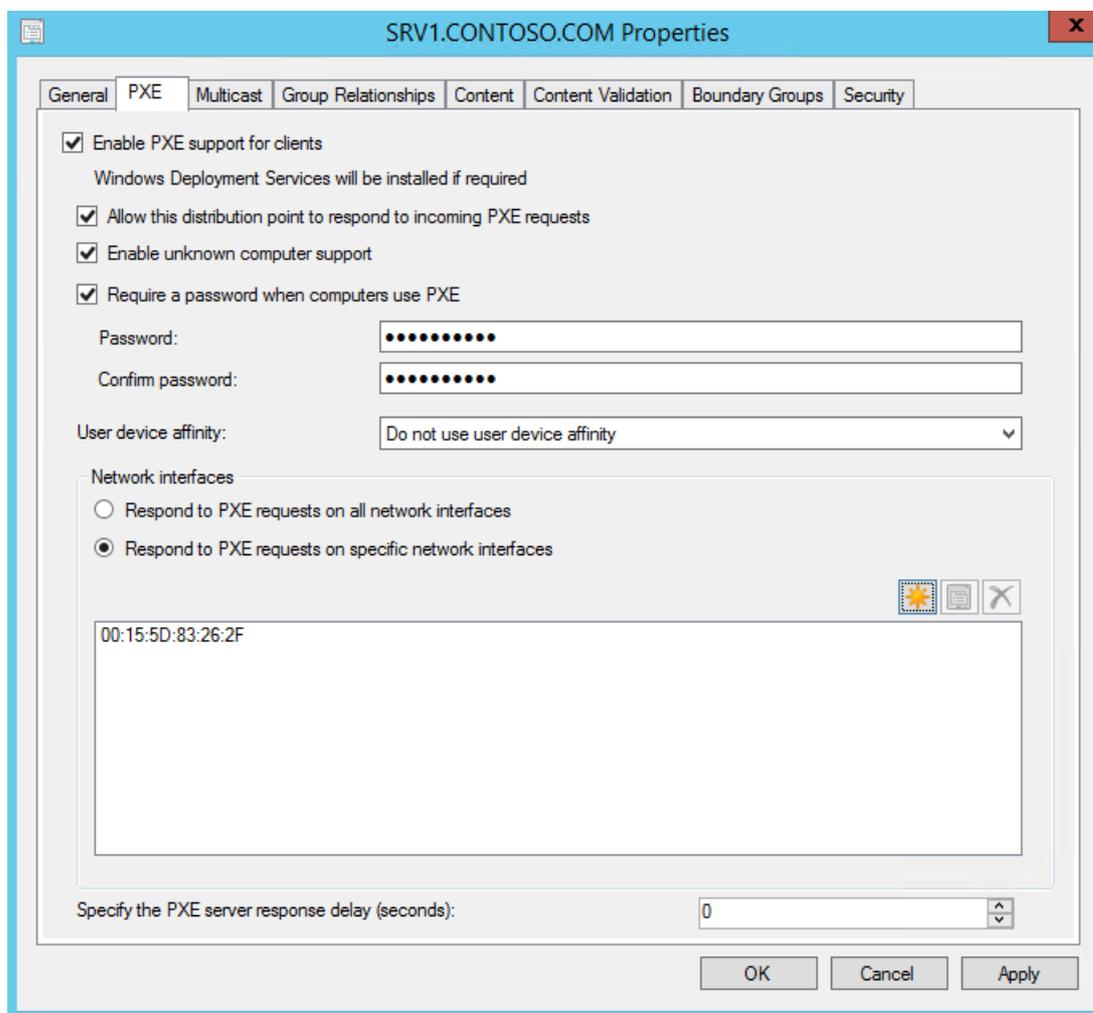
```
(Get-NetAdapter "Ethernet").MacAddress
```

If the internal network adapter, assigned an IP address of 192.168.0.2, is not named "Ethernet" then replace the name "Ethernet" in the previous command with the name of this network adapter. You can

review the names of network adapters and the IP addresses assigned to them by typing **ipconfig**.

- In the System Center Configuration Manager console, in the **Administration** workspace, click **Distribution Points**.
- In the display pane, right-click **SRV1.CONTOSO.COM** and then click **Properties**.
- On the PXE tab, select the following settings:
 - **Enable PXE support for clients**. Click **Yes** in the popup that appears.
 - **Allow this distribution point to respond to incoming PXE requests**
 - **Enable unknown computer support**. Click **OK** in the popup that appears.
 - **Require a password when computers use PXE**
 - **Password** and **Confirm password**: pass@word1
 - **Respond to PXE requests on specific network interfaces**: Click the yellow starburst and then enter the MAC address determined in the first step of this procedure.

See the following example:



- Click **OK**.
- Wait for a minute, then type the following command at an elevated Windows PowerShell prompt on SRV1, and verify that the files displayed are present:

```
cmd /c dir /b C:\RemoteInstall\SMSBoot\x64
```

```
abortpxe.com  
bootmgfw.efi  
bootmgr.exe  
pxeboot.com  
pxeboot.n12  
wdsmgfw.efi  
wdsnbp.com
```

If these files are not present in the C:\RemoteInstall directory, verify that the REMINST share is configured as C:\RemoteInstall. You can view the properties of this share by typing "net share REMINST" at a command prompt. If the share path is set to a different value, then replace C:\RemoteInstall with your REMINST share path. You can also type the following command at an elevated Windows PowerShell prompt to open the Configuration Manager Trace Log Tool. In the tool, click **File**, click **Open**, and then open the **dismgr.log** file. If errors are present, they will be highlighted in red:

```
Invoke-Item 'C:\Program Files\Microsoft Configuration Manager\tools\cmtrace.exe'
```

The log file will be updated continuously while Configuration Manager is running. Wait for Configuration Manager to repair any issues that are present, and periodically re-check that the files are present in the REMINST share location. Close the Configuration Manager Trace Log Tool when done. You will see the following line in dismgr.log that indicates the REMINST share is being populated with necessary files:

```
Running: WDSUTIL.exe /Initialize-Server /REMINST:"C:\RemoteInstall"
```

Once the files are present in the REMINST share location, you can close the cmtrace tool.

Create a branding image file

1. If you have a bitmap (.BMP) image for suitable use as a branding image, copy it to the C:\Sources\OSD\Branding folder on SRV1. Otherwise, use the following step to copy a simple branding image.
2. Type the following command at an elevated Windows PowerShell prompt:

```
copy "C:\ProgramData\Microsoft\User Account Pictures\user.bmp" "C:\Sources\OSD\Branding\contoso.bmp"
```

You can open C:\Sources\OSD\Branding\contoso.bmp in MSPaint.exe if desired to customize this image.

Create a boot image for Configuration Manager

1. In the Configuration Manager console, in the **Software Library** workspace, expand **Operating Systems**, right-click **Boot Images**, and then click **Create Boot Image using MDT**.
2. On the Package Source page, under **Package source folder to be created (UNC Path)**, type **\\SRV1\Sources\$\OSD\Boot\Zero Touch WinPE x64**, and then click **Next**.
 - The Zero Touch WinPE x64 folder does not yet exist. The folder will be created later.
3. On the General Settings page, type **Zero Touch WinPE x64** next to **Name**, and click **Next**.
4. On the Options page, under **Platform** choose **x64**, and click **Next**.
5. On the Components page, in addition to the default selection of **Microsoft Data Access Components**

(MDAC/ADO) support, select the **Microsoft Diagnostics and Recovery Toolkit (DaRT)** checkbox, and click **Next**.

- On the Customization page, select the **Use a custom background bitmap file** checkbox, and under **UNC path**, type or browse to `\\SRV1\Sources$\OSD\Branding\contoso.bmp`, and then click **Next** twice. It will take a few minutes to generate the boot image.
- Click **Finish**.
- In the console display pane, right-click the **Zero Touch WinPE x64** boot image, and then click **Distribute Content**.
- In the Distribute Content Wizard, click **Next**, click **Add** and select **Distribution Point**, select the **SRV1.CONTOSO.COM** checkbox, click **OK**, click **Next** twice, and then click **Close**.
- Use the CMTrace application to view the **distmgr.log** file again and verify that the boot image has been distributed. To open CMTrace, type the following command at an elevated Windows PowerShell prompt on SRV1:

```
Invoke-Item 'C:\Program Files\Microsoft Configuration Manager\tools\cmtrace.exe'
```

In the trace tool, click **Tools** on the menu and choose **Find**. Search for "**STATMSG: ID=2301**". For example:

```
STATMSG: ID=2301 SEV=I LEV=M SOURCE="SMS Server" COMP="SMS_DISTRIBUTION_MANAGER" SYS=SRV1.CONTOSO.COM  
SITE=PS1 PID=924 TID=1424 GMTDATE= Tue Oct 09 22:36:30.986 2018 ISTR0="Zero Touch WinPE x64"  
ISTR1="PS10000A" ISTR2="" ISTR3="" ISTR4="" ISTR5="" ISTR6="" ISTR7="" ISTR8="" ISTR9="" NUMATTRS=1  
AID0=400 AVAL0="PS10000A" SMS_DISTRIBUTION_MANAGER 10/9/2018 3:36:30 PM 1424 (0x0590)
```

- You can also review status by clicking the **Zero Touch WinPE x64** image, and then clicking **Content Status** under **Related Objects** in the bottom right-hand corner of the console, or by entering `\Monitoring\Overview\Distribution Status\Content Status` on the location bar in the console. Double-click **Zero Touch WinPE x64** under **Content Status** in the console tree and verify that a status of **Successfully distributed content** is displayed on the **Success** tab.
- Next, in the **Software Library** workspace, double-click **Zero Touch WinPE x64** and then click the **Data Source** tab.
- Select the **Deploy this boot image from the PXE-enabled distribution point** checkbox, and click **OK**.
- Review the `distmgr.log` file again for "**STATMSG: ID=2301**" and verify that there are three folders under `C:\RemoteInstall\SMSImages` with boot images. See the following example:

```
cmd /c dir /s /b C:\RemoteInstall\SMSImages  
  
C:\RemoteInstall\SMSImages\PS100004  
C:\RemoteInstall\SMSImages\PS100005  
C:\RemoteInstall\SMSImages\PS100006  
C:\RemoteInstall\SMSImages\PS100004\boot.PS100004.wim  
C:\RemoteInstall\SMSImages\PS100005\boot.PS100005.wim  
C:\RemoteInstall\SMSImages\PS100006\WinPE.PS100006.wim
```

The first two images (*.wim files) are default boot images. The third is the new boot image with DaRT.

Create a Windows 10 reference image

If you have already completed steps in [Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit](#) then

you have already created a Windows 10 reference image. In this case, skip to the next procedure in this guide: [Add a Windows 10 operating system image](#). If you have not yet created a Windows 10 reference image, complete the steps in this section.

1. In [Step by step guide: Deploy Windows 10 in a test lab](#) the Windows 10 Enterprise .iso file was saved to the c:\VHD directory as **c:\VHD\w10-enterprise.iso**. The first step in creating a deployment share is to mount this file on SRV1. To mount the Windows 10 Enterprise DVD on SRV1, open an elevated Windows PowerShell prompt on the Hyper-V host computer and type the following command:

```
Set-VMVDvdDrive -VMName SRV1 -Path c:\VHD\w10-enterprise.iso
```

2. Verify that the Windows Enterprise installation DVD is mounted on SRV1 as drive letter D.
3. The Windows 10 Enterprise installation files will be used to create a deployment share on SRV1 using the MDT deployment workbench. To open the deployment workbench, click **Start**, type **deployment**, and then click **Deployment Workbench**.
4. In the Deployment Workbench console, right-click **Deployment Shares** and select **New Deployment Share**.
5. Use the following settings for the New Deployment Share Wizard:
 - Deployment share path: **C:\MDTBuildLab**
 - Share name: **MDTBuildLab\$**
 - Deployment share description: **MDT build lab**
 - Options: click **Next** to accept the default
 - Summary: click **Next**
 - Progress: settings will be applied
 - Confirmation: click **Finish**
6. Expand the **Deployment Shares** node, and then expand **MDT build lab**.
7. Right-click the **Operating Systems** node, and then click **New Folder**. Name the new folder **Windows 10**. Complete the wizard using default values and click **Finish**.
8. Right-click the **Windows 10** folder created in the previous step, and then click **Import Operating System**.
9. Use the following settings for the Import Operating System Wizard:
 - OS Type: **Full set of source files**
 - Source: **D:**
 - Destination: **W10Ent_x64**
 - Summary: click **Next**
 - Confirmation: click **Finish**
10. For purposes of this test lab, we will not add applications, such as Microsoft Office, to the deployment share. For information about adding applications, see the [Add applications](#) section of the [Create a Windows 10 reference image](#) topic in the TechNet library.
11. The next step is to create a task sequence to reference the operating system that was imported. To create a task sequence, right-click the **Task Sequences** node under **MDT Build Lab** and then click **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - Task sequence ID: **REFW10X64-001**
 - Task sequence name: **Windows 10 Enterprise x64 Default Image**
 - Task sequence comments: **Reference Build**

- Template: **Standard Client Task Sequence**
 - Select OS: click **Windows 10 Enterprise Evaluation in W10Ent_x64 install.wim**
 - Specify Product Key: **Do not specify a product key at this time**
 - Full Name: **Contoso**
 - Organization: **Contoso**
 - Internet Explorer home page: <http://www.contoso.com>
 - Admin Password: **Do not specify an Administrator password at this time**
 - Summary: click **Next**
 - Confirmation: click **Finish**
12. Edit the task sequence to add the Microsoft NET Framework 3.5, which is required by many applications. To edit the task sequence, double-click **Windows 10 Enterprise x64 Default Image** that was created in the previous step.
 13. Click the **Task Sequence** tab. Under **State Restore** click **Tattoo** to highlight it, then click **Add** and choose **New Group**. A new group will be added under Tattoo.
 14. On the Properties tab of the group that was created in the previous step, change the Name from New Group to **Custom Tasks (Pre-Windows Update)** and then click **Apply**. To see the name change, click **Tattoo**, then click the new group again.
 15. Click the **Custom Tasks (Pre-Windows Update)** group again, click **Add**, point to **Roles**, and then click **Install Roles and Features**.
 16. Under **Select the roles and features that should be installed**, select **.NET Framework 3.5 (includes .NET 2.0 and 3.0)** and then click **Apply**.
 17. Enable Windows Update in the task sequence by clicking the **Windows Update (Post-Application Installation)** step, clicking the **Options** tab, and clearing the **Disable this step** checkbox.

Note: Since we are not installing applications in this test lab, there is no need to enable the Windows Update Pre-Application Installation step. However, you should enable this step if you are also installing applications.

18. Click **OK** to complete editing the task sequence.
19. The next step is to configure the MDT deployment share rules. To configure rules in the Deployment Workbench, right-click MDT build lab (C:\MDTBuildLab) and click **Properties**, and then click the **Rules** tab.
20. Replace the default rules with the following text:

```

[Settings]
Priority=Default

[Default]
_SMSTSORGNAME=Contoso
UserDataLocation=NONE
DoCapture=YES
OSInstall=Y
AdminPassword=pass@word1
TimeZoneName=Pacific Standard TimeZoneName
OSDComputername=#Left("PC-%SerialNumber%",7)#
JoinWorkgroup=WORKGROUP
HideShell=YES
FinishAction=SHUTDOWN
DoNotCreateExtraPartition=YES
ApplyGPOPack=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=YES
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=YES
SkipBitLocker=YES
SkipSummary=YES
SkipRoles=YES
SkipCapture=NO
SkipFinalSummary=NO

```

21. Click **Apply** and then click **Edit Bootstrap.ini**. Replace the contents of the Bootstrap.ini file with the following text, and save the file:

```

[Settings]
Priority=Default

[Default]
DeployRoot=\\SRV1\MDTBuildLab$
UserDomain=CONTOSO
UserID=MDT_BA
UserPassword=pass@word1
SkipBDDWelcome=YES

```

22. Click **OK** to complete the configuration of the deployment share.
23. Right-click **MDT build lab (C:\MDTBuildLab)** and then click **Update Deployment Share**.
24. Accept all default values in the Update Deployment Share Wizard by clicking **Next**. The update process will take 5 to 10 minutes. When it has completed, click **Finish**.
25. Copy **c:\MDTBuildLab\Boot\LiteTouchPE_x86.iso** on SRV1 to the **c:\VHD** directory on the Hyper-V host computer. Note that in MDT, the x86 boot image can deploy both x86 and x64 operating systems, except on computers based on Unified Extensible Firmware Interface (UEFI).

Hint: To copy the file, right-click the **LiteTouchPE_x86.iso** file and click **Copy** on SRV1, then open the **c:\VHD** folder on the Hyper-V host, right-click inside the folder and click **Paste**.

26. Open a Windows PowerShell prompt on the Hyper-V host computer and type the following commands:

```
New-VM -Name REFW10X64-001 -SwitchName poc-internal -NewVHDPATH "c:\VHD\REFW10X64-001.vhdx" -
NewVHDSIZEBYTES 60GB
Set-VMMemory -VMName REFW10X64-001 -DynamicMemoryEnabled $true -MinimumBytes 1024MB -MaximumBytes
1024MB -Buffer 20
Set-VMVDVDDrive -VMName REFW10X64-001 -Path c:\VHD\LiteTouchPE_x86.iso
Start-VM REFW10X64-001
vmconnect localhost REFW10X64-001
```

27. In the Windows Deployment Wizard, select **Windows 10 Enterprise x64 Default Image**, and then click **Next**.
28. Accept the default values on the Capture Image page, and click **Next**. Operating system installation will complete after 5 to 10 minutes and then the VM will reboot automatically. Allow the system to boot normally (do not press a key). The process is fully automated.

Additional system restarts will occur to complete updating and preparing the operating system. Setup will complete the following procedures:

- Install the Windows 10 Enterprise operating system.
- Install added applications, roles, and features.
- Update the operating system using Windows Update (or WSUS if optionally specified).
- Stage Windows PE on the local disk.
- Run System Preparation (Sysprep) and reboot into Windows PE.
- Capture the installation to a Windows Imaging (WIM) file.
- Turn off the virtual machine.

This step requires from 30 minutes to 2 hours, depending on the speed of the Hyper-V host and your network's download speed. After some time, you will have a Windows 10 Enterprise x64 image that is fully patched and has run through Sysprep. The image is located in the C:\MDTBuildLab\Captures folder on SRV1. The file name is **REFW10X64-001.wim**.

Add a Windows 10 operating system image

1. Type the following commands at an elevated Windows PowerShell prompt on SRV1:

```
New-Item -ItemType Directory -Path "C:\Sources\OSD\OS\Windows 10 Enterprise x64"
cmd /c copy /z "C:\MDTBuildLab\Captures\REFW10X64-001.wim" "C:\Sources\OSD\OS\Windows 10 Enterprise
x64"
```

2. In the Configuration Manager console, in the **Software Library** workspace, expand **Operating Systems**, right-click **Operating System Images**, and then click **Add Operating System Image**.
3. On the Data Source page, under **Path:**, type or browse to **\\SRV1\Sources\$\OSD\OS\Windows 10 Enterprise x64\REFW10X64-001.wim**, and click **Next**.
4. On the General page, next to **Name:**, type **Windows 10 Enterprise x64**, click **Next** twice, and then click **Close**.
5. Distribute the operating system image to the SRV1 distribution point by right-clicking the **Windows 10 Enterprise x64** operating system image and then clicking **Distribute Content**.
6. In the Distribute Content Wizard, click **Next**, click **Add**, click **Distribution Point**, add the **SRV1.CONTOSO.COM** distribution point, click **OK**, click **Next** twice and then click **Close**.
7. Enter **\Monitoring\Overview\Distribution Status\Content Status** on the location bar (be sure there is no space at the end of the location or you will get an error), click **Windows 10 Enterprise x64**, and monitor the status of content distribution until it is successful and no longer in progress. Refresh the view

with the F5 key or by right-clicking **Windows 10 Enterprise x64** and clicking **Refresh**. Processing of the image on the site server can take several minutes.

If content distribution is not successful, verify that sufficient disk space is available.

Create a task sequence

Complete this section slowly. There are a large number of similar settings from which to choose.

1. In the Configuration Manager console, in the **Software Library** workspace expand **Operating Systems**, right-click **Task Sequences**, and then click **Create MDT Task Sequence**.
2. On the Choose Template page, select the **Client Task Sequence** template and click **Next**.
3. On the General page, type **Windows 10 Enterprise x64** under **Task sequence name:** and then click **Next**.
4. On the Details page, enter the following settings:
 - Join a domain: **contoso.com**
 - Account: click **Set**
 - User name: **contoso\CM_JD**
 - Password: **pass@word1**
 - Confirm password: **pass@word1**
 - Click **OK**
 - Windows Settings
 - User name: **Contoso**
 - Organization name: **Contoso**
 - Product key: <blank>
 - Administrator Account: **Enable the account and specify the local administrator password**
 - Password: **pass@word1**
 - Confirm password: **pass@word1**
 - Click **Next**
5. On the Capture Settings page, accept the default settings and click **Next**.
6. On the Boot Image page, browse and select the **Zero Touch WinPE x64** boot image package, click **OK**, and then click **Next**.
7. On the MDT Package page, select **Create a new Microsoft Deployment Toolkit Files package**, under **Package source folder to be created (UNC Path):**, type `\\SRV1\Sources$\OSD\MDT\MDT` (MDT is repeated here, not a typo), and then click **Next**.
8. On the MDT Details page, next to **Name:** type **MDT** and then click **Next**.
9. On the OS Image page, browse and select the **Windows 10 Enterprise x64** package, click **OK**, and then click **Next**.
10. On the Deployment Method page, accept the default settings for **Zero Touch Installation** and click **Next**.
11. On the Client Package page, browse and select the **Microsoft Corporation Configuration Manager Client package**, click **OK**, and then click **Next**.
12. On the USMT Package page, browse and select the **Microsoft Corporation User State Migration Tool for Windows 10.0.14393.0** package, click **OK**, and then click **Next**.
13. On the Settings Package page, select **Create a new settings package**, and under **Package source folder**

to be created (UNC Path); type `\\SRV1\Sources$\OSD\Settings\Windows 10 x64 Settings`, and then click **Next**.

14. On the Settings Details page, next to **Name**; type **Windows 10 x64 Settings**, and click **Next**.
15. On the Sysprep Package page, click **Next** twice.
16. On the Confirmation page, click **Finish**.

Edit the task sequence

1. In the Configuration Manager console, in the **Software Library** workspace, click **Task Sequences**, right-click **Windows 10 Enterprise x64**, and then click **Edit**.
2. Scroll down to the **Install** group and click the **Set Variable for Drive Letter** action.
3. Change the Value under **OSDPreserveDriveLetter** from **False** to **True**, and then click **Apply**.
4. In the **State Restore** group, click the **Set Status 5** action, click **Add** in the upper left corner, point to **User State**, and click **Request State Store**. This adds a new action immediately after **Set Status 5**.
5. Configure the **Request State Store** action that was just added with the following settings:
 - Request state storage location to: **Restore state from another computer**
 - Select the **If computer account fails to connect to state store, use the Network Access account** checkbox.
 - Options tab: Select the **Continue on error** checkbox.
 - Add Condition: **Task Sequence Variable**:
 - Variable: **USMTLOCAL**
 - Condition: **not equals**
 - Value: **True**
 - Click **OK**.
 - Click **Apply**
6. In the **State Restore** group, click **Restore User State**, click **Add**, point to **User State**, and click **Release State Store**.
7. Configure the **Release State Store** action that was just added with the following settings:
 - Options tab: Select the **Continue on error** checkbox.
 - Add Condition: **Task Sequence Variable**:
 - Variable: **USMTLOCAL**
 - Condition: **not equals**
 - Value: **True**
 - Click **OK**.
 - Click **OK**

Finalize the operating system configuration

If you completed all procedures in [Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit](#) then the MDT deployment share is already present on SRV1. In this case, skip the first four steps below and begin with step 5 to edit CustomSettings.ini.

1. In the MDT deployment workbench on SRV1, right-click **Deployment Shares** and then click **New Deployment Share**.

2. Use the following settings for the New Deployment Share Wizard:

- Deployment share path: **C:\MDTProduction**
- Share name: **MDTProduction\$**
- Deployment share description: **MDT Production**
- Options: click **Next** to accept the default
- Summary: click **Next**
- Progress: settings will be applied
- Confirmation: click **Finish**

3. Right-click the **MDT Production** deployment share, and click **Properties**.

4. Click the **Monitoring** tab, select the **Enable monitoring for this deployment share** checkbox, and then click **OK**.

5. Type the following command at an elevated Windows PowerShell prompt on SRV1:

```
notepad "C:\Sources\OSD\Settings\Windows 10 x64 Settings\CustomSettings.ini"
```

6. Replace the contents of the file with the following text, and then save the file:

```
[Settings]
Priority=Default
Properties=OSDMigrateConfigFiles,OSDMigrateMode

[Default]
DoCapture=NO
ComputerBackupLocation=NONE
OSDMigrateMode=Advanced
OSDMigrateAdditionalCaptureOptions=/ue:* \* /ui:CONTOSO \*
OSDMigrateConfigFiles=Miguser.xml,Migapp.xml
SLSHARE=\\SRV1\Loggs$
EventService=http://SRV1:9800
ApplyGPOPack=NO
```

As noted previously, if you wish to migrate accounts other than those in the Contoso domain, then change the `OSDMigrateAdditionalCaptureOptions` option. For example, the following option will capture settings from all user accounts:

```
OSDMigrateAdditionalCaptureOptions=/all
```

7. Return to the Configuration Manager console, and in the Software Library workspace, expand **Application Management**, click **Packages**, right-click **Windows 10 x64 Settings**, and then click **Update Distribution Points**. Click **OK** in the popup that appears.
8. In the Software Library workspace, expand **Operating Systems**, click **Task Sequences**, right-click **Windows 10 Enterprise x64**, and then click **Distribute Content**.
9. In the Distribute Content Wizard, click **Next** twice, click **Add**, click **Distribution Point**, select the **SRV1.CONTOSO.COM** distribution point, click **OK**, click **Next** twice and then click **Close**.
10. Enter **\Monitoring\Overview\Distribution Status\Content Status\Windows 10 Enterprise x64** on the location bar, double-click **Windows 10 Enterprise x64**, and monitor the status of content distribution until it is successful and no longer in progress. Refresh the view with the F5 key or by right-clicking **Windows 10 Enterprise x64** and clicking **Refresh**.

Create a deployment for the task sequence

1. In the Software Library workspace, expand **Operating Systems**, click **Task Sequences**, right-click **Windows 10 Enterprise x64**, and then click **Deploy**.
2. On the General page, next to **Collection**, click **Browse**, select the **All Unknown Computers** collection, click **OK**, and then click **Next**.
3. On the Deployment Settings page, use the following settings:
 - Purpose: **Available**
 - Make available to the following: **Only media and PXE**
 - Click **Next**.
4. Click **Next** five times to accept defaults on the Scheduling, User Experience, Alerts, and Distribution Points pages.
5. Click **Close**.

Deploy Windows 10 using PXE and Configuration Manager

In this first deployment scenario, we will deploy Windows 10 using PXE. This scenario creates a new computer that does not have any migrated users or settings.

1. Type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
New-VM -Name "PC4" -NewVHDPATH "c:\vhd\pc4.vhdx" -NewVHDSIZEBYTES 40GB -SwitchName poc-internal -
BootDevice NetworkAdapter -Generation 2
Set-VMMemory -VMName "PC4" -DynamicMemoryEnabled $true -MinimumBytes 512MB -MaximumBytes 2048MB -
Buffer 20
Start-VM PC4
vmconnect localhost PC4
```

2. Press ENTER when prompted to start the network boot service.
3. In the Task Sequence Wizard, provide the password: **pass@word1**, and then click **Next**.
4. Before you click **Next** in the Task Sequence Wizard, press the **F8** key. A command prompt will open.
5. At the command prompt, type **explorer.exe** and review the Windows PE file structure.
6. The smsts.log file is critical for troubleshooting any installation problems that might be encountered. Depending on the deployment phase, the smsts.log file is created in different locations:
 - X:\windows\temp\SMSTSLog\smsts.log before disks are formatted.
 - x:\smstslog\smsts.log after disks are formatted.
 - c:_SMSTaskSequence\Log\Smstslog\smsts.log before the System Center Configuration Manager client is installed.
 - c:\windows\ccm\logs\Smstslog\smsts.log after the System Center Configuration Manager client is installed.
 - c:\windows\ccm\logs\smsts.log when the task sequence is complete.

Note: If a reboot is pending on the client, the reboot will be blocked as long as the command window is open.
7. In the explorer window, click **Tools** and then click **Map Network Drive**.
8. Do not map a network drive at this time. If you need to save the smsts.log file, you can use this method to

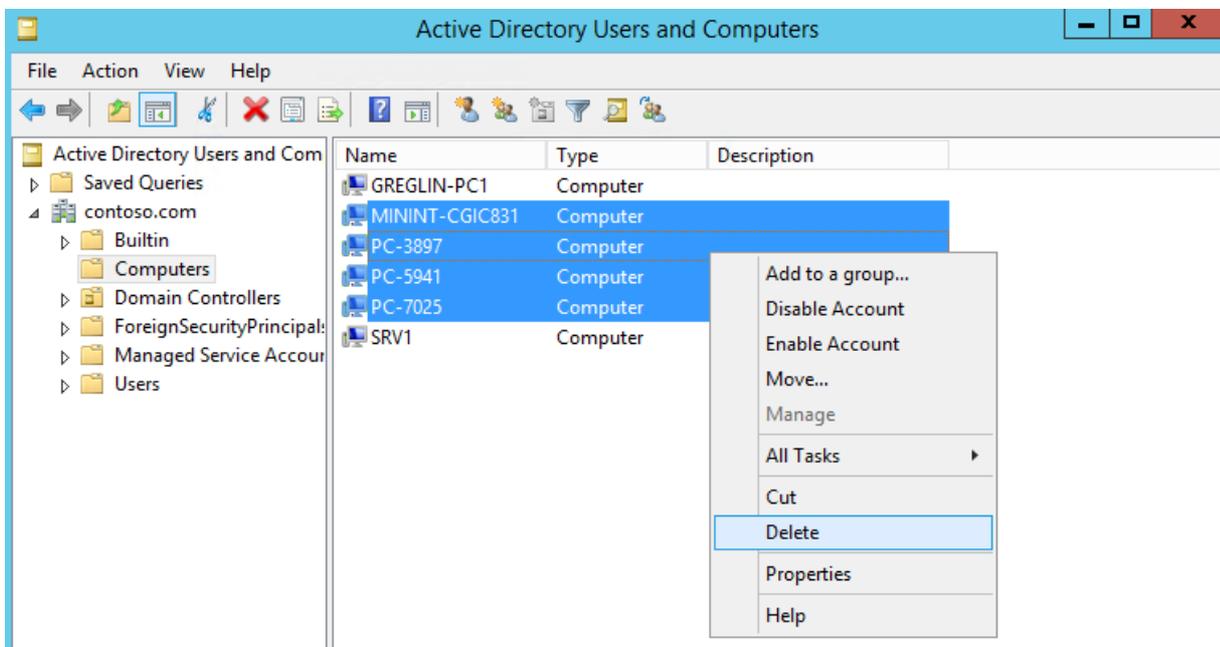
save the file to a location on SRV1.

9. Close the Map Network Drive window, the Explorer window, and the command prompt.
10. The **Windows 10 Enterprise x64** task sequence is selected in the Task Sequenc Wizard. Click **Next** to continue with the deployment.
11. The task sequence will require several minutes to complete. You can monitor progress of the task sequence using the MDT Deployment Workbench under Deployment Shares > MDTProduction > Monitoring. The task sequence will:
 - Install Windows 10
 - Install the Configuration Manager client and hotfix
 - Join the computer to the contoso.com domain
 - Install any applications that were specified in the reference image
12. When Windows 10 installation has completed, sign in to PC4 using the **contoso\administrator** account.
13. Right-click **Start**, click **Run**, type **control appwiz.cpl**, press ENTER, click **Turn Windows features on or off**, and verify that **.NET Framework 3.5 (includes .NET 2.0 and 3.0)** is installed. This is a feature included in the reference image.
14. Shut down the PC4 VM.

Note: The following two procedures 1) Replace a client with Windows 10 and 2) Refresh a client with Windows 10 have been exchanged in their order in this guide compared to the previous version. This is to avoid having to restore Hyper-V checkpoints to have access to PC1 before the OS is upgraded. If this is your first time going through this guide, you won't notice any change, but if you have tried the guide previously then this change should make it simpler to complete.

Replace a client with Windows 10 using Configuration Manager

Before starting this section, you can delete computer objects from Active Directory that were created as part of previous deployment procedures. Use the Active Directory Users and Computers console on DC1 to remove stale entries under contoso.com\Computers, but do not delete the computer account (hostname) for PC1. There should be at least two computer accounts present in the contoso.com\Computers container: one for SRV1, and one for the hostname of PC1. It is not required to delete the stale entries, this is only done to remove clutter.



In the replace procedure, PC1 will not be migrated to a new operating system. It is simplest to perform this procedure before performing the refresh procedure. After refreshing PC1, the operating system will be new. The next (replace) procedure does not install a new operating system on PC1 but rather performs a side-by-side migration of PC1 and another computer (PC4), to copy users and settings from PC1 to the new computer.

Create a replace task sequence

1. On SRV1, in the Configuration Manager console, in the Software Library workspace, expand **Operating Systems**, right-click **Task Sequences**, and then click **Create MDT Task Sequence**.
2. On the Choose Template page, select **Client Replace Task Sequence** and click **Next**.
3. On the General page, type the following:
 - Task sequence name: **Replace Task Sequence**
 - Task sequence comments: **USMT backup only**
4. Click **Next**, and on the Boot Image page, browse and select the **Zero Touch WinPE x64** boot image package. Click **OK** and then click **Next** to continue.
5. On the MDT Package page, browse and select the **MDT** package. Click **OK** and then click **Next** to continue.
6. On the USMT Package page, browse and select the **Microsoft Corporation User State Migration Tool for Windows** package. Click **OK** and then click **Next** to continue.
7. On the Settings Package page, browse and select the **Windows 10 x64 Settings** package. Click **OK** and then click **Next** to continue.
8. On the Summary page, review the details and then click **Next**.
9. On the Confirmation page, click **Finish**.

If an error is displayed at this stage it can be caused by a corrupt MDT integration. To repair it, close the Configuration Manager console, remove MDT integration, and then restore MDT integration.

Deploy PC4

Create a VM named PC4 to receive the applications and settings from PC1. This VM represents a new computer that will replace PC1. To create this VM, type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
New-VM -Name "PC4" -NewVHDPATH "c:\vhd\pc4.vhdx" -NewVHDSIZEBYTES 60GB -SwitchName poc-internal -BootDevice
NetworkAdapter -Generation 2
Set-VMMemory -VMName "PC4" -DynamicMemoryEnabled $true -MinimumBytes 1024MB -MaximumBytes 2048MB -Buffer 20
Set-VMNetworkAdapter -VMName PC4 -StaticMacAddress 00-15-5D-83-26-FF
```

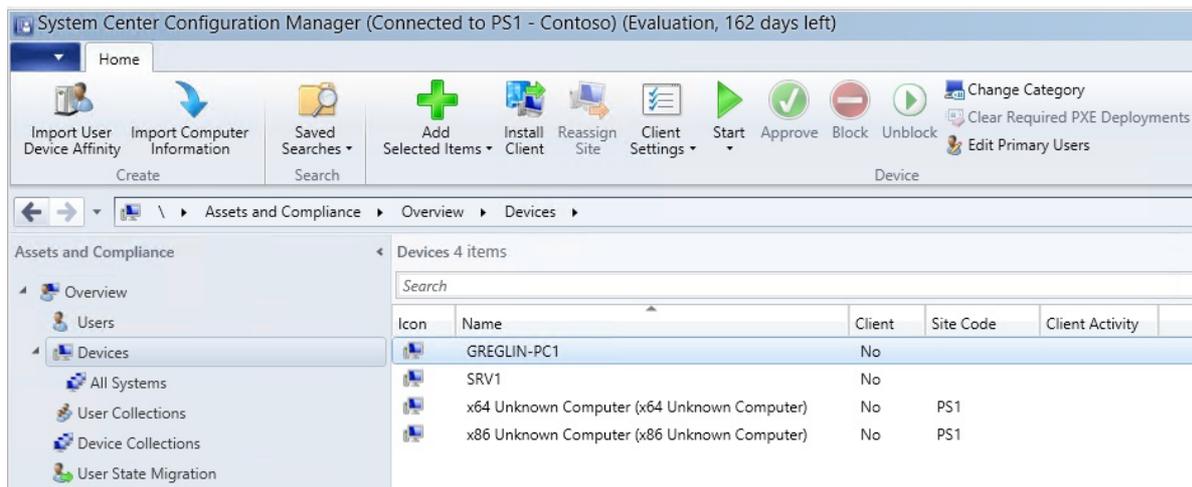
Hyper-V enables us to define a static MAC address on PC4. In a real-world scenario you must determine the MAC address of the new computer.

Install the Configuration Manager client on PC1

1. Verify that the PC1 VM is running and in its original state, which was saved as a checkpoint and then restored in [Deploy Windows 10 in a test lab using Microsoft Deployment Toolkit](#).
2. If a PC1 checkpoint has not already been saved, then save a checkpoint by typing the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Checkpoint-VM -Name PC1 -SnapshotName BeginState
```

3. On SRV1, in the Configuration Manager console, in the Administration workspace, expand **Hierarchy Configuration** and click on **Discovery Methods**.
4. Double-click **Active Directory System Discovery** and on the **General** tab select the **Enable Active Directory System Discovery** checkbox.
5. Click the yellow starburst, click **Browse**, select **contoso\Computers**, and then click **OK** three times.
6. When a popup dialog box asks if you want to run full discovery, click **Yes**.
7. In the Assets and Compliance workspace, click **Devices** and verify that the computer account names for SRV1 and PC1 are displayed. See the following example (GREGLIN-PC1 is the computer account name of PC1 in this example):



If you do not see the computer account for PC1, try clicking the **Refresh** button in the upper right corner of the console.

The **Client** column indicates that the Configuration Manager client is not currently installed. This procedure will be carried out next.

8. Sign in to PC1 using the contoso\administrator account and type the following at an elevated command prompt to remove any pre-existing client configuration, if it exists. Note: this command requires an elevated command prompt not an elevated Windows PowerShell prompt:

```
sc stop ccmsetup
"\\SRV1\c$\Program Files\Microsoft Configuration Manager\Client\CCMSetup.exe" /Uninstall
```

If PC1 still has Configuration Manager registry settings that were applied by Group Policy, startup scripts, or other policies in its previous domain, these might not all be removed by CCMSetup /Uninstall and can cause problems with installation or registration of the client in its new environment. It might be necessary to manually remove these settings if they are present. For more information, see [Manual removal of the SCCM client](#).

9. On PC1, temporarily stop Windows Update from queuing items for download and clear all BITS jobs from the queue:

```
net stop wuauerv
net stop BITS
```

Verify that both services were stopped successfully, then type the following at an elevated command prompt:

```
del "%ALLUSERSPROFILE%\Application Data\Microsoft\Network\Downloader\qmgr*.dat"
net start BITS
bitsadmin /list /allusers
```

Verify that BITSAdmin displays 0 jobs.

10. To install the Configuration Manager client as a standalone process, type the following at an elevated command prompt:

```
"\\SRV1\c$\Program Files\Microsoft Configuration Manager\Client\CCMSetup.exe" /mp:SRV1.contoso.com
/logon SMSITECODE=PS1
```

11. On PC1, using file explorer, open the **C:\Windows\ccmsetup** directory. During client installation, files will be downloaded here.
12. Installation progress will be captured in the file: **c:\windows\ccmsetup\logs\ccmsetup.log**. You can periodically open this file in notepad, or you can type the following command at an elevated Windows PowerShell prompt to monitor installation progress:

```
Get-Content -Path c:\windows\ccmsetup\logs\ccmsetup.log -Wait
```

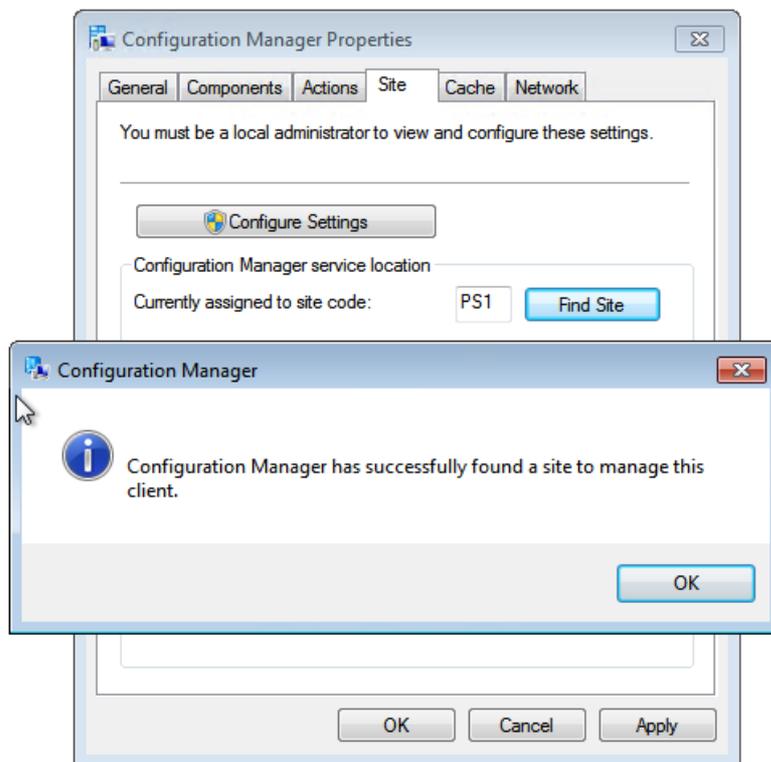
Installation might require several minutes, and display of the log file will appear to hang while some applications are installed. This is normal. When setup is complete, verify that **CcmSetup is existing with return code 0** is displayed on the last line of the ccmsetup.log file and then press **CTRL-C** to break out of the Get-Content operation (if you are viewing the log in Windows PowerShell the last line will be wrapped). A return code of 0 indicates that installation was successful and you should now see a directory created at **C:\Windows\CCM** that contains files used in registration of the client with its site.

13. On PC1, open the Configuration Manager control panel applet by typing the following command:

```
control smscfgrc
```

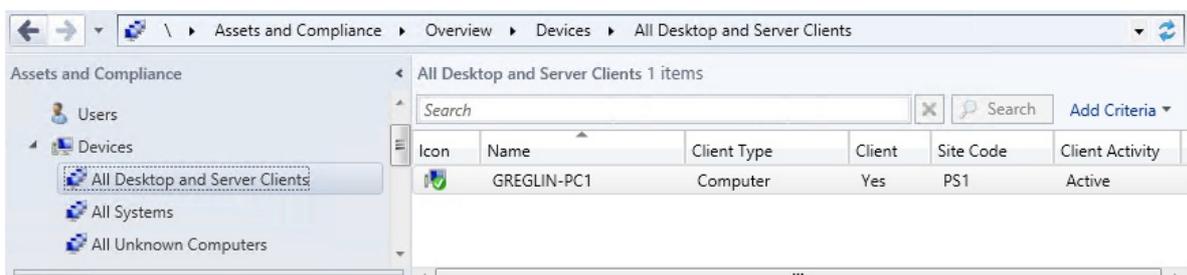
14. Click the **Site** tab, click **Configure Settings**, and click **Find Site**. The client will report that it has found the

PS1 site. See the following example:



If the client is not able to find the PS1 site, review any error messages that are displayed in **C:\Windows\CCM\Logs\ClientIDManagerStartup.log** and **LocationServices.log**. A common reason the site code is not located is because a previous configuration exists. For example, if a previous site code is configured at **HKLM\SOFTWARE\Microsoft\SMS\Mobile Client\GPRequestedSiteAssignmentCode** this must be deleted or updated.

15. On SRV1, in the Assets and Compliance workspace, click **Device Collections** and then double-click **All Desktop and Server Clients**. This node will be added under **Devices**.
16. Click **All Desktop and Server Clients** and verify that the computer account for PC1 is displayed here with **Yes** and **Active** in the **Client** and **Client Activity** columns, respectively. You might have to refresh the view and wait few minutes for the client to appear here. See the following example:



It might take several minutes for the client to fully register with the site and complete a client check. When it is complete you will see a green check mark over the client icon as shown above. To refresh the client, click it and then press **F5** or right-click the client and click **Refresh**.

Create a device collection and deployment

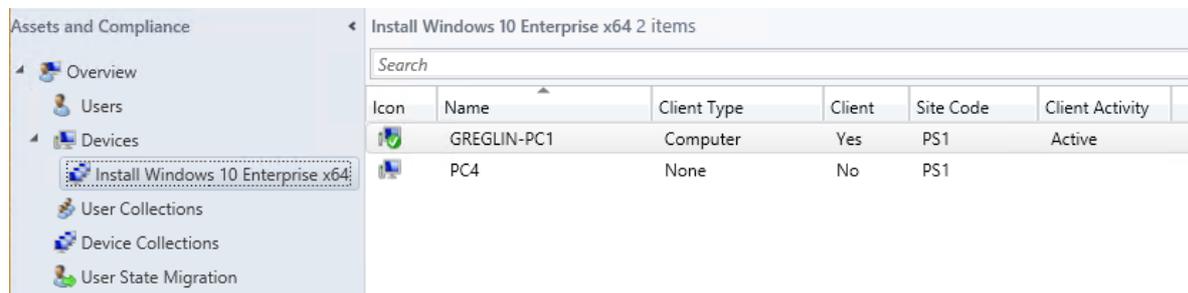
1. On SRV1, in the Configuration Manager console, in the Asset and Compliance workspace, right-click **Device Collections** and then click **Create Device Collection**.
2. Use the following settings in the **Create Device Collection Wizard**:
 - General > Name: **Install Windows 10 Enterprise x64**

- General > Limiting collection: **All Systems**
 - Membership Rules > Add Rule: **Direct Rule**
 - The **Create Direct Membership Rule Wizard** opens, click **Next**
 - Search for Resources > Resource class: **System Resource**
 - Search for Resources > Attribute name: **Name**
 - Search for Resources > Value: **%**
 - Select Resources > Value: Select the computername associated with the PC1 VM
 - Click **Next** twice and then click **Close** in both windows (Next, Next, Close, then Next, Next, Close)
3. Double-click the Install Windows 10 Enterprise x64 device collection and verify that the PC1 computer account is displayed.
 4. In the Software Library workspace, expand **Operating Systems**, click **Task Sequences**, right-click **Windows 10 Enterprise x64** and then click **Deploy**.
 5. Use the following settings in the Deploy Software wizard:
 - General > Collection: Click Browse and select **Install Windows 10 Enterprise x64**
 - Deployment Settings > Purpose: **Available**
 - Deployment Settings > Make available to the following: **Configuration Manager clients, media and PXE**
 - Scheduling > Click **Next**
 - User Experience > Click **Next**
 - Alerts > Click **Next**
 - Distribution Points > Click **Next**
 - Summary > Click **Next**
 - Verify that the wizard completed successfully and then click **Close**

Associate PC4 with PC1

1. On SRV1 in the Configuration Manager console, in the Assets and Compliance workspace, right-click **Devices** and then click **Import Computer Information**.
2. On the Select Source page, choose **Import single computer** and click **Next**.
3. On the Single Computer page, use the following settings:
 - Computer Name: **PC4**
 - MAC Address: **00:15:5D:83:26:FF**
 - Source Computer: <type the hostname of PC1, or click **Search** twice, click the hostname, and click **OK**>
4. Click **Next**, and on the User Accounts page choose **Capture and restore specified user accounts**, then click the yellow starburst next to **User accounts to migrate**.
5. Click **Browse** and then under Enter the object name to select type **user1** and click OK twice.
6. Click the yellow starburst again and repeat the previous step to add the **contoso\administrator** account.
7. Click **Next** twice, and on the Choose Target Collection page, choose **Add computers to the following collection**, click **Browse**, choose **Install Windows 10 Enterprise x64**, click **OK**, click **Next** twice, and then click **Close**.
8. In the Assets and Compliance workspace, click **User State Migration** and review the computer association in the display pane. The source computer will be the computername of PC1 (GREGLIN-PC1 in this example), the destination computer will be **PC4**, and the migration type will be **side-by-side**.
9. Right-click the association in the display pane and then click **Specify User Accounts**. You can add or remove user account here. Click **OK**.

10. Right-click the association in the display pane and then click **View Recovery Information**. Note that a recovery key has been assigned, but a user state store location has not. Click **Close**.
11. Click **Device Collections** and then double-click **Install Windows 10 Enterprise x64**. Verify that **PC4** is displayed in the collection. You might have to update and refresh the collection, or wait a few minutes, but do not proceed until PC4 is available. See the following example:



Create a device collection for PC1

1. On SRV1, in the Configuration Manager console, in the Assets and Compliance workspace, right-click **Device Collections** and then click **Create Device Collection**.
2. Use the following settings in the **Create Device Collection Wizard**:
 - General > Name: **USMT Backup (Replace)**
 - General > Limiting collection: **All Systems**
 - Membership Rules > Add Rule: **Direct Rule**
 - The **Create Direct Membership Rule Wizard** opens, click **Next**
 - Search for Resources > Resource class: **System Resource**
 - Search for Resources > Attribute name: **Name**
 - Search for Resources > Value: **%**
 - Select Resources > Value: Select the computername associated with the PC1 VM (GREGLIN-PC1 in this example).
 - Click **Next** twice and then click **Close** in both windows.
3. Click **Device Collections** and then double-click **USMT Backup (Replace)**. Verify that the computer name/hostname associated with PC1 is displayed in the collection. Do not proceed until this name is displayed.

Create a new deployment

In the Configuration Manager console, in the Software Library workspace under Operating Systems, click **Task Sequences**, right-click **Replace Task Sequence**, click **Deploy**, and use the following settings:

- General > Collection: **USMT Backup (Replace)**
- Deployment Settings > Purpose: **Available**
- Deployment Settings > Make available to the following: **Only Configuration Manager Clients**
- Scheduling: Click **Next**
- User Experience: Click **Next**
- Alerts: Click **Next**
- Distribution Points: Click **Next**
- Click **Next** and then click **Close**.

Verify the backup

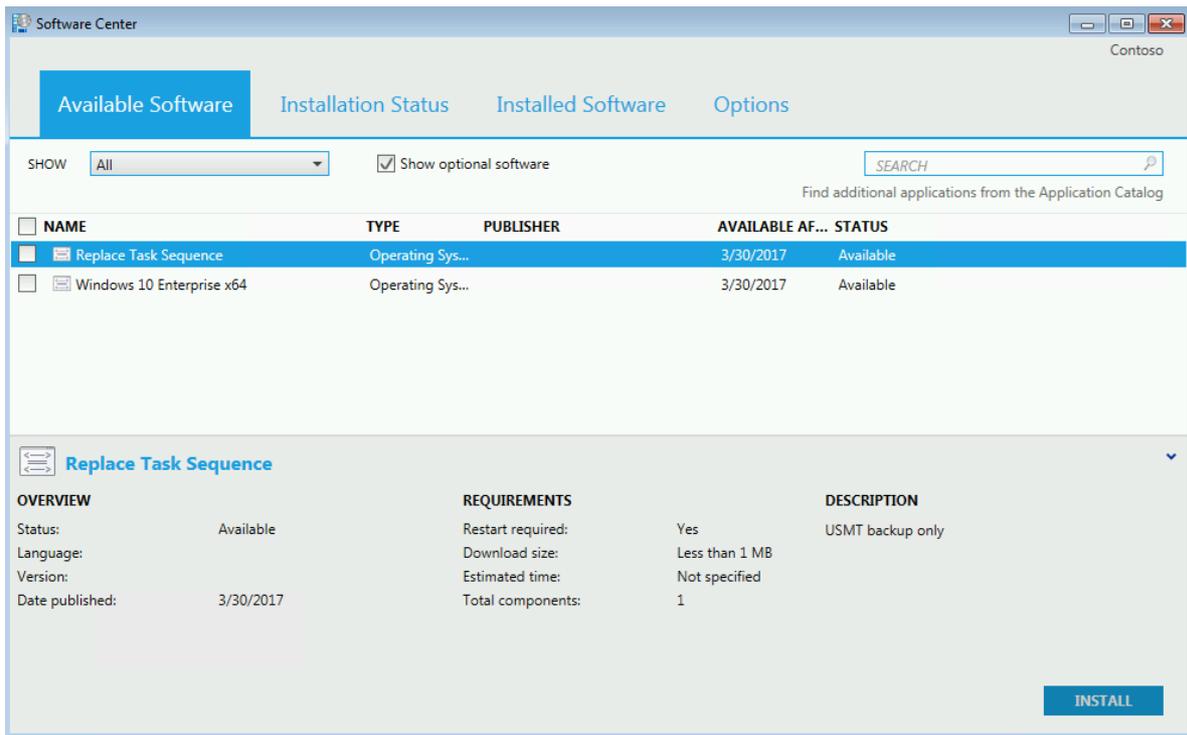
1. On PC1, open the Configuration Manager control panel applet by typing the following command:

```
control smscfgrc
```

2. On the **Actions** tab, click **Machine Policy Retrieval & Evaluation Cycle**, click **Run Now**, click **OK**, and then click **OK** again. This is one method that can be used to run a task sequence in addition to the Client Notification method that will be demonstrated in the computer refresh procedure.
3. Type the following at an elevated command prompt to open the Software Center:

```
C:\Windows\CCM\SCClient.exe
```

4. In the Software Center, click **Available Software** and then select the **Replace Task Sequence** checkbox. See the following example:



If you do not see any available software, try running step #2 again to start the Machine Policy Retrieval & Evaluation Cycle. You should see an alert that new software is available.

5. Click **INSTALL SELECTED** and then click **INSTALL OPERATING SYSTEM**.
6. Allow the **Replace Task Sequence** to complete, then verify that the C:\MigData folder on SRV1 contains the USMT backup.

Deploy the new computer

1. Start PC4 and press ENTER for a network boot when prompted. To start PC4, type the following commands at an elevated Windows Powershell prompt on the Hyper-V host:

```
Start-VM PC4  
vmconnect localhost PC4
```

2. In the **Welcome to the Task Sequence Wizard**, enter **pass@word1** and click **Next**.
3. Choose the **Windows 10 Enterprise X64** image.
4. Setup will install the operating system using the Windows 10 Enterprise x64 reference image, install the

configuration manager client, join PC4 to the domain, and restore users and settings from PC1.

5. Save checkpoints for all VMs if you wish to review their status at a later date. This is not required (checkpoints do take up space on the Hyper-V host). Note: the next procedure will install a new OS on PC1 update its status in Configuration Manager and in Active Directory as a Windows 10 device, so you cannot return to a previous checkpoint only on the PC1 VM without a conflict. Therefore, if you do create a checkpoint, you should do this for all VMs.

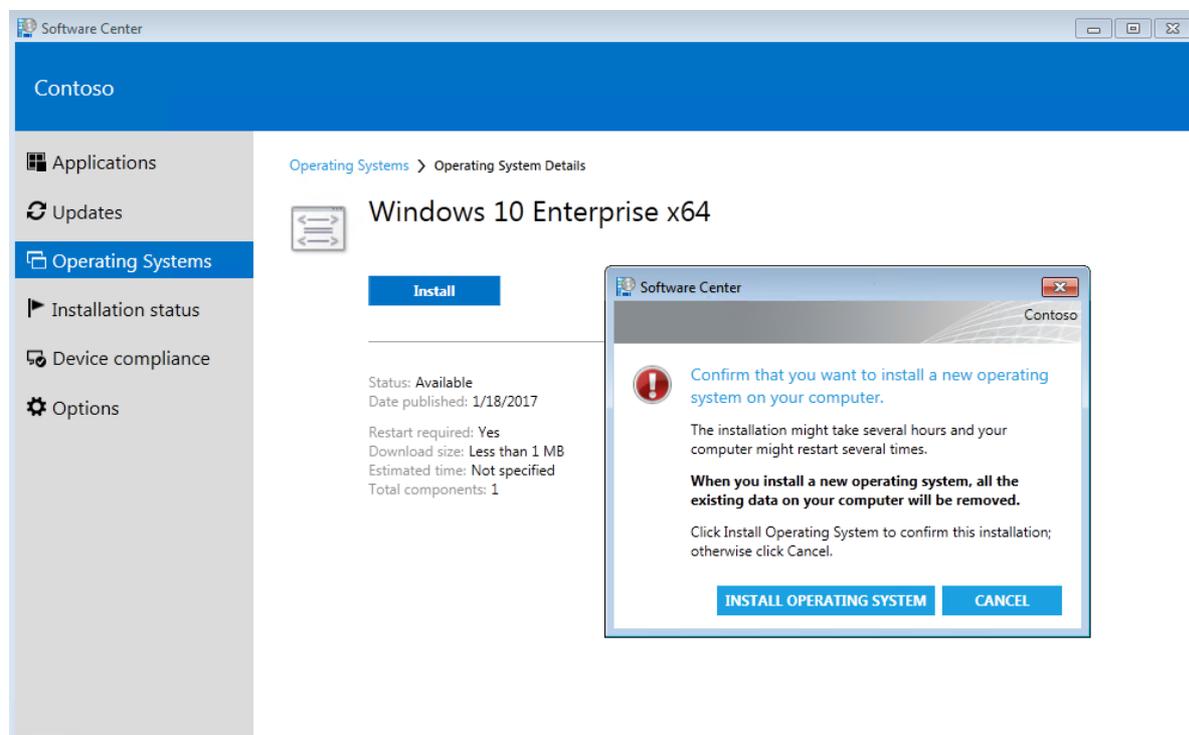
To save a checkpoint for all VMs, type the following commands at an elevated Windows PowerShell prompt on the Hyper-V host:

```
Checkpoint-VM -Name DC1 -SnapshotName cm-refresh
Checkpoint-VM -Name SRV1 -SnapshotName cm-refresh
Checkpoint-VM -Name PC1 -SnapshotName cm-refresh
```

Refresh a client with Windows 10 using Configuration Manager

Initiate the computer refresh

1. On SRV1, in the Assets and Compliance workspace, click **Device Collections** and then double-click **Install Windows 10 Enterprise x64**.
2. Right-click the computer account for PC1, point to **Client Notification**, click **Download Computer Policy**, and click **OK** in the popup dialog box.
3. On PC1, in the notification area, click **New software is available** and then click **Open Software Center**.
4. In the Software Center, click **Operating Systems**, click **Windows 10 Enterprise x64**, click **Install** and then click **INSTALL OPERATING SYSTEM**. See the following example:

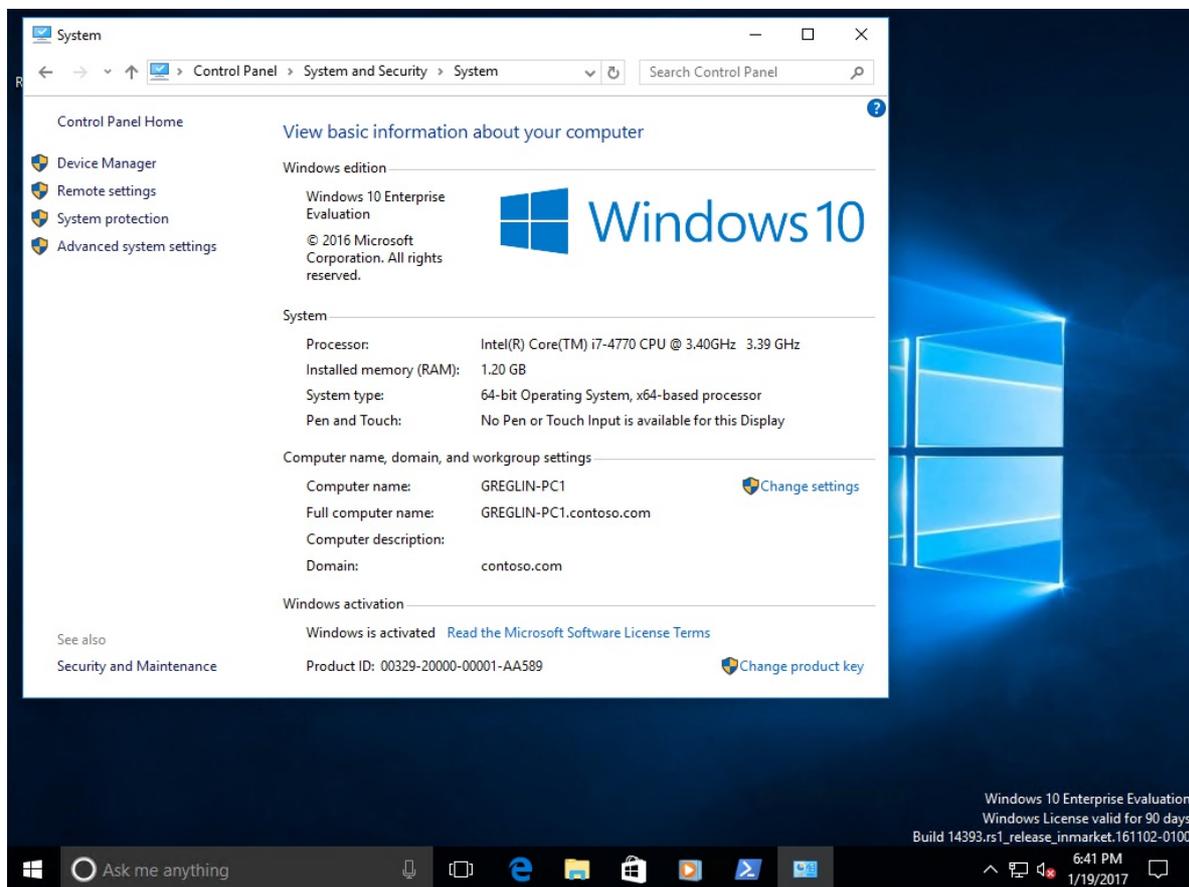


The computer will restart several times during the installation process. Installation includes downloading updates, reinstalling the Configuration Manager Client Agent, and restoring the user state. You can view status of the installation in the Configuration Manager console by accessing the Monitoring workspace, clicking **Deployments**, and then double-clicking the deployment associated with the **Install Windows 10 Enterprise x64** collection. Under **Asset Details**, right-click the device and then click **More Details**. Click the **Status** tab to see a list of tasks that have been performed. See the following example:

Asset Message						
Details		Status				
Execution Time	Step ^	Action Name	Group Name	Last Message Name	Last Message ID	Exit Code ^
1/19/2017 5:14...	41		Preinstall	The task sequence execution engine st...	11124	0
1/19/2017 5:14...	42		New Compu...	The task sequence execution engine s...	11122	0
1/19/2017 5:14...	55		Refresh Only	The task sequence execution engine st...	11124	0
1/19/2017 5:14...	56	Restart to Windows PE	Refresh Only	The task sequence execution engine s...	11134	0
1/19/2017 5:16...	56	Restart to Windows PE	Refresh Only	The task sequence execution engine p...	11142	0
1/19/2017 5:18...	57			The task sequence execution engine st...	11140	0
1/19/2017 5:19...	57	Use Toolkit Package	Refresh Only	The task sequence execution engine s...	11134	0
1/19/2017 5:19...	58	Gather	Refresh Only	The task sequence execution engine s...	11134	0
1/19/2017 5:19...	59		Refresh Only	The task sequence execution engine s...	11127	0
1/19/2017 5:19...	60	Set Status 1	Preinstall	The task sequence execution engine s...	11134	0
1/19/2017 5:19...	61		Offline USMT	The task sequence execution engine s...	11122	0
1/19/2017 5:19...	66	Backup	Preinstall	The task sequence execution engine s...	11134	0
1/19/2017 5:19...	67		Preinstall	The task sequence execution engine s...	11127	0
1/19/2017 5:19...	68		Install	The task sequence execution engine st...	11124	0
1/19/2017 5:19...	69	Set Status 2	Install	The task sequence execution engine s...	11134	0
1/19/2017 5:19...	70	Set Variable for Drive Letter	Install	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	71	Apply Operating System Image	Install	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	72	Use Toolkit Package	Install	The task sequence execution engine s...	11130	0
1/19/2017 5:50...	73		Install	The task sequence execution engine s...	11127	0
1/19/2017 5:50...	74		PostInstall	The task sequence execution engine st...	11124	0
1/19/2017 5:50...	75	Gather	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	76	Apply Windows Settings	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	77	Apply Network Settings	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	78	Configure	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	79	Auto Apply Drivers	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:50...	80	Set Status 3	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:51...	81	Setup Windows and ConfigMgr	PostInstall	The task sequence execution engine s...	11134	0
1/19/2017 5:51...	81	Setup Windows and ConfigMgr	PostInstall	The task sequence execution engine p...	11142	0

You can also monitor progress of the installation by using the MDT deployment workbench and viewing the **Monitoring** node under **Deployment Shares\MDT Production**.

When installation has completed, sign in using the contoso\administrator account or the contoso\user1 account and verify that applications and settings have been successfully backed up and restored to your new Windows 10 Enterprise operating system.



Related Topics

[System Center 2012 Configuration Manager Survival Guide](#)

Plan for Windows 10 deployment

6/18/2019 • 2 minutes to read • [Edit Online](#)

Windows 10 provides new deployment capabilities, scenarios, and tools by building on technologies introduced in Windows 7, and Windows 8.1, while at the same time introducing new Windows as a service concepts to keep the operating system up to date. Together, these changes require that you rethink the traditional deployment process.

In this section

TOPIC	DESCRIPTION
Windows 10 Enterprise: FAQ for IT professionals	Get answers to common questions around compatibility, installation, and support for Windows 10 Enterprise.
Windows 10 deployment considerations	There are new deployment options in Windows 10 that help you simplify the deployment process and automate migration of existing settings and applications.
Windows 10 compatibility	Windows 10 will be compatible with most existing PC hardware; most devices running Windows 7, Windows 8, or Windows 8.1 will meet the requirements for Windows 10.
Windows 10 infrastructure requirements	There are specific infrastructure requirements to deploy and manage Windows 10 that should be in place prior to significant Windows 10 deployments within your organization.
Features removed or planned for replacement	Information is provided about Windows 10 features and functionality that are removed or planned for replacement.
Application Compatibility Toolkit (ACT) Technical Reference	The Microsoft® Application Compatibility Toolkit (ACT) helps you determine whether the applications, devices, and computers in your organization are compatible with versions of the Windows® operating system.

Related topics

- [Windows 10 servicing options for updates and upgrades](#)
- [Deploy Windows 10 with MDT 2013 Update 1](#)
- [Deploy Windows 10 with Configuration Manager and MDT 2013 Update 1](#)
- [Upgrade to Windows 10 with MDT 2013 Update 1](#)
- [Upgrade to Windows 10 with Configuration Manager](#)
- [Windows Imaging and Configuration Designer](#)

Windows 10 Enterprise: FAQ for IT professionals

6/19/2019 • 8 minutes to read • [Edit Online](#)

Get answers to common questions around compatibility, installation, and support for Windows 10 Enterprise.

Download and requirements

Where can I download Windows 10 Enterprise?

If you have Windows volume licenses with Software Assurance, or if you have purchased licenses for Windows 10 Enterprise volume licenses, you can download 32-bit and 64-bit versions of Windows 10 Enterprise from the [Volume Licensing Service Center](#). If you do not have current Software Assurance for Windows and would like to purchase volume licenses for Windows 10 Enterprise, contact your preferred Microsoft Reseller or see [How to purchase through Volume Licensing](#).

What are the system requirements?

For details, see [Windows 10 Enterprise system requirements](#).

What are the hardware requirements for Windows 10?

Most computers that are compatible with Windows 8.1 will be compatible with Windows 10. You may need to install updated drivers in Windows 10 for your devices to properly function. See [Windows 10 specifications](#) for more information.

Can I evaluate Windows 10 Enterprise?

Yes, a 90-day evaluation of Windows 10 Enterprise is available through the [TechNet Evaluation Center](#). The evaluation is available in Chinese (Simplified), Chinese (Traditional), French, German, Italian, Japanese, Korean, Portuguese (Brazil), and Spanish (Spain, International Sort). We highly recommend that organizations make use of the Windows 10 Enterprise 90-day Evaluation to try out deployment and management scenarios, test compatibility with hardware and applications, and to get hands on experience with Windows 10 Enterprise features.

Drivers and compatibility

Where can I find drivers for my devices for Windows 10 Enterprise?

For many devices, drivers will be automatically installed in Windows 10 and there will be no need for additional action.

- For some devices, Windows 10 may be unable to install drivers that are required for operation. If your device drivers are not automatically installed, visit the manufacturer's support website for your device to download and manually install the drivers. If Windows 10 drivers are not available, the most up-to-date drivers for Windows 8.1 will often work in Windows 10.
- For some devices, the manufacturer may provide more up-to-date drivers or drivers that enable additional functionality than the drivers installed by Windows 10. Always follow the recommendations of the device manufacturer for optimal performance and stability.
- Some computer manufacturers provide packs of drivers for easy implementation in management and deployment solutions like the Microsoft Deployment Toolkit (MDT) or Microsoft System Center Configuration Manager. These driver packs contain all of the drivers needed for each device and can greatly simplify the process of deploying Windows to a new make or model of computer. Driver packs for some common manufacturers include:
 - [HP driver pack](#)

- [Dell driver packs for enterprise client OS deployment](#)
- [Lenovo Configuration Manager and MDT package index](#)
- [Panasonic Driver Pack for Enterprise](#)

Where can I find out if an application or device is compatible with Windows 10?

Many existing Win32 and Win64 applications already run reliably on Windows 10 without any changes. You can also expect strong compatibility and support for Web apps and devices. The [Ready for Windows](#) website lists software solutions that are supported and in use for Windows 10. You can find additional guidance to help with application compatibility at [Windows 10 application compatibility](#) on the Windows IT Center.

Is there an easy way to assess if my organization's devices are ready to upgrade to Windows 10?

[Windows Analytics Upgrade Readiness](#) (formerly known as Upgrade Analytics) provides powerful insights and recommendations about the computers, applications, and drivers in your organization, at no extra cost and without additional infrastructure requirements. This new service guides you through your upgrade and feature update projects using a workflow based on Microsoft recommended practices. Up-to-date inventory data allows you to balance cost and risk in your upgrade projects. You can find additional product information at [Windows Analytics](#).

Administration and deployment

Which deployment tools support Windows 10?

Updated versions of Microsoft deployment tools, including MDT, Configuration Manager, and the Windows Assessment and Deployment Kit (Windows ADK) have been released to support Windows 10.

- [MDT](#) is Microsoft's recommended collection of tools, processes, and guidance for automating desktop and server deployment.
- Configuration Manager simplifies the deployment and management of Windows 10. If you are not currently using Configuration Manager, you can download a free 180-day trial of [System Center Configuration Manager and Endpoint Protection \(current branch\)](#) from the TechNet Evaluation Center.
- The [Windows ADK](#) has tools that allow you to customize Windows images for large-scale deployment, and test system quality and performance. You can download the latest version of the Windows ADK for Windows 10 from the Hardware Dev Center.

Can I upgrade computers from Windows 7 or Windows 8.1 without deploying a new image?

Computers running Windows 7 or Windows 8.1 can be upgraded directly to Windows 10 through the in-place upgrade process without a need to reimage the device using MDT and/or Configuration Manager. For more information, see [Upgrade to Windows 10 with System Center Configuration Manager](#) or [Upgrade to Windows 10 with the Microsoft Deployment Toolkit](#).

Can I upgrade from Windows 7 Enterprise or Windows 8.1 Enterprise to Windows 10 Enterprise for free?

If you have Windows 7 Enterprise or Windows 8.1 Enterprise and current Windows 10 Enterprise E3 or E5 subscription, you are entitled to the upgrade to Windows 10 Enterprise through the rights of Software Assurance. You can find your product keys and installation media at the [Volume Licensing Service Center](#).

For devices that are licensed under a volume license agreement for Windows that does not include Software Assurance, new licenses will be required to upgrade these devices to Windows 10.

Managing updates

What is Windows as a service?

The Windows 10 operating system introduces a new way to build, deploy, and service Windows: Windows as a service. Microsoft has reimagined each part of the process, to simplify the lives of IT pros and maintain a consistent Windows 10 experience for its customers. These improvements focus on maximizing customer involvement in Windows development, simplifying the deployment and servicing of Windows client computers,

and leveling out the resources needed to deploy and maintain Windows over time. For more information, see [Overview of Windows as a service](#).

How is servicing different with Windows as a service?

Traditional Windows servicing has included several release types: major revisions (e.g., the Windows 8.1, Windows 8, and Windows 7 operating systems), service packs, and monthly updates. With Windows 10, there are two release types: feature updates that add new functionality two to three times per year, and quality updates that provide security and reliability fixes at least once a month.

What are the servicing channels?

To align with the new method of delivering feature updates and quality updates in Windows 10, Microsoft introduced the concept of servicing channels to allow customers to designate how aggressively their individual devices are updated. For example, an organization may have test devices that the IT department can update with new features as soon as possible, and then specialized devices that require a longer feature update cycle to ensure continuity. With that in mind, Microsoft offers two servicing channels for Windows 10: Semi-Annual Channel, and Long-Term Servicing Channel (LTSC). For details about the versions in each servicing channel, see [Windows 10 release information](#). For more information on each channel, see [servicing channels](#).

What tools can I use to manage Windows as a service updates?

There are many tools available. You can choose from these:

- Windows Update
- Windows Update for Business
- Windows Server Update Services
- System Center Configuration Manager

For more information on pros and cons for these tools, see [Servicing Tools](#).

User experience

Where can I find information about new features and changes in Windows 10 Enterprise?

For an overview of the new enterprise features in Windows 10 Enterprise, see [What's new in Windows 10](#) and [What's new in Windows 10, version 1703](#) in the Docs library.

Another place to track the latest information about new features of interest to IT professionals is the [Windows for IT Pros blog](#). Here you'll find announcements of new features, information on updates to the Windows servicing model, and details about the latest resources to help you more easily deploy and manage Windows 10.

To find out which version of Windows 10 is right for your organization, you can also [compare Windows editions](#).

How will people in my organization adjust to using Windows 10 Enterprise after upgrading from Windows 7 or Windows 8.1?

Windows 10 combines the best aspects of the user experience from Windows 8.1 and Windows 7 to make using Windows simple and straightforward. Users of Windows 7 will find the Start menu in the same location as they always have. In the same place, users of Windows 8.1 will find the live tiles from their Start screen, accessible by the Start button in the same way as they were accessed in Windows 8.1. To help you make the transition a seamless one, download the [Windows 10 Adoption Planning Kit](#) and see our [end user readiness](#) resources.

How does Windows 10 help people work with applications and data across a variety of devices?

The desktop experience in Windows 10 has been improved to provide a better experience for people that use a traditional mouse and keyboard. Key changes include:

- Start menu is a launching point for access to apps.
- Universal apps now open in windows instead of full screen.
- [Multitasking is improved with adjustable Snap](#), which allows you to have more than two windows side-by-side

on the same screen and to customize how those windows are arranged.

- Tablet Mode to simplify using Windows with a finger or pen by using touch input.

Help and support

Where can I ask a question about Windows 10?

Use the following resources for additional information about Windows 10.

- If you are an IT professional or if you have a question about administering, managing, or deploying Windows 10 in your organization or business, visit the [Windows 10 IT Professional forums](#) on TechNet.
- If you are an end user or if you have a question about using Windows 10, visit the [Windows 10 forums on Microsoft Community](#).
- If you are a developer or if you have a question about making apps for Windows 10, visit the [Windows Desktop Development forums](#) or [Windows and Windows phone apps forums](#) on MSDN.
- If you have a question about Internet Explorer, visit the [Internet Explorer forums](#) on TechNet.

Windows 10 deployment considerations

5/31/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

There are new deployment options in Windows 10 that help you simplify the deployment process and automate migration of existing settings and applications.

For many years, organizations have deployed new versions of Windows using a “wipe and load” deployment process. At a high level, this process captures existing data and settings from the existing device, deploys a new custom-built Windows image to a PC, injects hardware drivers, reinstalls applications, and finally restores the data and settings. With Windows 10, this process is still fully supported, and for some deployment scenarios is still necessary.

Windows 10 also introduces two additional scenarios that organizations should consider:

- **In-place upgrade**, which provides a simple, automated process that leverages the Windows setup process to automatically upgrade from an earlier version of Windows. This process automatically migrates existing data, settings, drivers, and applications.
- **Dynamic provisioning**, which enables organizations to configure new Windows 10 devices for organization use without having to deploy a new custom organization image to the device.

Both of these scenarios eliminate the image creation process altogether, which can greatly simplify the deployment process.

So how do you choose? At a high level:

CONSIDER ...	FOR THESE SCENARIOS
In-place upgrade	<ul style="list-style-type: none">• When you want to keep all (or at least most) existing applications• When you do not plan to significantly change the device configuration (for example, BIOS to UEFI) or operating system configuration (for example, x86 to x64, language changes, Administrators to non-Administrators, Active Directory domain consolidations)• To migrate from Windows 10 to a later Windows 10 release

CONSIDER ...	FOR THESE SCENARIOS
Traditional wipe-and-load	<ul style="list-style-type: none"> • When you upgrade significant numbers of applications along with the new Windows OS • When you make significant device or operating system configuration changes • When you “start clean”. For example, scenarios where it is not necessary to preserve existing apps or data (for example, call centers) or when you move from unmanaged to well-managed PCs • When you migrate from Windows Vista or other previous operating system versions
Dynamic provisioning	<ul style="list-style-type: none"> • For new devices, especially in “choose your own device” scenarios when simple configuration (not reimaging) is all that is required • When used in combination with a management tool (for example, an MDM service like Microsoft Intune) that enables self-service installation of user-specific or role-specific apps

Migration from previous Windows versions

For existing PCs running Windows 7 or Windows 8.1, in-place upgrade is the recommended method for Windows 10 deployment and should be used whenever possible. Although wipe-and-load (OS refresh) deployments are still fully supported (and necessary in some scenarios, as mentioned previously), in-place upgrade is simpler and faster, and enables a faster Windows 10 deployment overall.

Note that the original Windows 8 release is only supported until January 2016. Organizations that do not think they can complete a full Windows 10 migration by that date should deploy Windows 8.1 now and consider Windows 10 after Windows 8 has been removed from the environment.

For existing Windows PCs running Windows Vista, you can perform wipe-and-load (OS refresh) deployments when you use compatible hardware.

Note that to take advantage of the limited-time free upgrade offer for PCs running Windows 7, Windows 8, or Windows 8.1, you must leverage an in-place upgrade, either from Windows Update or by using the upgrade media available from the [Windows 10 software download page](#) to acquire a new Windows 10 license from the Windows Store. For more information, refer to the [Windows 10 FAQ](#).

For organizations with Software Assurance for Windows, both in-place upgrade or wipe-and-load can be leveraged (with in-place upgrade being the preferred method, as previously discussed).

For organizations that do not take advantage of the free upgrade offer and are not enrolled in Software Assurance for Windows, Windows 10 upgrade licenses are available for purchase through existing Volume License (VL) agreements.

Setup of new computers

For new computers acquired with Windows 10 preinstalled, you can leverage dynamic provisioning scenarios to transform the device from its initial state into a fully-configured organization PC. There are two primary dynamic provisioning scenarios you can use:

- **User-driven, from the cloud.** By joining a device into Azure Active Directory and leveraging the automatic mobile device management (MDM) provisioning capabilities at the same time, an end user can initiate the provisioning process themselves just by entering the Azure Active Directory account and password (called their “work or school account” within Windows 10). The MDM service can then transform the device into a fully-configured organization PC. For more information, see [Azure Active Directory integration with MDM](#).
- **IT admin-driven, using new tools.** Using the new Windows Imaging and Configuration Designer (ICD) tool, IT administrators can create provisioning packages that can be applied to a computer to transform it into a fully-configured organization PC. For more information, see [Windows Imaging and Configuration Designer](#).

In either of these scenarios, you can make a variety of configuration changes to the PC:

- Transform the edition (SKU) of Windows 10 that is in use.
- Apply configuration and settings to the device (for example, security settings, device restrictions, policies, Wi-Fi and VPN profiles, certificates, and so on).
- Install apps, language packs, and updates.
- Enroll the device in a management solution (applicable for IT admin-driven scenarios, configuring the device just enough to allow the management tool to take over configuration and ongoing management).

Stay up to date

For computers already running Windows 10 on the Semi-Annual Channel, new upgrades will periodically be deployed, approximately two to three times per year. You can deploy these upgrades by using a variety of methods:

- Windows Update or Windows Update for Business, for devices where you want to receive updates directly from the Internet.
- Windows Server Update Services (WSUS), for devices configured to pull updates from internal servers after they are approved (deploying like an update). Note that this will require updates to WSUS, which are only available for Windows Server 2012 and Windows Server 2012 R2, not previous versions.
- System Center Configuration Manager task sequences (with Configuration Manager 2012, 2012 R2, and later versions).
- System Center Configuration Manager vNext software update capabilities (deploying like an update).

Note that these upgrades (which are installed differently than monthly updates) will leverage an in-place upgrade process. Unlike updates, which are relatively small, these upgrades will include a full operating system image (around 3 GB for 64-bit operating systems), which requires time (1-2 hours) and disk space (approximately 10 GB) to complete. Ensure that the deployment method you use can support the required network bandwidth and/or disk space requirements.

Over time, this upgrade process will be optimized to reduce the overall time and network bandwidth consumed.

Related topics

[Windows 10 compatibility](#)

[Windows 10 infrastructure requirements](#)

Windows 10 compatibility

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Windows 10 will be compatible with most existing PC hardware; most devices running Windows 7, Windows 8, or Windows 8.1 will meet the requirements for Windows 10.

For full system requirements, see [Windows 10 specifications](#). Some driver updates may be required for Windows 10.

Existing desktop (Win32) application compatibility is also expected to be strong, with most existing applications working without any changes. Some applications that interface with Windows at a low level, those that use undocumented APIs, or those that do not follow recommended coding practices could experience issues.

Existing Windows Store (WinRT) apps created for Windows 8 and Windows 8.1 should also continue to work, because compatibility can be validated against all the apps that have been submitted to the Windows Store.

For web apps and sites, modern HTML5-based sites should also have a high degree of compatibility and excellent performance through the new Microsoft Edge browser, while older web apps and sites can continue to use Internet Explorer 11 and the Enterprise Mode features that were first introduced in Windows 7 and Windows 8.1 and are still present in Windows 10. For more information about Internet Explorer and Enterprise Mode, see the [Internet Explorer 11 Deployment Guide for IT Pros](#).

Recommended application testing process

Historically, organizations have performed extensive, and often exhaustive, testing of the applications they use before deployment of a new Windows version, service pack, or any other significant update. With Windows 10, organizations are encouraged to leverage more optimized testing processes, which reflects the higher levels of compatibility that are expected. At a high level:

- Identify mission-critical applications and websites, those that are absolutely essential to the organization's operations. Focus testing efforts on this subset of applications, early in the Windows development cycle (for example, with Windows Insider Program builds) to identify potential issues. Report any issues you encounter with the Windows Feedback tool, so that these issues can be addressed prior to the next Windows release.
- For less critical applications, leverage an "internal flighting" or pilot-based approach, by deploying new Windows upgrades to groups of machines, growing gradually in size and potential impact, to verify compatibility with hardware and software. Reactively address issues before you expand the pilot to more machines.

Related topics

[Windows 10 servicing options](#)

[Windows 10 deployment considerations](#)

[Windows 10 infrastructure requirements](#)

Windows 10 infrastructure requirements

5/31/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

There are specific infrastructure requirements to deploy and manage Windows 10 that should be in place prior to significant Windows 10 deployments within your organization.

High-level requirements

For initial Windows 10 deployments, as well as subsequent Windows 10 upgrades, ensure that sufficient disk space is available for distribution of the Windows 10 installation files (about 3 GB for Windows 10 x64 images, slightly smaller for x86). Also, be sure to take into account the network impact of moving these large images to each PC; you may need to leverage local server storage.

For persistent VDI environments, carefully consider the I/O impact from upgrading large numbers of PCs in a short period of time. Ensure that upgrades are performed in smaller numbers, or during off-peak time periods. (For pooled VDI environments, a better approach is to replace the base image with a new version.)

Deployment tools

A new version of the Assessment and Deployment Toolkit (ADK) has been released to support Windows 10. This new version, available for download [here](#), is required for Windows 10; you should not use earlier versions of the ADK to deploy Windows 10. It also supports the deployment of Windows 7, Windows 8, and Windows 8.1.

Significant enhancements in the ADK for Windows 10 include new runtime provisioning capabilities, which leverage the Windows Imaging and Configuration Designer (Windows ICD), as well as updated versions of existing deployment tools (DISM, USMT, Windows PE, and more).

Microsoft Deployment Toolkit 2013 Update 1, available for download [here](#), has also been updated to support Windows 10 and the new ADK; older versions do not support Windows 10. New in this release is task sequence support for Windows 10 in-place upgrades.

For System Center Configuration Manager, Windows 10 support is offered with various releases:

RELEASE	WINDOWS 10 MANAGEMENT?	WINDOWS 10 DEPLOYMENT?
System Center Configuration Manager 2007	Yes, with a hotfix	No
System Center Configuration Manager 2012	Yes, with SP2 and CU1	Yes, with SP2, CU1, and the ADK for Windows 10
System Center Configuration Manager 2012 R2	Yes, with SP1 and CU1	Yes, with SP1, CU1, and the ADK for Windows 10

Note: Configuration Manager 2012 supports Windows 10 version 1507 (build 10.0.10240) and 1511 (build 10.0.10586) for the lifecycle of these builds. Future releases of Windows 10 CB/CBB are not supported With Configuration Manager 2012, and will require System Center Configuration Manager current branch for supported management.

For more details about System Center Configuration Manager support for Windows 10, see [Deploy Windows 10 with System Center 2012 R2 Configuration Manager](#).

Management tools

In addition to System Center Configuration Manager, Windows 10 also leverages other tools for management. For Windows Server and Active Directory, existing supported versions are fully supported for Windows 10. New Group Policy templates will be needed to configure new settings available in Windows 10; these templates are available in the Windows 10 media images, and are available as a separate download [here](#). See [Group Policy settings reference](#) for a list of the new and modified policy settings. If you are using a central policy store, follow the steps outlined [here](#) to update the ADMX files stored in that central store.

No new Active Directory schema updates or specific functional levels are currently required for core Windows 10 product functionality, although subsequent upgrades could require these to support new features.

Microsoft Desktop Optimization Pack (MDOP) has been updated to support Windows 10. The minimum versions required to support Windows 10 are as follows:

PRODUCT	REQUIRED VERSION
Advanced Group Policy Management (AGPM)	AGPM 4.0 Service Pack 3
Application Virtualization (App-V)	App-V 5.1
Diagnostics and Recovery Toolkit (DaRT)	DaRT 10
Microsoft BitLocker Administration and Monitoring (MBAM)	MBAM 2.5 SP1 (2.5 is OK)
User Experience Virtualization (UE-V)	UE-V 2.1 SP1

For more information, see the [MDOP TechCenter](#).

For devices you manage with mobile device management (MDM) solutions such as Microsoft Intune, existing capabilities (provided initially in Windows 8.1) are fully supported in Windows 10; new Windows 10 MDM settings and capabilities will require updates to the MDM services. See [Mobile device management](#) for more information.

Windows Server Update Services (WSUS) requires some additional configuration to receive updates for Windows 10. Use the Windows Server Update Services admin tool and follow these instructions:

1. Select the **Options** node, and then click **Products and Classifications**.
2. In the **Products** tree, select the **Windows 10** and **Windows 10 LTSB** products and any other Windows 10-related items that you want. Click **OK**.
3. From the **Synchronizations** node, right-click and choose **Synchronize Now**.

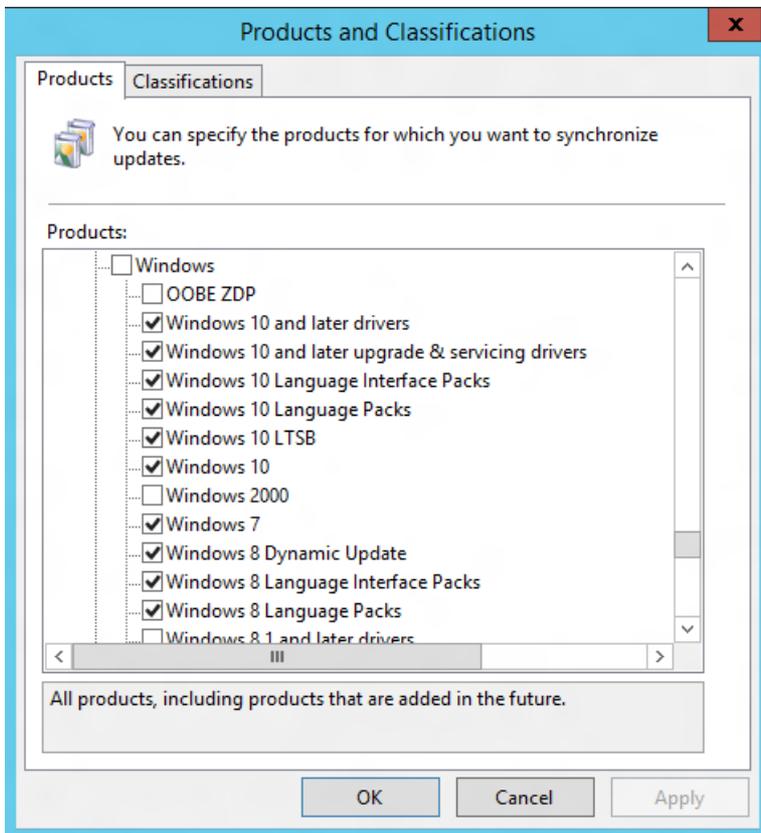


Figure 1. WSUS product list with Windows 10 choices

Because Windows 10 updates are cumulative in nature, each month's new update will supersede the previous month's. Consider leveraging "express installation" packages to reduce the size of the payload that needs to be sent to each PC each month; see [Express installation files](#) for more information. (Note that this will increase the amount of disk storage needed by WSUS, and impacts all operating systems being managed with WSUS.)

Activation

Windows 10 volume license editions of Windows 10 will continue to support all existing activation methods (KMS, MAK, and AD-based activation). An update will be required for existing KMS servers:

PRODUCT	REQUIRED UPDATE
Windows 10	None
Windows Server 2012 R2 and Windows 8.1	https://support.microsoft.com/kb/3058168
Windows Server 2012 and Windows 8	https://support.microsoft.com/kb/3058168
Windows Server 2008 R2 and Windows 7	https://support.microsoft.com/kb/3079821

Also see: [Windows Server 2016 Volume Activation Tips](#)

Additionally, new product keys will be needed for all types of volume license activation (KMS, MAK, and AD-based Activation); these keys are available on the Volume Licensing Service Center (VLSC) for customers with rights to the Windows 10 operating system. To find the needed keys:

- Sign into the [Volume Licensing Service Center \(VLSC\)](#) at with a Microsoft account that has appropriate rights.

- For KMS keys, click **Licenses** and then select **Relationship Summary**. Click the appropriate active license ID, and then select **Product Keys** near the right side of the page. For KMS running on Windows Server, find the **Windows Srv 2012R2 DataCtr/Std KMS for Windows 10** product key; for KMS running on client operating systems, find the **Windows 10** product key.
- For MAK keys, click **Downloads and Keys**, and then filter the list by using **Windows 10** as a product. Click the **Key** link next to an appropriate list entry (for example, **Windows 10 Enterprise** or **Windows 10 Enterprise LTSB**) to view the available MAK keys. (You can also find keys for KMS running on Windows 10 in this list. These keys will not work on Windows servers running KMS.)

Note that Windows 10 Enterprise and Windows 10 Enterprise LTSB installations use different MAK keys. But you can use the same KMS server or Active Directory-based activation environment for both; the KMS keys obtained from the Volume Licensing Service Center will work with both.

Related topics

[Windows 10 servicing options](#)

[Windows 10 deployment considerations](#)

[Windows 10 compatibility](#)

Volume Activation for Windows 10

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for volume licensing information?

- [Download the Volume Licensing Reference Guide for Windows 10 Desktop Operating System](#)

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

This guide is designed to help organizations that are planning to use volume activation to deploy and activate Windows 10, including organizations that have used volume activation for earlier versions of Windows. *Volume activation* is the process that Microsoft volume licensing customers use to automate and manage the activation of Windows operating systems, Microsoft Office, and other Microsoft products across large organizations. Volume licensing is available to customers who purchase software under various volume programs (such as Open and Select) and to participants in programs such as the Microsoft Partner Program and MSDN Subscriptions.

Volume activation is a configurable solution that helps automate and manage the product activation process on computers running Windows operating systems that have been licensed under a volume licensing program. Volume activation is also used with other software from Microsoft (most notably the Office suites) that are sold under volume licensing agreements and that support volume activation.

This guide provides information and step-by-step guidance to help you choose a volume activation method that suits your environment, and then to configure that solution successfully. This guide describes the volume activation features that are available in Windows 10 and Windows Server 2012 R2 and the tools that are provided in these versions of Windows and Windows Server to manage volume activation.

Because most organizations will not immediately switch all computers to Windows 10, practical volume activation strategies must also take in to account how to work with the Windows 8, Windows 7, Windows Server 2012, and Windows Server 2008 R2 operating systems. This guide discusses how the new volume activation tools can support earlier operating systems, but it does not discuss the tools that are provided with earlier operating system versions.

Volume activation—and the need for activation itself—is not new, and this guide does not review all of its concepts and history. You can find additional background in the appendices of this guide. For more information, see [Volume Activation Overview](#) in the TechNet Library.

If you would like additional information about planning a volume activation deployment specifically for Windows 7 and Windows Server 2008 R2, please see the [Volume Activation Planning Guide for Windows 7](#).

To successfully plan and implement a volume activation strategy, you must:

- Learn about and understand product activation.
- Review and evaluate the available activation types or models.
- Consider the connectivity of the clients to be activated.
- Choose the method or methods to be used with each type of client.
- Determine the types and number of product keys you will need.
- Determine the monitoring and reporting needs in your organization.
- Install and configure the tools required to support the methods selected.

Keep in mind that the method of activation does not change an organization's responsibility to the licensing requirements. You must ensure that all software used in your organization is properly licensed and activated in accordance with the terms of the licensing agreements in place.

In this guide:

- [Plan for volume activation](#)
- [Activate using Key Management Service](#)
- [Activate using Active Directory-based activation](#)
- [Activate clients running Windows 10](#)
- [Monitor activation](#)
- [Use the Volume Activation Management Tool](#)
- [Appendix: Information sent to Microsoft during activation](#)

Plan for volume activation

6/10/2019 • 18 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

Product activation is the process of validating software with the manufacturer after it has been installed on a specific computer. Activation confirms that the product is genuine—not a fraudulent copy—and that the product key or serial number is valid and has not been compromised or revoked. Activation also establishes a link or relationship between the product key and the particular installation.

During the activation process, information about the specific installation is examined. In the case of online activations, this information is sent to a server at Microsoft. This information may include the software version, the product key, the IP address of the computer, and information about the device. The activation methods that Microsoft uses are designed to help protect user privacy, and they cannot be used to track back to the computer or user. The gathered data confirms that the software is a legally licensed copy, and this data is used for statistical analysis. Microsoft does not use this information to identify or contact the user or the organization.

NOTE

The IP address is used only to verify the location of the request, because some editions of Windows (such as “Starter” editions) can only be activated within certain geographical target markets.

Distribution channels and activation

In general, Microsoft software is obtained through three main channels: retail, original equipment manufacturer (OEM), and volume licensing agreements. Different activations methods are available through each channel. Because organizations are free to obtain software through multiple channels (for example, buying some at retail and others through a volume licensing program) most organizations choose to use a combination of activation methods.

Retail activations

The retail activation method has not changed in several versions of Windows and Windows Server. Each purchased copy comes with one unique product key (often referred to as a retail key). The user enters this key during product installation. The computer uses this retail key to complete the activation after the installation is complete. Most activations are performed online, but telephone activation is also available. Recently, retail keys have been expanded into new distribution scenarios. Product key cards are available to activate products that have been preinstalled or downloaded. Programs such as Windows Anytime Upgrade and Get Genuine allow users to acquire legal keys separately from the software. These electronically distributed keys may come with media that

contains software, they can come as a software shipment, or they may be provided on a printed card or electronic copy. Products are activated the same way with any of these retail keys.

Original equipment manufacturer

Most original equipment manufacturers (OEMs) sell systems that include a standard build of the Windows operating system. The hardware vendor activates Windows by associating the operating system with the firmware (BIOS) of the computer. This occurs before the computer is sent to the customer, and no additional actions are required. OEM activation is valid as long as the customer uses the OEM-provided image on the system. OEM activation is available only for computers that are purchased through OEM channels and have the Windows operating system preinstalled.

Volume licensing

Volume licensing offers customized programs that are tailored to the size and purchasing preference of the organization. To become a volume licensing customer, the organization must set up a volume licensing agreement with Microsoft. There is a common misunderstanding about acquiring licenses for a new computer through volume licensing. There are two legal ways to acquire a full Windows client license for a new computer:

- Have the license preinstalled through the OEM.
- Purchase a fully packaged retail product.

The licenses that are provided through volume licensing programs such as Open License, Select License, and Enterprise Agreements cover upgrades to Windows client operating systems only. An existing retail or OEM operating system license is needed for each computer running Windows 10, Windows 8.1 Pro, Windows 8 Pro, Windows 7 Professional or Ultimate, or Windows XP Professional before the upgrade rights obtained through volume licensing can be exercised. Volume licensing is also available through certain subscription or membership programs, such as the Microsoft Partner Network and MSDN. These volume licenses may contain specific restrictions or other changes to the general terms applicable to volume licensing.

Note Some editions of the operating system, such as Windows 10 Enterprise, and some editions of application software are available only through volume licensing agreements or subscriptions.

Activation models

For a user or IT department, there are no significant choices about how to activate products that are acquired through retail or OEM channels. The OEM performs the activation at the factory, and the user or the IT department need take no activation steps.

With a retail product, the Volume Activation Management Tool (VAMT), which is discussed later in this guide, helps you track and manage keys. For each retail activation, you can choose:

- Online activation
- Telephone activation
- VAMT proxy activation

Telephone activation is primarily used in situations where a computer is isolated from all networks. VAMT proxy activation (with retail keys) is sometimes used when an IT department wants to centralize retail activations or when a computer with a retail version of the operating system is isolated from the Internet but connected to the LAN. For volume-licensed products, however, you must determine the best method or combination of methods to use in your environment. For Windows 10 Pro and Enterprise, you can choose from three models:

- MAKs
- KMS
- Active Directory-based activation

Note A specialized method, Token-based activation, is available for specific situations when approved customers

rely on a public key infrastructure in a completely isolated, and usually high-security, environment. For more information, contact your Microsoft Account Team or your service representative. Token-based Activation option is available for Windows 10 Enterprise LTSB editions (Version 1507 and 1607).

Multiple activation key

A Multiple Activation Key (MAK) is commonly used in small- or mid-sized organizations that have a volume licensing agreement, but they do not meet the requirements to operate a KMS or they prefer a simpler approach. A MAK also allows permanent activation of computers that are isolated from the KMS or are part of an isolated network that does not have enough computers to use the KMS.

To use a MAK, the computers to be activated must have a MAK installed. The MAK is used for one-time activation with the Microsoft online hosted activation services, by telephone, or by using VAMT proxy activation. In the simplest terms, a MAK acts like a retail key, except that a MAK is valid for activating multiple computers. Each MAK can be used a specific number of times. The VAMT can assist in tracking the number of activations that have been performed with each key and how many remain.

Organizations can download MAK and KMS keys from the [Volume Licensing Service Center](#) website. Each MAK has a preset number of activations, which are based on a percentage of the count of licenses the organization purchases; however, you can increase the number of activations that are available with your MAK by calling Microsoft.

Key Management Service

With the Key Management Service (KMS), IT pros can complete activations on their local network, eliminating the need for individual computers to connect to Microsoft for product activation. The KMS is a lightweight service that does not require a dedicated system and can easily be cohosted on a system that provides other services.

Volume editions of Windows 10 and Windows Server 2012 R2 (in addition to volume editions of operating system editions since Windows Vista and Windows Server 2008) automatically connect to a system that hosts the KMS to request activation. No action is required from the user.

The KMS requires a minimum number of computers (physical computers or virtual machines) in a network environment. The organization must have at least five computers to activate Windows Server 2012 R2 and at least 25 computers to activate client computers that are running Windows 10. These minimums are referred to as *activation thresholds*.

Planning to use the KMS includes selecting the best location for the KMS host and how many KMS hosts to have. One KMS host can handle a large number of activations, but organizations will often deploy two KMS hosts to ensure availability. Only rarely would more than two KMS hosts be used. The KMS can be hosted on a client computer or on a server, and it can be run on older versions of the operating system if proper configuration steps are taken. Setting up your KMS is discussed later in this guide.

Active Directory-based activation

Active Directory-based activation is the newest type of volume activation, and it was introduced in Windows 8. In many ways, Active Directory-based activation is similar to activation by using the KMS, but the activated computer does not need to maintain periodic connectivity with the KMS host. Instead, a domain-joined computer running Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, or Windows Server 2012 R2 queries AD DS for a volume activation object that is stored in the domain. The operating system checks the digital signatures that are contained in the activation object, and then activates the device.

Active Directory-based activation allows enterprises to activate computers through a connection to their domain. Many companies have computers at remote or branch locations, where it is impractical to connect to a KMS, or would not reach the KMS activation threshold. Rather than use MAKs, Active Directory-based activation provides a way to activate computers running Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, or Windows Server 2012 R2 as long as the computers can contact the company's domain. Active Directory-based activation offers the advantage of extending volume activation services everywhere you already have a domain presence.

Network and connectivity

A modern business network has many nuances and interconnections. This section examines evaluating your network and the connections that are available to determine how volume activations will occur.

Core network

Your core network is that part of your network that enjoys stable, high-speed, reliable connectivity to infrastructure servers. In many cases, the core network is also connected to the Internet, although that is not a requirement to use the KMS or Active Directory-based activation after the KMS server or AD DS is configured and active. Your core network likely consists of many network segments. In many organizations, the core network makes up the vast majority of the business network.

In the core network, a centralized KMS solution is usually recommended. You can also use Active Directory-based activation, but in many organizations, KMS will still be required to activate older client computers and computers that are not joined to the domain. Some administrators prefer to run both solutions to have the most flexibility, while others prefer to choose only a KMS-based solution for simplicity. Active Directory-based activation as the only solution is workable if all of the clients in your organization are running Windows 10, Windows 8.1, or Windows 8.

A typical core network that includes a KMS host is shown in Figure 1.

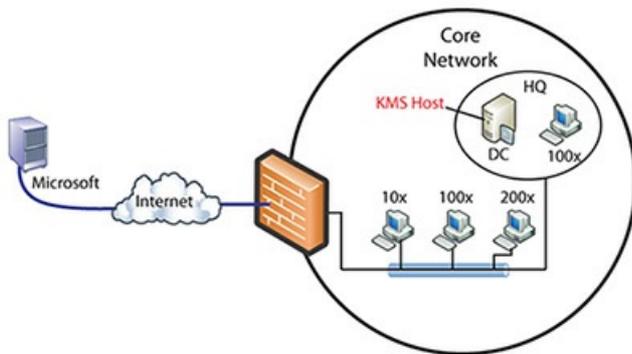


Figure 1. Typical core network

Isolated networks

In a large network, it is all but guaranteed that some segments will be isolated, either for security reasons or because of geography or connectivity issues.

Isolated for security

Sometimes called a *high-security zone*, a particular network segment may be isolated from the core network by a firewall or disconnected from other networks totally. The best solution for activating computers in an isolated network depends on the security policies in place in the organization.

If the isolated network can access the core network by using outbound requests on TCP port 1688, and it is allowed to receive remote procedure calls (RPCs), you can perform activation by using the KMS in the core network, thereby avoiding the need to reach additional activation thresholds.

If the isolated network participates fully in the corporate forest, and it can make typical connections to domain controllers, such as using Lightweight Directory Access Protocol (LDAP) for queries and Domain Name Service (DNS) for name resolution, this is a good opportunity to use Active Directory-based activation for Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, and Windows Server 2012 R2.

If the isolated network cannot communicate with the core network's KMS server, and it cannot use Active Directory-based activation, you can set up a KMS host in the isolated network. This configuration is shown in Figure 2. However, if the isolated network contains only a few computers, it will not reach the KMS activation threshold. In that case, you can activate by using MAKs.

If the network is fully isolated, MAK-independent activation would be the recommended choice, perhaps using the telephone option. But VAMT proxy activation may also be possible. You can also use MAKs to activate new computers during setup, before they are placed in the isolated network.

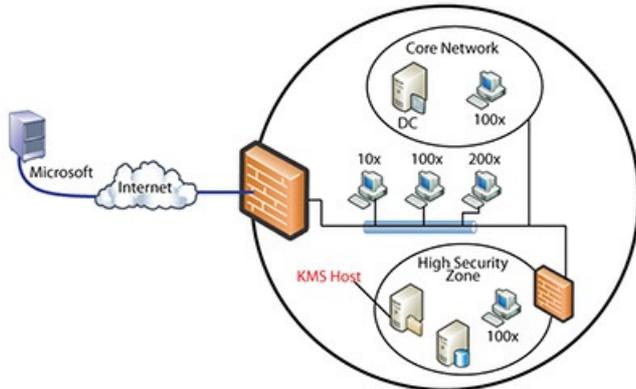


Figure 2. New KMS host in an isolated network

Branch offices and distant networks From mining operations to ships at sea, organizations often have a few computers that are not easily connected to the core network or the Internet. Some organizations have network segments at branch offices that are large and well-connected internally, but have a slow or unreliable WAN link to the rest of the organization. In these situations, you have several options:

- **Active Directory-based activation.** In any site where the client computers are running Windows 10, Active Directory-based activation is supported, and it can be activated by joining the domain.
- **Local KMS.** If a site has 25 or more client computers, it can activate against a local KMS server.
- **Remote (core) KMS.** If the remote site has connectivity to an existing KMS (perhaps through a virtual private network (VPN) to the core network), that KMS can be used. Using the existing KMS means that you only need to meet the activation threshold on that server.
- **MAK activation.** If the site has only a few computers and no connectivity to an existing KMS host, MAK activation is the best option.

Disconnected computers

Some users may be in remote locations or may travel to many locations. This scenario is common for roaming clients, such as the computers that are used by salespeople or other users who are offsite but not at branch locations. This scenario can also apply to remote branch office locations that have no connection to the core network. You can consider this an “isolated network,” where the number of computers is one. Disconnected computers can use Active Directory-based activation, the KMS, or MAK depending on the client version and how often the computers connect to the core network. If the computer is joined to the domain and running Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, or Windows Server 2012 R2 8, you can use Active Directory-based activation—directly or through a VPN—at least once every 180 days. If the computer connects to a network with a KMS host at least every 180 days, but it does not support Active Directory-based activation, you can use KMS activation. Otherwise for computers that rarely or never connect to the network, use MAK independent activation (by using the telephone or the Internet).

Test and development labs

Lab environments often have large numbers of virtual machines, and physical computers and virtual machines in labs are reconfigured frequently. Therefore, first determine whether the computers in test and development labs require activation. Editions of Windows 10 that include volume licensing will operate normally, even if they cannot activate immediately. If you have ensured that your test or development copies of the operating system are within the license agreement, you may not need to activate the lab computers if they will be rebuilt frequently. If you require that the lab computers be activated, treat the lab as an isolated network and use the methods described earlier in this guide. In labs that have a high turnover of computers and a small number of KMS clients, you must monitor the KMS activation count. You might need to adjust the time that the KMS caches the activation requests.

The default is 30 days.

Mapping your network to activation methods

Now it's time to assemble the pieces into a working solution. By evaluating your network connectivity, the numbers of computers you have at each site, and the operating system versions in use in your environment, you have collected the information you need to determine which activation methods will work best for you. You can fill-in information in Table 1 to help you make this determination.

Table 1. Criteria for activation methods

CRITERION	ACTIVATION METHOD
Number of domain-joined computers that support Active Directory-based activation (computers running Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, or Windows Server 2012 R2) and will connect to a domain controller at least every 180 days. Computers can be mobile, semi-isolated, or located in a branch office or the core network.	Active Directory-based activation
Number of computers in the core network that will connect (directly or through a VPN) at least every 180 days Note The core network must meet the KMS activation threshold.	KMS (central)
Number of computers that do not connect to the network at least once every 180 days (or if no network meets the activation threshold)	MAM
Number of computers in semi-isolated networks that have connectivity to the KMS in the core network	KMS (central)
Number of computers in isolated networks where the KMS activation threshold is met	KMS (local)
Number of computers in isolated networks where the KMS activation threshold is not met	MAK
Number of computers in test and development labs that will not be activated	None
Number of computers that do not have a retail volume license	Retail (online or phone)
Number of computers that do not have an OEM volume license	OEM (at factory)
Total number of computer activations Note This total should match the total number of licensed computers in your organization.	

Choosing and acquiring keys

When you know which keys you need, you must obtain them. Generally speaking, volume licensing keys are

collected in two ways:

- Go to the **Product Keys** section of the [Volume Licensing Service Center](#) for the following agreements: Open, Open Value, Select, Enterprise, and Services Provider License.
- Contact your [Microsoft Activation Center](#).

KMS host keys

A KMS host needs a key that activates, or authenticates, the KMS host with Microsoft. This key is usually referred to as the *KMS host key*, but it is formally known as a *Microsoft Customer Specific Volume License Key (CSVLK)*. Most documentation and Internet references earlier than Windows 8.1 use the term KMS key, but CSVLK is becoming more common in current documentation and management tools.

A KMS host running Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 can activate both Windows Server and Windows client operating systems. A KMS host key is also needed to create the activation objects in AD DS, as described later in this guide. You will need a KMS host key for any KMS that you want to set up and if you are going to use Active Directory-based activation.

Generic volume licensing keys

When you create installation media or images for client computers that will be activated by KMS or Active Directory-based activation, install a generic volume license key (GVLK) for the edition of Windows you are creating. GVLKs are also referred to as KMS client setup keys.

Installation media from Microsoft for Enterprise editions of the Windows operating system may already contain the GVLK. One GVLK is available for each type of installation. Note that the GVLK will not activate the software against Microsoft activation servers, only against a KMS or Active Directory-based activation object. In other words, the GVLK does not work unless a valid KMS host key can be found. GVLKs are the only product keys that do not need to be kept confidential.

Typically, you will not need to manually enter a GVLK unless a computer has been activated with a MAK or a retail key and it is being converted to a KMS activation or to Active Directory-based activation. If you need to locate the GVLK for a particular client edition, see [Appendix A: KMS Client Setup Keys](#).

Multiple activation keys

You will also need MAK keys with the appropriate number of activations available. You can see how many times a MAK has been used on the Volume Licensing Service Center website or in the VAMT.

Selecting a KMS host

The KMS does not require a dedicated server. It can be cohosted with other services, such as AD DS domain controllers and read-only domain controllers. KMS hosts can run on physical computers or virtual machines that are running any supported Windows operating system. A KMS host that is running Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 can activate any Windows client or server operating system that supports volume activation. A KMS host that is running Windows 10 can activate only computers running Windows 10, Windows 8.1, Windows 8, Windows 7, or Windows Vista. A single KMS host can support unlimited numbers of KMS clients, but Microsoft recommends deploying a minimum of two KMS hosts for failover purposes. However, as more clients are activated through Active Directory-based activation, the KMS and the redundancy of the KMS will become less important. Most organizations can use as few as two KMS hosts for their entire infrastructure.

The flow of KMS activation is shown in Figure 3, and it follows this sequence:

1. An administrator uses the VAMT console to configure a KMS host and install a KMS host key.
2. Microsoft validates the KMS host key, and the KMS host starts to listen for requests.
3. The KMS host updates resource records in DNS to allow clients to locate the KMS host. (Manually adding DNS records is required if your environment does not support DNS dynamic update protocol.)

4. A client configured with a GVLK uses DNS to locate the KMS host.
5. The client sends one packet to the KMS host.
6. The KMS host records information about the requesting client (by using a client ID). Client IDs are used to maintain the count of clients and detect when the same computer is requesting activation again. The client ID is only used to determine whether the activation thresholds are met. The IDs are not stored permanently or transmitted to Microsoft. If the KMS is restarted, the client ID collection starts again.
7. If the KMS host has a KMS host key that matches the products in the GVLK, the KMS host sends a single packet back to the client. This packet contains a count of the number of computers that have requested activation from this KMS host.
8. If the count exceeds the activation threshold for the product that is being activated, the client is activated. If the activation threshold has not yet been met, the client will try again.

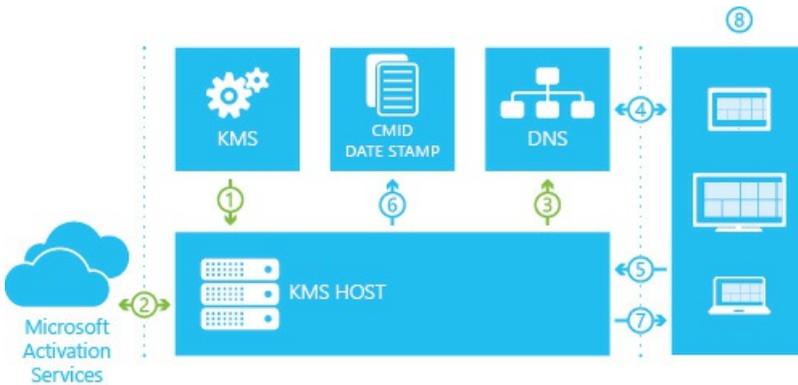


Figure 3. KMS activation flow

See also

- [Volume Activation for Windows 10](#)

Activate using Key Management Service

6/6/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

There are three possible scenarios for volume activation of Windows 10 or Windows Server 2012 R2 by using a Key Management Service (KMS) host:

- Host KMS on a computer running Windows 10
- Host KMS on a computer running Windows Server 2012 R2
- Host KMS on a computer running an earlier version of Windows

Check out [Windows 10 Volume Activation Tips](#).

Key Management Service in Windows 10

Installing a KMS host key on a computer running Windows 10 allows you to activate other computers running Windows 10 against this KMS host and earlier versions of the client operating system, such as Windows 8.1 or Windows 7. Clients locate the KMS server by using resource records in DNS, so some configuration of DNS may be required. This scenario can be beneficial if your organization uses volume activation for clients and MAK-based activation for a smaller number of servers. To enable KMS functionality, a KMS key is installed on a KMS host; then, the host is activated over the Internet or by phone using Microsoft's activation services.

Configure KMS in Windows 10

1. Open an elevated command prompt.
2. Enter one of the following commands.
 - To install a KMS key, type **slmgr.vbs /ipk <KmsKey>**.
 - To activate online, type **slmgr.vbs /ato**.
 - To activate by using the telephone, type **slui.exe 4**.
3. After activating the KMS key, restart the Software Protection Service.

For more information, see the information for Windows 7 in [Deploy KMS Activation](#).

Key Management Service in Windows Server 2012 R2

Installing a KMS host key on a computer running Windows Server allows you to activate computers running Windows Server 2012 R2, Windows Server 2008 R2, Windows Server 2008, Windows 10, Windows 8.1, Windows 7, and Windows Vista.

Note You cannot install a client KMS key into the KMS in Windows Server.

This scenario is commonly used in larger organizations that do not find the overhead of using a server a burden.

Note

If you receive error 0xC004F015 when trying to activate Windows 10 Enterprise, see [KB 3086418](#).

Configure KMS in Windows Server 2012 R2

1. Sign in to a computer running Windows Server 2012 R2 with an account that has local administrative credentials.
2. Launch Server Manager.
3. Add the Volume Activation Services role, as shown in Figure 4.

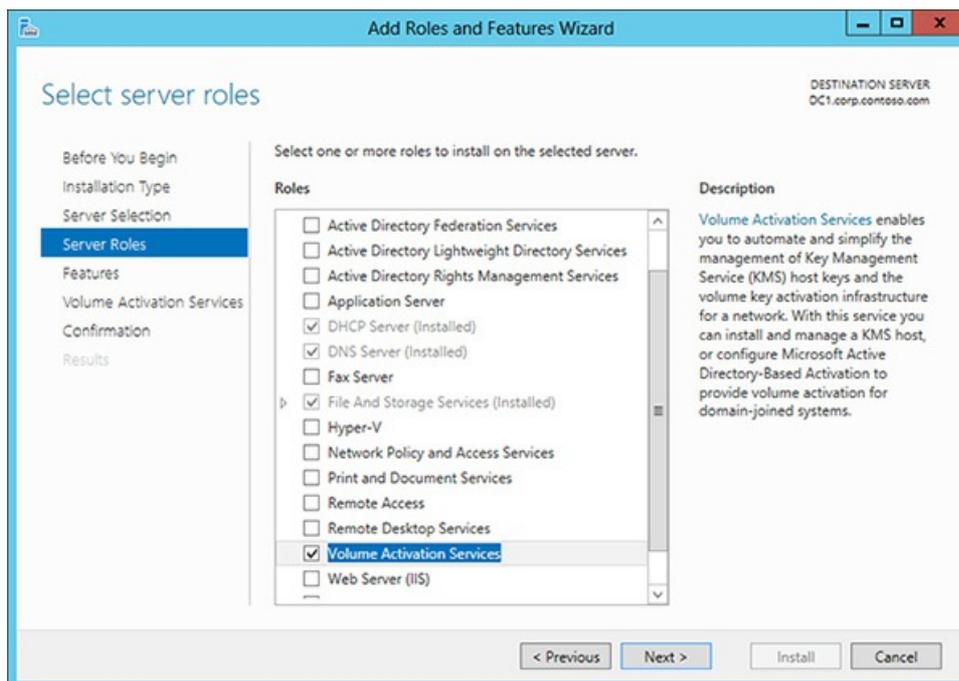


Figure 4. Adding the Volume Activation Services role in Server Manager\

4. When the role installation is complete, click the link to launch the Volume Activation Tools (Figure 5).

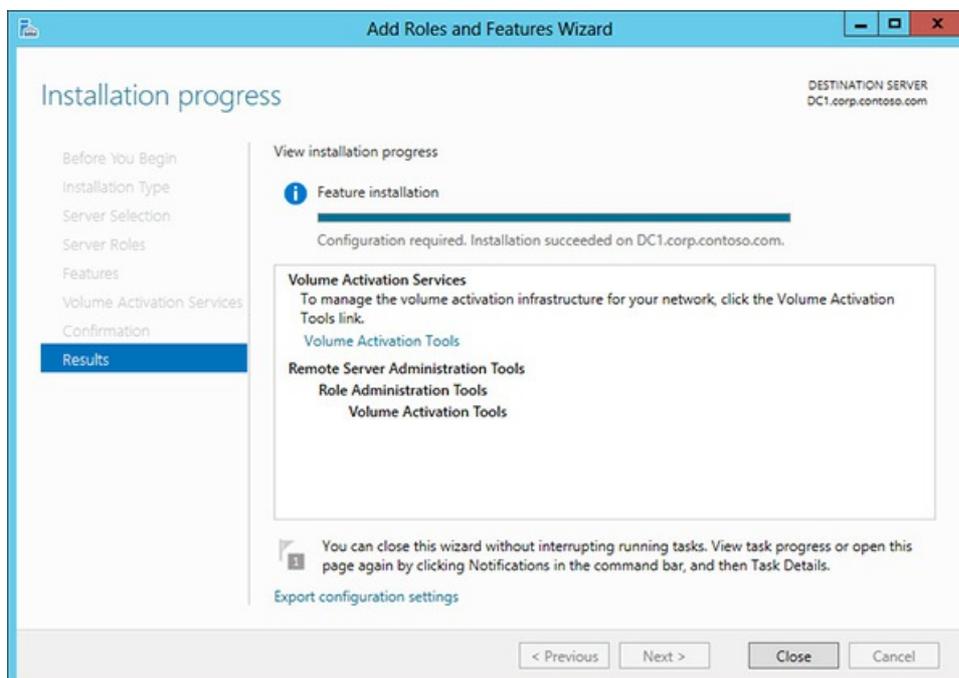


Figure 5. Launching the Volume Activation Tools

- e. Select the **Key Management Service (KMS)** option, and specify the computer that will act as the KMS host (Figure 6). This can be the same computer on which you installed the role or another computer. For example, it can be a client computer running Windows 10.

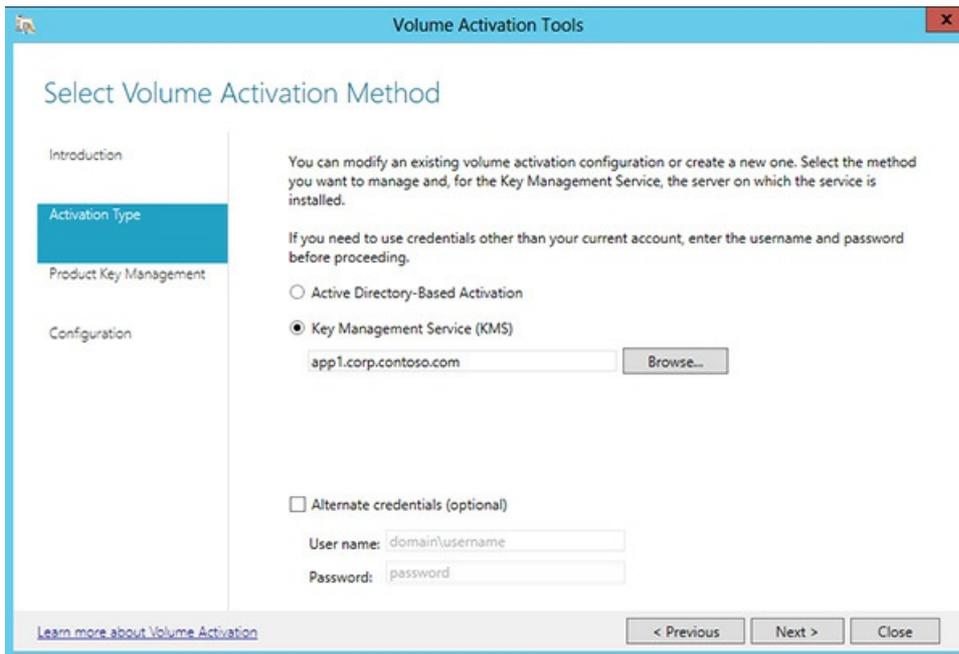


Figure 6. Configuring the computer as a KMS host

- 5. Install your KMS host key by typing it in the text box, and then click **Commit** (Figure 7).

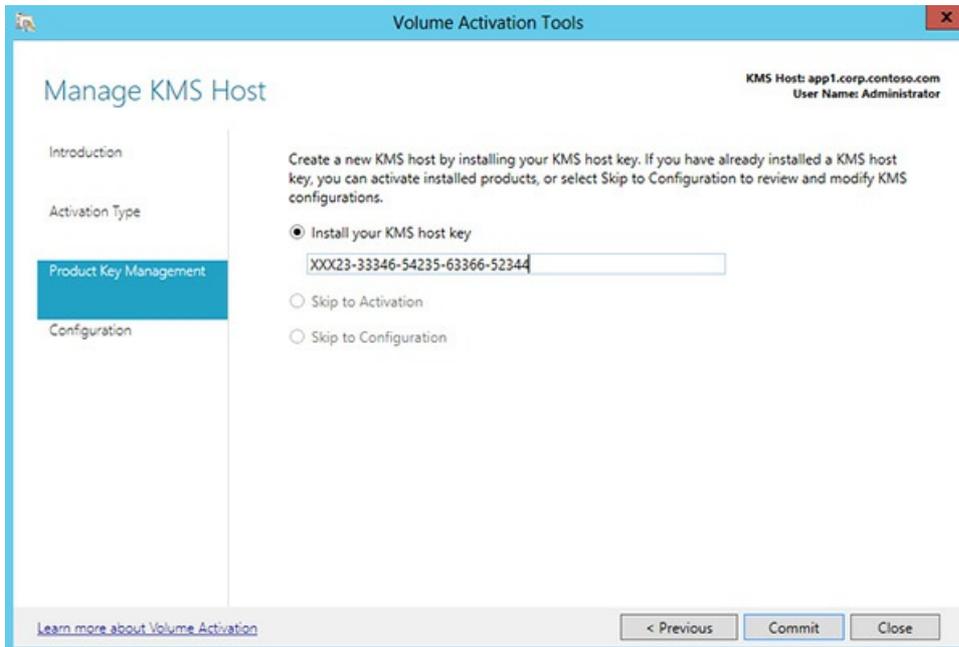


Figure 7. Installing your KMS host key

- 6. If asked to confirm replacement of an existing key, click **Yes**.
- 7. After the product key is installed, you must activate it. Click **Next** (Figure 8).

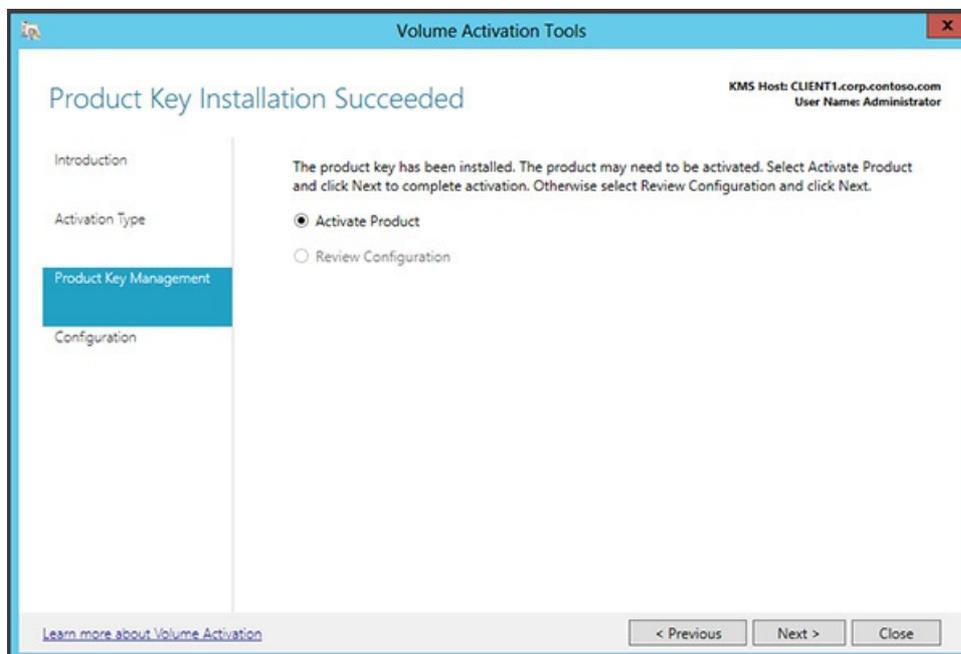


Figure 8. Activating the software

The KMS key can be activated online or by phone. See Figure 9.

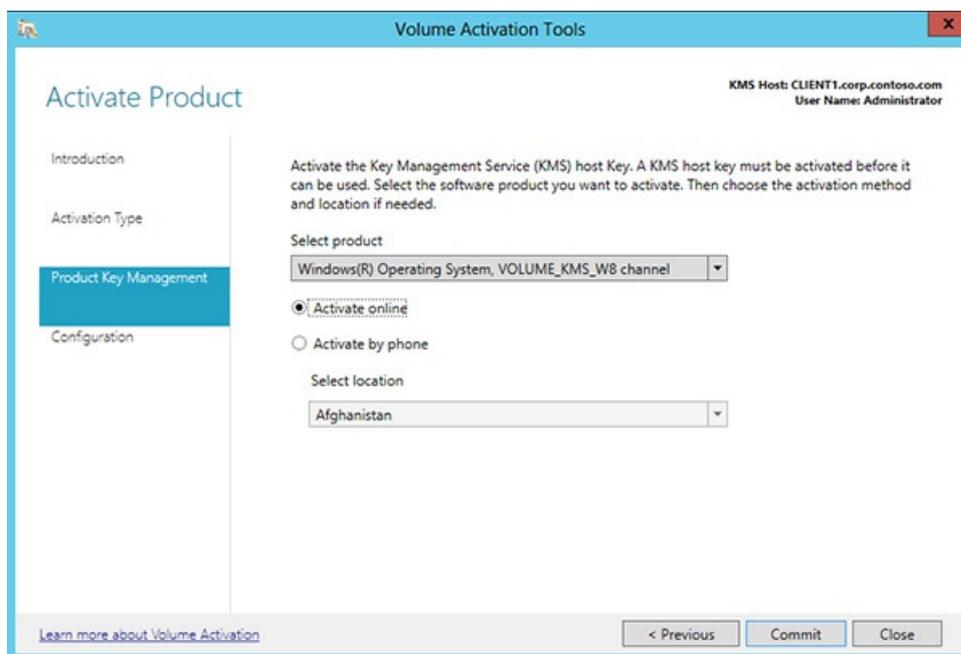


Figure 9. Choosing to activate online

Now that the KMS host is configured, it will begin to listen for activation requests. However, it will not activate clients successfully until the activation threshold is met.

Verifying the configuration of Key Management Service

You can verify KMS volume activation from the KMS host server or from the client computer. KMS volume activation requires a minimum threshold of 25 computers before activation requests will be processed. The verification process described here will increment the activation count each time a client computer contacts the KMS host, but unless the activation threshold is reached, the verification will take the form of an error message rather than a confirmation message. **Note**

If you configured Active Directory-based activation before configuring KMS activation, you must use a client computer that will not first try to activate itself by using Active Directory-based activation. You could use a workgroup computer that is not joined to a domain or a computer running Windows 7 or Windows

Server 2008 R2.

To verify that KMS volume activation works, complete the following steps:

1. On the KMS host, open the event log and confirm that DNS publishing is successful.
2. On a client computer, open a Command Prompt window, type **slmgr.vbs /ato**, and then press ENTER.
The **/ato** command causes the operating system to attempt activation by using whichever key has been installed in the operating system. The response should show the license state and detailed Windows version information.
3. On a client computer or the KMS host, open an elevated Command Prompt window, type **slmgr /dlv**, and then press ENTER.

The **/dlv** command displays the detailed licensing information. The response should return an error that states that the KMS activation count is too low. This confirms that KMS is functioning correctly, even though the client has not been activated.

For more information about the use and syntax of `slmgr.vbs`, see [Slmgr.vbs Options](#).

Key Management Service in earlier versions of Windows

If you have already established a KMS infrastructure in your organization for an earlier version of Windows, you may want to continue using that infrastructure to activate computers running Windows 10 or Windows Server 2012 R2. Your existing KMS host must be running Windows 7 or later. To upgrade your KMS host, complete the following steps:

1. Download and install the correct update for your current KMS host operating system. Restart the computer as directed.
2. Request a new KMS host key from the Volume Licensing Service Center.
3. Install the new KMS host key on your KMS host.
4. Activate the new KMS host key by running the `slmgr.vbs` script.

For detailed instructions, see [Update that enables Windows 8.1 and Windows 8 KMS hosts to activate a later version of Windows](#) and [Update that enables Windows 7 and Windows Server 2008 R2 KMS hosts to activate Windows 10](#).

See also

- [Volume Activation for Windows 10](#)

Activate using Active Directory-based activation

5/31/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2016

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

Active Directory-based activation is implemented as a role service that relies on AD DS to store activation objects. Active Directory-based activation requires that the forest schema be updated by adprep.exe on a computer running Windows Server 2012 or Windows Server 2012 R2, but after the schema is updated, older domain controllers can still activate clients. Any domain-joined computers running Windows 10, Windows 8.1, Windows 8, Windows Server 2012, or Windows Server 2012 R2 with a GVLK will be activated automatically and transparently. They will stay activated as long as they remain members of the domain and maintain periodic contact with a domain controller. Activation takes place after the Licensing service starts. When this service starts, the computer contacts AD DS automatically, receives the activation object, and is activated without user intervention. To allow computers with GVLKs to activate themselves, use the Volume Activation Tools console in Windows Server 2012 R2 or the VAMT in earlier versions of Windows Server to create an object in the AD DS forest. You create this activation object by submitting a KMS host key to Microsoft, as shown in Figure 10. The process proceeds as follows:

1. Perform one of the following tasks:
 - Install the Volume Activation Services server role on a domain controller running Windows Server 2012 R2, and add a KMS host key by using the Volume Activation Tools Wizard.
 - Extend the domain to the Windows Server 2012 R2 schema level, and add a KMS host key by using the VAMT.
2. Microsoft verifies the KMS host key, and an activation object is created.
3. Client computers are activated by receiving the activation object from a domain controller during startup.

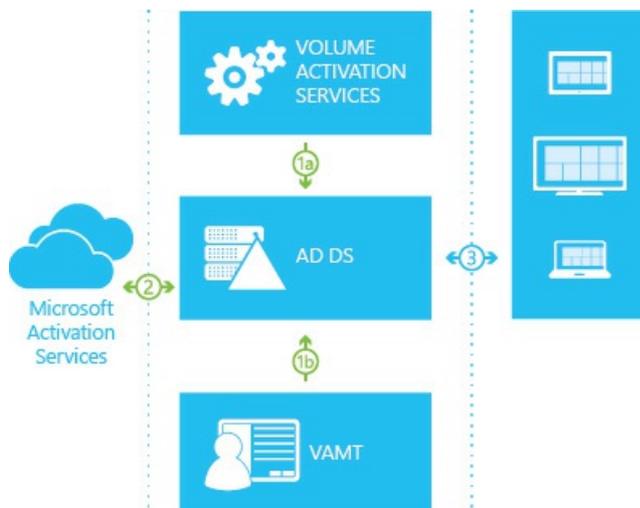


Figure 10. The Active Directory-based activation flow

For environments in which all computers are running Windows 10, Windows 8.1, Windows 8, Windows Server 2012, or Windows Server 2012 R2, and they are joined to a domain, Active Directory-based activation is the best option for activating all client computers and servers, and you may be able to remove any KMS hosts from your environment. If an environment will continue to contain earlier volume licensing operating systems and applications or if you have workgroup computers outside the domain, you need to maintain a KMS host to maintain activation status for earlier volume licensing editions of Windows and Office. Clients that are activated with Active Directory-based activation will maintain their activated state for up to 180 days since the last contact with the domain, but they will periodically attempt to reactivate before then and at the end of the 180day period. By default, this reactivation event occurs every seven days. When a reactivation event occurs, the client queries AD DS for the activation object. Client computers examine the activation object and compare it to the local edition as defined by the GVLK. If the object and GVLK match, reactivation occurs. If the AD DS object cannot be retrieved, client computers use KMS activation. If the computer is removed from the domain, when the computer or the Software Protection service is restarted, the operating system will change the status from activated to not activated, and the computer will try to activate with KMS.

Step-by-step configuration: Active Directory-based activation

Note You must be a member of the local Administrators group on all computers mentioned in these steps. You also need to be a member of the Enterprise Administrators group, because setting up Active Directory-based activation changes forest-wide settings. **To configure Active Directory-based activation on Windows Server 2012 R2, complete the following steps:**

1. Use an account with Domain Administrator and Enterprise Administrator credentials to sign in to a domain controller.
2. Launch Server Manager.
3. Add the Volume Activation Services role, as shown in Figure 11.

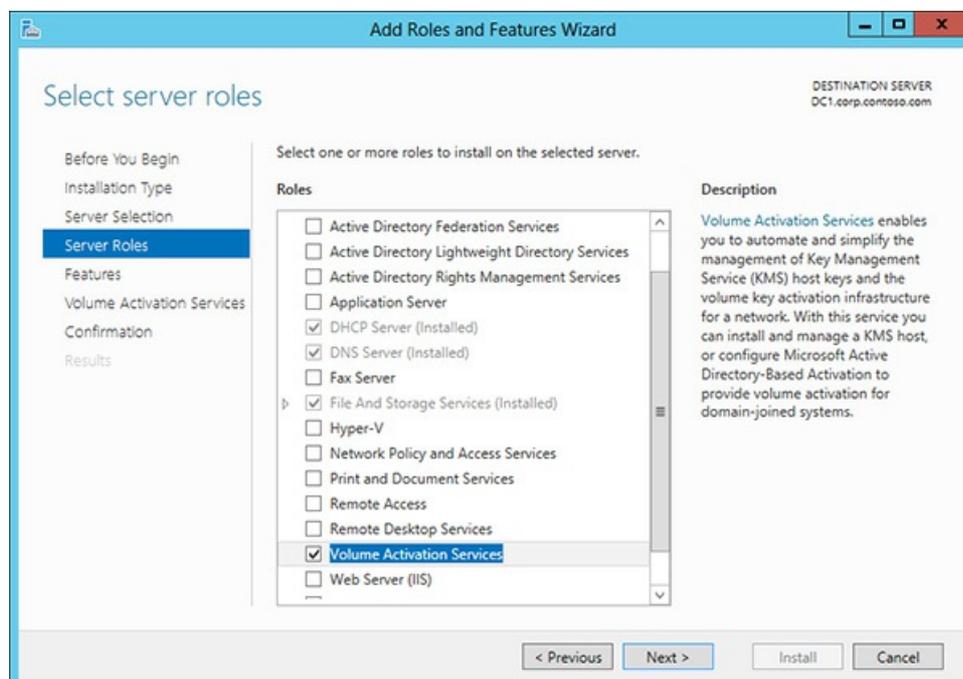


Figure 11. Adding the Volume Activation Services role

4. Click the link to launch the Volume Activation Tools (Figure 12).

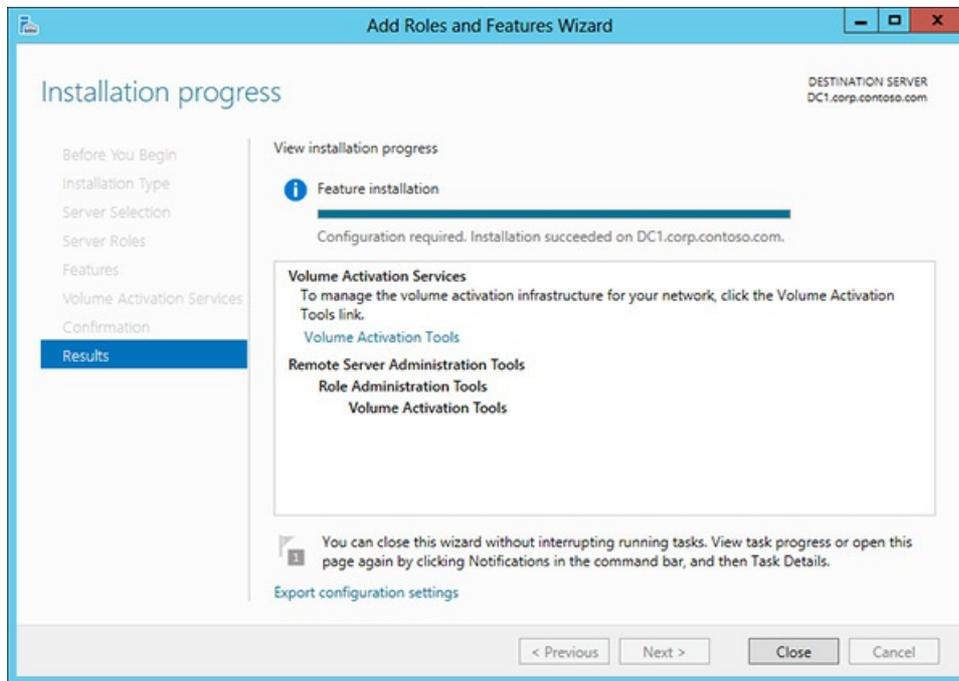


Figure 12. Launching the Volume Activation Tools

5. Select the **Active Directory-Based Activation** option (Figure 13).

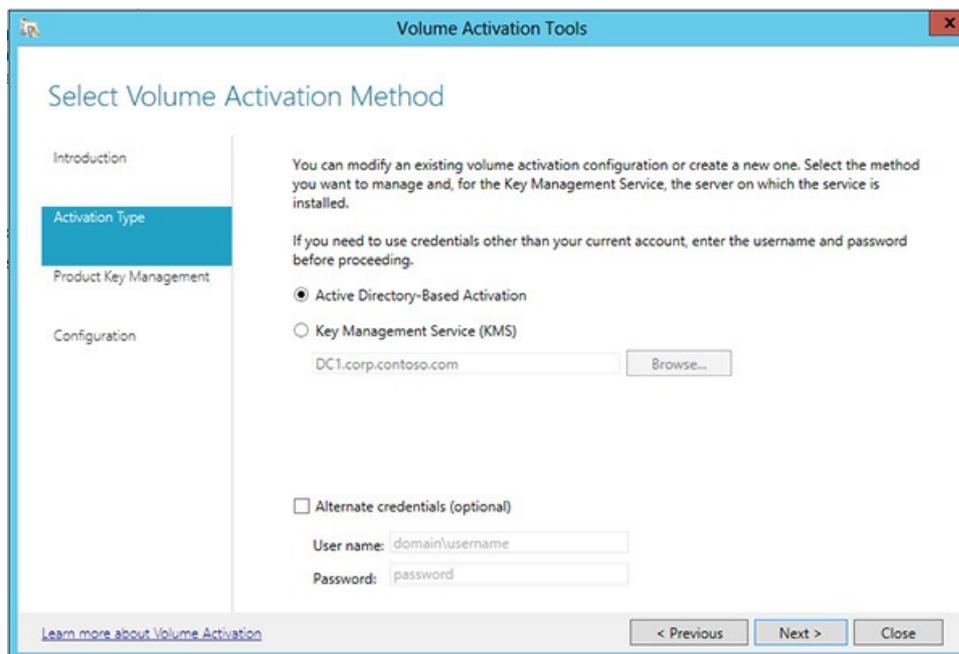


Figure 13. Selecting Active Directory-Based Activation

6. Enter your KMS host key and (optionally) a display name (Figure 14).

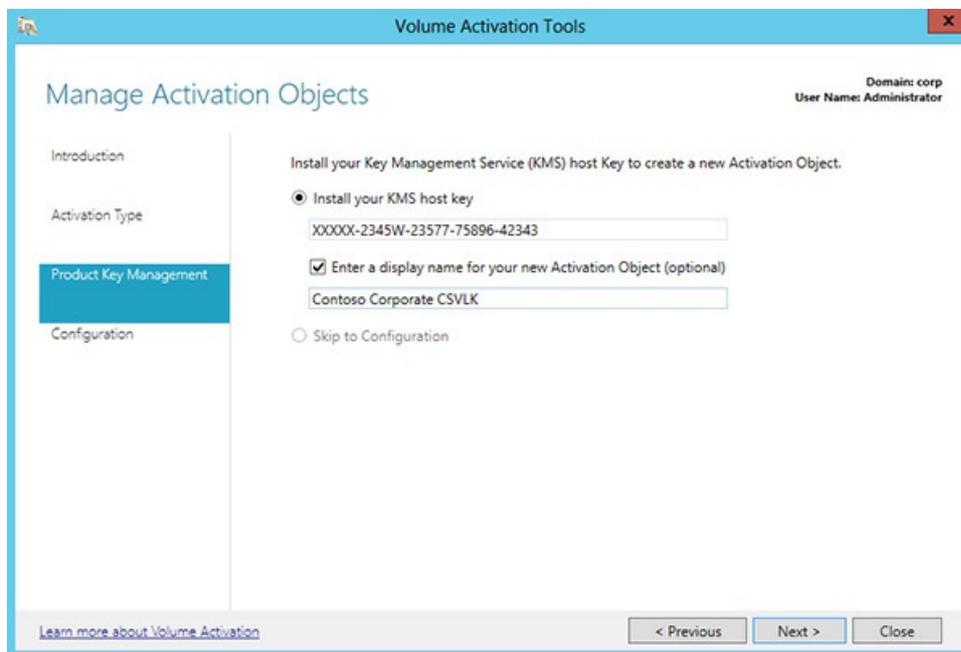


Figure 14. Entering your KMS host key

7. Activate your KMS host key by phone or online (Figure 15).

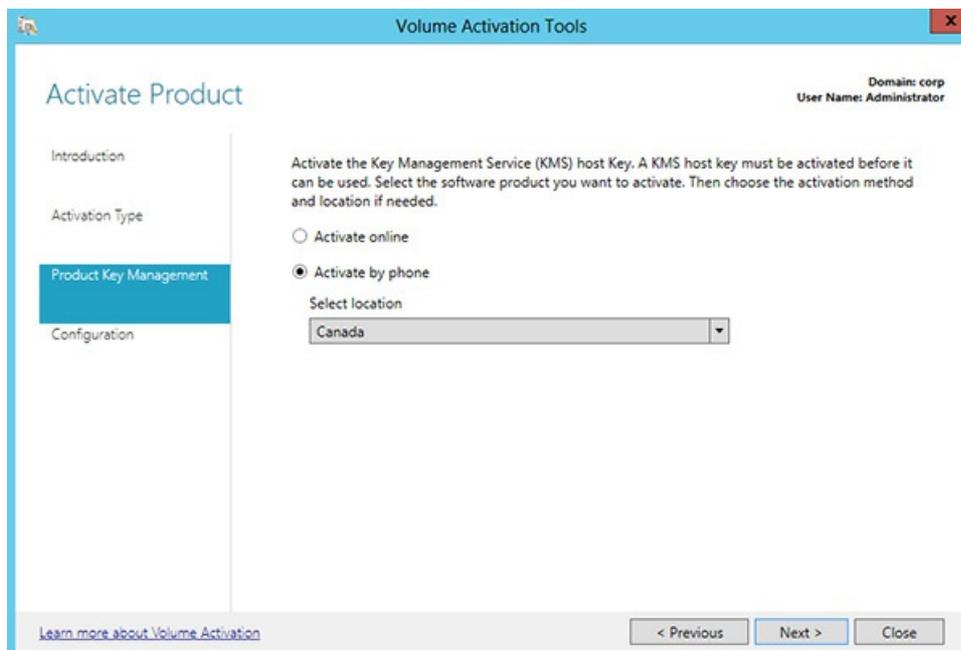


Figure 15. Choosing how to activate your product

8. After activating the key, click **Commit**, and then click **Close**.

Verifying the configuration of Active Directory-based activation

To verify your Active Directory-based activation configuration, complete the following steps:

1. After you configure Active Directory-based activation, start a computer that is running an edition of Windows that is configured by volume licensing.
2. If the computer has been previously configured with a MAK key, replace the MAK key with the GVLK by running the **slmgr.vbs /ipk** command and specifying the GLVK as the new product key.
3. If the computer is not joined to your domain, join it to the domain.
4. Sign in to the computer.

5. Open Windows Explorer, right-click **Computer**, and then click **Properties**.
6. Scroll down to the **Windows activation** section, and verify that this client has been activated.

Note

If you are using both KMS and Active Directory-based activation, it may be difficult to see whether a client has been activated by KMS or by Active Directory-based activation. Consider disabling KMS during the test, or make sure that you are using a client computer that has not already been activated by KMS. The **slmgr.vbs /dlv** command also indicates whether KMS has been used.

See also

- [Volume Activation for Windows 10](#)

Activate clients running Windows 10

5/31/2019 • 11 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

After you have configured Key Management Service (KMS) or Active Directory-based activation on your network, activating a client running Windows 10 is easy. If the computer has been configured with a Generic Volume License Key (GVLK), neither IT nor the user need take any action. It just works. Enterprise edition images and installation media should already be configured with the GVLK. When the client computer starts, the Licensing service examines the current licensing condition of the computer. If activation or reactivation is required, the following sequence occurs:

1. If the computer is a member of a domain, it asks a domain controller for a volume activation object. If Active Directory-based activation is configured, the domain controller returns the object. If the object matches the edition of the software that is installed and the computer has a matching GVLK, the computer is activated (or reactivated), and it will not need to be activated again for 180 days, although the operating system will attempt reactivation at much shorter, regular intervals.
2. If the computer is not a member of a domain or if the volume activation object is not available, the computer will issue a DNS query to attempt to locate a KMS server. If a KMS server can be contacted, activation occurs if the KMS has a key that matches the computer's GVLK.
3. The computer tries to activate against Microsoft servers if it is configured with a MAK.

If the client is not able to activate itself successfully, it will periodically try again. The frequency of the retry attempts depends on the current licensing state and whether the client computer has been successfully activated in the past. For example, if the client computer had been previously activated by Active Directory-based activation, it will periodically try to contact the domain controller at each restart.

How Key Management Service works

KMS uses a client-server topology. KMS client computers can locate KMS host computers by using DNS or a static configuration. KMS clients contact the KMS host by using RPCs carried over TCP/IP.

Key Management Service activation thresholds

You can activate physical computers and virtual machines by contacting a KMS host. To qualify for KMS activation, there must be a minimum number of qualifying computers (called the activation threshold). KMS clients will be activated only after this threshold has been met. Each KMS host counts the number of computers that have requested activation until the threshold is met.

A KMS host responds to each valid activation request from a KMS client with the count of how many computers

have already contacted the KMS host for activation. Client computers that receive a count below the activation threshold are not activated. For example, if the first two computers that contact the KMS host are running Windows 10, the first receives an activation count of 1, and the second receives an activation count of 2. If the next computer is a virtual machine on a computer running Windows 10, it receives an activation count of 3, and so on. None of these computers will be activated, because computers running Windows 10, like other client operating system versions, must receive an activation count of 25 or more. When KMS clients are waiting for the KMS to reach the activation threshold, they will connect to the KMS host every two hours to get the current activation count. They will be activated when the threshold is met.

In our example, if the next computer that contacts the KMS host is running Windows Server 2012 R2, it receives an activation count of 4, because activation counts are cumulative. If a computer running Windows Server 2012 R2 receives an activation count that is 5 or more, it is activated. If a computer running Windows 10 receives an activation count of 25 or more, it is activated.

Activation count cache

To track the activation threshold, the KMS host keeps a record of the KMS clients that request activation. The KMS host gives each KMS client a client ID designation, and the KMS host saves each client ID in a table. By default, each activation request remains in the table for up to 30 days. When a client renews its activation, the cached client ID is removed from the table, a new record is created, and the 30day period begins again. If a KMS client computer does not renew its activation within 30 days, the KMS host removes the corresponding client ID from the table and reduces the activation count by one. However, the KMS host only caches twice the number of client IDs that are required to meet the activation threshold. Therefore, only the 50 most recent client IDs are kept in the table, and a client ID could be removed much sooner than 30 days. The total size of the cache is set by the type of client computer that is attempting to activate. If a KMS host receives activation requests only from servers, the cache will hold only 10 client IDs (twice the required 5). If a client computer running Windows 10 contacts that KMS host, KMS increases the cache size to 50 to accommodate the higher threshold. KMS never reduces the cache size.

Key Management Service connectivity

KMS activation requires TCP/IP connectivity. By default, KMS hosts and clients use DNS to publish and find the KMS. The default settings can be used, which require little or no administrative action, or KMS hosts and client computers can be manually configured based on network configuration and security requirements.

Key Management Service activation renewal

KMS activations are valid for 180 days (the *activation validity interval*). To remain activated, KMS client computers must renew their activation by connecting to the KMS host at least once every 180 days. By default, KMS client computers attempt to renew their activation every 7 days. If KMS activation fails, the client computer retries every two hours. After a client computer's activation is renewed, the activation validity interval begins again.

Publication of the Key Management Service

The KMS uses service (SRV) resource records in DNS to store and communicate the locations of KMS hosts. KMS hosts use the DNS dynamic update protocol, if available, to publish the KMS service (SRV) resource records. If dynamic update is not available or the KMS host does not have rights to publish the resource records, the DNS records must be published manually, or you must configure client computers to connect to specific KMS hosts.

Client discovery of the Key Management Service

By default, KMS client computers query DNS for KMS information. The first time a KMS client computer queries DNS for KMS information, it randomly chooses a KMS host from the list of service (SRV) resource records that DNS returns. The address of a DNS server that contains the service (SRV) resource records can be listed as a suffixed entry on KMS client computers, which allows one DNS server to advertise the service (SRV) resource records for KMS, and KMS client computers with other primary DNS servers to find it. Priority and weight parameters can be added to the DnsDomainPublishList registry value for KMS. Establishing KMS host priority groupings and weighting within each group allows you to specify which KMS host the client computers should try first and balances traffic among multiple KMS hosts. Only Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2 provide these priority and weight

parameters. If the KMS host that a client computer selects does not respond, the KMS client computer removes that KMS host from its list of service (SRV) resource records and randomly selects another KMS host from the list. When a KMS host responds, the KMS client computer caches the name of the KMS host and uses it for subsequent activation and renewal attempts. If the cached KMS host does not respond on a subsequent renewal, the KMS client computer discovers a new KMS host by querying DNS for KMS service (SRV) resource records. By default, client computers connect to the KMS host for activation by using anonymous RPCs through TCP port 1688. (You can change the default port.) After establishing a TCP session with the KMS host, the client computer sends a single request packet. The KMS host responds with the activation count. If the count meets or exceeds the activation threshold for that operating system, the client computer is activated and the session is closed. The KMS client computer uses this same process for renewal requests. 250 bytes are used for communication each way.

Domain Name System server configuration

The default KMS automatic publishing feature requires the service (SRV) resource record and support for DNS dynamic update protocol. KMS client computer default behavior and the KMS service (SRV) resource record publishing are supported on a DNS server that is running Microsoft software or any other DNS server that supports service (SRV) resource records (per Internet Engineering Task Force [IETF] Request for Comments [RFC] 2782) and dynamic updates (per IETF RFC 2136). For example, Berkeley Internet Domain Name versions 8.x and 9.x support service (SRV) resource records and dynamic update. The KMS host must be configured so that it has the credentials needed to create and update the following resource records on the DNS servers: service (SRV), IPv4 host (A), and IPv6 host (AAAA), or the records need to be created manually. The recommended solution for giving the KMS host the needed credentials is to create a security group in AD DS, then add all KMS hosts to that group. On a DNS server that is running Microsoft software, ensure that this security group is given full control over the _VLMCS._TCP record in each DNS domain that will contain the KMS service (SRV) resource records.

Activating the first Key Management Service host

KMS hosts on the network need to install a KMS key, and then be activated with Microsoft. Installation of a KMS key enables the KMS on the KMS host. After installing the KMS key, complete the activation of the KMS host by telephone or online. Beyond this initial activation, a KMS host does not communicate any information to Microsoft. KMS keys are only installed on KMS hosts, never on individual KMS client computers.

Activating subsequent Key Management Service hosts

Each KMS key can be installed on up to six KMS hosts. These hosts can be physical computers or virtual machines. After activating a KMS host, the same host can be reactivated up to nine times with the same key. If the organization needs more than six KMS hosts, you can request additional activations for your organization's KMS key by calling a Microsoft Volume [Licensing Activation Center](#) to request an exception.

How Multiple Activation Key works

A MAK is used for one-time activation with Microsoft's hosted activation services. Each MAK has a predetermined number of allowed activations. This number is based on volume licensing agreements, and it might not match the organization's exact license count. Each activation that uses a MAK with the Microsoft hosted activation service counts toward the activation limit.

You can activate computers by using a MAK in two ways:

- **MAK independent activation.** Each computer independently connects and is activated with Microsoft over the Internet or by telephone. MAK independent activation is best suited to computers within an organization that do not maintain a connection to the corporate network. MAK independent activation is shown in Figure 16.

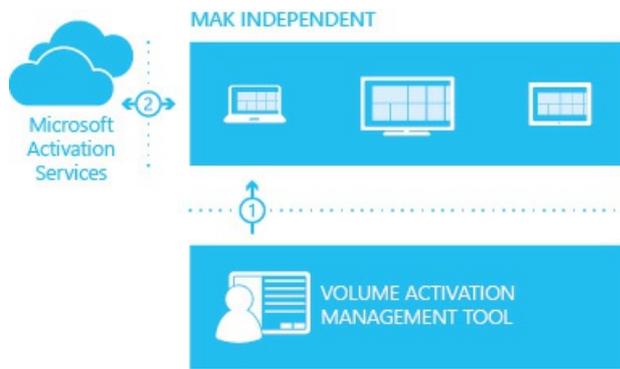


Figure 16. MAK independent activation

- MAK proxy activation.** MAK proxy activation enables a centralized activation request on behalf of multiple computers with one connection to Microsoft. You configure MAK proxy activation by using the VAMT. MAK proxy activation is appropriate for environments in which security concerns restrict direct access to the Internet or the corporate network. It is also suited for development and test labs that lack this connectivity. MAK proxy activation with the VAMT is shown in Figure 17.



Figure 17. MAK proxy activation with the VAMT

A MAK is recommended for computers that rarely or never connect to the corporate network and for environments in which the number of computers that require activation does not meet the KMS activation threshold.

You can use a MAK for individual computers or with an image that can be duplicated or installed by using Microsoft deployment solutions. You can also use a MAK on a computer that was originally configured to use KMS activation. This is useful for moving a computer off the core network to a disconnected environment.

Multiple Activation Key architecture and activation

MAK independent activation installs a MAK product key on a client computer. The key instructs that computer to activate itself with Microsoft servers over the Internet. In MAK proxy activation, the VAMT installs a MAK product key on a client computer, obtains the installation ID from the target computer, sends the installation ID to Microsoft on behalf of the client, and obtains a confirmation ID. The tool then activates the client computer by installing the confirmation ID.

Activating as a standard user

Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2 do not require administrator privileges for activation, but this change does not allow standard user accounts to remove computers running Windows 7 or Windows Server 2008 R2 from the activated state. An administrator account is still required for other activation- or license-related tasks, such as “rearm.”

See also

- [Volume Activation for Windows 10](#)

Monitor activation

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

You can monitor the success of the activation process for a computer running Windows 8.1 in several ways. The most popular methods include:

- Using the Volume Licensing Service Center website to track use of MAK keys.
- Using the **Slmgr /dlv** command on a client computer or on the KMS host. (For a full list of options, see [Slmgr.vbs Options](#).)
- Viewing the licensing status, which is exposed through Windows Management Instrumentation (WMI); therefore, it is available to non-Microsoft or custom tools that can access WMI. (Windows PowerShell can also access WMI information.)
- Most licensing actions and events are recorded in the Event log.
- Microsoft System Center Operations Manager and the KMS Management Pack can provide insight and information to users of System Center Operations Manager.
- The VAMT provides a single site from which to manage and monitor volume activations. This is explained in the next section.

See also

- [Volume Activation for Windows 10](#)

Use the Volume Activation Management Tool

5/31/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

The Volume Activation Management Tool (VAMT) provides several useful features, including the ability to perform VAMT proxy activation and to track and monitor several types of product keys.

By using the VAMT, you can automate and centrally manage the volume, retail, and MAK activation process for Windows, Office, and select other Microsoft products. The VAMT can manage volume activation by using MAKs or KMS. It is a standard Microsoft Management Console snap-in, and it can be installed on any computer running Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2.

The VAMT is distributed as part of the Windows Assessment and Deployment Kit (Windows ADK), which is a free download available from Microsoft Download Center. For more information, see [Windows Assessment and Deployment Kit \(Windows ADK\) for Windows 10](#).

In Windows Server 2012 R2, you can install the VAMT directly from Server Manager without downloading the Windows ADK by selecting the Volume Activation Services role or the Remote Server Administration Tools/Role Administration Tools/Volume Activation Tools feature.

Activating with the Volume Activation Management Tool

You can use the VAMT to complete the activation process in products by using MAK and retail keys, and you can work with computers individually or in groups. The VAMT enables two activation scenarios:

- **Online activation.** Online activation enables you to activate over the Internet any products that are installed with MAK, KMS host, or retail product keys. You can activate one or more connected computers within a network. This process requires that each product communicate activation information directly to Microsoft.
- **Proxy activation.** This activation method enables you to perform volume activation for products that are installed on client computers that do not have Internet access. The VAMT host computer distributes a MAK, KMS host key, or retail product key to one or more client products and collects the installation ID from each client product. The VAMT host sends the installation IDs to Microsoft on behalf of the client products and obtains the corresponding confirmation IDs. The VAMT host then installs the confirmation IDs on the client products to complete their activation. By using this method, only the VAMT host computer requires Internet access. Proxy activation by using the VAMT is beneficial for isolated network segments and for cases where your organization has a mix of retail, MAK, and KMS-based activations.

Tracking products and computers with the Volume Activation Management Tool

The VAMT provides an overview of the activation and licensing status of computers across your network, as shown in Figure 18. Several prebuilt reports are also available to help you proactively manage licensing.

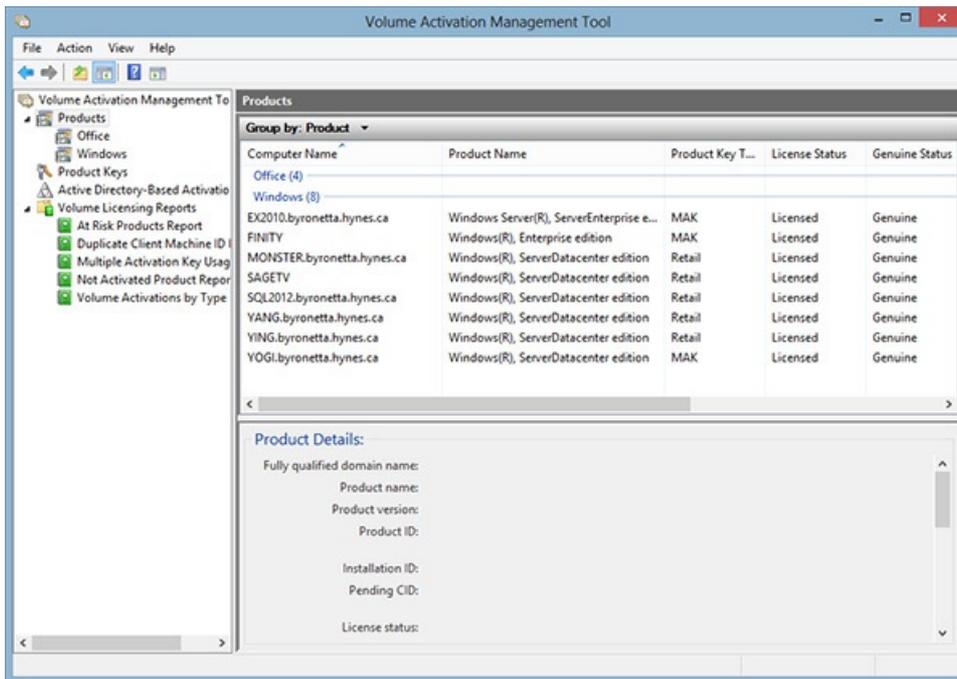


Figure 18. The VAMT showing the licensing status of multiple computers

Tracking key usage with the Volume Activation Management Tool

The VAMT makes it easier to track the various keys that are issued to your organization. You can enter each key into VAMT, and then the VAMT can use those keys for online or proxy activation of clients. The tool can also describe what type of key it is and to which product group it belongs. The VAMT is the most convenient way to quickly determine how many activations remain on a MAK. Figure 19 shows an example of key types and usage.

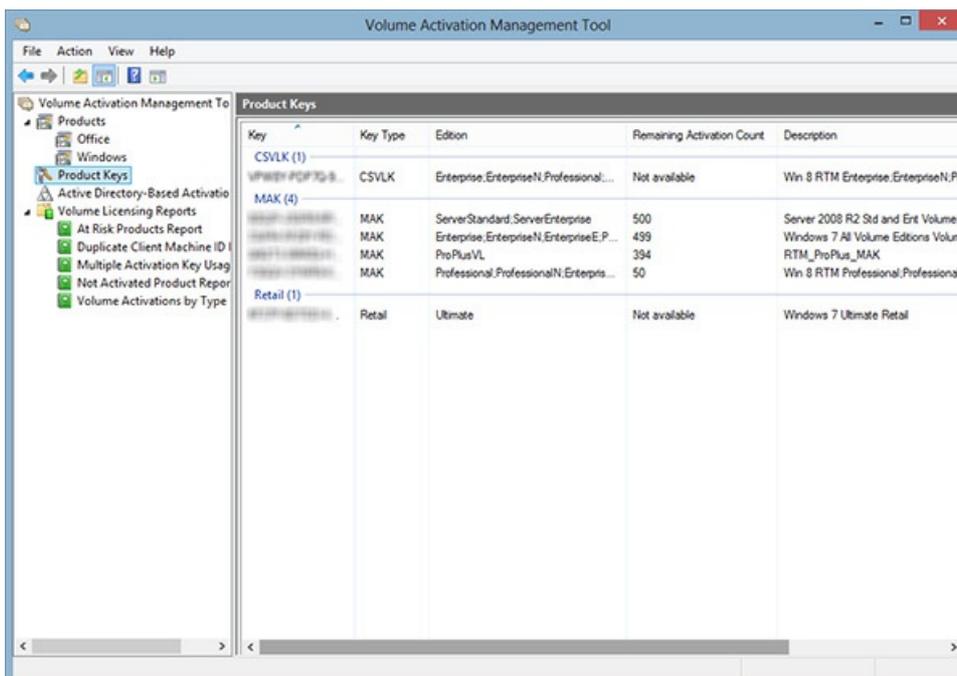


Figure 19. The VAMT showing key types and usage

Other Volume Activation Management Tool features

The VAMT stores information in a Microsoft SQL Server database for performance and flexibility, and it provides a single graphical user interface for managing activations and performing other activation-related tasks, such as:

- **Adding and removing computers.** You can use the VAMT to discover computers in the local environment. The VAMT can discover computers by querying AD DS, workgroups, or individual computer names or IP addresses, or through a general LDAP query.
- **Discovering products.** You can use the VAMT to discover Windows, Windows Server, Office, and select other products that are installed on the client computers.
- **Managing activation data.** The VAMT stores activation data in a SQL Server database. The tool can export this data in XML format to other VAMT hosts or to an archive.

For more information, see:

- [Volume Activation Management Tool \(VAMT\) Overview](#)
- [VAMT Step-by-Step Scenarios](#)

See also

- [Volume Activation for Windows 10](#)

Appendix: Information sent to Microsoft during activation

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Looking for retail activation?

- [Get Help Activating Microsoft Windows](#)

When you activate a computer running Windows 10, the following information is sent to Microsoft:

- The Microsoft product code (a five-digit code that identifies the Windows product you are activating)
- A channel ID or site code that identifies how the Windows product was originally obtained

For example, a channel ID or site code identifies whether the product was originally purchased from a retail store, obtained as an evaluation copy, obtained through a volume licensing program, or preinstalled by a computer manufacturer.

- The date of installation and whether the installation was successful
- Information that helps confirm that your Windows product key has not been altered
- Computer make and model
- Version information for the operating system and software
- Region and language settings
- A unique number called a *globally unique identifier*, which is assigned to your computer
- Product key (hashed) and product ID
- BIOS name, revision number, and revision date
- Volume serial number (hashed) of the hard disk drive
- The result of the activation check

This includes error codes and the following information about any activation exploits and related malicious or unauthorized software that was found or disabled:

- The activation exploit's identifier
- The activation exploit's current state, such as cleaned or quarantined
- Computer manufacturer's identification
- The activation exploit's file name and hash in addition to a hash of related software components that may

indicate the presence of an activation exploit

- The name and a hash of the contents of your computer's startup instructions file
- If your Windows license is on a subscription basis, information about how your subscription works

Standard computer information is also sent, but your computer's IP address is only retained temporarily.

Use of information

Microsoft uses the information to confirm that you have a licensed copy of the software. Microsoft does not use the information to contact individual consumers. For additional details, see [Windows 10 Privacy Statement](#).

See also

- [Volume Activation for Windows 10](#)

Application Compatibility Toolkit (ACT) Technical Reference

6/18/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10, version 1607

IMPORTANT

We've replaced the majority of functionality included in the Application Compatibility Toolkit (ACT) with [Windows Analytics](#), a solution in the Microsoft Operations Management Suite. Windows Analytics gives enterprises the tools to plan and manage the upgrade process end to end, allowing them to adopt new Windows releases more quickly. With new Windows versions being released multiple times a year, ensuring application and driver compatibility on an ongoing basis is key to adopting new Windows versions as they are released.

Microsoft developed Windows Analytics in response to demand from enterprise customers looking for additional direction and details about upgrading to Windows 10. Windows Analytics was built taking into account multiple channels of customer feedback, testing, and Microsoft's experience upgrading millions of devices to Windows 10.

With Windows diagnostic data enabled, Windows Analytics collects system, application, and driver data for analysis. We then identify compatibility issues that can block an upgrade and suggest fixes when they are known to Microsoft.

Use Windows Analytics to get:

- A visual workflow that guides you from pilot to production
- Detailed computer and application inventory
- Powerful computer level search and drill-downs
- Guidance and insights into application and driver compatibility issues, with suggested fixes
- Data driven application rationalization tools
- Application usage information, allowing targeted validation; workflow to track validation progress and decisions
- Data export to commonly used software deployment tools, including System Center Configuration Manager

The Windows Analytics workflow steps you through the discovery and rationalization process until you have a list of computers that are ready to be upgraded.

At the same time, we've kept the Standard User Analyzer tool, which helps you test your apps and to monitor API calls for potential compatibility issues, and the Compatibility Administrator, which helps you to resolve potential compatibility issues.

In this section

TOPIC	DESCRIPTION
Standard User Analyzer (SUA) User's Guide	The Standard User Analyzer (SUA) helps you test your applications and monitor API calls to detect compatibility issues related to the User Account Control (UAC) feature in Windows.

TOPIC	DESCRIPTION
Compatibility Administrator User's Guide	The Compatibility Administrator tool helps you resolve potential application-compatibility issues before deploying a new version of Windows to your organization.
Compatibility Fixes for Windows 10, Windows 8, Windows 7, and Windows Vista	You can fix some compatibility issues that are due to the changes made between Windows operating system versions. These issues can include User Account Control (UAC) restrictions.

SUA User's Guide

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

You can use Standard User Analyzer (SUA) to test your applications and monitor API calls to detect compatibility issues related to the User Account Control (UAC) feature in Windows.

You can use SUA in either of the following ways:

- **Standard User Analyzer Wizard.** A wizard that guides you through a step-by-step process to locate and fix issues, without options for additional analysis.
- **Standard User Analyzer Tool.** A full-function tool in which you can perform in-depth analysis and fix issues.

In this section

TOPIC	DESCRIPTION
Using the SUA Wizard	The Standard User Analyzer (SUA) Wizard works much like the SUA tool to evaluate User Account Control (UAC) issues. However, the SUA Wizard does not offer detailed analysis, and it cannot disable virtualization or elevate your permissions.
Using the SUA Tool	By using the Standard User Analyzer (SUA) tool, you can test your applications and monitor API calls to detect compatibility issues with the User Account Control (UAC) feature.

Using the SUA Wizard

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Standard User Analyzer (SUA) Wizard works much like the SUA tool to evaluate User Account Control (UAC) issues. However, the SUA Wizard does not offer detailed analysis, and it cannot disable virtualization or elevate your permissions.

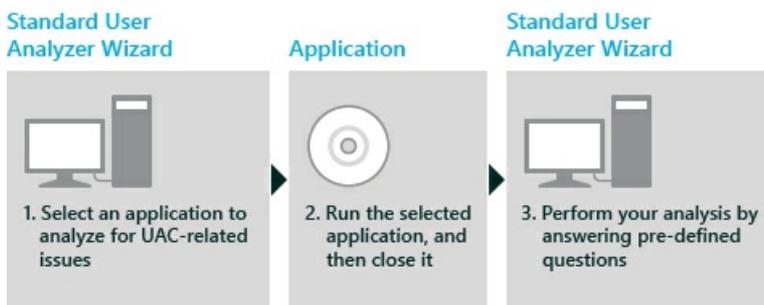
For information about the SUA tool, see [Using the SUA Tool](#).

Testing an Application by Using the SUA Wizard

You must install Application Verifier before you can use the SUA Wizard. If Application Verifier is not installed on the computer that is running the SUA Wizard, the SUA Wizard notifies you. You must also install the Microsoft® .NET Framework 3.5 or later before you can use the SUA Wizard.

The following flowchart shows the process of using the SUA Wizard.

Standard User Analyzer Wizard



To test an application by using the SUA Wizard

1. On the computer where the SUA Wizard is installed, log on by using a non-administrator account.
2. Run the Standard User Analyzer Wizard.
3. Click **Browse for Application**, browse to the folder that contains the application that you want to test, and then double-click the executable file for the application.
4. Click **Launch**.

If you are prompted, elevate your permissions. The SUA Wizard may require elevation of permissions to correctly diagnose the application.

If a **Permission denied** dialog box appears, click **OK**. The application starts, despite the warning.

5. In the application, exercise the functionality that you want to test.

6. After you finish testing, exit the application.

The SUA Wizard displays a message that asks whether the application ran without any issues.

7. Click **No**.

The SUA Wizard shows a list of potential remedies that you might use to fix the application.

8. Select the fixes that you want to apply, and then click **Launch**.

The application appears again, with the fixes applied.

9. Test the application again, and after you finish testing, exit the application.

The SUA Wizard displays a message that asks whether the application ran without any issues.

10. If the application ran correctly, click **Yes**.

The SUA Wizard closes the issue as resolved on the local computer.

If the remedies do not fix the issue with the application, click **No** again, and the wizard may offer additional remedies. If the additional remedies do not fix the issue, the wizard informs you that there are no more remedies available. For information about how to run the SUA tool for additional investigation, see [Using the SUA Tool](#).

Related topics

[SUA User's Guide](#)

Using the SUA Tool

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

By using the Standard User Analyzer (SUA) tool, you can test your applications and monitor API calls to detect compatibility issues with the User Account Control (UAC) feature.

The SUA Wizard also addresses UAC-related issues. In contrast to the SUA tool, the SUA Wizard guides you through the process step by step, without the in-depth analysis of the SUA tool. For information about the SUA Wizard, see [Using the SUA Wizard](#).

In the SUA tool, you can turn virtualization on and off. When you turn virtualization off, the tested application may function more like the way it does in earlier versions of Windows®.

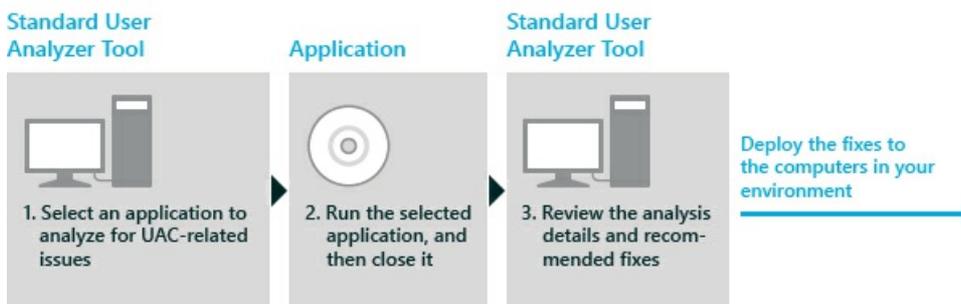
In the SUA tool, you can choose to run the application as **Administrator** or as **Standard User**. Depending on your selection, you may locate different types of UAC-related issues.

Testing an Application by Using the SUA Tool

Before you can use the SUA tool, you must install Application Verifier. You must also install the Microsoft® .NET Framework 3.5 or later.

The following flowchart shows the process of using the SUA tool.

Standard User Analyzer Tool



To collect UAC-related issues by using the SUA tool

1. Close any open instance of the SUA tool or SUA Wizard on your computer.

If there is an existing SUA instance on the computer, the SUA tool opens in log viewer mode instead of normal mode. In log viewer mode, you cannot start applications, which prevents you from collecting UAC issues.

2. Run the Standard User Analyzer.

3. In the **Target Application** box, browse to the executable file for the application that you want to analyze,

and then double-click to select it.

4. Clear the **Elevate** check box, and then click **Launch**.

If a **Permission denied** dialog box appears, click **OK**. The application starts, despite the warning.

5. Exercise the aspects of the application for which you want to gather information about UAC issues.
6. Exit the application.
7. Review the information from the various tabs in the SUA tool. For information about each tab, see [Tabs on the SUA Tool Interface](#).

To review and apply the recommended mitigations

1. In the SUA tool, on the **Mitigation** menu, click **Apply Mitigations**.
2. Review the recommended compatibility fixes.
3. Click **Apply**.

The SUA tool generates a custom compatibility-fix database and automatically applies it to the local computer, so that you can test the fixes to see whether they worked.

Related topics

[Tabs on the SUA Tool Interface](#)

[Showing Messages Generated by the SUA Tool](#)

[Applying Filters to Data in the SUA Tool](#)

[Fixing Applications by Using the SUA Tool](#)

Tabs on the SUA Tool Interface

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The tabs in the Standard User Analyzer (SUA) tool show the User Account Control (UAC) issues for the applications that you analyze.

The following table provides a description of each tab on the user interface for the SUA tool.

TAB NAME	DESCRIPTION
App Info	<p>Provides the following information for the selected application:</p> <ul style="list-style-type: none">• Debugging information• Error, warning, and informational messages (if they are enabled)• Options for running the application
File	<p>Provides information about access to the file system.</p> <p>For example, this tab might show an attempt to write to a file that only administrators can typically access.</p>
Registry	<p>Provides information about access to the system registry.</p> <p>For example, this tab might show an attempt to write to a registry key that only administrators can typically access.</p>
INI	<p>Provides information about WriteProfile API issues.</p> <p>For example, in the Calculator tool (Calc.exe) in Windows® XP, when you change the view from Standard to Scientific, Calc.exe calls the WriteProfile API to write to the Windows\Win.ini file. The Win.ini file is writable only for administrators.</p>
Token	<p>Provides information about access-token checking.</p> <p>For example, this tab might show an explicit check for the Builtin\Administrators security identifier (SID) in the user's access token. This operation may not work for a standard user.</p>

TAB NAME	DESCRIPTION
Privilege	<p>Provides information about permissions.</p> <p>For example, this tab might show an attempt to explicitly enable permissions that do not work for a standard user.</p>
Name Space	<p>Provides information about creation of system objects.</p> <p>For example, this tab might show an attempt to create a new system object, such as an event or a memory map, in a restricted namespace. Applications that attempt this kind of operation do not function for a standard user.</p>
Other Objects	<p>Provides information related to applications accessing objects other than files and registry keys.</p>
Process	<p>Provides information about process elevation.</p> <p>For example, this tab might show the use of the CreateProcess API to open an executable (.exe) file that, in turn, requires process elevation that will not function for a standard user.</p>

Showing Messages Generated by the SUA Tool

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

On the user interface for the Standard User Analyzer (SUA) tool, you can show the messages that the tool has generated.

To show the messages that the SUA tool has generated

1. Use the SUA tool to test an application. For more information, see [Using the SUA Tool](#).
2. After you finish testing, in the SUA tool, click the **App Info** tab.
3. On the **View** menu, click the command that corresponds to the messages that you want to see. The following table describes the commands.

VIEW MENU COMMAND	DESCRIPTION
Error Messages	When this command is selected, the user interface shows error messages that the SUA tool has generated. Error messages are highlighted in pink. This command is selected by default.
Warning Messages	When this command is selected, the user interface shows warning messages that the SUA tool has generated. Warning messages are highlighted in yellow.
Information Messages	When this command is selected, the user interface shows informational messages that the SUA tool has generated. Informational messages are highlighted in green.
Detailed Information	When this command is selected, the user interface shows information that the SUA tool has generated, such as debug, stack trace, stop code, and severity information.

Applying Filters to Data in the SUA Tool

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

On the user interface for the Standard User Analyzer (SUA) tool, you can apply filters to the issues that the tool has found so that you can view only the information that interests you.

To apply filters to data in the SUA tool

1. Use the SUA tool to test an application. For more information, see [Using the SUA Tool](#).
2. After you finish testing, in the SUA tool, click a tab that shows issues that the SUA tool has found. All tabs except the **App Info** tab can show issues.
3. On the **Options** menu, click a command that corresponds to the filter that you want to apply. The following table describes the commands.

OPTIONS MENU COMMAND	DESCRIPTION
Filter Noise	Filters noise from the issues. This command is selected by default.
Load Noise Filter File	Opens the Open Noise Filter File dialog box, in which you can load an existing noise filter (.xml) file.
Export Noise Filter File	Opens the Save Noise Filter File dialog box, in which you can save filter settings as a noise filter (.xml) file.
Only Display Records with Application Name in StackTrace	Filters out records that do not have the application name in the stack trace. However, because the SUA tool captures only the first 32 stack frames, this command can also filter out real issues with the application where the call stack is deeper than 32 frames.
Show More Details in StackTrace	Shows additional stack frames that are related to the SUA tool, but not related to the diagnosed application.

OPTIONS MENU COMMAND	DESCRIPTION
Warn Before Deleting AppVerifier Logs	<p>Displays a warning message before the SUA tool deletes all of the existing SUA-related log files on the computer.</p> <p>This command is selected by default.</p>
Logging	<p>Provides the following logging-related options:</p> <ul style="list-style-type: none">• Show or hide log errors.• Show or hide log warnings.• Show or hide log information. <p>To maintain a manageable file size, we recommend that you do not select the option to show informational messages.</p>

Fixing Applications by Using the SUA Tool

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

On the user interface for the Standard User Analyzer (SUA) tool, you can apply fixes to an application.

To fix an application by using the SUA tool

1. Use the SUA tool to test an application. For more information, see [Using the SUA Tool](#).
2. After you finish testing, open the SUA tool.
3. On the **Mitigation** menu, click the command that corresponds to the action that you want to take. The following table describes the commands.

MITIGATION MENU COMMAND	DESCRIPTION
Apply Mitigations	Opens the Mitigate AppCompat Issues dialog box, in which you can select the fixes that you intend to apply to the application.
Undo Mitigations	Removes the application fixes that you just applied. This option is available only after you apply an application fix and before you close the SUA tool. Alternatively, you can manually remove application fixes by using Programs and Features in Control Panel.
Export Mitigations as Windows Installer file	Exports your application fixes as a Windows® Installer (.msi) file, which can then be deployed to other computers that are running the application.

Compatibility Administrator User's Guide

6/6/2019 • 2 minutes to read • [Edit Online](#)

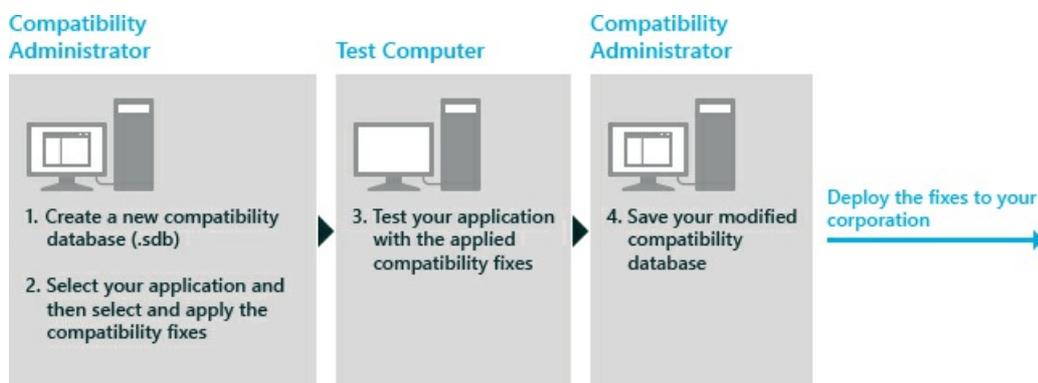
Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Compatibility Administrator tool helps you resolve potential application-compatibility issues before deploying a new version of Windows to your organization. Compatibility Administrator provides the following:

- Compatibility fixes, compatibility modes, AppHelp messages that you can use to resolve specific compatibility issues.
- Tools for creating customized compatibility fixes, compatibility modes, AppHelp messages, and compatibility databases.
- A query tool that you can use to search for installed compatibility fixes on your local computers.

The following flowchart shows the steps for using the Compatibility Administrator tool to create your compatibility fixes, compatibility modes, and AppHelp messages.



Important Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to create and work with custom databases for 32-bit applications, and the 64-bit version to create and work with custom databases for 64-bit applications.

In this section

TOPIC	DESCRIPTION
Using the Compatibility Administrator Tool	This section provides information about using the Compatibility Administrator tool.

TOPIC	DESCRIPTION
Managing Application-Compatibility Fixes and Custom Fix Databases	This section provides information about managing your application-compatibility fixes and custom-compatibility fix databases. This section explains the reasons for using compatibility fixes and how to deploy custom-compatibility fix databases.
Using the Sdbinst.exe Command-Line Tool	You must deploy your customized database (.sdb) files to other computers in your organization before your compatibility fixes, compatibility modes, and AppHelp messages are applied. You can deploy your customized database files in several ways, including by using a logon script, by using Group Policy, or by performing file copy operations.

Using the Compatibility Administrator Tool

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

This section provides information about using the Compatibility Administrator tool.

In this section

TOPIC	DESCRIPTION
Available Data Types and Operators in Compatibility Administrator	The Compatibility Administrator tool provides a way to query your custom-compatibility databases.
Searching for Fixed Applications in Compatibility Administrator	With the search functionality in Compatibility Administrator, you can locate specific executable (.exe) files with previously applied compatibility fixes, compatibility modes, or AppHelp messages. This is particularly useful if you are trying to identify applications with a specific compatibility fix or identifying which fixes are applied to a specific application.
Searching for Installed Compatibility Fixes with the Query Tool in Compatibility Administrator	You can access the Query tool from within Compatibility Administrator. The Query tool provides the same functionality as using the Search feature.
Creating a Custom Compatibility Fix in Compatibility Administrator	The Compatibility Administrator tool uses the term <i>fix</i> to describe the combination of compatibility information added to a customized database for a specific application. This combination can include single application fixes, groups of fixes that work together as a compatibility mode, and blocking and non-blocking AppHelp messages.
Creating a Custom Compatibility Mode in Compatibility Administrator	Windows® provides several <i>compatibility modes</i> , groups of compatibility fixes found to resolve many common application-compatibility issues. While working with Compatibility Administrator, you might decide to group some of your individual compatibility fixes into a custom-compatibility mode, which you can then deploy and use on any of your compatibility databases.

TOPIC	DESCRIPTION
Creating an AppHelp Message in Compatibility Administrator	<p>The Compatibility Administrator tool enables you to create an AppHelp text message. This is a blocking or non-blocking message that appears when a user starts an application that you know has major functionality issues on the Windows® operating system.</p>
Viewing the Events Screen in Compatibility Administrator	<p>The Events screen enables you to record and to view your activities in the Compatibility Administrator tool, provided that the screen is open while you perform the activities.</p>
Enabling and Disabling Compatibility Fixes in Compatibility Administrator	<p>You can disable and enable individual compatibility fixes in your customized databases for testing and troubleshooting purposes.</p>
Installing and Uninstalling Custom Compatibility Databases in Compatibility Administrator	<p>The Compatibility Administrator tool enables the creation and the use of custom-compatibility and standard-compatibility databases. Both the custom databases and the standard databases store the known compatibility fixes, compatibility modes, and AppHelp messages. They also store the required application-matching information for installation on your local computers.</p>

Available Data Types and Operators in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Compatibility Administrator tool provides a way to query your custom-compatibility databases.

Available Data Types

Customized-compatibility databases in Compatibility Administrator contain the following data types.

- **Integer.** A numerical value with no fractional part. All integers are unsigned because none of the attributes can have a negative value.
- **String.** A series of alphanumeric characters manipulated as a group.
- **Boolean.** A value of True or False.

Available Attributes

The following table shows the attributes you can use for querying your customized-compatibility databases in Compatibility Administrator.

ATTRIBUTE	DESCRIPTION	DATA TYPE
APP_NAME	Name of the application.	String
DATABASE_GUID	Unique ID for your compatibility database.	String
DATABASE_INSTALLED	Specifies if you have installed the database.	Boolean
DATABASE_NAME	Descriptive name of your database.	String
DATABASE_PATH	Location of the database on your computer.	String

ATTRIBUTE	DESCRIPTION	DATA TYPE
FIX_COUNT	Number of compatibility fixes applied to a specific application.	Integer
FIX_NAME	Name of your compatibility fix.	String
MATCH_COUNT	Number of matching files for a specific, fixed application.	Integer
MATCHFILE_NAME	Name of a matching file used to identify a specific, fixed application.	String
MODE_COUNT	Number of compatibility modes applied to a specific, fixed application.	Integer
MODE_NAME	Name of your compatibility mode.	String
PROGRAM_APPHELPTYPE	Type of AppHelp message applied to an entry. The value can be 1 or 2, where 1 enables the program to run and 2 blocks the program.	Integer
PROGRAM_DISABLED	Specifies if you disabled the compatibility fix for an application. If True, Compatibility Administrator does not apply the fixes to the application.	Boolean
PROGRAM_GUID	Unique ID for an application.	String
PROGRAM_NAME	Name of the application that you are fixing.	String

Available Operators

The following table shows the operators that you can use for querying your customized-compatibility databases in the Compatibility Administrator.

SYMBOL	DESCRIPTION	DATA TYPE	PRECEDENCE
>	Greater than	Integer or string	1
>=	Greater than or equal to	Integer or string	1

SYMBOL	DESCRIPTION	DATA TYPE	PRECEDENCE
<	Less than	Integer or string	1
<=	Less than or equal to	Integer or string	1
<>	Not equal to	Integer or string	1
=	Equal to	Integer, string, or Boolean	1
HAS	A special SQL operator used to check if the left-hand operand contains a substring specified by the right-hand operand.	<p>Left-hand operand. MATCHFILE_NAME, MODE_NAME, FIX_NAME</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Note Only the HAS operator can be applied to the MATCHFILE_NAME, MODE_NAME, and FIX_NAME attributes.</p> </div> <p>Right-hand operand. String</p>	1
OR	Logical OR operator	Boolean	2
AND	Logical AND operator	Boolean	2

Related topics

[Using the Compatibility Administrator Tool](#)

Searching for Fixed Applications in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

With the search functionality in Compatibility Administrator, you can locate specific executable (.exe) files with previously applied compatibility fixes, compatibility modes, or AppHelp messages. This is particularly useful if you are trying to identify applications with a specific compatibility fix or identifying which fixes are applied to a specific application.

The **Query Compatibility Databases** tool provides additional search options. For more information, see [Searching for Installed Compatibility Fixes with the Query Tool in Compatibility Administrator](#).

Searching for Previously Applied Compatibility Fixes

Important You must perform your search with the correct version of the Compatibility Administrator tool. If you are searching for a 32-bit custom database, you must use the 32-bit version of Compatibility Administrator. If you are searching for a 64-bit custom database, you must use the 64-bit version of Compatibility Administrator.

To search for previous fixes

1. On the Compatibility Administrator toolbar, click **Search**.
2. Click **Browse** to locate the directory location to search for .exe files.
3. Select at least one check box from **Entries with Compatibility Fixes**, **Entries with Compatibility Modes**, or **Entries with AppHelp**.
4. Click **Find Now**.

The query runs, returning your results in the lower pane.

Viewing Your Query Results

Your query results display the affected files, the application location, the application name, the type of compatibility fix, and the custom database that provided the fix.

Exporting Your Query Results

You can export your search results to a text (.txt) file for later review or archival.

To export your search results

1. In the **Search for Fixes** dialog box, click **Export**.

2. Browse to the location where you want to store your search result file, and then click **Save**.

Related topics

[Compatibility Administrator User's Guide](#)

Searching for Installed Compatibility Fixes with the Query Tool in Compatibility Administrator

6/6/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

You can access the Query tool from within Compatibility Administrator. The Query tool provides the same functionality as using the Search feature.

For information about the Search feature, see [Searching for Fixed Applications in Compatibility Administrator](#). However, the Query tool provides more detailed search criteria, including tabs that enable you to search the program properties, the compatibility fix properties, and the fix description. You can perform a search by using SQL SELECT and WHERE clauses, in addition to searching specific types of databases.

Important You must perform your search with the correct version of the Compatibility Administrator tool. To use the Query tool to search for a 32-bit custom database, you must use the 32-bit version of Compatibility Administrator. To use the Query tool to search for a 64-bit custom database, you must use the 64-bit version of Compatibility Administrator.

Querying by Using the Program Properties Tab

You can use the **Program Properties** tab of the Query tool to search for any compatibility fix, compatibility mode, or AppHelp for a specific application.

To query by using the Program Properties tab

1. On the Compatibility Administrator toolbar, click **Query**.
2. In the **Look in** drop-down list, select the appropriate database type to search.
3. Type the location of the application you are searching for into the **Search for the Application** field.

This name should be the same as the name in the **Applications** area (left pane) of Compatibility Administrator.

4. Type the application executable (.exe) file name into the **Search for the File** box. If you leave this box blank, the percent (%) sign appears as a wildcard to search for any file.

You must designate the executable name that was given when the compatibility fix was added to the database.

5. Optionally, select the check box for one of the following types of compatibility fix:
 - **Compatibility Modes**
 - **Compatibility Fixes**

- **Application Helps**

Important If you do not select any of the check boxes, the search will look for all types of compatibility fixes. Do not select multiple check boxes because only applications that match all of the requirements will appear.

6. Click **Find Now**.

The query runs and the results of the query are displayed in the lower pane.

Querying by Using the Fix Properties Tab

You can use the **Fix Properties** tab of the Query tool to search for any application affected by a specific compatibility fix or a compatibility mode. For example, you can search for any application affected by the ProfilesSetup compatibility mode.

To query by using the Fix Properties tab

1. On the Compatibility Administrator toolbar, click **Query**.
2. Click the **Fix Properties** tab.
3. In the **Look in** drop-down list, select the appropriate database type to search.
4. Type the name of the compatibility fix or compatibility mode into the **Search for programs fixed using** field.

Note You can use the percent (%) symbol as a wildcard in your fix-properties query, as a substitute for any string of zero or more characters.

5. Select the check box for either **Search in Compatibility Fixes** or **Search in Compatibility Modes**.

Important Your text must match the type of compatibility fix or mode for which you are performing the query. For example, entering the name of a compatibility fix and selecting the compatibility mode check box will not return any results. Additionally, if you select both check boxes, the query will search for the fix by compatibility mode and compatibility fix. Only applications that match both requirements appear.

6. Click **Find Now**.

The query runs and the results of the query are displayed in the lower pane.

Querying by Using the Fix Description Tab

You can use the **Fix Description** tab of the Query tool to add parameters that enable you to search your compatibility databases by application title or solution description text.

To query by using the Fix Description tab

1. On the Compatibility Administrator toolbar, click **Query**.
2. Click the **Fix Description** tab.
3. In the **Look in** drop-down list, select the appropriate database type to search.
4. Type your search keywords into the box **Words to look for**. Use commas to separate multiple keywords.

Important You cannot use wildcards as part of the Fix Description search query because the default behavior is to search for any entry that meets your search criteria.

5. Refine your search by selecting **Match any word** or **Match all words** from the drop-down list.
6. Click **Find Now**.

The query runs and the results of the query are displayed in the lower pane.

Querying by Using the Fix Description Tab

You can use the **Fix Description** tab of the Query tool to add additional SQL Server SELECT and WHERE clauses to your search criteria.

To query by using the Advanced tab

1. On the Compatibility Administrator toolbar, click **Query**.
2. Click the **Advanced** tab.
3. In the **Look in** drop-down list, select the appropriate database type to search.
4. Select the appropriate SELECT clause for your search from the **Select clauses** box. For example, **APP_NAME**.

The **APP_NAME** clause appears in the **SELECT** field. You can add as many additional clauses as you require. They will appear as columns in your search results.

5. Select the appropriate WHERE clause for your search from the **Where clauses** box. For example, **DATABASE_NAME**.

The **DATABASE_NAME =** clause appears in the **WHERE** box.

6. Type the appropriate clause criteria after the equal (=) sign in the **WHERE** box. For example, **DATABASE_NAME = "Custom_Database"**.

You must surround your clause criteria text with quotation marks (") for the clause to function properly.

7. Click **Find Now**.

The query runs and the results of the query are displayed in the lower pane.

Exporting Your Search Results

You can export any of your search results into a tab-delimited text (.txt) file for later review or for archival purposes.

To export your results

1. After you have completed your search by using the Query tool, click **Export**.

The **Save results to a file** dialog box appears.

2. Browse to the location where you intend to store the search results file, and then click **Save**.

Related topics

[Compatibility Administrator User's Guide](#)

Creating a Custom Compatibility Fix in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Compatibility Administrator tool uses the term *fix* to describe the combination of compatibility information added to a customized database for a specific application. This combination can include single application fixes, groups of fixes that work together as a compatibility mode, and blocking and non-blocking AppHelp messages.

Important Fixes apply to a single application only; therefore, you must create multiple fixes if you need to fix the same issue in multiple applications.

What is a Compatibility Fix?

A compatibility fix, previously known as a shim, is a small piece of code that intercepts API calls from applications. The fix transforms the API calls so that the current version of the operating system supports the application in the same way as previous versions of the operating system. This can mean anything from disabling a new feature in the current version of the operating system to emulating a particular behavior of an older version of the Windows API.

Searching for Existing Compatibility Fixes

The Compatibility Administrator tool has preloaded fixes for many common applications, including known compatibility fixes, compatibility modes, and AppHelp messages. Before you create a new compatibility fix, you can search for an existing application and then copy and paste the known fixes into your customized database.

Important Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to create custom databases for 32-bit applications and the 64-bit version to create custom databases for 64-bit applications.

To search for an existing application

1. In the left-side pane of Compatibility Administrator, expand the **Applications** folder and search for your application name.
2. Click the application name to view the preloaded compatibility fixes, compatibility modes, or AppHelp messages.

Creating a New Compatibility Fix

If you are unable to find a preloaded compatibility fix for your application, you can create a new one for use by your customized database.

To create a new compatibility fix

1. In the left-side pane of Compatibility Administrator underneath the **Custom Databases** heading, right-click the name of the database to which you want to apply the compatibility fix, click **Create New**, and then click **Application Fix**.
2. Type the name of the application to which the compatibility fix applies, type the name of the application vendor, browse to the location of the application file (.exe) on your computer, and then click **Next**.
3. Select the operating system for which your compatibility fix applies, click any applicable compatibility modes to apply to your compatibility fix, and then click **Next**.
4. Select any additional compatibility fixes to apply to your compatibility fix, and then click **Next**.
5. Select any additional criteria to use to match your applications to the AppHelp message, and then click **Finish**.

By default, Compatibility Administrator selects the basic matching criteria for your application. As a best practice, use a limited set of matching information to represent your application, because it reduces the size of the database. However, make sure you have enough information to correctly identify your application.

Related topics

[Compatibility Administrator User's Guide](#)

Creating a Custom Compatibility Mode in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

Windows® provides several *compatibility modes*, groups of compatibility fixes found to resolve many common application-compatibility issues. While working with Compatibility Administrator, you might decide to group some of your individual compatibility fixes into a custom-compatibility mode, which you can then deploy and use on any of your compatibility databases.

What Is a Compatibility Mode?

A compatibility mode is a group of compatibility fixes. A compatibility fix, previously known as a shim, is a small piece of code that intercepts API calls from applications. The fix transforms the API calls so that the current version of the operating system supports the application in the same way as previous versions of the operating system. This can be anything from disabling a new feature in Windows to emulating a particular behavior of an older version of the Windows API.

Searching for Existing Compatibility Modes

The Compatibility Administrator tool has preloaded fixes for many common applications, including known compatibility fixes, compatibility modes, and AppHelp messages. Before you create a new compatibility mode, you can search for an existing application and then copy and paste the known fixes into your custom database.

Important

Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to create custom databases for 32-bit applications and the 64-bit version to create custom databases for 64-bit applications.

To search for an existing application

1. In the left-side pane of Compatibility Administrator, expand the **Applications** folder and search for your application name.
2. Click the application name to view the preloaded compatibility modes, compatibility fixes, or AppHelp messages.

Creating a New Compatibility Mode

If you are unable to find a preloaded compatibility mode for your application, you can create a new one for use by your custom database.

Important

A compatibility mode includes a set of compatibility fixes and must be deployed as a group. Therefore, you should include only fixes that you intend to deploy together to the database.

To create a new compatibility mode

1. In the left-side pane of Compatibility Administrator, underneath the **Custom Databases** heading, right-click the name of the database to which you will apply the compatibility mode, click **Create New**, and then click **Compatibility Mode**.
2. Type the name of your custom-compatibility mode into the **Name of the compatibility mode** text box.
3. Select each of the available compatibility fixes to include in your custom-compatibility mode and then click **>**.

Important

If you are unsure which compatibility fixes to add, you can click **Copy Mode**. The **Select Compatibility Mode** dialog box appears and enables you to select from the preloaded compatibility modes. After you select a compatibility mode and click **OK**, any compatibility fixes that are included in the preloaded compatibility mode will be automatically added to your custom-compatibility mode.

If you have any compatibility fixes that require additional parameters, you can select the fix, and then click **Parameters**. The **Options for <Compatibility_Fix_Name>** dialog box appears, enabling you to update the parameter fields.

4. After you are done selecting the compatibility fixes to include, click **OK**.

The compatibility mode is added to your custom database.

Related topics

[Compatibility Administrator User's Guide](#)

Creating an AppHelp Message in Compatibility Administrator

6/6/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Compatibility Administrator tool enables you to create an AppHelp text message. This is a blocking or non-blocking message that appears when a user starts an application that you know has major functionality issues on the Windows® operating system.

Blocking Versus Non-Blocking AppHelp Messages

A blocking AppHelp message prevents the application from starting and displays a message to the user. You can define a specific URL where the user can download an updated driver or other fix to resolve the issue. When using a blocking AppHelp message, you must also define the file-matching information to identify the version of the application and enable the corrected version to continue.

A non-blocking AppHelp message does not prevent the application from starting, but provides a message to the user including information such as security issues, updates to the application, or changes to the location of network resources.

Searching for Existing Compatibility Fixes

The Compatibility Administrator tool has preloaded fixes for many common applications, including known compatibility fixes, compatibility modes, and AppHelp messages. Before you create a new AppHelp message, you can search for an existing application and then copy and paste the known fixes into your custom database.

Important Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to create custom databases for 32-bit applications and the 64-bit version to create custom databases for 64-bit applications.

To search for an existing application

1. In the left-side pane of Compatibility Administrator, expand the **Applications** folder and search for your application name.
2. Click the application name to view the preloaded AppHelp messages, compatibility fixes, and compatibility modes.

Creating a New AppHelp Message

If you are unable to find a preloaded AppHelp message for your application, you can create a new one for use by your custom database.

To create a new AppHelp message

1. In the left-side pane of Compatibility Administrator, below the **Custom Databases** heading, right-click the name of the database to which you will apply the AppHelp message, click **Create New**, and then click **AppHelp Message**.

2. Type the name of the application to which this AppHelp message applies, type the name of the application vendor, browse to the location of the application file (.exe) on your computer, and then click **Next**.

The wizard shows the known **Matching Information**, which is used for program identification.

3. Select any additional criteria to use to match your applications to the AppHelp message, and then click **Next**.

By default, Compatibility Administrator selects the basic matching criteria for your application.

The wizard shows the **Enter Message Type** options.

4. Click one of the following options:

- **Display a message and allow this program to run.** This is a non-blocking message, which means that you can alert the user that there might be a problem, but the application is not prevented from starting.
- **Display a message and do not allow this program to run.** This is a blocking message, which means that the application will not start. Instead, this message points the user to a location that provides more information about fixing the issue.

5. Click **Next**.

The wizard then shows the **Enter Message Information** fields.

6. Type the website URL and the message text to appear when the user starts the application, and then click **Finish**.

Issues with AppHelp Messages and Computers Running Windows 2000

The following issues might occur with computers running Windows 2000:

- You might be unable to create a custom AppHelp message.
- The AppHelp message text used for system database entries might not appear.
- Copying an AppHelp entry for a system database or a custom-compatibility fix from a system database might cause Compatibility Administrator to hide the descriptive text.

Related topics

[Compatibility Administrator User's Guide](#)

Viewing the Events Screen in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The **Events** screen enables you to record and to view your activities in the Compatibility Administrator tool, provided that the screen is open while you perform the activities.

Important The **Events** screen only records your activities when the screen is open. If you perform an action before opening the **Events** screen, the action will not appear in the list.

To open the Events screen

- On the **View** menu, click **Events**.

Handling Multiple Copies of Compatibility Fixes

Compatibility Administrator enables you to copy your compatibility fixes from one database to another, which can become confusing after adding multiple fixes, compatibility modes, and databases. For example, you can copy a fix called MyFix from Database 1 to Database 2. However, if there is already a fix called MyFix in Database 2, Compatibility Administrator renames the fix as MyFix (1) to avoid duplicate names.

If you open the **Events** screen and then perform the copy operation, you can see a description of the action, along with the time stamp, which enables you to view your fix information without confusion.

Related topics

[Creating a Custom Compatibility Mode in Compatibility Administrator](#)

[Compatibility Administrator User's Guide](#)

Enabling and Disabling Compatibility Fixes in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

You can disable and enable individual compatibility fixes in your customized databases for testing and troubleshooting purposes.

Disabling Compatibility Fixes

Customized compatibility databases can become quite complex as you add your fixes for the multiple applications found in your organization. Over time, you may find you need to disable a particular fix in your customized database. For example, if a software vendor releases a fix for an issue addressed in one of your compatibility fixes, you must validate that the vendor's fix is correct and that it resolves your issue. To do this, you must temporarily disable the compatibility fix and then test your application.

Important Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to work with custom databases for 32-bit applications and the 64-bit version to work with custom databases for 64-bit applications.

To disable a compatibility fix within a database

1. In the left-side pane of Compatibility Administrator, expand the custom database that includes the compatibility fix that you want to disable, and then select the specific compatibility fix.

The compatibility fix details appear in the right-hand pane.

2. On the **Database** menu, click **Disable Entry**.

Important When you disable an entry, it will remain disabled even if you do not save the database file.

Enabling Compatibility Fixes

You can enable your disabled compatibility fixes at any time.

To enable a compatibility fix within a database

1. In the left-side pane of Compatibility Administrator, expand the custom database that includes the compatibility fix that you want to enable, and then select the specific compatibility fix.

The compatibility fix details appear in the right-side pane.

2. On the **Database** menu, click **Enable Entry**.

Related topics

[Compatibility Administrator User's Guide](#)

Installing and Uninstalling Custom Compatibility Databases in Compatibility Administrator

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

The Compatibility Administrator tool enables the creation and the use of custom-compatibility and standard-compatibility databases. Both the custom databases and the standard databases store the known compatibility fixes, compatibility modes, and AppHelp messages. They also store the required application-matching information for installation on your local computers.

By default, the Windows® operating system installs a System Application Fix database for use with the Compatibility Administrator. This database can be updated through Windows Update, and is stored in the %WINDIR% \AppPatch directory. Your custom databases are automatically stored in the %WINDIR% \AppPatch\Custom directory and are installed by using the Sdbinst.exe tool provided with the Compatibility Administrator.

Important Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator tool. You must use the 32-bit version to work with custom databases for 32-bit applications and the 64-bit version to work with custom databases for 64-bit applications.

In addition, you must deploy your databases to your organization's computers before the included fixes will have any effect on the application issue. For more information about deploying your database, see [Using the Sdbinst.exe Command-Line Tool](#).

Installing a Custom Database

Installing your custom-compatibility database enables you to fix issues with your installed applications.

To install a custom database

1. In the left-side pane of Compatibility Administrator, click the custom database to install to your local computers.
2. On the **File** menu, click **Install**.

The Compatibility Administrator installs the database, which appears in the **Installed Databases** list.

The relationship between your database file and an included application occurs in the registry. Every time you start an application, the operating system checks the registry for compatibility-fix information and, if found, retrieves the information from your customized database file.

Uninstalling a Custom Database

When a custom database is no longer necessary, either because the applications are no longer used or because the

vendor has provided a fix that resolves the compatibility issues, you can uninstall the custom database.

To uninstall a custom database

1. In the **Installed Databases** list, which appears in the left-side pane of Compatibility Administrator, click the database to uninstall from your local computers.
2. On the **File** menu, click **Uninstall**.

Related topics

[Compatibility Administrator User's Guide](#)

Managing Application-Compatibility Fixes and Custom Fix Databases

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

This section provides information about managing your application-compatibility fixes and custom-compatibility fix databases. This section explains the reasons for using compatibility fixes and how to deploy custom-compatibility fix databases.

In this section

TOPIC	DESCRIPTION
Understanding and Using Compatibility Fixes	As the Windows operating system evolves to support new technology and functionality, the implementations of some functions may change. This can cause problems for applications that relied upon the original implementation. You can avoid compatibility issues by using the Microsoft Windows Application Compatibility (Compatibility Fix) infrastructure to create a specific application fix for a particular version of an application.
Compatibility Fix Database Management Strategies and Deployment	After you determine that you will use compatibility fixes in your application-compatibility mitigation strategy, you must define a strategy to manage your custom compatibility-fix database. Typically, you can use one of two approaches:
Testing Your Application Mitigation Packages	This topic provides details about testing your application-mitigation packages, including recommendations about how to report your information and how to resolve any outstanding issues.

Related topics

[Compatibility Administrator User's Guide](#)

[Using the Compatibility Administrator Tool](#)

Understanding and Using Compatibility Fixes

6/6/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

As the Windows operating system evolves to support new technology and functionality, the implementations of some functions may change. This can cause problems for applications that relied upon the original implementation. You can avoid compatibility issues by using the Microsoft Windows Application Compatibility (Compatibility Fix) infrastructure to create a specific application fix for a particular version of an application.

How the Compatibility Fix Infrastructure Works

The Compatibility Fix infrastructure uses the linking ability of APIs to redirect an application from Windows code directly to alternative code that implements the compatibility fix.

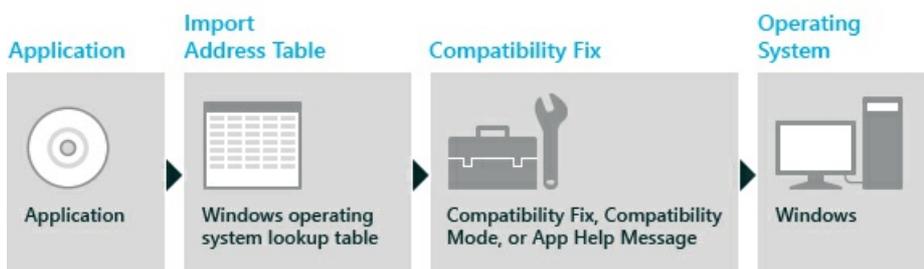
The Windows Portable Executable File Format includes headers that contain the data directories that are used to provide a layer of indirection between the application and the linked file. API calls to the external binary files take place through the Import Address Table (IAT), which then directly calls the Windows operating system, as shown in the following figure.

Compatibility Fix Infrastructure without Fix



Specifically, the process modifies the address of the affected Windows function in the IAT to point to the compatibility fix code, as shown in the following figure.

Compatibility Fix Infrastructure with Fix



Note For statically linked DLLs, the code redirection occurs as the application loads. You can also fix dynamically linked DLLs by hooking into the GetProcAddress API.

Design Implications of the Compatibility Fix Infrastructure

There are important considerations to keep in mind when determining your application fix strategy, due to certain characteristics of the Compatibility Fix infrastructure.

- The compatibility fix is not part of the Windows operating system (as shown in the previous figure). Therefore, the same security restrictions apply to the compatibility fix as apply to the application code, which means that you cannot use compatibility fixes to bypass any of the security mechanisms of the operating system. Therefore, compatibility fixes do not increase your security exposure, nor do you need to lower your security settings to accommodate compatibility fixes.
- The Compatibility Fix infrastructure injects additional code into the application before it calls the operating system. This means that any remedy that can be accomplished by a compatibility fix can also be addressed by fixing the application code.
- The compatibility fixes run as user-mode code inside of a user-mode application process. This means that you cannot use a compatibility fix to fix kernel-mode code issues. For example, you cannot use a compatibility fix to resolve device-driver issues.

Note Some antivirus, firewall, and anti-spyware code runs in kernel mode.

Determining When to Use a Compatibility Fix

The decision to use compatibility fixes to remedy your compatibility issues may involve more than just technical issues. The following scenarios reflect other common reasons for using a compatibility fix.

Scenario 1

The compatibility issue exists on an application which is no longer supported by the vendor.

As in many companies, you may run applications for which the vendor has ended support. In this situation, you cannot have the vendor make the fix, nor can you access the source code to modify the issue yourself. However, it is possible that the use of a compatibility fix might resolve the compatibility issue.

Scenario 2

The compatibility issue exists on an internally created application.

While it is preferable to fix the application code to resolve the issue, this is not always possible. Your internal team might not be able to fix all of the issues prior to the deployment of the new operating system. Instead, they might choose to employ a compatibility fix anywhere that it is possible. They can then fix the code only for issues that cannot be resolved in this manner. Through this method, your team can modify the application as time permits, without delaying the deployment of the new operating system into your environment.

Scenario 3

The compatibility issue exists on an application for which a compatible version is to be released in the near future, or an application that is not critical to the organization, regardless of its version.

In the situation where an application is either unimportant to your organization, or for which a newer, compatible version is to be released shortly, you can use a compatibility fix as a temporary solution. This means that you can continue to use the application without delaying the deployment of a new operating system, with the intention of updating your configuration as soon as the new version is released.

Determining Which Version of an Application to Fix

You can apply a compatibility fix to a particular version of an application, either by using the "up to or including" clause or by selecting that specific version. This means that the next version of the application will not have the compatibility fix automatically applied. This is important, because it allows you to continue to use your application,

but it also encourages the vendor to fix the application.

Support for Compatibility Fixes

Compatibility fixes are shipped as part of the Windows operating system and are updated by using Windows Update. Therefore, they receive the same level of support as Windows itself.

You can apply the compatibility fixes to any of your applications. However, Microsoft does not provide the tools to use the Compatibility Fix infrastructure to create your own custom fixes.

Related topics

[Managing Application-Compatibility Fixes and Custom Fix Databases](#)

Compatibility Fix Database Management Strategies and Deployment

6/6/2019 • 7 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

After you determine that you will use compatibility fixes in your application-compatibility mitigation strategy, you must define a strategy to manage your custom compatibility-fix database. Typically, you can use one of two approaches:

- Deploying your compatibility fixes as part of an application-installation package.
- Deploying your compatibility fixes through a centralized compatibility-fix database.

Regardless of which approach you decide to use in your organization, Microsoft provides the following general recommendations for improving the management of your custom compatibility-fix databases:

- **Define standards for when you will apply compatibility fixes.**

You must define the standards and scenarios for using compatibility fixes, based on your specific business and technology needs.

- **Define standards for your custom compatibility-fix databases.**

You must define how to associate your compatibility fixes to particular applications. For example, you might want to ensure that your compatibility fixes always include a version check, so that a fix will not be applied to newer versions of your applications.

- **Define your resources responsible for addressing questions and enforcing your standards.**

You must determine who will be responsible for staying current with the technology and standards related to your compatibility fixes and custom compatibility-fix databases. As your databases are managed over time, you must ensure that someone in your organization stays current with the relevant technology.

Strategies for Deploying Your Compatibility Fixes

We recommend that you use one of two strategies to deploy your compatibility fixes into your organization. They are:

- Deploying your compatibility fixes as part of an application-installation package.
- Deploying your compatibility fixes through a centralized compatibility-fix database.

You must determine which method best meets your organization's deployment needs.

Deploying Fixes as Part of an Application-Installation Package

One strategy for deploying compatibility fixes is to create a custom compatibility-fix database that contains a single entry that is applied directly to the application-installation package. While this is the most straightforward method of deployment, it has been shown that this method can become overly complex, especially if you are fixing a large number of applications.

If the following considerations apply to your organization, you should avoid this strategy and instead consider using a centralized compatibility-fix database, as described in the next section.

- **How many applications require compatibility fixes?**

Custom compatibility-fix databases are actual databases. Therefore, if you have 1000 applications to be fixed, it will take longer to open and query 1000 single-row databases for a match, instead of a single database with 1000 rows.

- **Will you be able to track which applications are installed on which computer?**

You might determine that your initial set of compatibility fixes is not comprehensive, and that you must deploy an updated version of the compatibility-fix database to resolve the additional issues. If you deployed the initial set by using the application-installation package, you will be required to locate each client computer that is running the application and replace the compatibility fix.

Deploying Fixes Through a Centralized Compatibility-Fix Database

The other recommended strategy for deploying compatibility fixes into your organization is to create and manage either a single custom compatibility-fix database, or else to create and manage several custom databases for large subsets of your organization. This strategy will help to enforce your company policy and to provide consistent updates for application fixes that you discover later.

This approach tends to work best for organizations that have a well-developed deployment infrastructure in place, with centralized ownership of the process. We recommend that you consider the following before using this approach:

- Does your organization have the tools required to deploy and update a compatibility-fix database for all of the effected computers?

If you intend to manage a centralized compatibility-fix database, you must verify that your organization has the required tools to deploy and update all of the affected computers in your organization.

- Do you have centralized resources that can manage and update the centralized compatibility-fix database?

You must ensure that you have identified the appropriate owners for the deployment process, for the applications, and for the database updates, in addition to determining the process by which compatibility issues can be deployed to specific computers.

Merging Centralized Compatibility-Fix Databases

If you decide to use the centralized compatibility-fix database deployment strategy, you can merge any of your individual compatibility-fix databases. This enables you to create a single custom compatibility-fix database that can be used to search for and determine whether Windows® should apply a fix to a specific executable (.exe) file. We recommend merging your databases based on the following process.

To merge your custom-compatibility databases

1. Verify that your application-compatibility testers are performing their tests on computers with the latest version of your compatibility-fix database. For example, Custom DB1.
2. If the tester determines that an application requires an additional compatibility fix that is not a part of the original compatibility-fix database, he or she must create a new custom compatibility database with all of the required information for that single fix. For example, Custom DB2.
3. The tester applies the new Custom DB2 information to the application and then tests for both the

functionality and integration, to ensure that the compatibility issues are addressed.

4. After the application passes all of the required functionality and integration tests, the tester can send Custom DB2 to the team that manages the central compatibility-fix database.
5. The team that manages the centralized database opens Custom DB1 and uses the Compatibility Administrator to include the new compatibility fixes that were included in Custom DB2.

Note

Custom DB1 contains a unique GUID that makes updating the database easier. For example, if you install a new version of the custom compatibility-fix database that uses the same GUID as the previous version, the computer will automatically uninstall the old version.

6. The centralized management team then redeploys the new version of Custom DB1 to all of the end users in your organization.

Deploying Your Custom Compatibility-Fix Databases

Deploying your custom compatibility-fix database into your organization requires you to perform the following actions:

1. Store your custom compatibility-fix database (.sdb file) in a location that is accessible to all of your organization's computers.
2. Use the Sdbinst.exe command-line tool to install the custom compatibility-fix database locally.

In order to meet the two requirements above, we recommend that you use one of the following two methods:

- **Using a Windows Installer package and a custom script**

You can package your .sdb file and a custom deployment script into an .msi file, and then deploy the .msi file into your organization.

Important

You must ensure that you mark your custom script so that it does not impersonate the calling user. For example, if you use Microsoft® Visual Basic® Scripting Edition (VBScript), the custom action type would be:

```
``` syntax
msidbCustomActionTypeVBScript + msidbCustomActionTypeInScript + msidbCustomActionTypeNoImpersonate = 0x0006 +
0x0400 + 0x0800 = 0x0C06 = 3078 decimal
```
```

- **Using a network share and a custom script**

You can store your .sdb file on your network share and then call to a script that resides on your specified computers.

Important

You must ensure that you call the script at a time when it will receive elevated rights. For example, you should call the script by using computer startup scripts instead of a user logon script. You must also ensure that the installation of the custom compatibility-fix database occurs with Administrator rights.

Example Script for an Installation of the .sdb File based on an .msi File

The following examples show an installation of a custom compatibility-fix database based on an .msi file.

```
'InstallSDB.vbs
Function Install
Dim WshShell
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run "sdbinst.exe -q " & CHR(34) & "%ProgramFiles%\MyOrganizationSDB\MyOrg.sdb" & CHR(34), 0, true
WshShell.Run "cmd.exe /c " & CHR(34) & "del " & CHR(34) & "%ProgramFiles%\MyOrganizationSDB\MyOrg.sdb" &
CHR(34) & CHR(34), 0
WshShell.Run "reg.exe delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{guidFromMyOrgsSdb}.sdb
/f", 0
End Function

Function UnInstall
Dim WshShell
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run "sdbinst.exe -q -u -g {guidFromMyOrgsSdb}", 0
End Function
```

Initial Deployment and Updates

Most of your testing of application-compatibility issues will happen prior to the deployment of a new Windows operating system into your environment. As such, a common approach is to include the custom compatibility-fix database, which includes all of your known issues, in your corporate image. Then, as you update your compatibility-fix database, you can provide the updates by using one of the two mechanisms described in the "Deploying Your Custom Compatibility Fix Databases" section earlier in this topic.

Related topics

[Managing Application-Compatibility Fixes and Custom Fix Databases](#)

Testing Your Application Mitigation Packages

6/6/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

This topic provides details about testing your application-mitigation packages, including recommendations about how to report your information and how to resolve any outstanding issues.

Testing Your Application Mitigation Packages

Testing your application mitigation package strategies is an iterative process, whereby the mitigation strategies that prove unsuccessful will need to be revised and retested. The testing process includes a series of tests in the test environment and one or more pilot deployments in the production environment.

To test your mitigation strategies

1. Perform the following steps for each of the applications for which you have developed mitigations.
 - a. Test the mitigation strategy in your test environment.
 - b. If the mitigation strategy is unsuccessful, revise the mitigation strategy and perform step 1 again.

At the end of this step, you will have successfully tested all of your mitigation strategies in your test environment and can move to your pilot deployment environment.

2. Perform the following steps in the pilot deployments for each of the applications for which you have developed mitigations.
 - a. Test the mitigation strategy in your pilot deployment.
 - b. If the mitigation strategy is unsuccessful, revise the mitigation strategy and perform Step 2 again.

At the end of this step, you will have successfully tested all of your mitigation strategies in your pilot environment.

Reporting the Compatibility Mitigation Status to Stakeholders

After testing your application mitigation package, you must communicate your status to the appropriate stakeholders before deployment begins. We recommend that you perform this communication by using the following status ratings.

- **Resolved application compatibility issues.** This status indicates that the application compatibility issues are resolved and that these applications represent no risk to your environment.
- **Unresolved application compatibility issues.** This status indicates that there are unresolved issues for the specifically defined applications. Because these applications are a risk to your environment, more discussion is required before you can resolve the compatibility issues.

- **Changes to user experience.** This status indicates that the fix will change the user experience for the defined applications, possibly requiring your staff to receive further training. More investigation is required before you can resolve the compatibility issues.
- **Changes in help desk procedures and processes.** This status indicates that the fix will require changes to your help desk's procedures and processes, possibly requiring your support staff to receive further training. More investigation is required before you can resolve the compatibility issues.

Resolving Outstanding Compatibility Issues

At this point, you probably cannot resolve any unresolved application compatibility issues by automated mitigation methods or by modifying the application. Resolve any outstanding application compatibility issues by using one of the following methods.

- Apply specific compatibility modes, or run the program as an Administrator, by using the Compatibility Administrator tool.

Note For more information about using Compatibility Administrator to apply compatibility fixes and compatibility modes, see [Using the Compatibility Administrator Tool](#).

- Run the application in a virtual environment.

Run the application in a version of Windows supported by the application in a virtualized environment. This method ensures application compatibility, because the application is running on a supported operating system.

- Resolve application compatibility by using non-Microsoft tools.

If the application was developed in an environment other than Microsoft Visual Studio®, you must use non-Microsoft debugging and analysis tools to help resolve the remaining application compatibility issues.

- Outsource the application compatibility mitigation.

If your developers have insufficient resources to resolve the application compatibility issues, outsource the mitigation effort to another organization within your company.

Related topics

[Managing Application-Compatibility Fixes and Custom Fix Databases](#)

Using the Sdbinst.exe Command-Line Tool

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2016
- Windows Server 2012
- Windows Server 2008 R2

You must deploy your customized database (.sdb) files to other computers in your organization before your compatibility fixes, compatibility modes, and AppHelp messages are applied. You can deploy your customized database files in several ways, including by using a logon script, by using Group Policy, or by performing file copy operations.

After you deploy and store the customized databases on each of your local computers, you must register the database files. Until you register the database files, the operating system is unable to identify the available compatibility fixes when starting an application.

Command-Line Options for Deploying Customized Database Files

Sample output from the command `Sdbinst.exe /?` in an elevated CMD window:

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>Sdbinst.exe /?
Usage: Sdbinst.exe [-?] [-q] [-u] [-g] [-p] [-n[:WIN32|WIN64]] myfile.sdb | {guid} | "name"

    -? - print this help text.
    -p - Allow SDBs containing patches.
    -q - Quiet mode: prompts are auto-accepted.
    -u - Uninstall.
    -g {guid} - GUID of file (uninstall only).
    -n "name" - Internal name of file (uninstall only).

C:\Windows\system32>_
```

The command-line options use the following conventions:

```
Sdbinst.exe [-?] [-p] [-q] [-u] [-g] [-u filepath] [-g GUID] [-n "name"]
```

The following table describes the available command-line options.

| OPTION | DESCRIPTION |
|--------|-------------|
|--------|-------------|

| OPTION | DESCRIPTION |
|--------------------|--|
| -? | <p>Displays the Help for the Sdbinst.exe tool.</p> <p>For example,</p> <pre data-bbox="823 280 997 309">sdbinst.exe -?</pre> |
| -p | <p>Allows SDBs installation with Patches</p> <p>For example,</p> <pre data-bbox="823 495 1337 524">sdbinst.exe -p C:\Windows\AppPatch\Myapp.sdb</pre> |
| -q | <p>Performs a silent installation with no visible window, status, or warning information. Fatal errors appear only in Event Viewer (Eventvwr.exe).</p> <p>For example,</p> <pre data-bbox="823 768 997 797">sdbinst.exe -q</pre> |
| -u <i>filepath</i> | <p>Performs an uninstallation of the specified database.</p> <p>For example,</p> <pre data-bbox="823 976 1166 1005">sdbinst.exe -u C:\example.sdb</pre> |
| -g <i>GUID</i> | <p>Specifies the customized database to uninstall by a globally unique identifier (GUID).</p> <p>For example,</p> <pre data-bbox="823 1216 1394 1267">sdbinst.exe -g 6586cd8f-edc9-4ea8-ad94-afabea7f62e3</pre> |
| -n " <i>name</i> " | <p>Specifies the customized database to uninstall by file name.</p> <p>For example,</p> <pre data-bbox="823 1480 1155 1509">sdbinst.exe -n "My_Database"</pre> |

Related topics

[Compatibility Administrator User's Guide](#)

Compatibility Fixes for Windows 10, Windows 8, Windows 7, and Windows Vista

6/6/2019 • 28 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012
- Windows Server 2008 R2

You can fix some compatibility issues that are due to the changes made between Windows operating system versions. These issues can include User Account Control (UAC) restrictions.

Important

The Application Compatibility Toolkit (ACT) installs a 32-bit and a 64-bit version of the Compatibility Administrator. You must use the 32-bit version for 32-bit applications and the 64-bit version to work for 64-bit applications. You will receive an error message if you try to use the wrong version.

If you start the Compatibility Administrator as an Administrator (with elevated privileges), all repaired applications can run successfully; however, virtualization and redirection might not occur as expected. To verify that a compatibility fix addresses an issue, you must test the repaired application by running it under the destination user account.

Compatibility Fixes

The following table lists the known compatibility fixes for all Windows operating systems that have been released from Windows Vista through Windows 10. The fixes are listed in alphabetical order.

| FIX | FIX DESCRIPTION |
|------------------------|---|
| 8And16BitAggregateBlts | Applications that are mitigated by 8/16-bit mitigation can exhibit performance issues. This layer aggregates all the blt operations and improves performance. |
| 8And16BitDXMaxWinMode | Applications that use DX8/9 and are mitigated by the 8/16-bit mitigation are run in a maximized windowed mode. This layer mitigates applications that exhibit graphical corruption in full screen mode. |
| 8And16BitGDIDraw | This fix repairs applications that use GDI and that work in 8-bit color mode. The application is forced to repaint its window on RealizePalette. |
| AccelGdipFlush | This fix increases the speed of GdipFlush, which has perf issues in DWM. |

| FIX | FIX DESCRIPTION |
|--|--|
| AoaMp4Converter | <p>This fix resolves a display issue for the AoA Mp4 Converter.</p> |
| BIOSRead | <p>This problem is indicated when an application cannot access the Device\PhysicalMemory object beyond the kernel-mode drivers, on any of the Windows Server® 2003 operating systems.</p> <p>The fix enables OEM executable (.exe) files to use the GetSystemFirmwareTable function instead of the NtOpenSection function when the BIOS is queried for the \Device\Physical memory information..</p> |
| BlockRunasInteractiveUser | <p>This problem occurs when InstallShield creates installers and uninstallers that fail to complete and that generate error messages or warnings.</p> <p>The fix blocks InstallShield from setting the value of RunAs registry keys to InteractiveUser Because InteractiveUser no longer has Administrator rights.</p> <div data-bbox="821 891 1433 1048" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the BlockRunAsInteractiveUser Fix.</p> </div> |
| ChangeFolderPathToXPStyle | <p>This fix is required when an application cannot return shell folder paths when it uses the SHGetFolder API.</p> <p>The fix intercepts the SHGetFolder path request to the common appdata file path and returns the Windows® XP-style file path instead of the Windows Vista-style file path.</p> |
| ClearLastErrorStatusonIntializeCriticalSection | <p>This fix is indicated when an application fails to start.</p> <p>The fix modifies the InitializeCriticalSection function call so that it checks the NTSTATUS error code, and then sets the last error to ERROR_SUCCESS.</p> |

| FIX | FIX DESCRIPTION |
|---------------------------------|---|
| CopyHKCUSettingsFromOtherUsers | <p>This problem occurs when an application's installer must run in elevated mode and depends on the HKCU settings that are provided for other users.</p> <p>The fix scans the existing user profiles and tries to copy the specified keys into the HKEY_CURRENT_USER registry area.</p> <p>You can control this fix further by entering the relevant registry keys as parameters that are separated by the ^ Symbol; for example:</p> <pre>Software\MyCompany\Key1^Software\MyCompany\Key2 .</pre> <div data-bbox="821 555 1433 712" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the CopyHKCUSettingsFromOtherUsers Fix.</p> </div> |
| CorrectCreateBrushIndirectHatch | <p>The problem is indicated by an access violation error message that displays and when the application fails when you select or crop an image.</p> <p>The fix corrects the brush style hatch value, which is passed to the CreateBrushIndirect() function and enables the information to be correctly interpreted.</p> |
| CorrectFilePaths | <p>The problem is indicated when an application tries to write files to the hard disk and is denied access or receives a file not found or path not found error message.</p> <p>The fix modifies the file path names to point to a new location on the hard disk.</p> <div data-bbox="821 1272 1433 1527" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about the CorrectFilePaths application fix, see Using the CorrectFilePaths Fix. We recommend that you use this fix together with the CorrectFilePathsUninstall fix if you are applying it to a setup installation file.</p> </div> |
| CorrectFilePathsUninstall | <p>This problem occurs when an uninstalled application leaves behind files, directories, and links.</p> <p>The fix corrects the file paths that are used by the uninstallation process of an application.</p> <div data-bbox="821 1774 1433 1998" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this fix, see Using the CorrectFilePathsUninstall Fix. We recommend that you use this fix together with the CorrectFilePaths fix if you are applying it to a setup installation file.</p> </div> |

| FIX | FIX DESCRIPTION |
|-------------------------|--|
| CorrectShellExecuteHWND | <p>This problem occurs when you start an executable (.exe) and a taskbar item blinks instead of an elevation prompt being opened, or when the application does not provide a valid HWND value when it calls the ShellExecute(Ex) function.</p> <p>The fix intercepts the ShellExecute(Ex) calls, and then inspects the HWND value. If the value is invalid, this fix enables the call to use the currently active HWND value.</p> <div data-bbox="821 470 1433 660" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about the CorrectShellExecuteHWND application fix, see Using the CorrectShellExecuteHWND Fix.</p> </div> |
| CustomNCRender | <p>This fix instructs DWM to not render the non-client area, thereby forcing the application to do its own NC rendering. This often gives windows an XP look.</p> |
| DelayApplyFlag | <p>This fix applies a KERNEL, USER, or PROCESS flag if the specified DLL is loaded.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <p>DLL_Name;Flag_Type;Hexidecimal_Value</p> <p>Where the DLL_Name is the name of the specific DLL, including the file extension. Flag_Type is KERNEL, USER, or PROCESS, and a Hexidecimal_Value, starting with 0x and up to 64-bits long.</p> <div data-bbox="821 1272 1433 1429" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>The PROCESS flag type can have a 32-bit length only. You can separate multiple entries with a backslash ().</p> </div> |

| FIX | FIX DESCRIPTION |
|------------------------------|---|
| <p>DeprecatedServiceShim</p> | <p>The problem is indicated when an application tries to install a service that has a dependency on a deprecated service. An error message displays.</p> <p>The fix intercepts the CreateService function calls and removes the deprecated dependency service from the lpDependencies parameter.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>Deprecated_Service\App_Service/Deprecated_Service2 \App_Service2</pre> <p>Where Deprecated_Service is the name of the service that has been deprecated and App_Service is the name of the specific application service that is to be modified; for example, NtLmSsp\WMI.</p> <div data-bbox="821 719 1433 909" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>If you do not provide an App_Service name, the deprecated service will be removed from all newly created services.</p> </div> <div data-bbox="821 936 1433 1093" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>You can separate multiple entries with a forward slash (/).</p> </div> |
| <p>DirectXVersionLie</p> | <p>This problem occurs when an application fails because it does not find the correct version number for DirectX®.</p> <p>The fix modifies the DXDIAGN GetProp function call to return the correct DirectX version.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>MAJORVERSION.MINORVERSION.LETTER</pre> <p>For example, 9.0.c .</p> |
| <p>DetectorDWM8And16Bit</p> | <p>This fix offers mitigation for applications that work in 8/16-bit display color mode because these legacy color modes are not supported in Windows 8 .</p> |
| <p>Disable8And16BitD3D</p> | <p>This fix improves performance of 8/16-bit color applications that render using D3D and do not mix directdraw.</p> |
| <p>Disable8And16BitModes</p> | <p>This fix disables 8/16-bit color mitigation and enumeration of 8/16-bit color modes.</p> |

| FIX | FIX DESCRIPTION |
|-------------------------|---|
| DisableDWM | <p>The problem occurs when some objects are not drawn or object artifacts remain on the screen in an application.</p> <p>The fix temporarily disables the Windows Aero menu theme functionality for unsupported applications.</p> <div data-bbox="821 342 1436 504" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the DisableDWM Fix.</p> </div> |
| DisableFadeAnimations | <p>The problem is indicated when an application fade animations, buttons, or other controls do not function properly.</p> <p>The fix disables the fade animations functionality for unsupported applications.</p> |
| DisableThemeMenus | <p>The problem is indicated by an application that behaves unpredictably when it tries to detect and use the correct Windows settings.</p> <p>The fix temporarily disables the Windows Aero menu theme functionality for unsupported applications.</p> |
| DisableWindowsDefender | <p>The fix disables Windows Defender for security applications that do not work with Windows Defender.</p> |
| DWM8And16BitMitigation | <p>The fix offers mitigation for applications that work in 8/16-bit display color mode because these legacy color modes are not supported in Windows 8.</p> |
| DXGICompat | <p>The fix allows application-specific compatibility instructions to be passed to the DirectX engine.</p> |
| DXMaximizedWindowedMode | <p>Applications that use DX8/9 are run in a maximized windowed mode. This is required for applications that use GDI/DirectDraw in addition to Direct3D.</p> |

| FIX | FIX DESCRIPTION |
|-------------------------|--|
| ElevateCreateProcess | <p>The problem is indicated when installations, de-installations, or updates fail because the host process calls the CreateProcess function and it returns an ERROR_ELEVATION_REQUIRED error message.</p> <p>The fix handles the error code and attempts to recall the CreateProcess function together with requested elevation. If the fixed application already has a UAC manifest, the error code will be returned unchanged.</p> <div data-bbox="821 472 1433 629" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the ElevateCreateProcess Fix.</p> </div> |
| EmulateOldPathIsUNC | <p>The problem occurs when an application fails because of an incorrect UNC path.</p> <p>The fix changes the PathIsUNC function to return a value of True for UNC paths in Windows.</p> |
| EmulateGetDiskFreeSpace | <p>The problem is indicated when an application fails to install or to run, and it generates an error message that there is not enough free disk space to install or use the application, even though there is enough free disk space to meet the application requirements.</p> <p>The fix determines the amount of free space, so that if the amount of free space is larger than 2 GB, the compatibility fix returns a value of 2 GB, but if the amount of free space is smaller than 2 GB, the compatibility fix returns the actual free space amount.</p> <div data-bbox="821 1283 1433 1440" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the EmulateGetDiskFreeSpace Fix.</p> </div> |
| EmulateSorting | <p>The problem occurs when an application experiences search functionality issues.</p> <p>The fix forces applications that use the CompareStringW/LCMapString sorting table to use an older version of the table.</p> <div data-bbox="821 1720 1433 1877" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this e application fix, see Using the EmulateSorting Fix.</p> </div> |
| EmulateSortingWindows61 | <p>The fix emulates the sorting order of Windows 7 and Windows Server 2008 R2 for various APIs.</p> |

| FIX | FIX DESCRIPTION |
|--------------------------|---|
| EnableRestarts | <p>The problem is indicated when an application and computer appear to hang because processes cannot end to allow the computer to complete its restart processes.</p> <p>The fix enables the computer to restart and finish the installation process by verifying and enabling that the SeShutdownPrivilege service privilege exists.</p> <div data-bbox="821 407 1433 566" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the EnableRestarts Fix.</p> </div> |
| ExtraAddRefDesktopFolder | <p>The problem occurs when an application invokes the Release() method too many times and causes an object to be prematurely destroyed.</p> <p>The fix counteracts the application's tries to obtain the shell desktop folder by invoking the AddRef() method on the Desktop folder, which is returned by the SHGetDesktopFolder function.</p> |
| FailObsoleteShellAPIs | <p>The problem occurs when an application fails because it generated deprecated API calls.</p> <p>The fix either fully implements the obsolete functions or implements the obsolete functions with stubs that fail.</p> <div data-bbox="821 1124 1433 1314" style="border: 1px solid black; padding: 5px;"> <p>Note
You can type FailAll=1 at the command prompt to suppress the function implementation and force all functions to fail.</p> </div> |
| FailRemoveDirectory | <p>The problem occurs when an application uninstallation process does not remove all of the application files and folders.</p> <p>This fix fails calls to RemoveDirectory() when called with a path matching the one specified in the shim command-line. Only a single path is supported. The path can contain environment variables, but must be an exact path – no partial paths are supported.</p> <p>The fix can resolve an issue where an application expects RemoveDirectory() to delete a folder immediately even though a handle is open to it.</p> |

| FIX | FIX DESCRIPTION |
|------------------------------|--|
| FakeLunaTheme | <p>The problem occurs when a theme application does not properly display: the colors are washed out or the user interface is not detailed.</p> <p>The fix intercepts the GetCurrentThemeName API and returns the value for the Windows XP default theme, (Luna).</p> <div data-bbox="821 407 1433 600" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about the FakeLunaTheme application fix, see Using the FakeLunaTheme Fix.</p> </div> |
| FlushFile | <p>This problem is indicated when a file is updated and changes do not immediately appear on the hard disk. Applications cannot see the file changes.</p> <p>The fix enables the WriteFile function to call to the FlushFileBuffers APIs, which flush the file cache onto the hard disk.</p> |
| FontMigration | <p>The fix replaces an application-requested font with a better font selection, to avoid text truncation.</p> |
| ForceAdminAccess | <p>The problem occurs when an application fails to function during an explicit administrator check.</p> <p>The fix allows the user to temporarily imitate being a part of the Administrators group by returning a value of True during the administrator check.</p> <div data-bbox="821 1290 1433 1451" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the ForceAdminAccess Fix.</p> </div> |
| ForceInvalidateOnClose | <p>The fix invalidates any windows that exist under a closing or hiding window for applications that rely on the invalidation messages.</p> |
| ForceLoadMirrorDrvMitigation | <p>The fix loads the Windows 8 mirror driver mitigation for applications where the mitigation is not automatically applied.</p> |
| FreestyleBMX | <p>The fix resolves an application race condition that is related to window message order.</p> |

| FIX | FIX DESCRIPTION |
|-------------------------------|--|
| GetDriveTypeWHook | <p>The application presents unusual behavior during installation; for example, the setup program states that it cannot install to a user-specified location.</p> <p>The fix changes GetDriveType() so that only the root information appears for the file path. This is required when an application passes an incomplete or badly-formed file path when it tries to retrieve the drive type on which the file path exists.</p> |
| GlobalMemoryStatusLie | <p>The problem is indicated by a Computer memory full error message that displays when you start an application.</p> <p>The fix modifies the memory status structure, so that it reports a swap file that is 400 MB, regardless of the true swap file size.</p> |
| HandleBadPtr | <p>The problem is indicated by an access violation error message that displays because an API is performing pointer validation before it uses a parameter.</p> <p>The fix supports using lpBuffer validation from the InternetSetOptionA and InternetSetOptionW functions to perform the additional parameter validation.</p> |
| HandleMarkedContentNotIndexed | <p>The problem is indicated by an application that fails when it changes an attribute on a file or directory.</p> <p>The fix intercepts any API calls that return file attributes and directories that are invoked from the %TEMP% directory, and resets the FILE_ATTRIBUTE_NOT_CONTENT_INDEXED attribute to its original state.</p> |
| HeapClearAllocation | <p>The problem is indicated when the allocation process shuts down unexpectedly.</p> <p>The fix uses zeros to clear out the heap allocation for an application.</p> |
| IgnoreAltTab | <p>The problem occurs when an application fails to function when special key combinations are used.</p> <p>The fix intercepts the RegisterRawInputDevices API and prevents the delivery of the WM_INPUT messages. This delivery failure forces the included hooks to be ignored and forces DInput to use Windows-specific hooks.</p> <div data-bbox="821 1818 1433 1977" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the IgnoreAltTab Fix.</p> </div> |
| IgnoreChromeSandbox | <p>The fix allows Google Chrome to run on systems that have ntdll loaded above 4GB.</p> |

| FIX | FIX DESCRIPTION |
|------------------------------------|--|
| IgnoreDirectoryJunction | <p>The problem is indicated by a read or access violation error message that displays when an application tries to find or open files.</p> <p>The fix links the FindNextFileW, FindNextFileA, FindFirstFileExW, FindFirstFileExA, FindFirstFileW and FindFirstFileA APIs to prevent them from returning directory junctions.</p> <div data-bbox="821 439 1437 566" style="border: 1px solid black; padding: 5px;"> <p>Note
Symbolic links appear starting in Windows Vista.</p> </div> |
| IgnoreException | <p>The problem is indicated when an application stops functioning immediately after it starts, or the application starts with only a cursor appearing on the screen.</p> <p>The fix enables the application to ignore specified exceptions. By default, this fix ignores privileged-mode exceptions; however, it can be configured to ignore any exception.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <p>Exception1;Exception2</p> <p>Where Exception1 and Exception2 are specific exceptions to be ignored. For example:
ACCESS_VIOLATION_READ:1;ACCESS_VIOLATION_WRITE:1.</p> <div data-bbox="821 1184 1437 1406" style="border: 1px solid black; padding: 5px;"> <p>Important
You should use this compatibility fix only if you are certain that it is acceptable to ignore the exception. You might experience additional compatibility issues if you choose to incorrectly ignore an exception.</p> </div> <div data-bbox="821 1435 1437 1592" style="border: 1px solid black; padding: 5px;"> <p>Note
For more detailed information about this application fix, see Using the IgnoreException Fix.</p> </div> |
| IgnoreFloatingPointRoundingControl | <p>This fix enables an application to ignore the rounding control request and to behave as expected in previous versions of the application.</p> <p>Before floating point SSE2 support in the C runtime library, the rounding control request was being ignored which would use round to nearest option by default. This shim ignores the rounding control request to support applications relying on old behavior.</p> |

| FIX | FIX DESCRIPTION |
|---------------------|---|
| IgnoreFontQuality | <p>The problem occurs when application text appears to be distorted.</p> <p>The fix enables color-keyed fonts to properly work with anti-aliasing.</p> |
| IgnoreMessageBox | <p>The problem is indicated by a message box that displays with debugging or extraneous content when the application runs on an unexpected operating system.</p> <p>The fix intercepts the MessageBox* APIs and inspects them for specific message text. If matching text is found, the application continues without showing the message box.</p> <div data-bbox="821 658 1434 815" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the IgnoreMessageBox Fix.</p> </div> |
| IgnoreMSOXMLMF | <p>The problem is indicated by an error message that states that the operating system cannot locate the MSVCR80D.DLL file.</p> <p>The fix ignores the registered MSOXMLMF.DLL object, which Microsoft® Office 2007 loads into the operating system any time that you load an XML file, and then it fails the CoGetObject for its CLSID. This compatibility fix will just ignore the registered MSOXMLMF and fail the CoGetObject for its CLSID.</p> |
| IgnoreSetROP2 | <p>The fix ignores read-modify-write operations on the desktop to avoid performance issues.</p> |
| InstallComponent | <p>The fix prompts the user to install .Net 3.5 or .Net 2.0 because .Net is not included with Windows 8.</p> |
| LoadLibraryRedirect | <p>The fix forces an application to load system versions of libraries instead of loading redistributable versions that shipped with the application.</p> |
| LocalMappedObject | <p>The problem occurs when an application unsuccessfully tries to create an object in the Global namespace.</p> <p>The fix intercepts the function call to create the object and replaces the word Global with Local.</p> <div data-bbox="821 1877 1434 2033" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the LocalMappedObject Fix.</p> </div> |

| FIX | FIX DESCRIPTION |
|-----------------------------|--|
| MakeShortcutRunas | <p>The problem is indicated when an application fails to uninstall because of access-related errors.</p> <p>The fix locates any RunDLL.exe-based uninstallers and forces them to run with different credentials during the application installation. After it applies this fix, the installer will create a shortcut that specifies a matching string to run during the application installation, thereby enabling the uninstallation to occur later.</p> <div data-bbox="821 470 1433 629" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the MakeShortcutRunas Fix</p> </div> |
| ManageLinks | <p>The fix intercepts common APIs that are going to a directory or to an executable (.exe) file, and then converts any symbolic or directory junctions before passing it back to the original APIs.</p> |
| MirrorDriverWithComposition | <p>The fix allows mirror drivers to work properly with acceptable performance with desktop composition.</p> |
| MoveToCopyFileShim | <p>The problem occurs when an application experiences security access issues during setup.</p> <p>The fix forces the CopyFile APIs to run instead of the MoveFile APIs. CopyFile APIs avoid moving the security descriptor, which enables the application files to get the default descriptor of the destination folder and prevents the security access issue.</p> |
| OpenDirectoryAd | <p>The problem is indicated by an error message that states that you do not have the appropriate permissions to access the application.</p> <p>The fix reduces the security privilege levels on a specified set of files and folders.</p> <div data-bbox="821 1554 1433 1713" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the OpenDirectoryACL Fix.</p> </div> |
| PopCapGamesForceResPerf | <p>The fix resolves the performance issues in PopCap games like Bejeweled2. The performance issues are visible in certain low-end cards at certain resolutions where the 1024x768 buffer is scaled to fit the display resolution.</p> |
| PreInstallDriver | <p>The fix preinstalls drivers for applications that would otherwise try to install or start drivers during the initial start process.</p> |

| FIX | FIX DESCRIPTION |
|-------------------------|--|
| PreInstallSmarteSECURE | <p>The fix preinstalls computer-wide CLSIDs for applications that use SmartSECURE copy protection, which would otherwise try to install the CLSIDs during the initial start process.</p> |
| ProcessPerfData | <p>The problem is indicated by an Unhandled Exception error message because the application tried to read the process performance data registry value to determine if another instance of the application is running.</p> <p>The fix handles the failure case by passing a fake process performance data registry key, so that the application perceives that it is the only instance running.</p> <div data-bbox="821 638 1433 795" style="border: 1px solid black; padding: 5px;"> <p>Note
This issue seems to occur most frequently with .NET applications.</p> </div> |
| PromoteDAM | <p>The fix registers an application for power state change notifications.</p> |
| PropagateProcessHistory | <p>The problem occurs when an application incorrectly fails to apply an application fix.</p> <p>The fix sets the _PROCESS_HISTORY environment variable so that child processes can look in the parent directory for matching information while searching for application fixes.</p> |
| ProtectedAdminCheck | <p>The problem occurs when an application fails to run because of incorrect Protected Administrator permissions.</p> <p>The fix addresses the issues that occur when applications use non-standard Administrator checks, thereby generating false positives for user accounts that are being run as Protected Administrators. In this case, the associated SID exists, but it is set as deny-only.</p> |
| RedirectCRTTempFile | <p>The fix intercepts failing CRT calls that try to create a temporary file at the root of the volume, thereby redirecting the calls to a temporary file in the user's temporary directory.</p> |
| RedirectHKCUKeys | <p>The problem occurs when an application cannot be accessed because of User Account Control (UAC) restrictions.</p> <p>The fix duplicates any newly created HKCU keys to other users' HKCU accounts. This fix is generic for UAC restrictions, whereby the HKCU keys are required, but are unavailable to an application at runtime.</p> |

| FIX | FIX DESCRIPTION |
|------------------|--|
| RedirectMP3Codec | <p>This problem occurs when you cannot play MP3 files.</p> <p>The fix intercepts the CoCreateInstance call for the missing filter and then redirects it to a supported version.</p> |
| RedirectShortcut | <p>The problem occurs when an application cannot be accessed by its shortcut, or application shortcuts are not removed during the application uninstallation process.</p> <p>The fix redirects all of the shortcuts created during the application setup to appear according to a specified path.</p> <ul style="list-style-type: none"> • Start Menu shortcuts: Appear in the \ProgramData\Microsoft\Windows\Start Menu directory for all users. • Desktop or Quick Launch shortcuts: You must manually place the shortcuts on the individual user's desktop or Quick Launch bar. <p>This issue occurs because of UAC restrictions: specifically, when an application setup runs by using elevated privileges and stores the shortcuts according to the elevated user's context. In this situation, a restricted user cannot access the shortcuts.</p> <p>You cannot apply this fix to an .exe file that includes a manifest and provides a runlevel.</p> |
| RelaunchElevated | <p>The problem occurs when installers, uninstallers, or updaters fail when they are started from a host application.</p> <p>The fix enables a child .exe file to run with elevated privileges when it is difficult to determine the parent process with either the ElevateCreateProcess fix or by marking the .exe files to RunAsAdmin.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RelaunchElevated Fix.</p> </div> |

| FIX | FIX DESCRIPTION |
|---|---|
| <p>RetryOpenSCManagerWithReadAccess</p> | <p>The problem occurs when an application tries to open the Service Control Manager (SCM) and receives an Access Denied error message.</p> <p>The fix retries the call and requests a more restricted set of rights that include the following:</p> <ul style="list-style-type: none"> • SC_MANAGER_CONNECT • SC_MANAGER_ENUMERATE_SERVICE • SC_MANAGER_QUERY_LOCK_STATUS • STANDARD_READ_RIGHTS <div data-bbox="853 577 1396 770" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RetryOpenSCManagerwithReadAccess Fix.</p> </div> |
| <p>RetryOpenServiceWithReadAccess</p> | <p>The problem occurs when an Unable to open service due to your application using the OpenService() API to test for the existence of a particular service error message displays.</p> <p>The fix retries the OpenService() API call and verifies that the user has Administrator rights, is not a Protected Administrator, and by using read-only access. Applications can test for the existence of a service by calling the OpenService() API but some applications ask for all access when making this check. This fix retries the call but only asking for read-only access. The user needs to be an administrator for this to work</p> <div data-bbox="821 1267 1434 1429" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RetryOpenServiceWithReadAccess Fix.</p> </div> |
| <p>RunAsAdmin</p> | <p>The problem occurs when an application fails to function by using the Standard User or Protected Administrator account.</p> <p>The fix enables the application to run by using elevated privileges. The fix is the equivalent of specifying requireAdministrator in an application manifest.</p> <div data-bbox="821 1733 1434 1895" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RunAsAdmin Fix.</p> </div> |

| FIX | FIX DESCRIPTION |
|--------------|--|
| RunAsHighest | <p>The problem occurs when administrators cannot view the read/write version of an application that presents a read-only view to standard users.</p> <p>The fix enables the application to run by using the highest available permissions. This is the equivalent of specifying highestAvailable in an application manifest.</p> <div data-bbox="821 407 1433 566" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RunAsHighest Fix.</p> </div> |
| RunAsInvoker | <p>The problem occurs when an application is not detected as requiring elevation.</p> <p>The fix enables the application to run by using the privileges that are associated with the creation process, without requiring elevation. This is the equivalent of specifying asInvoker in an application manifest.</p> <div data-bbox="821 875 1433 1034" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the RunAsInvoker Fix.</p> </div> |
| SecuROM7 | <p>The fix repairs applications by using SecuROM7 for copy protection.</p> |
| SessionShim | <p>The fix intercepts API calls from applications that are trying to interact with services that are running in another session, by using the terminal service name prefix (Global or Local) as the parameter.</p> <p>At the command prompt, you can supply a list of objects to modify, separating the values by a double backslash (). Or, you can choose not to include any parameters, so that all of the objects are modified.</p> <div data-bbox="821 1541 1433 1731" style="border: 1px solid black; padding: 5px;"> <p>Important</p> <p>Users cannot log in as Session 0 (Global Session) in Windows Vista and later. Therefore, applications that require access to Session 0 automatically fail.</p> </div> <div data-bbox="821 1753 1433 1912" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the SessionShim Fix.</p> </div> |

| FIX | FIX DESCRIPTION |
|-------------------------------|--|
| SetProtocolHandler | <p>The fix registers an application as a protocol handler.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>Client;Protocol;App</pre> <p>Where the Client is the name of the email protocol, Protocol is mailto, and App is the name of the application.</p> <div data-bbox="821 448 1433 638" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>Only the mail client and the mailto protocol are supported. You can separate multiple clients by using a backslash ().</p> </div> |
| SetupCommitFileQueueIgnoreWow | <p>The problem occurs when a 32-bit setup program fails to install because it requires 64-bit drivers.</p> <p>The fix disables the Wow64 file system that is used by the 64-bit editions of Windows, to prevent 32-bit applications from accessing 64-bit file systems during the application setup.</p> |
| SharePointDesigner2007 | <p>The fix resolves an application bug that severely slows the application when it runs in DWM.</p> |
| ShimViaEAT | <p>The problem occurs when an application fails, even after applying a compatibility fix that is known to fix an issue. Applications that use unicows.dll or copy protection often present this issue.</p> <p>The fix applies the specified compatibility fixes by modifying the export table and by nullifying the use of module inclusion and exclusion.</p> <div data-bbox="821 1393 1433 1552" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more information about this application fix, see Using the ShimViaEAT Fix.</p> </div> |
| ShowWindowIE | <p>The problem occurs when a web application experiences navigation and display issues because of the tabbing feature.</p> <p>The fix intercepts the ShowWindow API call to address the issues that can occur when a web application determines that it is in a child window. This fix calls the real ShowWindow API on the top-level parent window.</p> |
| SierraWirelessHideCDROM | <p>The fix repairs the Sierra Wireless Driver installation, thereby preventing bugcheck.</p> |

| FIX | FIX DESCRIPTION |
|------------------------|---|
| Sonique2 | <p>The application uses an invalid window style, which breaks in DWM. This fix replaces the window style with a valid value.</p> |
| SpecificInstaller | <p>The problem occurs when an application installation file fails to be picked up by the GenericInstaller function.</p> <p>The fix flags the application as being an installer file (for example, setup.exe), and then prompts for elevation.</p> <div data-bbox="821 510 1434 669" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the SpecificInstaller Fix.</p> </div> |
| SpecificNonInstaller | <p>The problem occurs when an application that is not an installer (and has sufficient privileges) generates a false positive from the GenericInstaller function.</p> <p>The fix flags the application to exclude it from detection by the GenericInstaller function.</p> <div data-bbox="821 943 1434 1102" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the SpecificNonInstaller Fix.</p> </div> |
| SystemMetricsLie | <p>The fix replaces SystemMetrics values and SystemParametersInfo values with the values of previous Windows versions.</p> |
| TextArt | <p>The application receives different mouse coordinates with DWM ON versus DWM OFF, which causes the application to hang. This fix resolves the issue.</p> |
| TrimDisplayDeviceNames | <p>The fix trims the names of the display devices that are returned by the EnumDisplayDevices API.</p> |
| UIPICompatLogging | <p>The fix enables the logging of Windows messages from Internet Explorer and other processes.</p> |

| FIX | FIX DESCRIPTION |
|---------------------------|--|
| UIPIEnableCustomMsgs | <p>The problem occurs when an application does not properly communicate with other processes because customized Windows messages are not delivered.</p> <p>The fix enables customized Windows messages to pass through to the current process from a lower Desktop integrity level. This fix is the equivalent of calling the RegisterWindowMessage function, followed by the ChangeWindowMessageFilter function in the code.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>MessageString1 MessageString2</pre> <p>Where MessageString1 and MessageString2 reflect the message strings that can pass.</p> <div data-bbox="821 689 1434 878" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>Multiple message strings must be separated by spaces. For more detailed information about this application fix, see Using the UIPIEnableCustomMsgs Fix.</p> </div> |
| UIPIEnableStandardMsgs | <p>The problem occurs when an application does not communicate properly with other processes because standard Windows messages are not delivered.</p> <p>The fix enables standard Windows messages to pass through to the current process from a lower Desktop integrity level. This fix is the equivalent of calling the ChangeWindowMessageFilter function in the code.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>1055 1056 1069</pre> <p>Where 1055 reflects the first message ID, 1056 reflects the second message ID, and 1069 reflects the third message ID that can pass.</p> <div data-bbox="821 1467 1434 1657" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>Multiple messages can be separated by spaces. For more detailed information about this application fix, see Using the UIPIEnableStandardMsgs Fix [act].</p> </div> |
| VirtualizeDeleteFileLayer | <p>The fix virtualizes DeleteFile operations for applications that try to delete protected files.</p> |
| VirtualizeDesktopPainting | <p>This fix improves the performance of a number of operations on the Desktop DC while using DWM.</p> |

| FIX | FIX DESCRIPTION |
|---------------------------|---|
| VirtualRegistry | <p>The problem is indicated when a Component failed to be located error message displays when an application is started.</p> <p>The fix enables the registry functions to allow for virtualization, redirection, expansion values, version spoofing, the simulation of performance data counters, and so on.</p> <p>For more detailed information about this application fix, see Using the VirtualRegistry Fix.</p> |
| VirtualizeDeleteFile | <p>The problem occurs when several error messages display and the application cannot delete files.</p> <p>The fix makes the application's DeleteFile function call a virtual call in an effort to remedy the UAC and file virtualization issues that were introduced with Windows Vista. This fix also links other file APIs (for example, GetFileAttributes) to ensure that the virtualization of the file is deleted.</p> <div data-bbox="821 869 1433 1025" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the VirtualizeDeleteFile Fix.</p> </div> |
| VirtualizeHKCRLite | <p>The problem occurs when an application fails to register COM components at runtime.</p> <p>The fix redirects the HKCR write calls (HKLM) to the HKCU hive for a per-user COM registration. This operates much like the VirtualRegistry fix when you use the VirtualizeHKCR parameter; however, VirtualizeHKCRLite provides better performance.</p> <p>HKCR is a virtual merge of the HKCU\Software\Classes and HKLM\Software\Classes directories. The use of HKCU is preferred if an application is not elevated and is ignored if the application is elevated.</p> <p>You typically will use this compatibility fix in conjunction with the VirtualizeRegisterTypeLib fix.</p> <p>For more detailed information about this application fix, see Using the VirtualizeHKCRLite Fix.</p> |
| VirtualizeRegisterTypeLib | <p>The fix, when it is used with the VirtualizeHKCRLite fix, ensures that the type library and the COM class registration happen simultaneously. This functions much like the RegistryTypeLib fix when the RegisterTypeLibForUser parameter is used.</p> <div data-bbox="821 1908 1433 2065" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>For more detailed information about this application fix, see Using the VirtualizeRegisterTypelib Fix.</p> </div> |

| FIX | FIX DESCRIPTION |
|---------------------------|--|
| WaveOutIgnoreBadFormat | <p>This problem is indicated by an error message that states: Unable to initialize sound device from your audio driver; the application then closes.</p> <p>The fix enables the application to ignore the format error and continue to function properly.</p> |
| WerDisableReportException | <p>The fix turns off the silent reporting of exceptions to the Windows Error Reporting tool, including those that are reported by Object Linking and Embedding-Database (OLE DB). The fix intercepts the RtlReportException API and returns a STATUS_NOT_SUPPORTED error message.</p> |
| Win7RTM/Win8RTM | <p>The layer provides the application with Windows 7/Windows 8 compatibility mode.</p> |
| WinxxRTMVersionLie | <p>The problem occurs when an application fails because it does not find the correct version number for the required Windows operating system.</p> <p>All version lie compatibility fixes address the issue whereby an application fails to function because it is checking for, but not finding, a specific version of the operating system. The version lie fix returns the appropriate operating system version information. For example, the VistaRTMVersionLie returns the Windows Vista version information to the application, regardless of the actual operating system version that is running on the computer.</p> |
| Wing32SystoSys32 | <p>The problem is indicated by an error message that states that the WinG library was not properly installed.</p> <p>The fix detects whether the WinG32 library exists in the correct directory. If the library is located in the wrong location, this fix copies the information (typically during the runtime of the application) into the %WINDIR%\system32 directory.</p> <div data-bbox="821 1496 1433 1659" style="border: 1px solid black; padding: 5px;"> <p>Important</p> <p>The application must have Administrator privileges for this fix to work.</p> </div> |
| WinSrv08R2RTM | |

| FIX | FIX DESCRIPTION |
|--------------------|--|
| WinXPSP2VersionLie | <p>The problem occurs when an application experiences issues because of a VB runtime DLL.</p> <p>The fix forces the application to follow these steps:</p> <ol style="list-style-type: none">1. Open the Compatibility Administrator, and then select None for Operating System Mode.2. On the Compatibility Fixes page, click WinXPSP2VersionLie, and then click Parameters. <p>The Options for <fix_name> dialog box appears.</p> <ol style="list-style-type: none">3. Type vbrun60.dll into the Module Name box, click Include, and then click Add.4. Save the custom database. <div data-bbox="853 694 1396 884" style="border: 1px solid black; padding: 5px;"><p>Note</p><p>For more information about the WinXPSP2VersionLie application fix, see Using the WinXPSP2VersionLie Fix.</p></div> |
| WRPDllRegister | <p>The application fails when it tries to register a COM component that is released together with Windows Vista and later.</p> <p>The fix skips the processes of registering and unregistering WRP-protected COM components when calling the DLLRegisterServer and DLLUnregisterServer functions.</p> <p>You can control this fix further by typing the following command at the command prompt:</p> <pre>Component1.dll;Component2.dll</pre> <p>Where Component1.dll and Component2.dll reflect the components to be skipped.</p> <div data-bbox="821 1411 1428 1568" style="border: 1px solid black; padding: 5px;"><p>Note</p><p>For more detailed information about this application fix, see Using the WRPDllRegister Fix.</p></div> |
| WRPMitigation | <p>The problem is indicated when an access denied error message displays when the application tries to access a protected operating system resource by using more than read-only access.</p> <p>The fix emulates the successful authentication and modification of file and registry APIs, so that the application can continue.</p> <div data-bbox="821 1904 1428 2060" style="border: 1px solid black; padding: 5px;"><p>Note</p><p>For more detailed information about WRPMitigation, see Using the WRPMitigation Fix.</p></div> |

| FIX | FIX DESCRIPTION |
|---------------------|--|
| WRPRegDeleteKey | <p>The problem is indicated by an access denied error message that displays when the application tries to delete a registry key.</p> <p>The fix verifies whether the registry key is WRP-protected. If the key is protected, this fix emulates the deletion process.</p> |
| XPAfxIsValidAddress | <p>The fix emulates the behavior of Windows XP for MFC42!AfxIsValidAddress.</p> |

Compatibility Modes

The following table lists the known compatibility modes.

| COMPATIBILITY MODE NAME | DESCRIPTION | INCLUDED COMPATIBILITY FIXES |
|-------------------------|--|---|
| WinSrv03 | Emulates the Windows Server 2003 operating system. | <ul style="list-style-type: none"> • Win2k3RTMVersionLie • VirtualRegistry • ElevateCreateProcess • EmulateSorting • FailObsoleteShellAPIs • LoadLibraryCWD • HandleBadPtr • GlobalMemoryStatus2GB • RedirectMP3Codec • EnableLegacyExceptionHandlinginOLE • NoGhost • HardwareAudioMixer |
| WinSrv03Sp1 | Emulates the Windows Server 2003 with Service Pack 1 (SP1) operating system. | <ul style="list-style-type: none"> • Win2K3SP1VersionLie • VirtualRegistry • ElevateCreateProcess • EmulateSorting • FailObsoleteShellAPIs • LoadLibraryCWD • HandleBadPtr • EnableLegacyExceptionHandlinginOLE • RedirectMP3Codec • HardwareAudioMixer |

Deploy Windows 10 with the Microsoft Deployment Toolkit

6/14/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This guide will walk you through the process of deploying Windows 10 in an enterprise environment using the Microsoft Deployment Toolkit (MDT).

The Microsoft Deployment Toolkit is a unified collection of tools, processes, and guidance for automating desktop and server deployment. In addition to reducing deployment time and standardizing desktop and server images, MDT enables you to more easily manage security and ongoing configurations. MDT builds on top of the core deployment tools in the Windows Assessment and Deployment Kit (Windows ADK) with additional guidance and features designed to reduce the complexity and time required for deployment in an enterprise environment. MDT supports the deployment of Windows 10, as well as Windows 7, Windows 8, Windows 8.1, and Windows Server 2012 R2. It also includes support for zero-touch installation (ZTI) with Microsoft System Center 2012 R2 Configuration Manager.

To download the latest version of MDT, visit the [MDT resource page](#).

In this section

- [Get started with the Microsoft Deployment Toolkit \(MDT\)](#)
- [Create a Windows 10 reference image](#)
- [Deploy a Windows 10 image using MDT](#)
- [Build a distributed environment for Windows 10 deployment](#)
- [Refresh a Windows 7 computer with Windows 10](#)
- [Replace a Windows 7 computer with a Windows 10 computer](#)
- [Configure MDT settings](#)

Proof-of-concept environment

For the purposes of this guide, and the topics discussed herein, we will use the following servers and client machines: DC01, MDT01, CM01, PC0001, and PC0002.



Figure 1. The servers and machines used for examples in this guide.

DC01 is a domain controller; the other servers and client machines are members of the domain contoso.com for the fictitious Contoso Corporation.

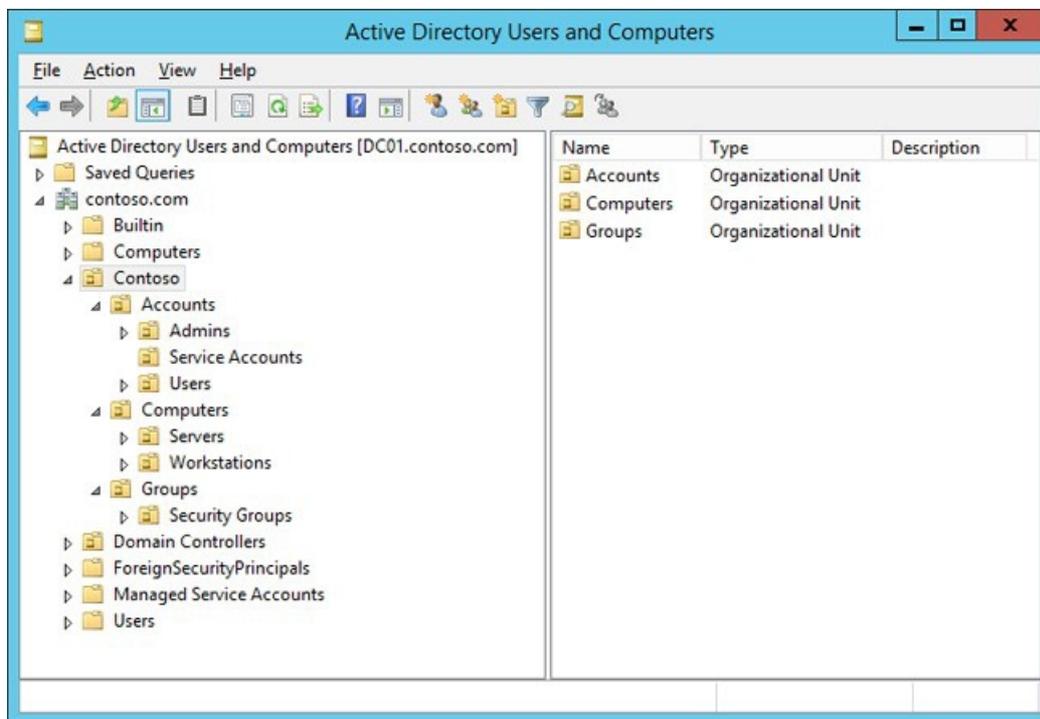


Figure 2. The organizational unit (OU) structure used in this guide.

Server details

- **DC01.** A Windows Server 2012 R2 Standard machine, fully patched with the latest security updates, and configured as Active Directory Domain Controller, DNS Server, and DHCP Server in the contoso.com domain.
 - Server name: DC01
 - IP Address: 192.168.1.200
 - Roles: DNS, DHCP, and Domain Controller
- **MDT01.** A Windows Server 2012 R2 Standard machine, fully patched with the latest security updates, and configured as a member server in the contoso.com domain.
 - Server name: MDT01
 - IP Address: 192.168.1.210
- **CM01.** A Windows Server 2012 R2 Standard machine, fully patched with the latest security updates, and configured as a member server in the contoso.com domain.
 - Server name: CM01
 - IP Address: 192.168.1.214

Client machine details

- **PC0001.** A Windows 10 Enterprise x64 machine, fully patched with the latest security updates, and configured as a member in the contoso.com domain. This machine is referenced as the admin workstation.
 - Client name: PC0001
 - IP Address: DHCP
- **PC0002.** A Windows 7 SP1 Enterprise x64 machine, fully patched with the latest security updates, and configured as a member in the contoso.com domain. This machine is referenced during the migration scenarios.
 - Client name: PC0002
 - IP Address: DHCP

Sample files

The information in this guide is designed to help you deploy Windows 10. In order to help you put the

information you learn into practice more quickly, we recommend that you download a small set of sample files for the fictitious Contoso Corporation:

- [Gather.ps1](#). This sample Windows PowerShell script performs the MDT Gather process in a simulated MDT environment. This allows you to test the MDT gather process and check to see if it is working correctly without performing a full Windows deployment.
- [Set-OUPermissions.ps1](#). This sample Windows PowerShell script creates a domain account and then configures OU permissions to allow the account to join machines to the domain in the specified OU.
- [MDTSample.zip](#). This sample web service shows you how to configure a computer name dynamically using MDT.

Related topics

[Microsoft Deployment Toolkit downloads and resources](#)

[Windows 10 deployment scenarios](#)

[Windows 10 deployment tools](#)

[Deploy Windows 10 with System Center 2012 R2 Configuration Manager](#)

[Deploy Windows To Go in your organization](#)

[Sideload apps in Windows 10](#)

[Volume Activation for Windows 10](#)

Get started with the Microsoft Deployment Toolkit (MDT)

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic will help you gain a better understanding of how to use the Microsoft Deployment Toolkit (MDT), as part of a Windows operating system deployment. MDT is one of the most important tools available to IT professionals today. You can use it to create reference images or as a complete deployment solution. MDT also can be used to extend the operating system deployment features available in Microsoft System Center 2012 R2 Configuration Manager.

In addition to familiarizing you with the features and options available in MDT, this topic will walk you through the process of preparing for deploying Windows 10 using MDT by configuring Active Directory, creating an organizational unit (OU) structure, creating service accounts, configuring log files and folders, and installing the tools needed to view the logs and continue with the deployment process.

For the purposes of this topic, we will use two machines: DC01 and MDT01. DC01 is a domain controller and MDT01 is a Windows Server 2012 R2 standard server. MDT01 is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

In this section

- [Key features in MDT](#)
- [MDT Lite Touch components](#)
- [Prepare for deployment with MDT](#)

Related topics

[Microsoft Deployment Toolkit downloads and documentation](#)

[Create a Windows 10 reference image](#)

[Deploy a Windows 10 image using MDT](#)

[Build a distributed environment for Windows 10 deployment](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

[Configure MDT settings](#)

Key features in MDT

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

The Microsoft Deployment Toolkit (MDT) has been in existence since 2003, when it was first introduced as Business Desktop Deployment (BDD) 1.0. The toolkit has evolved, both in functionality and popularity, and today it is considered fundamental to Windows operating system and enterprise application deployment.

MDT has many useful features, the most important of which are:

- **Windows Client support.** Supports Windows 7, Windows 8, Windows 8.1, and Windows 10.
- **Windows Server support.** Supports Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2.
- **Additional operating systems support.** Supports Windows Thin PC and Windows Embedded POSReady 7, as well as Windows 8.1 Embedded Industry.
- **UEFI support.** Supports deployment to machines using Unified Extensible Firmware Interface (UEFI) version 2.3.1.
- **GPT support.** Supports deployment to machines that require the new GUID (globally unique identifier) partition table (GPT) format. This is related to UEFI.
- **Enhanced Windows PowerShell support.** Provides support for running PowerShell scripts.

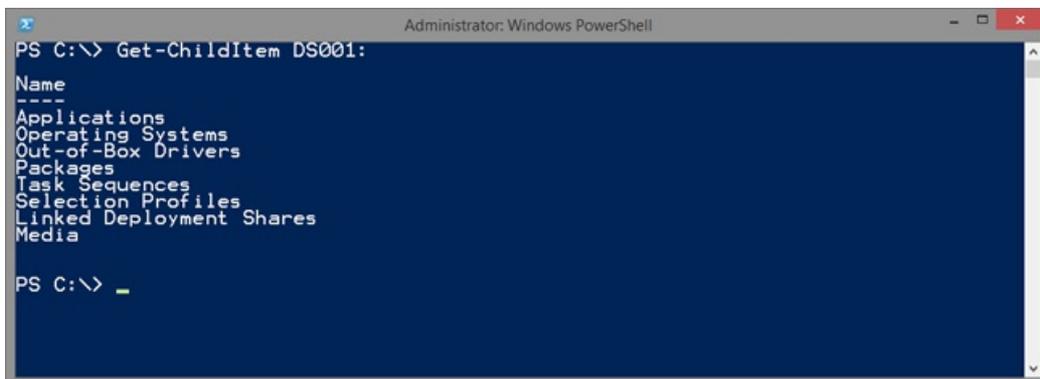


Figure 2. The deployment share mounted as a standard PSDrive allows for administration using PowerShell.

- **Add local administrator accounts.** Allows you to add multiple user accounts to the local Administrators group on the target computers, either via settings or the deployment wizard.
- **Automated participation in CEIP and WER.** Provides configuration for participation in Windows Customer Experience Improvement Program (CEIP) and Windows Error Reporting (WER).
- **Deploy Windows RE.** Enables deployment of a customized Windows Recovery Environment (Windows RE) as part of the task sequence.
- **Deploy to VHD.** Provides ready-made task sequence templates for deploying Windows into a virtual hard disk (VHD) file.
- **Improved deployment wizard.** Provides additional progress information and a cleaner UI for the Lite

Touch Deployment Wizard.

- **Monitoring.** Allows you to see the status of currently running deployments.
- **Apply GPO Pack.** Allows you to deploy local group policy objects created by Microsoft Security Compliance Manager (SCM).
- **Partitioning routines.** Provides improved partitioning routines to ensure that deployments work regardless of the current hard drive structure.
- **Offline BitLocker.** Provides the capability to have BitLocker enabled during the Windows Preinstallation Environment (Windows PE) phase, thus saving hours of encryption time.
- **USMT offline user-state migration.** Provides support for running the User State Migration Tool (USMT) capture offline, during the Windows PE phase of the deployment.

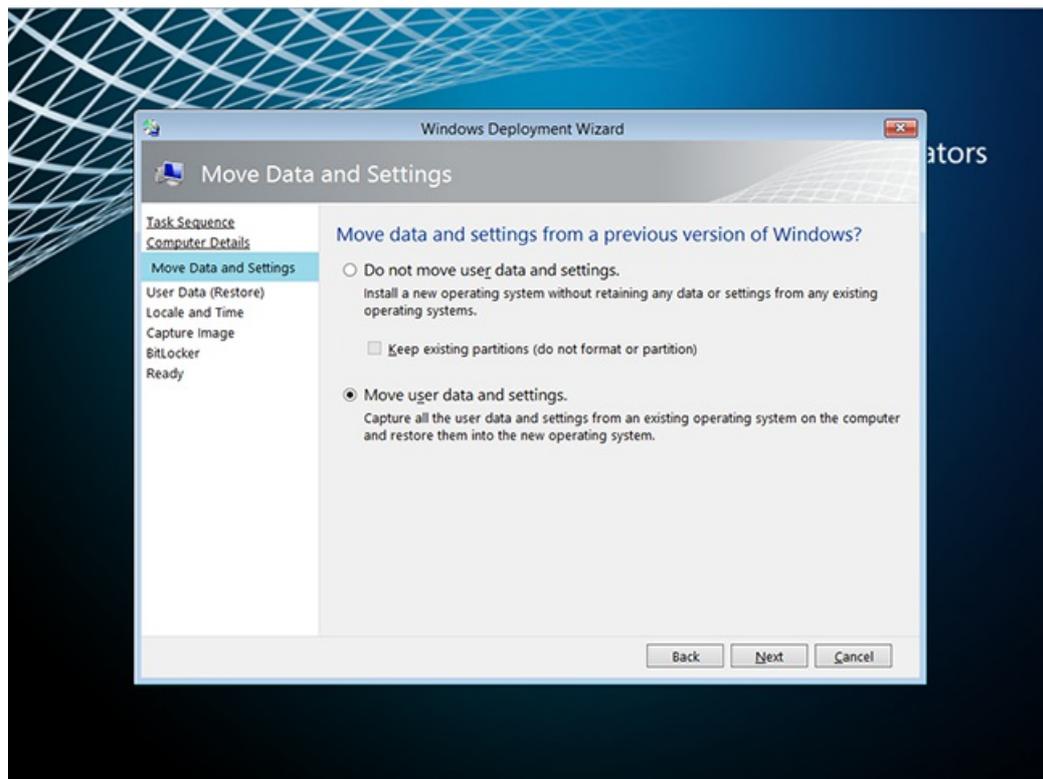


Figure 3. The offline USMT backup in action.

- **Install or uninstall Windows roles or features.** Enables you to select roles and features as part of the deployment wizard. MDT also supports uninstall of roles and features.
- **Microsoft System Center 2012 Orchestrator integration.** Provides the capability to use Orchestrator runbooks as part of the task sequence.
- **Support for DaRT.** Supports optional integration of the DaRT components into the boot image.
- **Support for Office 2013.** Provides added support for deploying Microsoft Office Professional Plus 2013.
- **Support for Modern UI app package provisioning.** Provisions applications based on the new Windows app package standard, which is used in Windows 8 and later.
- **Extensibility.** Provides the capability to extend MDT far beyond the built-in features by adding custom scripts, web services, System Center Orchestrator runbooks, PowerShell scripts, and VBScripts.
- **Upgrade task sequence.** Provides a new upgrade task sequence template that you can use to upgrade existing Windows 7, Windows 8, and Windows 8.1 systems directly to Windows 10, automatically preserving all data, settings, applications, and drivers. For more information about using this new upgrade

task sequence, refer to the [Microsoft Deployment Toolkit resource page](#).

Related topics

[Prepare for deployment with MDT](#)

[MDT Lite Touch components](#)

MDT Lite Touch components

6/14/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic provides an overview of the features in the Microsoft Deployment Toolkit (MDT) that support Lite Touch Installation (LTI) for Windows 10. An LTI deployment strategy requires very little infrastructure or user interaction, and can be used to deploy an operating system from a network share or from a physical media, such as a USB flash drive or disc. When deploying the Windows operating system using MDT, most of the administration and configuration is done through the Deployment Workbench, but you also can perform many of the tasks using Windows PowerShell. The easiest way to find out how to use PowerShell in MDT is to use the Deployment Workbench to perform an operation and at the end of that task, click View Script. That will give you the PowerShell command.

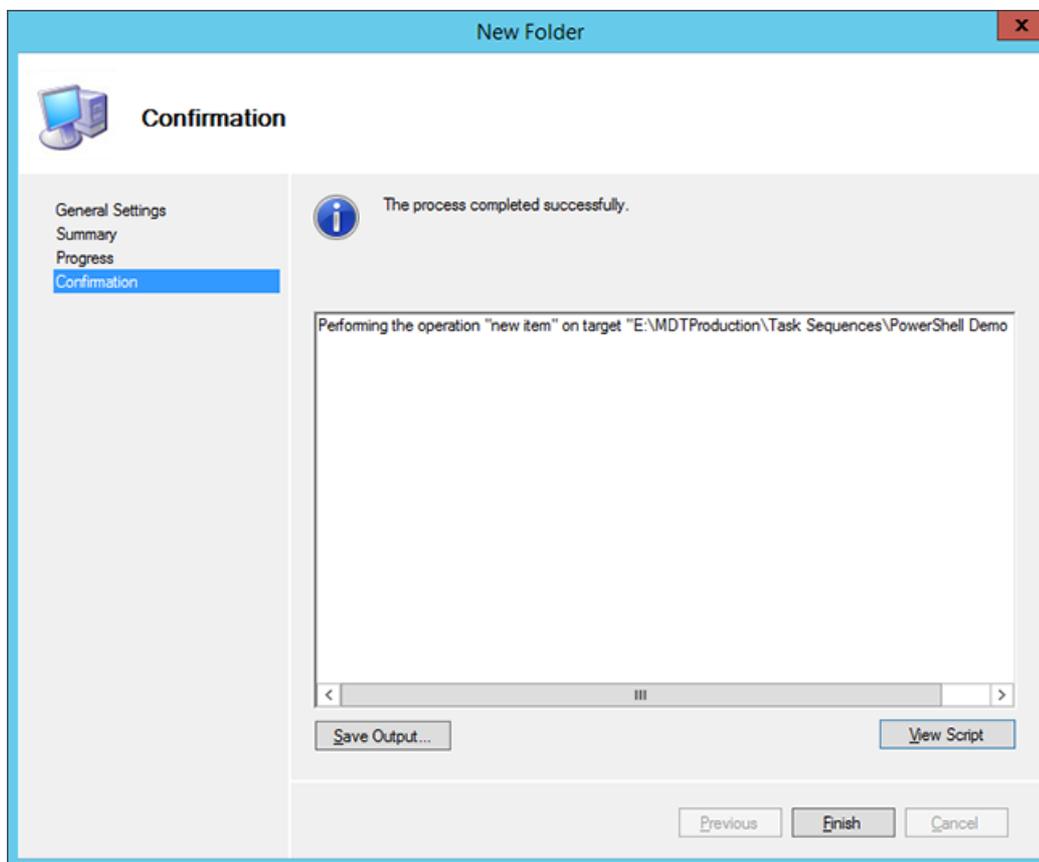


Figure 4. If you click **View Script** on the right side, you will get the PowerShell code that was used to perform the task.

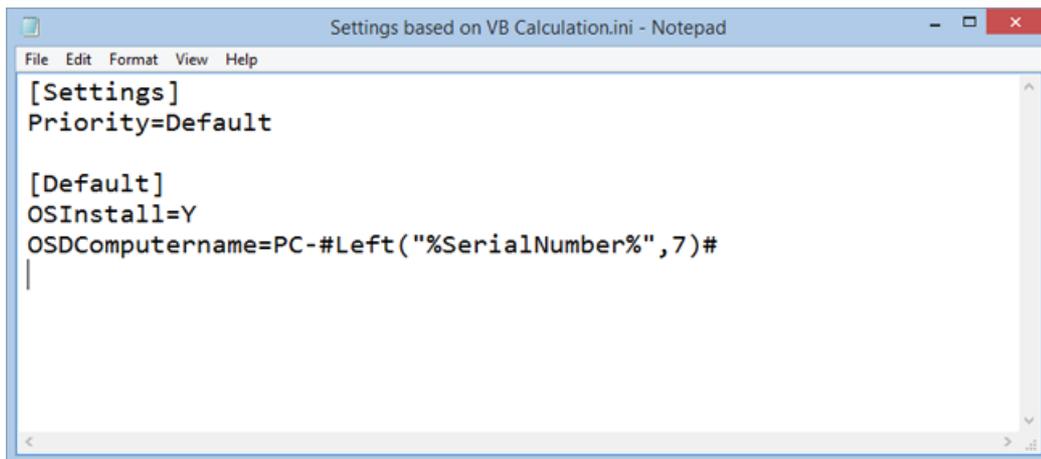
Deployment shares

A deployment share is essentially a folder on the server that is shared and contains all the setup files and scripts needed for the deployment solution. It also holds the configuration files (called rules) that are gathered when a machine is deployed. These configuration files can reach out to other sources, like a database, external script, or web server to get additional settings for the deployment. For Lite Touch deployments, it is common to have two deployment shares: one for creating the reference images and one for deployment. For Zero Touch, it is common to have only the deployment share for creating reference images because Microsoft System Center 2012 R2 Configuration Manager deploys the image in the production environment.

Rules

The rules (CustomSettings.ini and Bootstrap.ini) make up the brain of MDT. The rules control the Windows Deployment Wizard on the client and, for example, can provide the following settings to the machine being deployed:

- Computer name
- Domain to join, and organizational unit (OU) in Active Directory to hold the computer object
- Whether to enable BitLocker
- Regional settings You can manage hundreds of settings in the rules. For more information, see the [Microsoft Deployment Toolkit resource center](#).

A screenshot of a Notepad window titled "Settings based on VB Calculation.ini - Notepad". The window contains the following text:

```
[Settings]
Priority=Default

[Default]
OSInstall=Y
OSDComputername=PC-#Left("%SerialNumber%",7)#
```

Figure 5. Example of a MDT rule. In this example, the new computer name is being calculated based on PC- plus the first seven (Left) characters from the serial number

Boot images

Boot images are the Windows Preinstallation Environment (Windows PE) images that are used to start the deployment. They can be started from a CD or DVD, an ISO file, a USB device, or over the network using a Pre-Boot Execution Environment (PXE) server. The boot images connect to the deployment share on the server and start the deployment.

Operating systems

Using the Deployment Workbench, you import the operating systems you want to deploy. You can import either the full source (like the full Windows 10 DVD/ISO) or a custom image that you have created. The full-source operating systems are primarily used to create reference images; however, they also can be used for normal deployments.

Applications

Using the Deployment Workbench, you also add the applications you want to deploy. MDT supports virtually every executable Windows file type. The file can be a standard .exe file with command-line switches for an unattended install, a Microsoft Windows Installer (MSI) package, a batch file, or a VBScript. In fact, it can be just about anything that can be executed unattended. MDT also supports the new Universal Windows apps.

Driver repository

You also use the Deployment Workbench to import the drivers your hardware needs into a driver repository that lives on the server, not in the image.

Packages

With the Deployment Workbench, you can add any Microsoft packages that you want to use. The most commonly added packages are language packs, and the Deployment Workbench Packages node works well for those. You also can add security and other updates this way. However, we generally recommend that you use Windows Server Update Services (WSUS) for operating system updates. The rare exceptions are critical hotfixes that are not available via WSUS, packages for the boot image, or any other package that needs to be deployed before the WSUS update process starts.

Task sequences

Task sequences are the heart and soul of the deployment solution. When creating a task sequence, you need to select a template. The templates are located in the Templates folder in the MDT installation directory, and they determine which default actions are present in the sequence.

You can think of a task sequence as a list of actions that need to be executed in a certain order. Each action can also have conditions. Some examples of actions are as follows:

- **Gather.** Reads configuration settings from the deployment server.
- **Format and Partition.** Creates the partition(s) and formats them.
- **Inject Drivers.** Finds out which drivers the machine needs and downloads them from the central driver repository.
- **Apply Operating System.** Uses ImageX to apply the image.
- **Windows Update.** Connects to a WSUS server and updates the machine.

Task sequence templates

MDT comes with nine default task sequence templates. You can also create your own templates. As long as you store them in the Templates folder, they will be available when you create a new task sequence.

- **Sysprep and Capture task sequence.** Used to run the System Preparation (Sysprep) tool and capture an image of a reference computer.

Note It is preferable to use a complete build and capture instead of the Sysprep and Capture task sequence. A complete build and capture can be automated, whereas Sysprep and Capture cannot.

- **Standard Client task sequence.** The most frequently used task sequence. Used for creating reference images and for deploying clients in production.
- **Standard Client Replace task sequence.** Used to run User State Migration Tool (USMT) backup and the optional full Windows Imaging (WIM) backup action. Can also be used to do a secure wipe of a machine that is going to be decommissioned.
- **Custom task sequence.** As the name implies, a custom task sequence with only one default action (one Install Application action).
- **Standard Server task sequence.** The default task sequence for deploying operating system images to servers. The main difference between this template and the Standard Client task sequence template is that it does not contain any USMT actions because USMT is not supported on servers.
- **Lite Touch OEM task sequence.** Used to preload operating systems images on the computer hard drive. Typically used by computer original equipment manufacturers (OEMs) but some enterprise organizations also use this feature.
- **Post OS Installation task sequence.** A task sequence prepared to run actions after the operating system has been deployed. Very useful for server deployments but not often used for client deployments.

- **Deploy to VHD Client task sequence.** Similar to the Standard Client task sequence template but also creates a virtual hard disk (VHD) file on the target computer and deploys the image to the VHD file.
- **Deploy to VHD Server task sequence.** Same as the Deploy to VHD Client task sequence but for servers.
- **Standard Client Upgrade task sequence.** A simple task sequence template used to perform an in-place upgrade from Windows 7, Windows 8, or Windows 8.1 directly to Windows 10, automatically preserving existing data, settings, applications, and drivers.

Selection profiles

Selection profiles, which are available in the Advanced Configuration node, provide a way to filter content in the Deployment Workbench. Selection profiles are used for several purposes in the Deployment Workbench and in Lite Touch deployments. For example, they can be used to:

- Control which drivers and packages are injected into the Lite Touch (and generic) boot images.
- Control which drivers are injected during the task sequence.
- Control what is included in any media that you create.
- Control what is replicated to other deployment shares.
- Filter which task sequences and applications are displayed in the Deployment Wizard.

Logging

MDT uses many log files during operating system deployments. By default the logs are client side, but by configuring the deployment settings, you can have MDT store them on the server, as well.

Note The easiest way to view log files is to use Configuration Manager Trace (CMTrace), which is included in the [System Center 2012 R2 Configuration Manager Toolkit](#).

Monitoring

On the deployment share, you also can enable monitoring. After you enable monitoring, you will see all running deployments in the Monitor node in the Deployment Workbench.

Related topics

[Key features in MDT](#)

[Prepare for deployment with MDT](#)

Prepare for deployment with MDT

6/14/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic will walk you through the steps necessary to create the server structure required to deploy the Windows 10 operating system using the Microsoft Deployment Toolkit (MDT). It covers the installation of the necessary system prerequisites, the creation of shared folders and service accounts, and the configuration of security permissions in the files system and in Active Directory.

For the purposes of this topic, we will use two machines: DC01 and MDT01. DC01 is a domain controller and MDT01 is a Windows Server 2012 R2 standard server. MDT01 is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

System requirements

MDT requires the following components:

- Any of the following operating systems:
 - Windows 7
 - Windows 8
 - Windows 8.1
 - Windows 10
 - Windows Server 2008 R2
 - Windows Server 2012
 - Windows Server 2012 R2
- Windows Assessment and Deployment Kit (ADK) for Windows 10
- Windows PowerShell
- Microsoft .NET Framework

Install Windows ADK for Windows 10

These steps assume that you have the MDT01 member server installed and configured and that you have downloaded [Windows ADK for Windows 10](#) to the E:\Downloads\ADK folder.

1. On MDT01, log on as Administrator in the CONTOSO domain using a password of **P@ssw0rd**.
2. Start the **ADK Setup** (E:\Downloads\ADK\adksetup.exe), and on the first wizard page, click **Continue**.
3. On the **Select the features you want to change** page, select the features below and complete the wizard using the default settings:
 - a. Deployment Tools
 - b. Windows Preinstallation Environment (Windows PE)
 - c. User State Migration Tool (USMT)

IMPORTANT

Starting with Windows 10, version 1809, Windows PE is released separately from the ADK. See [Download and install the Windows ADK](#) for more information.

Install MDT

These steps assume that you have downloaded [MDT](#) to the E:\Downloads\MDT folder on MDT01.

1. On MDT01, log on as Administrator in the CONTOSO domain using a password of **P@ssw0rd**.
2. Install **MDT** (E:\Downloads\MDT\MicrosoftDeploymentToolkit_x64.msi) with the default settings.

Create the OU structure

If you do not have an organizational unit (OU) structure in your Active Directory, you should create one. In this section, you create an OU structure and a service account for MDT.

1. On DC01, using Active Directory User and Computers, in the contoso.com domain level, create a top-level OU named **Contoso**.
2. In the **Contoso** OU, create the following OUs:
 - a. Accounts
 - b. Computers
 - c. Groups
3. In the **Contoso / Accounts** OU, create the following underlying OUs:
 - a. Admins
 - b. Service Accounts
 - c. Users
4. In the **Contoso / Computers** OU, create the following underlying OUs:
 - a. Servers
 - b. Workstations
5. In the **Contoso / Groups** OU, create the following OU:
 - Security Groups

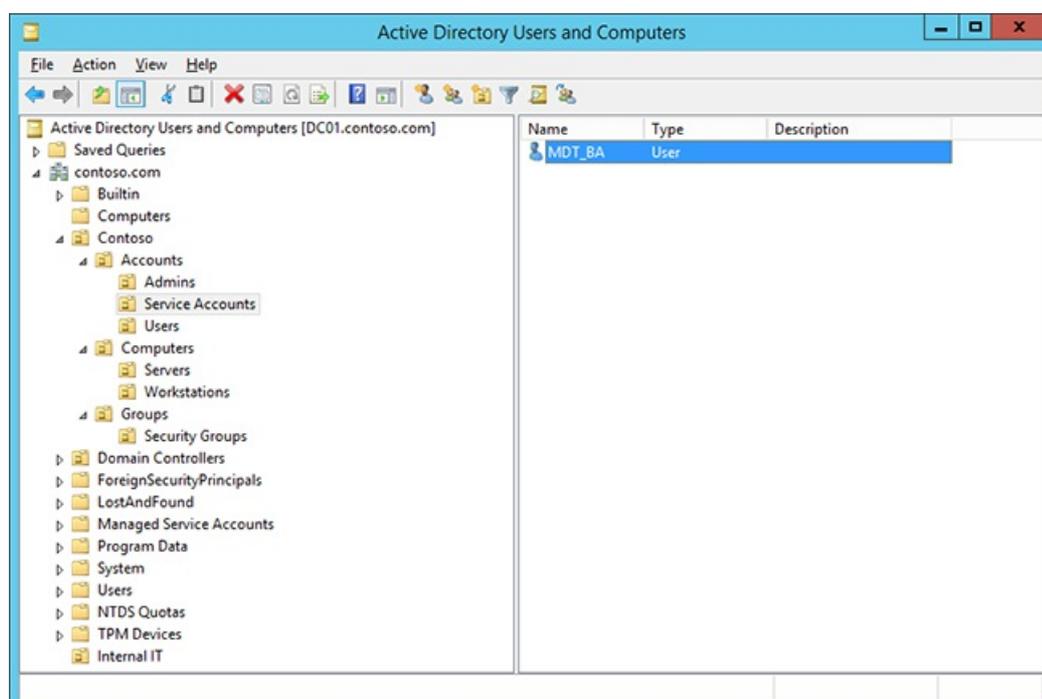


Figure 6. A sample of how the OU structure will look after all the OUs are created.

Create the MDT service account

When creating a reference image, you need an account for MDT. The MDT Build Account is used for Windows Preinstallation Environment (Windows PE) to connect to MDT01.

1. On DC01, using Active Directory User and Computers, browse to **contoso.com / Contoso / Service Accounts**.
2. Select the **Service Accounts** OU and create the **MDT_BA** account using the following settings:
 - a. Name: MDT_BA
 - b. User logon name: MDT_BA
 - c. Password: P@ssw0rd
 - d. User must change password at next logon: Clear
 - e. User cannot change password: Selected
 - f. Password never expires: Selected

Create and share the logs folder

By default MDT stores the log files locally on the client. In order to capture a reference image, you will need to enable server-side logging and, to do that, you will need to have a folder in which to store the logs. For more information, see [Create a Windows 10 reference image](#).

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create and share the **E:\Logs** folder by running the following commands in an elevated Windows PowerShell prompt:

```
New-Item -Path E:\Logs -ItemType directory
New-SmbShare -Name Logs$ -Path E:\Logs -ChangeAccess EVERYONE
icacls E:\Logs /grant '"MDT_BA":(OI)(CI)(M)'
```

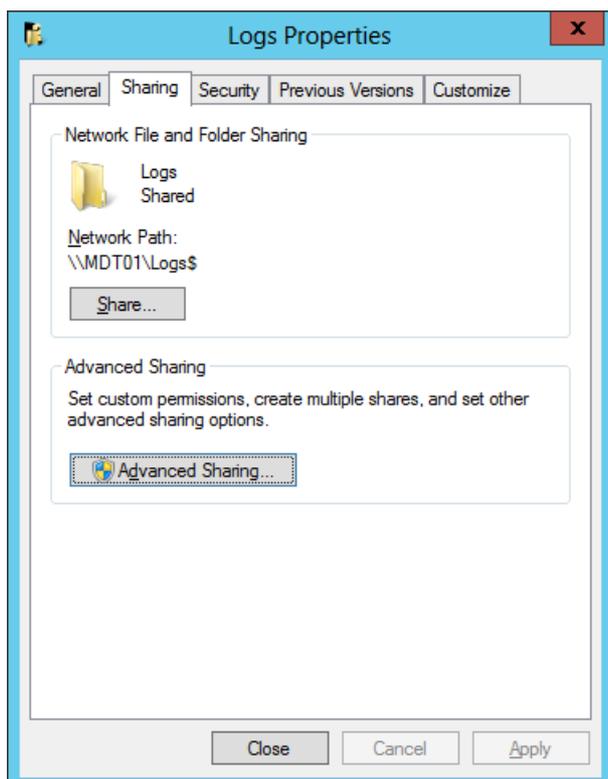


Figure 7. The Sharing tab of the E:\Logs folder after sharing it with PowerShell.

Use CMTrace to read log files (optional)

The log files in MDT Lite Touch are formatted to be read by Configuration Manager Trace (CMTrace), which is available as part of [Microsoft System Center 2012 R2 Configuration Manager Toolkit](#). You can use Notepad, but CMTrace formatting makes the logs easier to read.

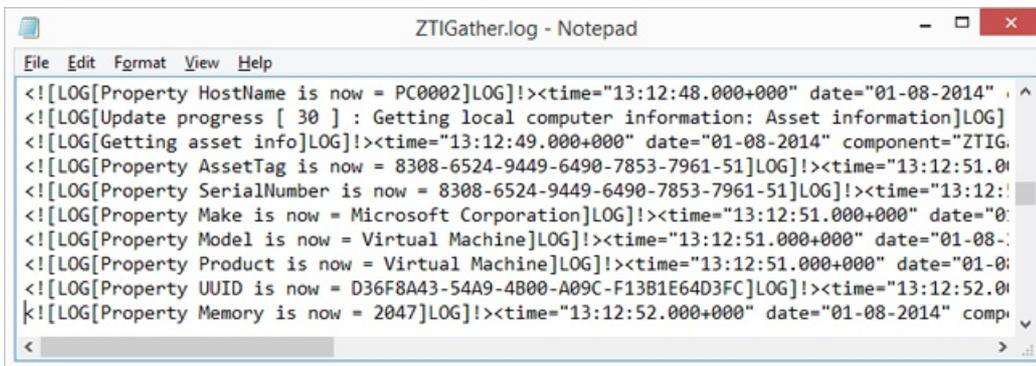


Figure 8. An MDT log file opened in Notepad.

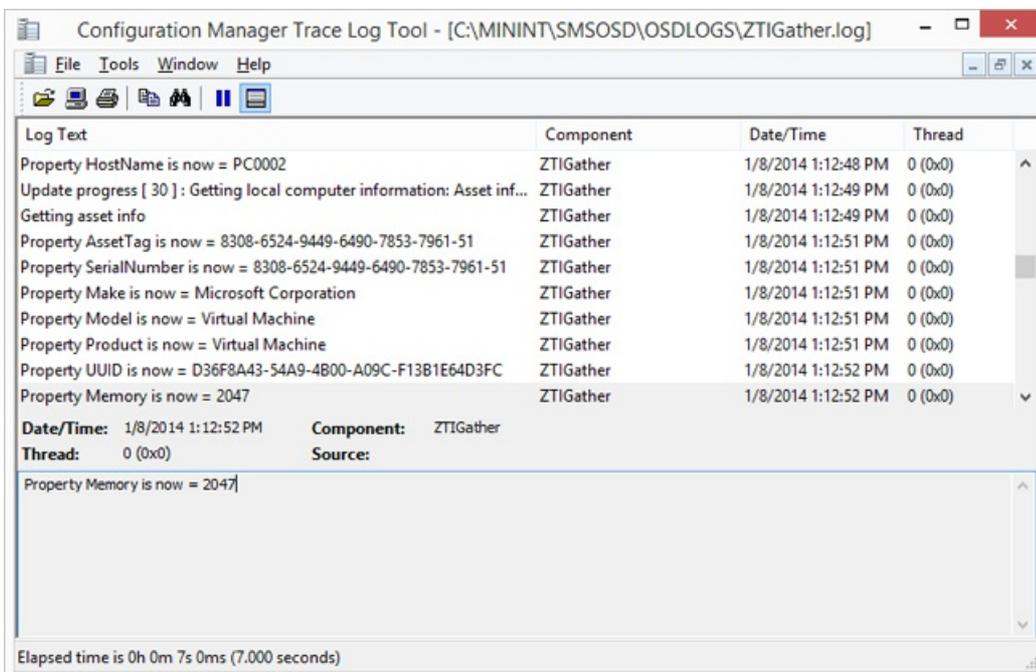


Figure 9. The same log file, opened in CMTrace, is much easier to read.

Related topics

[Key features in MDT](#)

[MDT Lite Touch components](#)

Create a Windows 10 reference image

6/14/2019 • 28 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Creating a reference image is important because that image serves as the foundation for the devices in your organization. In this topic, you will learn how to create a Windows 10 reference image using the Microsoft Deployment Toolkit (MDT). You will create a deployment share, configure rules and settings, and import all the applications and operating system files required to build a Windows 10 reference image. After completing the steps outlined in this topic, you will have a Windows 10 reference image that can be used in your deployment solution. For the purposes of this topic, we will use four machines: DC01, MDT01, HV01, and PC0001. DC01 is a domain controller, PC0001 is a Windows 10 Enterprise x64 client, and MDT01 is a Windows Server 2012 R2 standard server. HV01 is a Hyper-V host server, but HV01 could be replaced by PC0001 as long as PC0001 has enough memory and is capable of running Hyper-V. MDT01, HV01, and PC0001 are members of the domain contoso.com for the fictitious Contoso Corporation.

NOTE

For important details about the setup for the steps outlined in this article, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

The reference image

The reference image described in this documentation is designed primarily for deployment to physical machines. However, the reference image is created on a virtual platform, before being automatically run through the System Preparation (Sysprep) tool process and captured to a Windows Imaging (WIM) file. The reasons for creating the reference image on a virtual platform are the following:

- You reduce development time and can use snapshots to test different configurations quickly.
- You rule out hardware issues. You simply get the best possible image, and if you have a problem, it's not likely to be hardware related.
- It ensures that you won't have unwanted applications that could be installed as part of a driver install but not removed by the Sysprep process.
- It's easy to move between lab, test, and production.

Set up the MDT build lab deployment share

With Windows 10, there is no hard requirement to create reference images; however, to reduce the time needed for deployment, you may want to create a reference image that contains a few base applications as well as all of the latest updates. This section will show you how to create and configure the MDT Build Lab deployment share to create a Windows 10 reference image. Because reference images will be deployed only to virtual machines during the creation process and have specific settings (rules), you should always create a separate deployment

share specifically for this process.

Create the MDT build lab deployment share

- On MDT01, log on as Administrator in the CONTOSO domain using a password of **P@ssw0rd**.
- Using the Deployment Workbench, right-click **Deployment Shares** and select **New Deployment Share**.
- Use the following settings for the New Deployment Share Wizard:
 - Deployment share path: E:\MDTBuildLab
 - Share name: MDTBuildLab\$
 - Deployment share description: MDT Build Lab
 - <default>
- Verify that you can access the \\MDT01\MDTBuildLab\$ share.

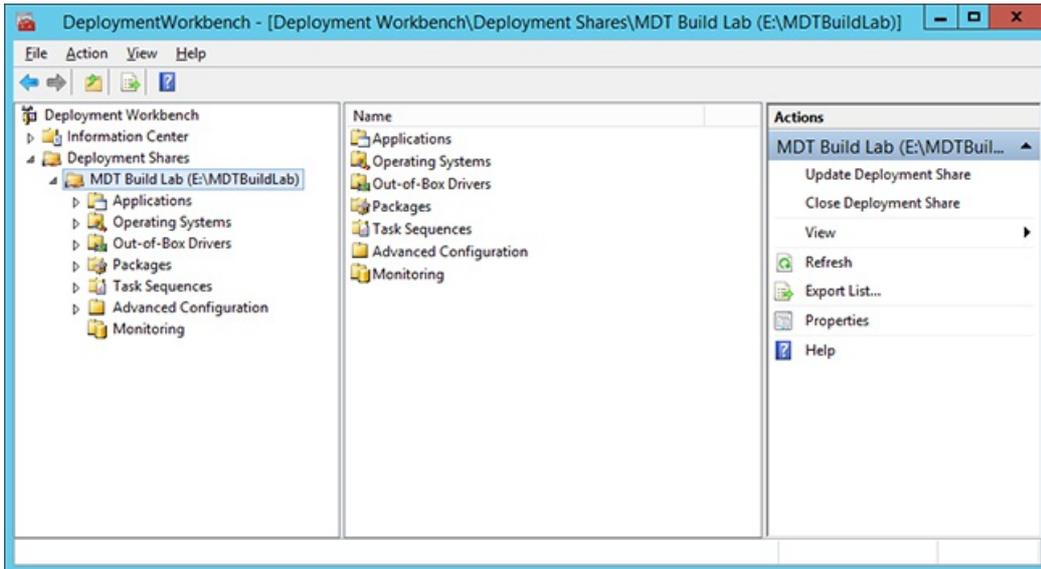


Figure 2. The Deployment Workbench with the MDT Build Lab deployment share created.

Configure permissions for the deployment share

In order to write the reference image back to the deployment share, you need to assign Modify permissions to the MDT Build Account (MDT_BA) for the **Captures** subfolder in the **E:\MDTBuildLab** folder

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Modify the NTFS permissions for the **E:\MDTBuildLab\Captures** folder by running the following command in an elevated Windows PowerShell prompt:

```
icacls E:\MDTBuildLab\Captures /grant '"MDT_BA":(OI)(CI)(M)'
```

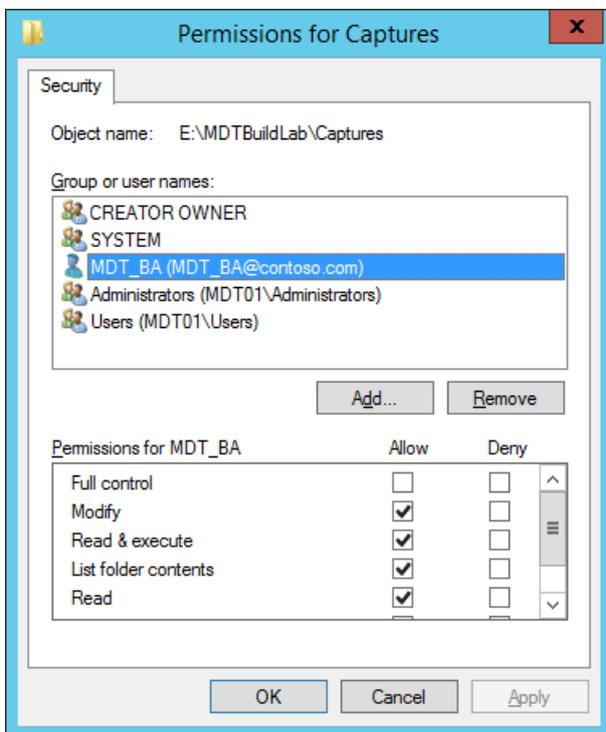


Figure 3. Permissions configured for the MDT_BA user.

Add the setup files

This section will show you how to populate the MDT deployment share with the Windows 10 operating system source files, commonly referred to as setup files, which will be used to create a reference image. Setup files are used during the reference image creation process and are the foundation for the reference image.

Add the Windows 10 installation files

MDT supports adding both full source Windows 10 DVDs (ISOs) and custom images that you have created. In this case, you create a reference image, so you add the full source setup files from Microsoft.

NOTE

Due to the Windows limits on path length, we are purposely keeping the operating system destination directory short, using the folder name W10EX64RTM rather than a more descriptive name like Windows 10 Enterprise x64 RTM.

Add Windows 10 Enterprise x64 (full source)

In these steps we assume that you have copied the content of a Windows 10 Enterprise x64 ISO to the **E:\Downloads\Windows 10 Enterprise x64** folder.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Using the Deployment Workbench, expand the **Deployment Shares** node, and then expand **MDT Build Lab**.
3. Right-click the **Operating Systems** node, and create a new folder named **Windows 10**.
4. Expand the **Operating Systems** node, right-click the **Windows 10** folder, and select **Import Operating System**. Use the following settings for the Import Operating System Wizard:
5. Full set of source files
6. Source directory: E:\Downloads\Windows 10 Enterprise x64
7. Destination directory name: W10EX64RTM
8. After adding the operating system, in the **Operating Systems / Windows 10** folder, double-click the added operating system name in the **Operating System** node and change the name to the following: **Windows 10 Enterprise x64 RTM Default Image**

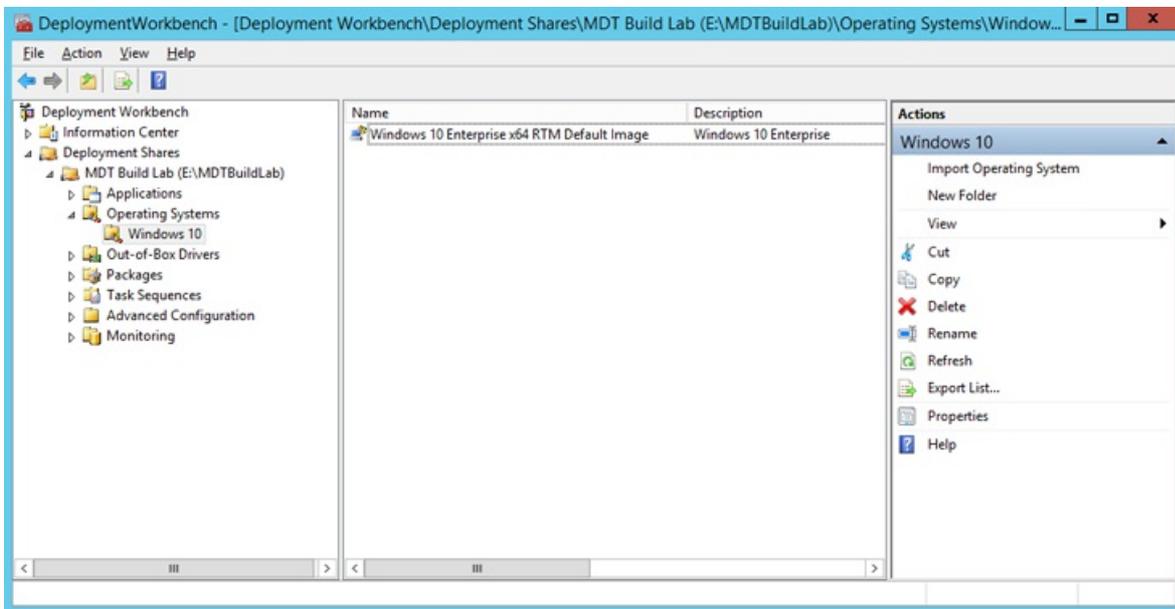


Figure 4. The imported Windows 10 operating system after renaming it.

Add applications

Before you create an MDT task sequence, you need to add all of the applications and other sample scripts to the MDT Build Lab share.

The steps in this section use a strict naming standard for your MDT applications. You add the "Install - " prefix for typical application installations that run a setup installer of some kind, and you use the "Configure - " prefix when an application configures a setting in the operating system. You also add an " - x86", " - x64", or " - x86-x64" suffix to indicate the application's architecture (some applications have installers for both architectures). Using a script naming standard is always recommended when using MDT as it helps maintain order and consistency. By storing configuration items as MDT applications, it is easy to move these objects between various solutions, or between test and production environments. In this topic's step-by-step sections, you will add the following applications:

- Install - Microsoft Office 2013 Pro Plus - x86
- Install - Microsoft Silverlight 5.0 - x64
- Install - Microsoft Visual C++ 2005 SP1 - x86
- Install - Microsoft Visual C++ 2005 SP1 - x64
- Install - Microsoft Visual C++ 2008 SP1 - x86
- Install - Microsoft Visual C++ 2008 SP1 - x64
- Install - Microsoft Visual C++ 2010 SP1 - x86
- Install - Microsoft Visual C++ 2010 SP1 - x64
- Install - Microsoft Visual C++ 2012 Update 4 - x86
- Install - Microsoft Visual C++ 2012 Update 4 - x64

In these examples, we assume that you downloaded the software in this list to the E:\Downloads folder. The first application is added using the UI, but because MDT supports Windows PowerShell, you add the other applications using Windows PowerShell.

NOTE

All the Microsoft Visual C++ downloads can be found on [The latest supported Visual C++ downloads](#).

Create the install: Microsoft Office Professional Plus 2013 x86

You can customize Office 2013. In the volume license versions of Office 2013, there is an Office Customization Tool you can use to customize the Office installation. In these steps we assume you have copied the Office 2013 installation files to the E:\Downloads\Office2013 folder.

Add the Microsoft Office Professional Plus 2013 x86 installation files

After adding the Microsoft Office Professional Plus 2013 x86 application, you then automate its setup by running the Office Customization Tool. In fact, MDT detects that you added the Office Professional Plus 2013 x86 application and creates a shortcut for doing this. You also can customize the Office installation using a Config.xml file. But we recommend that you use the Office Customization Tool as described in the following steps, as it provides a much richer way of controlling Office 2013 settings.

1. Using the Deployment Workbench in the MDT Build Lab deployment share, expand the **Applications / Microsoft** node, and double-click **Install - Microsoft Office 2013 Pro Plus x86**.
2. In the **Office Products** tab, click **Office Customization Tool**, and click **OK** in the **Information** dialog box.

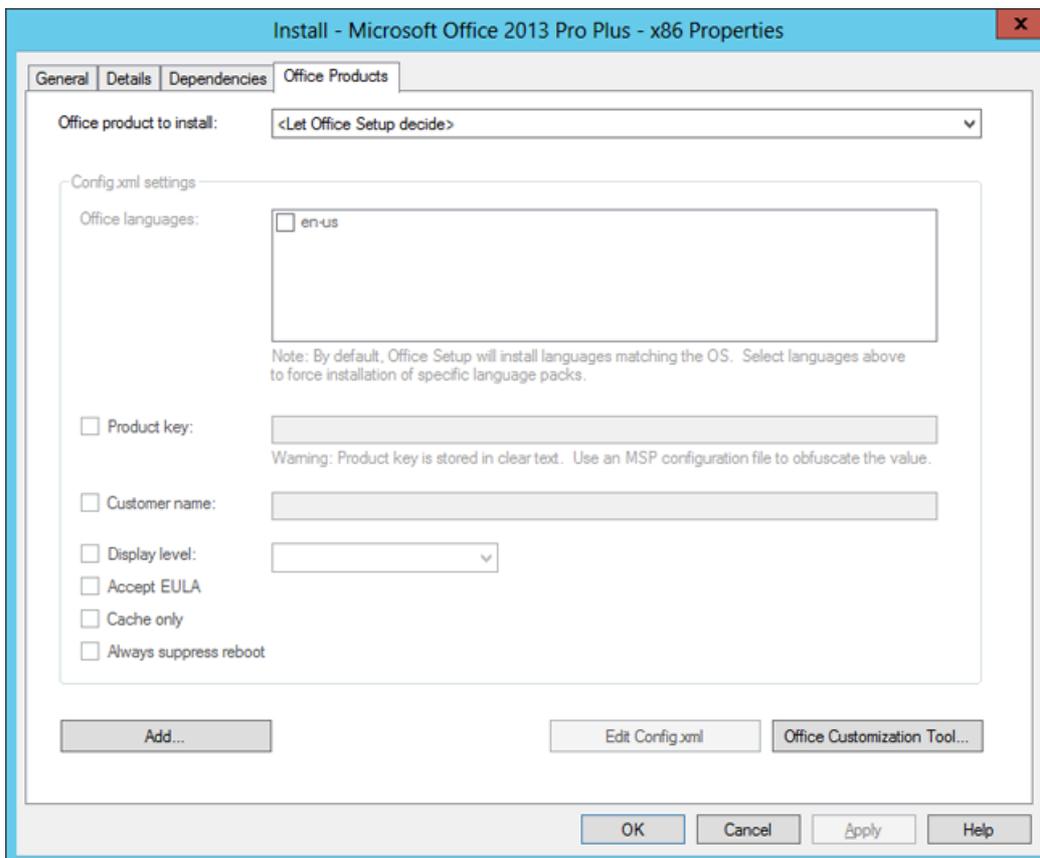


Figure 5. The Install - Microsoft Office 2013 Pro Plus - x86 application properties.

NOTE

If you don't see the Office Products tab, verify that you are using a volume license version of Office. If you are deploying Office 365, you need to download the Admin folder from Microsoft.

3. In the Office Customization Tool dialog box, select the Create a new Setup customization file for the following product option, select the Microsoft Office Professional Plus 2013 (32-bit) product, and click OK.
4. Use the following settings to configure the Office 2013 setup to be fully unattended:
 - a. Install location and organization name

- Organization name: Contoso
- b. Licensing and user interface
 - a. Select Use KMS client key
 - b. Select I accept the terms in the License Agreement.
 - c. Select Display level: None

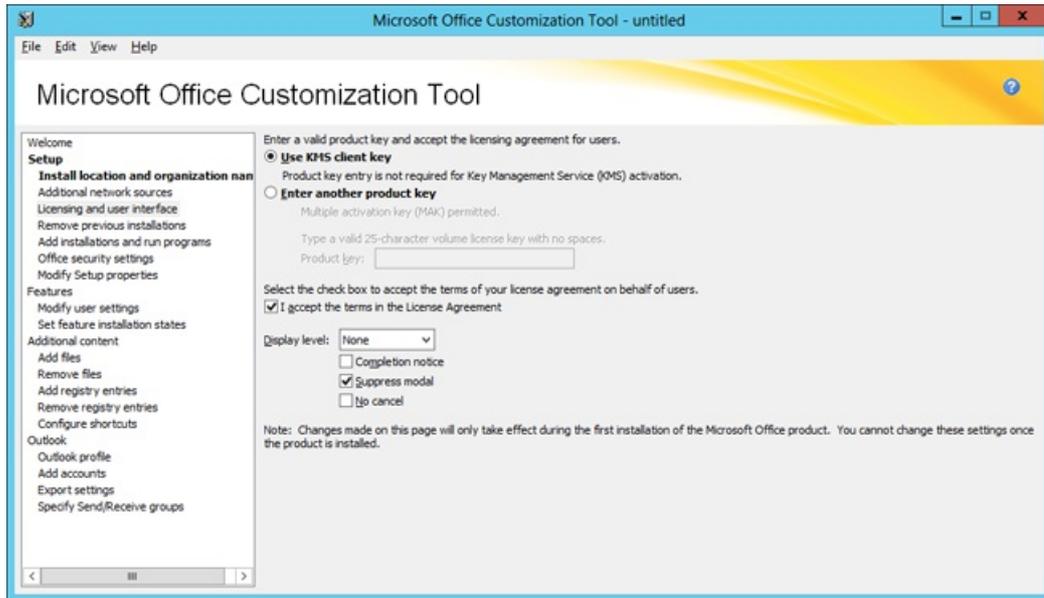


Figure 6. The licensing and user interface screen in the Microsoft Office Customization Tool

- c. Modify Setup properties
 - Add the **SETUP_REBOOT** property and set the value to **Never**.
 - d. Modify user settings
 - In the **Microsoft Office 2013** node, expand **Privacy**, select **Trust Center**, and enable the **Disable Opt-in Wizard on first run** setting.
5. From the **File** menu, select **Save**, and save the configuration as **0_Office2013ProPlusx86.msp** in the **E:\MDTBuildLab\Applications\Install - Microsoft Office 2013 Pro Plus - x86\Updates** folder.

NOTE

The reason for naming the file with a 0 (zero) at the beginning is that the Updates folder also handles Microsoft Office updates, and they are installed in alphabetical order. The Office 2013 setup works best if the customization file is installed before any updates.

6. Close the Office Customization Tool, click Yes in the dialog box, and in the **Install - Microsoft Office 2013 Pro Plus - x86 Properties** window, click **OK**.

Connect to the deployment share using Windows PowerShell

If you need to add many applications, you can take advantage of the PowerShell support that MDT has. To start using PowerShell against the deployment share, you must first load the MDT PowerShell snap-in and then make the deployment share a PowerShell drive (PSDrive).

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Import the snap-in and create the PSDrive by running the following commands in an elevated PowerShell prompt:

```
Import-Module "C:\Program Files\Microsoft Deployment Toolkit\bin\MicrosoftDeploymentToolkit.psd1"
New-PSDrive -Name "DS001" -PSProvider MDTProvider -Root "E:\MDTBuildLab"
```

Create the install: Microsoft Visual C++ 2005 SP1 x86

In these steps we assume that you have downloaded Microsoft Visual C++ 2005 SP1 x86. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2005SP1x86.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2005 SP1 - x86"
$CommandLine = "vcredist_x86.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2005SP1x86"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2005 SP1 x64

In these steps we assume that you have downloaded Microsoft Visual C++ 2005 SP1 x64. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2005SP1x64.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2005 SP1 - x64"
$CommandLine = "vcredist_x64.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2005SP1x64"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2008 SP1 x86

In these steps we assume that you have downloaded Microsoft Visual C++ 2008 SP1 x86. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2008SP1x86.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2008 SP1 - x86"
$CommandLine = "vcredist_x86.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2008SP1x86"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2008 SP1 x64

In these steps we assume that you have downloaded Microsoft Visual C++ 2008 SP1 x64. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2008SP1x64.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2008 SP1 - x64"
$CommandLine = "vcredist_x64.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2008SP1x64"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2010 SP1 x86

In these steps we assume that you have downloaded Microsoft Visual C++ 2010 SP1 x86. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2010SP1x86.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2010 SP1 - x86"
$CommandLine = "vcredist_x86.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2010SP1x86"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2010 SP1 x64

In these steps we assume that you have downloaded Microsoft Visual C++ 2010 SP1 x64. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2010SP1x64.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2010 SP1 - x64"
$CommandLine = "vcredist_x64.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2010SP1x64"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2012 Update 4 x86

In these steps we assume that you have downloaded Microsoft Visual C++ 2012 Update 4 x86. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is

set to E:\Downloads\VC++2012Ux86.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2012 Update 4 - x86"
$CommandLine = "vcredist_x86.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2012Ux86"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the install: Microsoft Visual C++ 2012 Update 4 x64

In these steps we assume that you have downloaded Microsoft Visual C++ 2012 Update 4 x64. You might need to modify the path to the source folder to reflect your current environment. In this example, the source path is set to E:\Downloads\VC++2012Ux64.

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create the application by running the following commands in an elevated PowerShell prompt:

```
$ApplicationName = "Install - Microsoft Visual C++ 2012 Update 4 - x64"
$CommandLine = "vcredist_x64.exe /Q"
$ApplicationSourcePath = "E:\Downloads\VC++2012Ux64"
Import-MDTApplication -Path "DS001:\Applications\Microsoft" -Enable "True" -Name $ApplicationName -
ShortName $ApplicationName -CommandLine $CommandLine -WorkingDirectory
".\Applications\$ApplicationName" -ApplicationSourcePath $ApplicationSourcePath -DestinationFolder
$ApplicationName
-Verbose
```

Create the reference image task sequence

In order to build and capture your Windows 10 reference image for deployment using MDT, you will create a task sequence. The task sequence will reference the operating system and applications that you previously imported into the MDT Build Lab deployment share to build a Windows 10 reference image. After creating the task sequence, you configure it to enable patching against the Windows Server Update Services (WSUS) server. The Task Sequence Windows Update action supports getting updates directly from Microsoft Update, but you get more stable patching if you use a local WSUS server. WSUS also allows for an easy process of approving the patches that you are deploying.

Drivers and the reference image

Because we use modern virtual platforms for creating our reference images, we don't need to worry about drivers when creating reference images for Windows 10. We use Hyper-V in our environment, and Windows Preinstallation Environment (Windows PE) already has all the needed drivers built-in for Hyper-V.

Create a task sequence for Windows 10 Enterprise

To create a Windows 10 reference image task sequence, the process is as follows:

1. Using the Deployment Workbench in the MDT Build Lab deployment share, right-click **Task Sequences**, and create a new folder named **Windows 10**.
2. Expand the **Task Sequences** node, right-click the new **Windows 10** folder and select **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - a. Task sequence ID: REFW10X64-001

- b. Task sequence name: Windows 10 Enterprise x64 RTM Default Image
- c. Task sequence comments: Reference Build
- d. Template: Standard Client Task Sequence
- e. Select OS: Windows 10 Enterprise x64 RTM Default Image
- f. Specify Product Key: Do not specify a product key at this time
- g. Full Name: Contoso
- h. Organization: Contoso
- i. Internet Explorer home page: <http://www.contoso.com>
- j. Admin Password: Do not specify an Administrator Password at this time

Edit the Windows 10 task sequence

The steps below walk you through the process of editing the Windows 10 reference image task sequence to include the actions required to update the reference image with the latest updates from WSUS, install roles and features, and utilities, and install Microsoft Office 2013.

1. In the Task Sequences / Windows 10 folder, right-click the Windows 10 Enterprise x64 RTM Default Image task sequence, and select Properties.
2. On the **Task Sequence** tab, configure the Windows 10 Enterprise x64 RTM Default Image task sequence with the following settings:
 - a. State Restore. Enable the Windows Update (Pre-Application Installation) action. **Note** Enable an action by going to the Options tab and clearing the Disable this step check box.
 - b. State Restore. Enable the Windows Update (Post-Application Installation) action.
 - c. State Restore. Enable the Windows Update (Post-Application Installation) action. State Restore. After the **Tattoo** action, add a new **Group** action with the following setting:
 - Name: Custom Tasks (Pre-Windows Update)
 - d. State Restore. After Windows Update (Post-Application Installation) action, rename Custom Tasks to Custom Tasks (Post-Windows Update). **Note** The reason for adding the applications after the Tattoo action but before running Windows Update is simply to save time during the deployment. This way we can add all applications that will upgrade some of the built-in components and avoid unnecessary updating.
 - e. State Restore / Custom Tasks (Pre-Windows Update). Add a new Install Roles and Features action with the following settings:
 - a. Name: Install - Microsoft NET Framework 3.5.1
 - b. Select the operating system for which roles are to be installed: Windows 10
 - c. Select the roles and features that should be installed: .NET Framework 3.5 (includes .NET 2.0 and 3.0)

IMPORTANT

This is probably the most important step when creating a reference image. Many applications need the .NET Framework, and we strongly recommend having it available in the image. The one thing that makes this different from other components is that .NET Framework 3.5.1 is not included in the WIM file. It is installed from the **Sources\SxS** folder on the media, and that makes it more difficult to add after the image has been deployed.

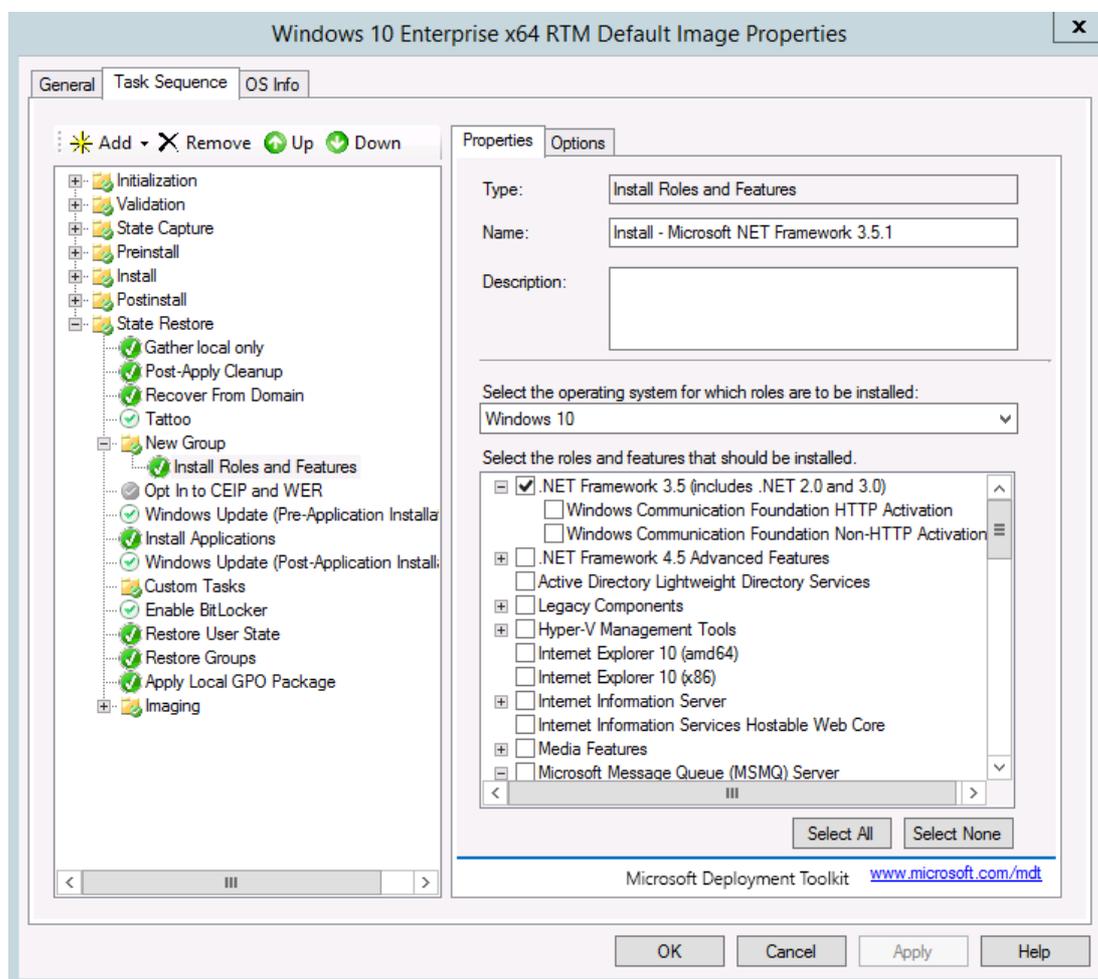


Figure 7. The task sequence after creating the Custom Tasks (Pre-Windows Update) group and adding the Install - Microsoft NET Framework 3.5.1 action.

- f. State Restore - Custom Tasks (Pre-Windows Update). After the **Install - Microsoft NET Framework 3.5.1** action, add a new **Install Application** action with the following settings:
 - a. Name: Install - Microsoft Visual C++ 2005 SP1 - x86
 - b. Install a Single Application: Install - Microsoft Visual C++ 2005 SP1 - x86-x64
- g. Repeat the previous step (add a new **Install Application**) to add the following applications:
 - a. Install - Microsoft Visual C++ 2005 SP1 - x64
 - b. Install - Microsoft Visual C++ 2008 SP1 - x86
 - c. Install - Microsoft Visual C++ 2008 SP1 - x64
 - d. Install - Microsoft Visual C++ 2010 SP1 - x86
 - e. Install - Microsoft Visual C++ 2010 SP1 - x64
 - f. Install - Microsoft Visual C++ 2012 Update 4 - x86
 - g. Install - Microsoft Visual C++ 2012 Update 4 - x64
 - h. Install - Microsoft Office 2013 Pro Plus - x86
- h. After the Install - Microsoft Office 2013 Pro Plus - x86 action, add a new Restart computer action.

3. Click **OK**.

Optional configuration: Add a suspend action

The goal when creating a reference image is of course to automate everything. But sometimes you have a special configuration or application setup that is too time-consuming to automate. If you need to do some manual configuration, you can add a little-known feature called Lite Touch Installation (LTI) Suspend. If you add the LTISuspend.wsf script as a custom action in the task sequence, it will suspend the task sequence until you

click the Resume Task Sequence shortcut icon on the desktop. In addition to using the LTI Suspend feature for manual configuration or installation, you can also use it simply for verifying a reference image before you allow the task sequence to continue and use Sysprep and capture the virtual machine.

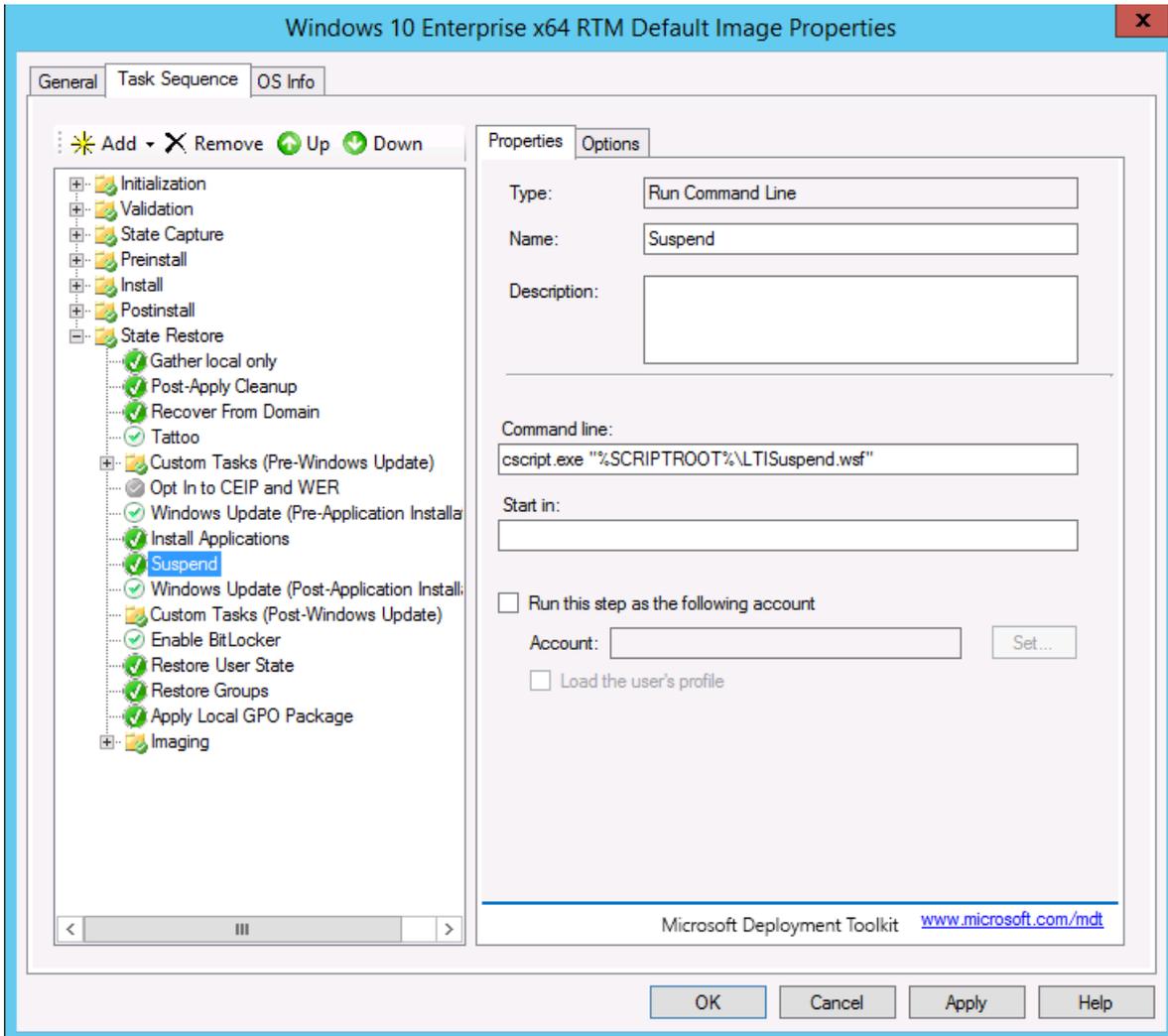


Figure 8. A task sequence with optional Suspend action (LTISuspend.wsf) added.



Figure 9. The Windows 10 desktop with the Resume Task Sequence shortcut.

Edit the Unattend.xml file for Windows 10 Enterprise

When using MDT, you don't need to edit the Unattend.xml file very often because most configurations are taken care of by MDT. However if, for example, you want to configure Internet Explorer 11 behavior, then you can edit the Unattend.xml for this. Editing the Unattend.xml for basic Internet Explorer settings is easy, but for more advanced settings, you will want to use Internet Explorer Administration Kit (IEAK).

WARNING

Do not use **SkipMachineOOBE** or **SkipUserOOBE** in your Unattend.xml file. These settings are deprecated and can have unintended effects if used.

NOTE

You also can use the Unattend.xml to enable components in Windows 10, like the Telnet Client or Hyper-V client. Normally we prefer to do this via the **Install Roles and Features** action, or using Deployment Image Servicing and Management (DISM) command-line tools, because then we can add that as an application, being dynamic, having conditions, and so forth. Also, if you are adding packages via Unattend.xml, it is version specific, so Unattend.xml must match the exact version of the operating system you are servicing.

Follow these steps to configure Internet Explorer settings in Unattend.xml for the Windows 10 Enterprise x64 RTM Default Image task sequence:

1. Using the Deployment Workbench, right-click the **Windows 10 Enterprise x64 RTM Default Image** task sequence and select **Properties**.
2. In the **OS Info** tab, click **Edit Unattend.xml**. MDT now generates a catalog file. This will take a few minutes, and then Windows System Image Manager (Windows SIM) will start.
3. In Windows SIM, expand the **4 specialize** node in the **Answer File** pane and select the amd64_Microsoft-

Windows-IE-InternetExplorer_neutral entry.

- In the **amd64_Microsoft-Windows-IE-InternetExplorer_neutral properties** window (right-hand window), set the following values:
 - DisableDevTools: true
- Save the Unattend.xml file, and close Windows SIM.
- On the Windows 10 Enterprise x64 RTM Default Image Properties, click **OK**.

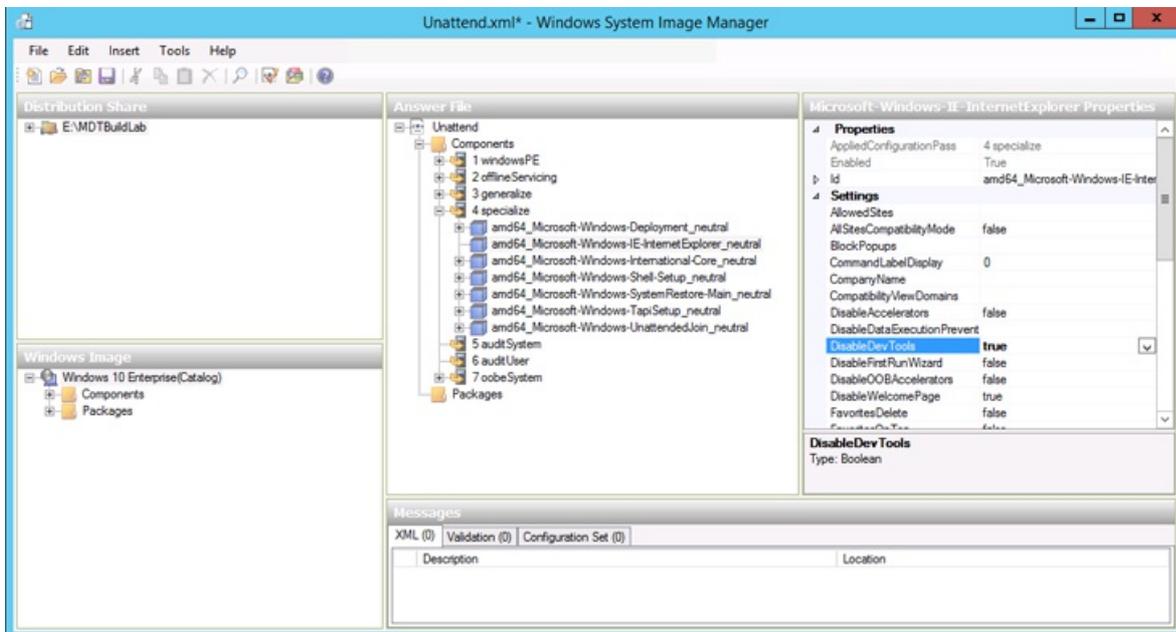


Figure 10. Windows System Image Manager with the Windows 10 Unattend.xml.

Configure the MDT deployment share rules

Understanding rules is critical to successfully using MDT. Rules are configured using the Rules tab of the deployment share's properties. The Rules tab is essentially a shortcut to edit the CustomSettings.ini file that exists in the E:\MDTBuildLab\Control folder. This section discusses how to configure the MDT deployment share rules as part of your Windows 10 Enterprise deployment.

MDT deployment share rules overview

In MDT, there are always two rule files: the CustomSettings.ini file and the Bootstrap.ini file. You can add almost any rule to either; however, the Bootstrap.ini file is copied from the Control folder to the boot image, so the boot image needs to be updated every time you change that file. For that reason, add only a minimal set of rules to Bootstrap.ini, such as which deployment server and share to connect to - the DEPLOYROOT value. Put the other rules in CustomSettings.ini because that file is updated immediately when you click OK. By taking the following steps, you will configure the rules for the MDT Build Lab deployment share:

- Using the Deployment Workbench, right-click the **MDT Build Lab deployment share** and select **Properties**.
- Select the **Rules** tab and modify using the following information:

```

[Settings]
Priority=Default
[Default]
_SMSTSORGNAME=Contoso
UserDataLocation=NONE
DoCapture=YES
OSInstall=Y
AdminPassword=P@ssw0rd
TimeZoneName=Pacific Standard Time
JoinWorkgroup=WORKGROUP
HideShell=YES
FinishAction=SHUTDOWN
DoNotCreateExtraPartition=YES
WSUSServer=http://mdt01.contoso.com:8530
ApplyGPOPack=NO
SLSHARE=\\MDT01\Logs$
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=YES
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=YES
SkipBitLocker=YES
SkipSummary=YES
SkipRoles=YES
SkipCapture=NO
SkipFinalSummary=YES

```

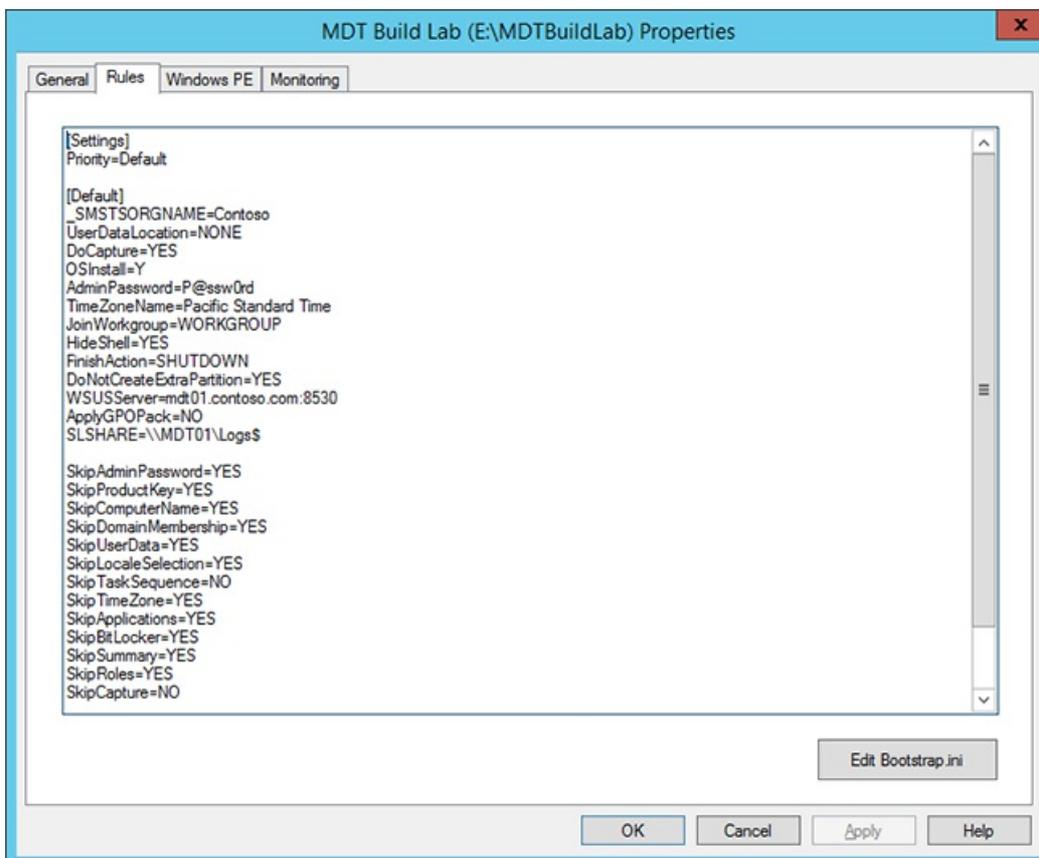


Figure 11. The server-side rules for the MDT Build Lab deployment share.

3. Click **Edit Bootstrap.ini** and modify using the following information:

```
[Settings]
Priority=Default
[Default]
DeployRoot=\\MDT01\MDTBuildLab$
UserDomain=CONTOSO
UserID=MDT_BA
UserPassword=P@ssw0rd
SkipBDDWelcome=YES
```

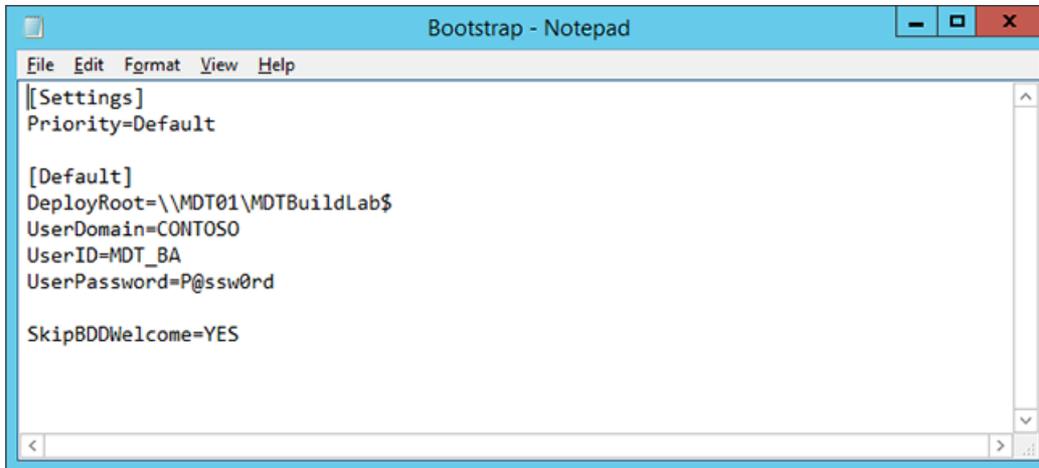


Figure 12. The boot image rules for the MDT Build Lab deployment share.

NOTE

For security reasons, you normally don't add the password to the Bootstrap.ini file; however, because this deployment share is for creating reference image builds only, and should not be published to the production network, it is acceptable to do so in this situation.

4. In the **Windows PE** tab, in the **Platform** drop-down list, select **x86**.
5. In the **Lite Touch Boot Image Settings** area, configure the following settings:
 - a. Image description: MDT Build Lab x86
 - b. ISO file name: MDT Build Lab x86.iso
6. In the **Windows PE** tab, in the **Platform** drop-down list, select **x64**.
7. In the **Lite Touch Boot Image Settings** area, configure the following settings:
 - a. Image description: MDT Build Lab x64
 - b. ISO file name: MDT Build Lab x64.iso
8. Click **OK**.

NOTE

In MDT, the x86 boot image can deploy both x86 and x64 operating systems (except on computers based on Unified Extensible Firmware Interface).

Update the deployment share

After the deployment share has been configured, it needs to be updated. This is the process when the Windows PE boot images are created.

1. Using the Deployment Workbench, right-click the **MDT Build Lab deployment share** and select **Update Deployment Share**.

2. Use the default options for the Update Deployment Share Wizard.

NOTE

The update process will take 5 to 10 minutes.

The rules explained

Now that the MDT Build Lab deployment share (the share used to create the reference images) has been configured, it is time to explain the various settings used in the Bootstrap.ini and CustomSettings.ini files.

The Bootstrap.ini and CustomSettings.ini files work together. The Bootstrap.ini file is always present on the boot image and is read first. The basic purpose for Bootstrap.ini is to provide just enough information for MDT to find the CustomSettings.ini.

The CustomSettings.ini file is normally stored on the server, in the Deployment share\Control folder, but also can be stored on the media (when using offline media).

NOTE

The settings, or properties, that are used in the rules (CustomSettings.ini and Bootstrap.ini) are listed in the MDT documentation, in the Microsoft Deployment Toolkit Reference / Properties / Property Definition section.

The Bootstrap.ini file

The Bootstrap.ini file is available via the deployment share's Properties dialog box, or via the E:\MDTBuildLab\Control folder on MDT01.

```
[Settings]
Priority=Default
[Default]
DeployRoot=\\MDT01\MDTBuildLab$
UserDomain=CONTOSO
UserID=MDT_BA
UserPassword=P@ssw0rd
SkipBDDWelcome=YES
```

So, what are these settings?

- **Priority.** This determines the order in which different sections are read. This Bootstrap.ini has only one section, named [Default].
- **DeployRoot.** This is the location of the deployment share. Normally, this value is set by MDT, but you need to update the DeployRoot value if you move to another server or other share. If you don't specify a value, the Windows Deployment Wizard prompts you for a location.
- **UserDomain, UserID, and UserPassword.** These values are used for automatic log on to the deployment share. Again, if they are not specified, the wizard prompts you.

WARNING

Caution is advised. These values are stored in clear text on the boot image. Use them only for the MDT Build Lab deployment share and not for the MDT Production deployment share that you learn to create in the next topic.

- **SkipBDDWelcome.** Even if it is nice to be welcomed every time we start a deployment, we prefer to skip the initial welcome page of the Windows Deployment Wizard.

NOTE

All properties beginning with "Skip" control only whether to display that pane in the Windows Deployment Wizard. Most of the panes also require you to actually set one or more values.

The CustomSettings.ini file

The CustomSettings.ini file, whose content you see on the Rules tab of the deployment share Properties dialog box, contains most of the properties used in the configuration.

```
[Settings]
Priority=Default
[Default]
_SMSTSORGNAME=Contoso
UserDataLocation=NONE
DoCapture=YES
OSInstall=Y
AdminPassword=P@ssw0rd
TimeZoneName=Pacific Standard Time
JoinWorkgroup=WORKGROUP
HideShell=YES
FinishAction=SHUTDOWN
DoNotCreateExtraPartition=YES
WSUSServer=http://mdt01.contoso.com:8530
ApplyGPOPack=NO
LSHARE=\\MDT01\Logs$
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=YES
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=YES
SkipBitLocker=YES
SkipSummary=YES
SkipRoles=YES
SkipCapture=NO
SkipFinalSummary=YES
```

- **Priority.** Has the same function as in Bootstrap.ini. Priority determines the order in which different sections are read. This CustomSettings.ini has only one section, named [Default]. In general, if you have multiple sections that set the same value, the value from the first section (higher priority) wins. The rare exceptions are listed in the ZTIGather.xml file.
- **_SMSTSORGNAME.** The organization name displayed in the task sequence progress bar window during deployment.
- **UserDataLocation.** Controls the settings for user state backup. You do not need to use when building and capturing a reference image.
- **DoCapture.** Configures the task sequence to run the System Preparation (Sysprep) tool and capture the image to a file when the operating system is installed.
- **OSInstall.** Must be set to Y or YES (the code actually just looks for the Y character) for the setup to proceed.
- **AdminPassword.** Sets the local Administrator account password.
- **TimeZoneName.** Establishes the time zone to use. Don't confuse this value with TimeZone, which is

only for legacy operating systems (Windows 7 and Windows Server 2003).

Note The easiest way to find the current time zone name on a Windows 10 machine is to run `tzutil /g` in a command prompt. You can also run `tzutil /l` to get a listing of all available time zone names.

- **JoinWorkgroup.** Configures Windows to join a workgroup.
- **HideShell.** Hides the Windows Shell during deployment. This is especially useful for Windows 10 deployments in which the deployment wizard will otherwise appear behind the tiles.
- **FinishAction.** Instructs MDT what to do when the task sequence is complete.
- **DoNotCreateExtraPartition.** Configures the task sequence not to create the extra partition for BitLocker. There is no need to do this for your reference image.
- **WSUSServer.** Specifies which Windows Server Update Services (WSUS) server (and port, if needed) to use during the deployment. Without this option MDT will use Microsoft Update directly, which will increase deployment time and limit your options of controlling which updates are applied.
- **SLSHARE.** Instructs MDT to copy the log files to a server share if something goes wrong during deployment, or when a deployment is successfully completed.
- **ApplyGPOPack.** Allows you to deploy local group policies created by Microsoft Security Compliance Manager (SCM).
- **SkipAdminPassword.** Skips the pane that asks for the Administrator password.
- **SkipProductKey.** Skips the pane that asks for the product key.
- **SkipComputerName.** Skips the Computer Name pane.
- **SkipDomainMemberShip.** Skips the Domain Membership pane. If set to Yes, you need to configure either the `JoinWorkgroup` value or the `JoinDomain`, `DomainAdmin`, `DomainAdminDomain`, and `DomainAdminPassword` properties.
- **SkipUserData.** Skips the pane for user state migration.
- **SkipLocaleSelection.** Skips the pane for selecting language and keyboard settings.
- **SkipTimeZone.** Skips the pane for setting the time zone.
- **SkipApplications.** Skips the Applications pane.
- **SkipBitLocker.** Skips the BitLocker pane.
- **SkipSummary.** Skips the initial Windows Deployment Wizard summary pane.
- **SkipRoles.** Skips the Install Roles and Features pane.
- **SkipCapture.** Skips the Capture pane.
- **SkipFinalSummary.** Skips the final Windows Deployment Wizard summary. Because you use `FinishAction=Shutdown`, you don't want the wizard to stop in the end so that you need to click OK before the machine shuts down.

Build the Windows 10 reference image

Once you have created your task sequence, you are ready to create the Windows 10 reference image. This will be performed by launching the task sequence from a virtual machine which will then automatically perform the reference image creation and capture process. This steps below outline the process used to boot a virtual machine using an ISO boot image created by MDT, and then execute the reference image task sequence image

to create and capture the Windows 10 reference image.

1. Copy the E:\MDTBuildLab\Boot\MDT Build Lab x86.iso on MDT01 to C:\ISO on the Hyper-V host.

Note Remember, in MDT you can use the x86 boot image to deploy both x86 and x64 operating system images. That's why you can use the x86 boot image instead of the x64 boot image.

2. Create a virtual machine with the following settings:

- a. Name: REFW10X64-001
- b. Location: C:\VMs
- c. Memory: 1024 MB
- d. Network: External (The network that is connected to the same infrastructure as MDT01 is)
- e. Hard disk: 60 GB (dynamic disk)
- f. Image file: C:\ISO\MDT Build Lab x86.iso

3. Take a snapshot of the REFW10X64-001 virtual machine, and name it **Clean with MDT Build Lab x86 ISO**.

Note Taking a snapshot is useful if you need to restart the process and want to make sure you can start clean.

4. Start the REFW10X64-001 virtual machine. After booting into Windows PE, complete the Windows Deployment Wizard using the following settings:

- a. Select a task sequence to execute on this computer: Windows 10 Enterprise x64 RTM Default Image
- b. Specify whether to capture an image: Capture an image of this reference computer
 - Location: \\MDT01\MDTBuildLab\Captures
- c. File name: REFW10X64-001.wim

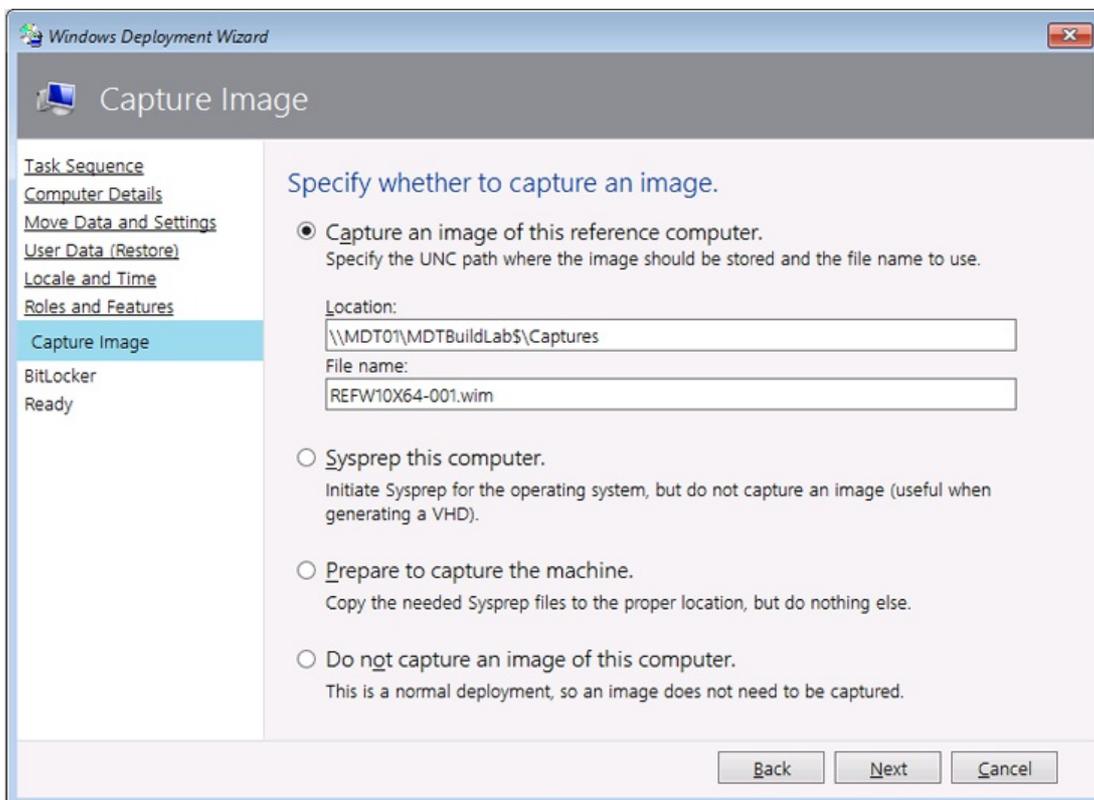


Figure 13. The Windows Deployment Wizard for the Windows 10 reference image.

5. The setup now starts and does the following:

- a. Installs the Windows 10 Enterprise operating system.
- b. Installs the added applications, roles, and features.
- c. Updates the operating system via your local Windows Server Update Services (WSUS) server.
- d. Stages Windows PE on the local disk.
- e. Runs System Preparation (Sysprep) and reboots into Windows PE.
- f. Captures the installation to a Windows Imaging (WIM) file.
- g. Turns off the virtual machine.

After some time, you will have a Windows 10 Enterprise x64 image that is fully patched and has run through Sysprep, located in the E:\MDTBuildLab\Captures folder on your deployment server. The file name is REFW10X64-001.wim.

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Deploy a Windows 10 image using MDT](#)

[Build a distributed environment for Windows 10 deployment](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

[Configure MDT settings](#)

Deploy a Windows 10 image using MDT

6/14/2019 • 26 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic will show you how to take your reference image for Windows 10, and deploy that image to your environment using the Microsoft Deployment Toolkit (MDT). You will prepare for this by creating a MDT deployment share that is used solely for image deployment. Separating the processes of creating reference images from the processes used to deploy them in production allows greater control of on both processes. You will then configure the deployment share, create a new task sequence, add applications, add drivers, add rules, and configure Active Directory permissions for deployment.

For the purposes of this topic, we will use three machines: DC01, MDT01, and PC0005. DC01 is a domain controller, MDT01 is a Windows Server 2012 R2 standard server, and PC0005 is a blank machine to which you deploy Windows 10. MDT01 and PC0005 are members of the domain contoso.com for the fictitious Contoso Corporation.



Figure 1. The machines used in this topic.

NOTE

For important details about the setup for the steps outlined in this article, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Step 1: Configure Active Directory permissions

These steps will show you how to configure an Active Directory account with the permissions required to deploy a Windows 10 machine to the domain using MDT. These steps assume you have downloaded the sample [Set-OUPermissions.ps1 script](#) and copied it to C:\Setup\Scripts on DC01. The account is used for Windows Preinstallation Environment (Windows PE) to connect to MDT01. In order for MDT to join machines into the contoso.com domain you need to create an account and configure permissions in Active Directory.

1. On DC01, using Active Directory User and Computers, browse to **contoso.com / Contoso / Service Accounts**.
2. Select the **Service Accounts** organizational unit (OU) and create the MDT_JD account using the following settings:
 - a. Name: MDT_JD
 - b. User logon name: MDT_JD
 - c. Password: P@ssw0rd
 - d. User must change password at next logon: Clear
 - e. User cannot change password: Select
 - f. Password never expires: Select
3. In an elevated Windows PowerShell prompt (run as Administrator), run the following commands and press **Enter** after each command:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Force
Set-Location C:\Setup\Scripts
.\Set-OUPermissions.ps1 -Account MDT_JD -TargetOU "OU=Workstations,OU=Computers,OU=Contoso"
```

4. The Set-OUPermissions.ps1 script allows the MDT_JD user account permissions to manage computer accounts in the Contoso / Computers OU. Below you find a list of the permissions being granted:
 - a. Scope: This object and all descendant objects
 - a. Create Computer objects
 - b. Delete Computer objects
 - b. Scope: Descendant Computer objects
 - a. Read All Properties
 - b. Write All Properties
 - c. Read Permissions
 - d. Modify Permissions
 - e. Change Password
 - f. Reset Password
 - g. Validated write to DNS host name
 - h. Validated write to service principal name

Step 2: Set up the MDT production deployment share

When you are ready to deploy Windows 10 in a production environment, you will first create a new MDT deployment share. You should not use the same deployment share that you used to create the reference image for a production deployment. For guidance on creating a custom Windows 10 image, see [Create a Windows 10 reference image](#).

Create the MDT production deployment share

The steps for creating the deployment share for production are the same as when you created the deployment share for creating the custom reference image:

1. On MDT01, log on as Administrator in the CONTOSO domain using a password of **P@ssw0rd**.
2. Using the Deployment Workbench, right-click **Deployment Shares** and select **New Deployment Share**.
3. On the **Path** page, in the **Deployment share path** text box, type **E:\MDTProduction** and click **Next**.
4. On the **Share** page, in the **Share name** text box, type **MDTProduction\$** and click **Next**.
5. On the **Descriptive Name** page, in the **Deployment share description** text box, type **MDT Production** and click **Next**.
6. On the **Options** page, accept the default settings and click **Next** twice, and then click **Finish**.
7. Using File Explorer, verify that you can access the **\\MDT01\MDTProduction\$** share.

Step 3: Add a custom image

The next step is to add a reference image into the deployment share with the setup files required to successfully deploy Windows 10. When adding a custom image, you still need to copy setup files (an option in the wizard) because Windows 10 stores additional components in the Sources\SxS folder which is outside the image and may be required when installing components.

Add the Windows 10 Enterprise x64 RTM custom image

In these steps, we assume that you have completed the steps in the [Create a Windows 10 reference image](#) topic, so you have a Windows 10 reference image in the E:\MDTBuildLab\Captures folder on MDT01.

1. Using the Deployment Workbench, expand the **Deployment Shares** node, and then expand **MDT**

- Production**; select the **Operating Systems** node, and create a folder named **Windows 10**.
- Right-click the **Windows 10** folder and select **Import Operating System**.
 - On the **OS Type** page, select **Custom image file** and click **Next**.
 - On the **Image** page, in the **Source file** text box, browse to **E:\MDTBuildLab\Captures\REFW10X64-001.wim** and click **Next**.
 - On the **Setup** page, select the **Copy Windows 7, Windows Server 2008 R2, or later setup files from the specified path** option; in the **Setup source directory** text box, browse to **E:\MDTBuildLab\Operating Systems\W10EX64RTM** and click **Next**.
 - On the **Destination** page, in the **Destination directory name** text box, type **W10EX64RTM**, click **Next** twice, and then click **Finish**.
 - After adding the operating system, double-click the added operating system name in the **Operating Systems / Windows 10** node and change the name to match the following: **Windows 10 Enterprise x64 RTM Custom Image**.

NOTE

The reason for adding the setup files has changed since earlier versions of MDT. MDT 2010 used the setup files to install Windows. MDT uses DISM to apply the image; however, you still need the setup files because some components in roles and features are stored outside the main image.

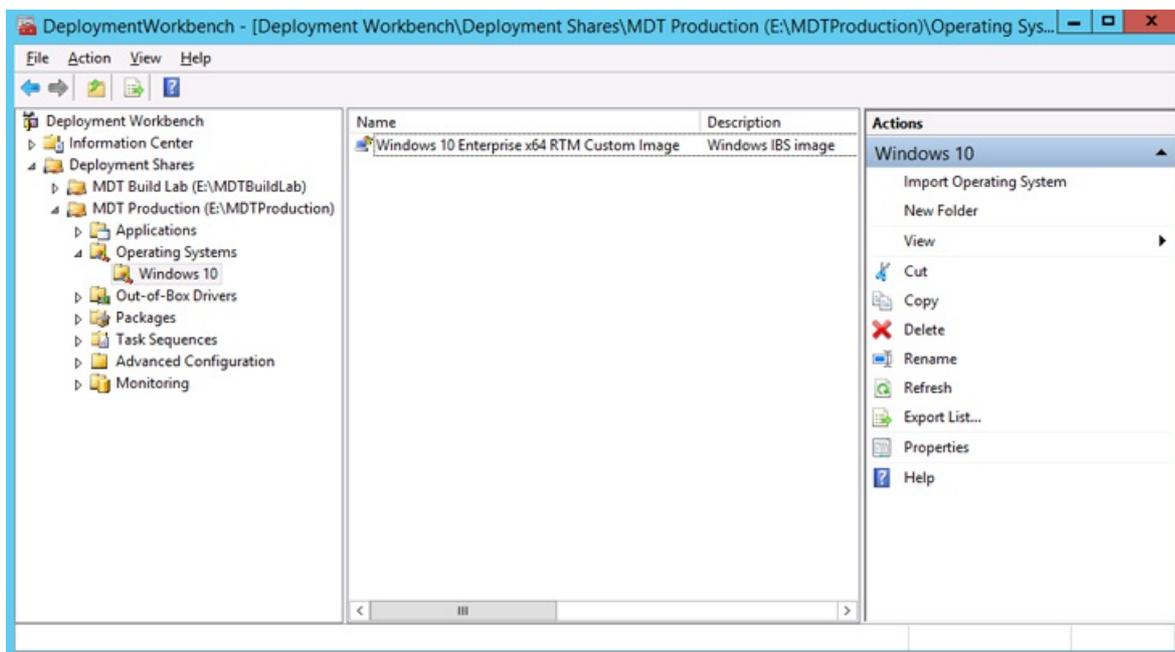


Figure 2. The imported operating system after renaming it.

Step 4: Add an application

When you configure your MDT Build Lab deployment share, you will also add any applications to the new deployment share before creating your task sequence. This section walks you through the process of adding an application to the MDT Production deployment share using Adobe Reader as an example.

Create the install: Adobe Reader XI x86

In this example, we assume that you have downloaded the Adobe Reader XI installation file (AdbeRdr11000_eu_ES.msi) to E:\Setup\Adobe Reader on MDT01.

- Using the Deployment Workbench, expand the **MDT Production** node and navigate to the **Applications** node.
- Right-click the **Applications** node, and create a new folder named **Adobe**.

3. In the **Applications** node, right-click the **Adobe** folder and select **New Application**.
4. On the **Application Type** page, select the **Application with source files** option and click **Next**.
5. On the **Details** page, in the **Application** name text box, type **Install - Adobe Reader XI - x86** and click **Next**.
6. On the **Source** page, in the **Source Directory** text box, browse to **E:\Setup\Adobe Reader XI** and click **Next**.
7. On the **Destination** page, in the **Specify the name of the directory that should be created** text box, type **Install - Adobe Reader XI - x86** and click **Next**.
8. On the **Command Details** page, in the **Command Line** text box, type **msiexec /i AdbRdr11000_eu_ES.msi /q**, click **Next** twice, and then click **Finish**.

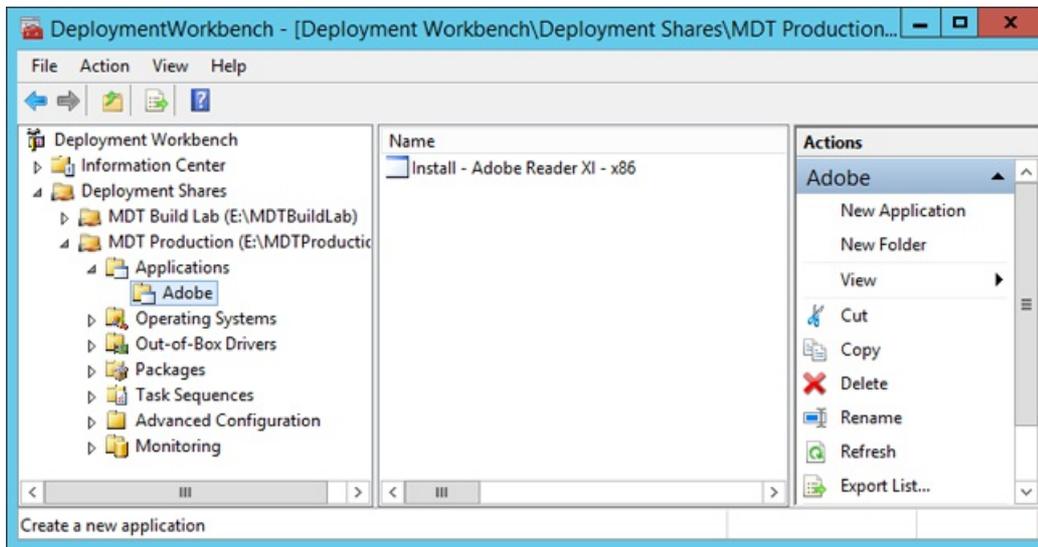


Figure 3. The Adobe Reader application added to the Deployment Workbench.

Step 5: Prepare the drivers repository

In order to deploy Windows 10 with MDT successfully, you need drivers for the boot images and for the actual operating system. This section will show you how to add drivers for the boot image and operating system, using the following hardware models as examples:

- Lenovo ThinkPad T420
- Dell Latitude E6440
- HP EliteBook 8560w
- Microsoft Surface Pro For boot images, you need to have storage and network drivers; for the operating system, you need to have the full suite of drivers.

NOTE

You should only add drivers to the Windows PE images if the default drivers don't work. Adding drivers that are not necessary will only make the boot image larger and potentially delay the download time.

Create the driver source structure in the file system

The key to successful management of drivers for MDT, as well as for any other deployment solution, is to have a really good driver repository. From this repository, you import drivers into MDT for deployment, but you should always maintain the repository for future use.

1. On MDT01, using File Explorer, create the **E:\Drivers** folder.
2. In the **E:\Drivers** folder, create the following folder structure:

- a. WinPE x86
 - b. WinPE x64
 - c. Windows 10 x64
3. In the new Windows 10 x64 folder, create the following folder structure:
- Dell
 - Latitude E6440
 - HP
 - HP EliteBook 8560w
 - Lenovo
 - ThinkPad T420 (4178)
 - Microsoft Corporation
 - Surface Pro 3

NOTE

Even if you are not going to use both x86 and x64 boot images, we still recommend that you add the support structure for future use.

Create the logical driver structure in MDT

When you import drivers to the MDT driver repository, MDT creates a single instance folder structure based on driver class names. However, you can, and should, mimic the driver structure of your driver source repository in the Deployment Workbench. This is done by creating logical folders in the Deployment Workbench.

1. On MDT01, using Deployment Workbench, select the **Out-of-Box Drivers** node.
2. In the **Out-Of-Box Drivers** node, create the following folder structure:
 - a. WinPE x86
 - b. WinPE x64
 - c. Windows 10 x64
3. In the **Windows 10 x64** folder, create the following folder structure:
 - Dell Inc.
 - Latitude E6440
 - Hewlett-Packard
 - HP EliteBook 8560w
 - Lenovo
 - 4178
 - Microsoft Corporation
 - Surface Pro 3

The preceding folder names are selected because they match the actual make and model values that MDT reads from the machines during deployment. You can find out the model values for your machines via the following command in Windows PowerShell:

```
Get-WmiObject -Class:Win32_ComputerSystem
```

Or, you can use this command in a normal command prompt:

```
wmic csproduct get name
```

If you want a more standardized naming convention, try the ModelAliasExit.vbs script from the Deployment Guys

blog post entitled [Using and Extending Model Aliases for Hardware Specific Application Installation](#).

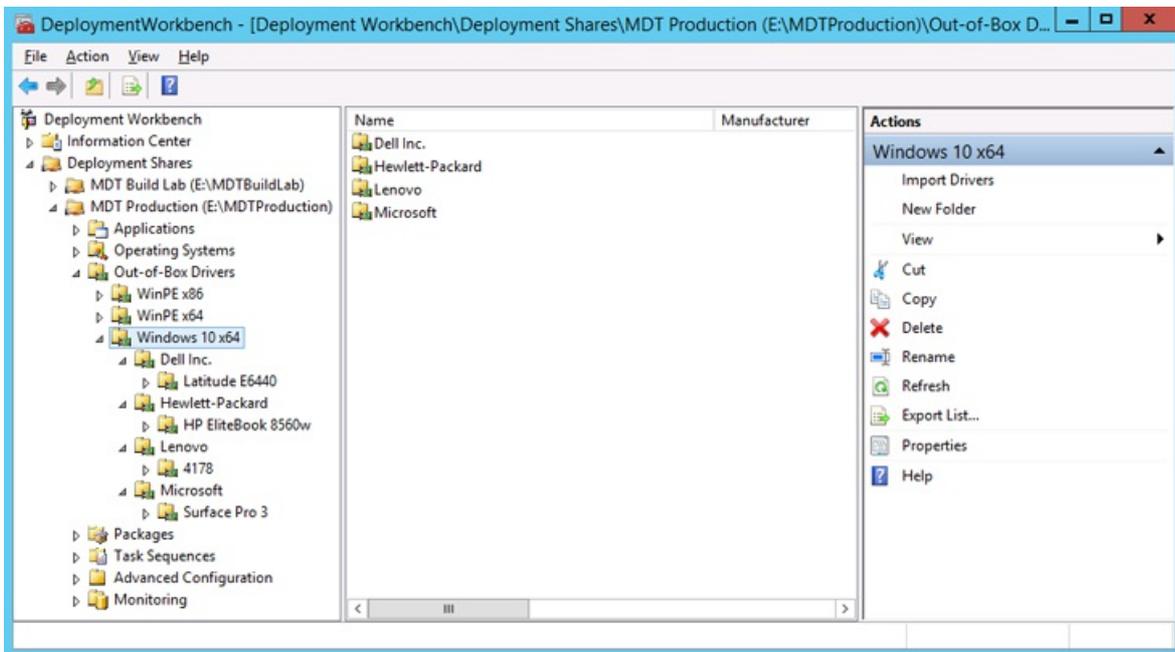


Figure 4. The Out-of-Box Drivers structure in Deployment Workbench.

Create the selection profiles for boot image drivers

By default, MDT adds any storage and network drivers that you import to the boot images. However, you should add only the drivers that are necessary to the boot image. You can control which drivers are added by using selection profiles. The drivers that are used for the boot images (Windows PE) are Windows 10 drivers. If you can't locate Windows 10 drivers for your device, a Windows 7 or Windows 8.1 driver will most likely work, but Windows 10 drivers should be your first choice.

1. On MDT01, using the Deployment Workbench, in the **MDT Production** node, expand the **Advanced Configuration** node, right-click the **Selection Profiles** node, and select **New Selection Profile**.
2. In the New Selection Profile Wizard, create a selection profile with the following settings:
 - a. Selection Profile name: WinPE x86
 - b. Folders: Select the WinPE x86 folder in Out-of-Box Drivers.
3. Again, right-click the **Selection Profiles** node, and select **New Selection Profile**.
4. In the New Selection Profile Wizard, create a selection profile with the following settings:
 - a. Selection Profile name: WinPE x64
 - b. Folders: Select the WinPE x64 folder in Out-of-Box Drivers.

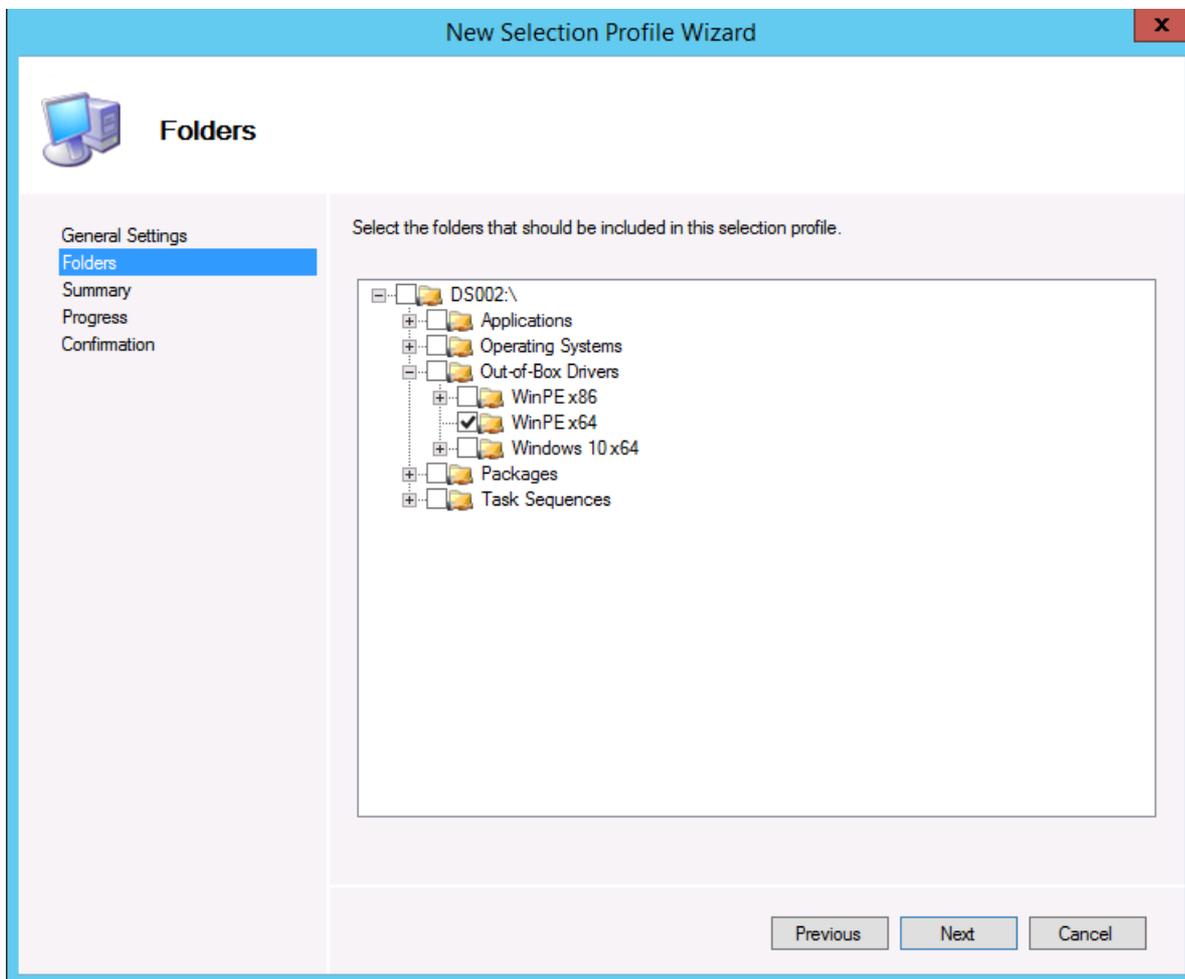


Figure 5. Creating the WinPE x64 selection profile.

Extract and import drivers for the x64 boot image

Windows PE supports all the hardware models that we have, but here you learn to add boot image drivers to accommodate any new hardware that might require additional drivers. In this example, you add the latest Intel network drivers to the x64 boot image. In these steps, we assume you have downloaded PROWinx64.exe from Intel.com and saved it to a temporary folder.

1. Extract PROWinx64.exe to a temporary folder - in this example to the **C:\Tmp\ProWinx64** folder.
2. Using File Explorer, create the **E:\Drivers\WinPE x64\Intel PRO1000** folder.
3. Copy the content of the **C:\Tmp\PROWinx64\PRO1000\Winx64\NDIS64** folder to the **E:\Drivers\WinPE x64\Intel PRO1000** folder.
4. Using Deployment Workbench, expand the **Out-of-Box Drivers** node, right-click the **WinPE x64** node, and select **Import Drivers**. Use the following setting for the Import Drivers Wizard:
 - Driver source directory: **E:\Drivers\WinPE x64\Intel PRO1000**

Download, extract, and import drivers

For the ThinkPad T420

For the Lenovo T420 model, you use the Lenovo ThinkVantage Update Retriever software to download the drivers. With Update Retriever, you need to specify the correct Lenovo Machine Type for the actual hardware (the first four characters of the model name). As an example, the Lenovo T420 model has the 4178B9G model name, meaning the Machine Type is 4178.

To get the updates, you download the drivers from the Lenovo ThinkVantage Update Retriever using its export function. You can download the drivers from the [Lenovo website](#).

In these steps, we assume you have downloaded and extracted the drivers using ThinkVantage Update Retriever v5.0 to the E:\Drivers\Lenovo\ThinkPad T420 (4178) folder.

1. On MDT01, using the Deployment Workbench, in the **MDT Production** node, expand the **Out-Of-Box Drivers** node, and expand the **Lenovo** node.
2. Right-click the **4178** folder and select **Import Drivers**; use the following setting for the Import Drivers Wizard:
 - Driver source directory: **E:\Drivers\Windows 10 x64\Lenovo\ThinkPad T420 (4178)**

For the Latitude E6440

For the Dell Latitude E6440 model, you use the Dell Driver CAB file, which is accessible via the [Dell TechCenter website](#).

In these steps, we assume you have downloaded and extracted the CAB file for the Latitude E6440 model to the E:\Drivers\Dell\Latitude E6440 folder.

1. On **MDT01**, using the **Deployment Workbench**, in the **MDT Production** node, expand the **Out-Of-Box Drivers** node, and expand the **Dell** node.
2. Right-click the **Latitude E6440** folder and select **Import Drivers**; use the following setting for the Import Drivers Wizard:
 - Driver source directory: **E:\Drivers\Windows 10 x64\Dell\Latitude E6440**

For the HP EliteBook 8560w

For the HP EliteBook 8560w, you use HP SoftPaq Download Manager to get the drivers. The HP SoftPaq Download Manager can be accessed on the [HP Support site](#).

In these steps, we assume you have downloaded and extracted the drivers for the HP EliteBook 8650w model to the E:\Drivers\Windows 10 x64\HP\HP EliteBook 8560w folder.

1. On **MDT01**, using the **Deployment Workbench**, in the **MDT Production** node, expand the **Out-Of-Box Drivers** node, and expand the **Hewlett-Packard** node.
2. Right-click the **HP EliteBook 8560w** folder and select **Import Drivers**; use the following setting for the Import Drivers Wizard:
 - Driver source directory: **E:\Drivers\Windows 10 x64\HP\HP EliteBook 8560w**

For the Microsoft Surface Pro 3

For the Microsoft Surface Pro model, you find the drivers on the Microsoft website. In these steps we assume you have downloaded and extracted the Surface Pro 3 drivers to the E:\Drivers\Windows 10 x64\Microsoft\Surface Pro 3 folder.

1. On MDT01, using the Deployment Workbench, in the **MDT Production** node, expand the **Out-Of-Box Drivers** node, and expand the **Microsoft** node.
2. Right-click the **Surface Pro 3** folder and select **Import Drivers**; use the following setting for the Import Drivers Wizard:
 - Driver source directory: **E:\Drivers\Windows 10 x64\Microsoft\Surface Pro 3**

Step 6: Create the deployment task sequence

This section will show you how to create the task sequence used to deploy your production Windows 10 reference image. You will then configure the tasks sequence to enable patching via a Windows Server Update Services (WSUS) server.

Create a task sequence for Windows 10 Enterprise

1. Using the Deployment Workbench, select **Task Sequences** in the **MDT Production** node, and create a folder named **Windows 10**.
2. Right-click the new **Windows 10** folder and select **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:

- a. Task sequence ID: W10-X64-001
- b. Task sequence name: Windows 10 Enterprise x64 RTM Custom Image
- c. Task sequence comments: Production Image
- d. Template: Standard Client Task Sequence
- e. Select OS: Windows 10 Enterprise x64 RTM Custom Image
- f. Specify Product Key: Do not specify a product key at this time
- g. Full Name: Contoso
- h. Organization: Contoso
- i. Internet Explorer home page: about:blank
- j. Admin Password: Do not specify an Administrator Password at this time

Edit the Windows 10 task sequence

3. Right-click the **Windows 10 Enterprise x64 RTM Custom Image** task sequence, and select **Properties**.
4. On the **Task Sequence** tab, configure the **Windows 10 Enterprise x64 RTM Custom Image** task sequence with the following settings:
 - a. Preinstall. After the **Enable BitLocker (Offline)** action, add a **Set Task Sequence Variable** action with the following settings:
 - a. Name: Set DriverGroup001
 - b. Task Sequence Variable: DriverGroup001
 - c. Value: Windows 10 x64\%Make%\%Model%
 - b. Configure the **Inject Drivers** action with the following settings:
 - a. Choose a selection profile: Nothing
 - b. Install all drivers from the selection profile

NOTE

The configuration above indicates that MDT should only use drivers from the folder specified by the DriverGroup001 property, which is defined by the "Choose a selection profile: Nothing" setting, and that MDT should not use plug and play to determine which drivers to copy, which is defined by the "Install all drivers from the selection profile" setting.

- c. State Restore. Enable the **Windows Update (Pre-Application Installation)** action.
 - d. State Restore. Enable the **Windows Update (Post-Application Installation)** action.
5. Click **OK**.

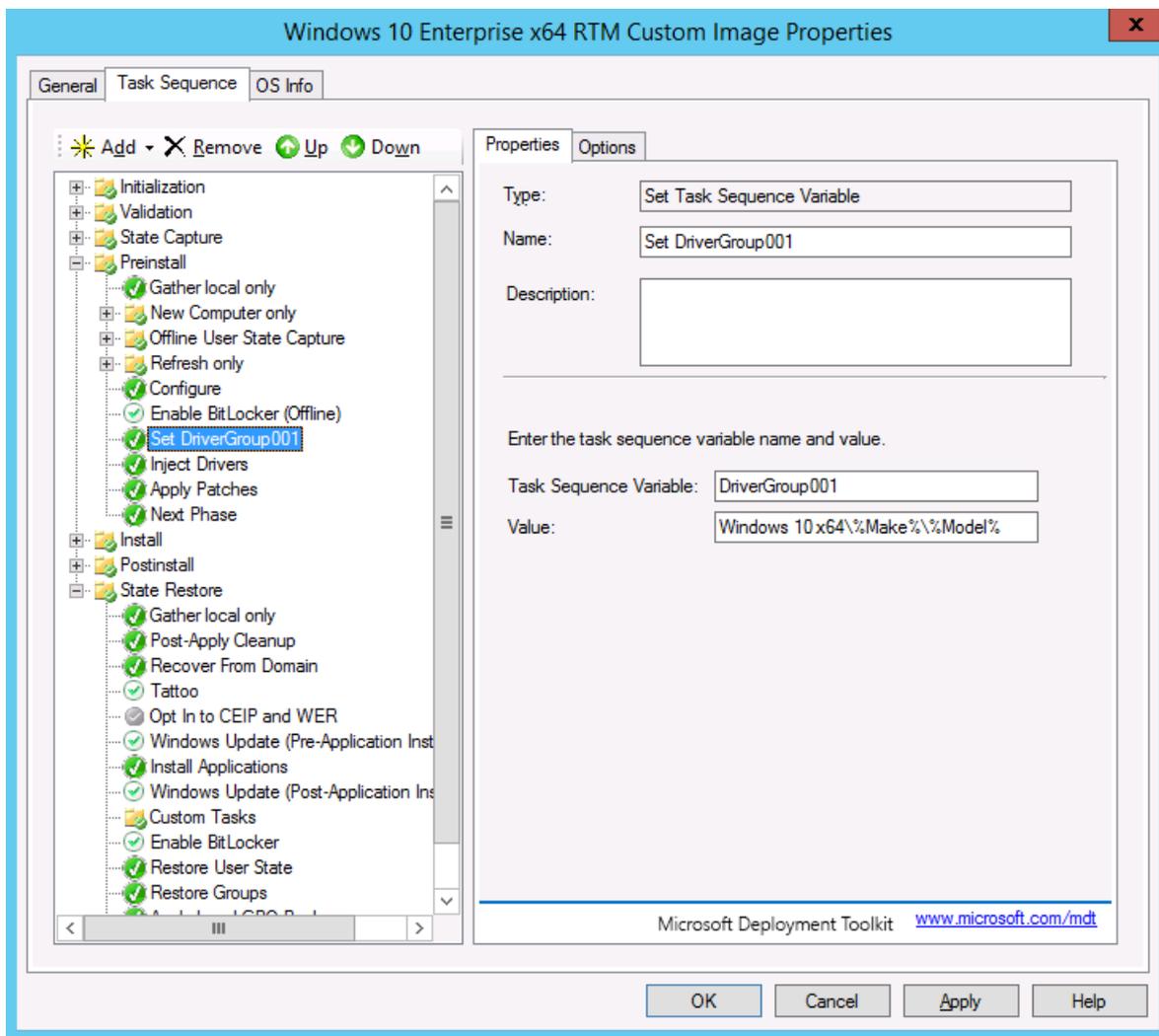


Figure 6. The task sequence for production deployment.

Step 7: Configure the MDT production deployment share

In this section, you will learn how to configure the MDT Build Lab deployment share with the rules required to create a simple and dynamic deployment process. This includes configuring commonly used rules and an explanation of how these rules work.

Configure the rules

1. On MDT01, using File Explorer, copy the following files from the **D:\Setup\Sample Files\MDT Production\Control** folder to **E:\MDTProduction\Control**. Overwrite the existing files.
 - a. Bootstrap.ini
 - b. CustomSettings.ini
2. Right-click the **MDT Production** deployment share and select **Properties**.
3. Select the **Rules** tab and modify using the following information:

```

[Settings]
Priority=Default
[Default]
_SMSTSORGNAME=Contoso
OSInstall=YES
UserDataLocation=AUTO
TimeZoneName=Pacific Standard Time
AdminPassword=P@ssw0rd
JoinDomain=contoso.com
DomainAdmin=CONTOSO\MDT_JD
DomainAdminPassword=P@ssw0rd
MachineObjectOU=OU=Workstations,OU=Computers,OU=Contoso,DC=contoso,DC=com
SLShare=\\MDT01\Logos$
ScanStateArgs=/ue:* \* /ui:CONTOSO\*
USMTMigFiles001=MigApp.xml
USMTMigFiles002=MigUser.xml
HideShell=YES
ApplyGPOPack=NO
WSUSServer=mdt01.contoso.com:8530
SkipAppsOnUpgrade=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=NO
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=NO
SkipBitLocker=YES
SkipSummary=YES
SkipCapture=YES
SkipFinalSummary=NO

```

4. Click **Edit Bootstrap.ini** and modify using the following information:

```

[Settings]
Priority=Default
[Default]
DeployRoot=\\MDT01\MDTProduction$
UserDomain=CONTOSO
UserID=MDT_BA
SkipBDDWelcome=YES

```

5. In the **Windows PE** tab, in the **Platform** drop-down list, make sure **x86** is selected.
6. In the **General** sub tab, configure the following settings:
- In the **Lite Touch Boot Image Settings** area:
 - a. Image description: MDT Production x86
 - b. ISO file name: MDT Production x86.iso

NOTE

Because you are going to use Pre-Boot Execution Environment (PXE) later to deploy the machines, you do not need the ISO file; however, we recommend creating ISO files because they are useful when troubleshooting deployments and for quick tests.

7. In the **Drivers and Patches** sub tab, select the **WinPE x86** selection profile and select the **Include all drivers from the selection profile** option.

8. In the **Windows PE** tab, in the **Platform** drop-down list, select **x64**.
9. In the **General** sub tab, configure the following settings:
 - In the **Lite Touch Boot Image Settings** area:
 - a. Image description: MDT Production x64
 - b. ISO file name: MDT Production x64.iso
10. In the **Drivers and Patches** sub tab, select the **WinPE x64** selection profile and select the **Include all drivers from the selection profile** option.
11. In the **Monitoring** tab, select the **Enable monitoring for this deployment share** check box.
12. Click **OK**.

NOTE

It will take a while for the Deployment Workbench to create the monitoring database and web service.

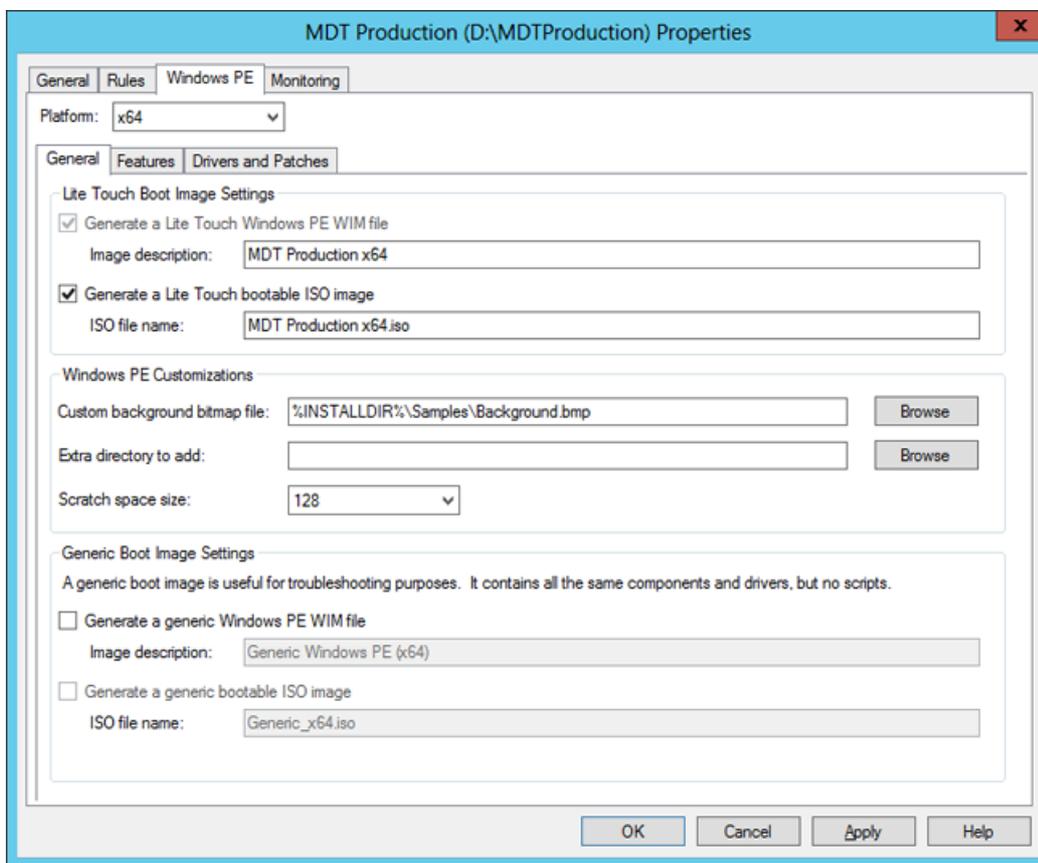


Figure 7. The Windows PE tab for the x64 boot image.

The rules explained

The rules for the MDT Production deployment share are somewhat different from those for the MDT Build Lab deployment share. The biggest differences are that you deploy the machines into a domain instead of a workgroup and that you do not automate the logon.

The Bootstrap.ini file

This is the MDT Production Bootstrap.ini without the user credentials (except domain information):

```
[Settings]
Priority=Default
[Default]
DeployRoot=\\MDT01\MDTProduction$
UserDomain=CONTOSO
UserID=MDT_BA
SkipBDDWelcome=YES
```

The CustomSettings.ini file

This is the CustomSettings.ini file with the new join domain information:

```
[Settings]
Priority=Default
[Default]
_SMSTSORGNAME=Contoso
OSInstall=Y
UserDataLocation=AUTO
TimeZoneName=Pacific Standard Time
AdminPassword=P@ssw0rd
JoinDomain=contoso.com
DomainAdmin=CONTOSO\MDT_JD
DomainAdminPassword=P@ssw0rd
MachineObjectOU=OU=Workstations,OU=Computers,OU=Contoso,DC=contoso,DC=com
SLShare=\\MDT01\Logs$
ScanStateArgs=/ue:* \* /ui:CONTOSO\*
USMTMigFiles001=MigApp.xml
USMTMigFiles002=MigUser.xml
HideShell=YES
ApplyGPOPack=NO
WSUSServer=http://mdt01.contoso.com:8530
SkipAppsOnUpgrade=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=NO
SkipDomainMembership=YES
SkipUserData=YES
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=NO
SkipBitLocker=YES
SkipSummary=YES
SkipCapture=YES
SkipFinalSummary=NO
EventService=http://MDT01:9800
```

The additional properties to use in the MDT Production rules file are as follows:

- **JoinDomain.** The domain to join.
- **DomainAdmin.** The account to use when joining the machine to the domain.
- **DomainAdminDomain.** The domain for the join domain account.
- **DomainAdminPassword.** The password for the join domain account.
- **MachineObjectOU.** The organizational unit (OU) to which to add the computer account.
- **ScanStateArgs.** Arguments for the User State Migration Tool (USMT) ScanState command.
- **USMTMigFiles(*).** List of USMT templates (controlling what to backup and restore).
- **EventService.** Activates logging information to the MDT monitoring web service.

Optional deployment share configuration

If your organization has a Microsoft Software Assurance agreement, you also can subscribe to the additional Microsoft Desktop Optimization Package (MDOP) license (at an additional cost). Included in MDOP is Microsoft

Diagnostics and Recovery Toolkit (DaRT), which contains tools that can help you troubleshoot MDT deployments, as well as troubleshoot Windows itself.

Add DaRT 10 to the boot images

If you have licensing for MDOP and DaRT, you can add DaRT to the boot images using the steps in this section. If you do not have DaRT licensing, or don't want to use it, simply skip to the next section, [Update the Deployment Share](#). To enable the remote connection feature in MDT, you need to do the following:

- Install DaRT 10 (part of MDOP 2015 R1).
- Copy the two tools CAB files (Toolsx86.cab and Toolsx64.cab) to the deployment share.
- Configure the deployment share to add DaRT. In these steps, we assume that you downloaded MDOP 2015 R1 and copied DaRT 10 to the E:\Setup\DaRT 10 folder on MDT01.
- On MDT01, install DaRT 10 (MSDaRT10.msi) using the default settings.
- Using File Explorer, navigate to the **C:\Program Files\Microsoft DaRT\v10** folder.
- Copy the Toolsx64.cab file to **E:\MDTProduction\Tools\x64**.
- Copy the Toolsx86.cab file to **E:\MDTProduction\Tools\x86**.
- Using the Deployment Workbench, right-click the **MDT Production** deployment share and select **Properties**.
- In the **Windows PE** tab, in the **Platform** drop-down list, make sure **x86** is selected.
- In the **Features** sub tab, select the **Microsoft Diagnostics and Recovery Toolkit (DaRT)** check box.

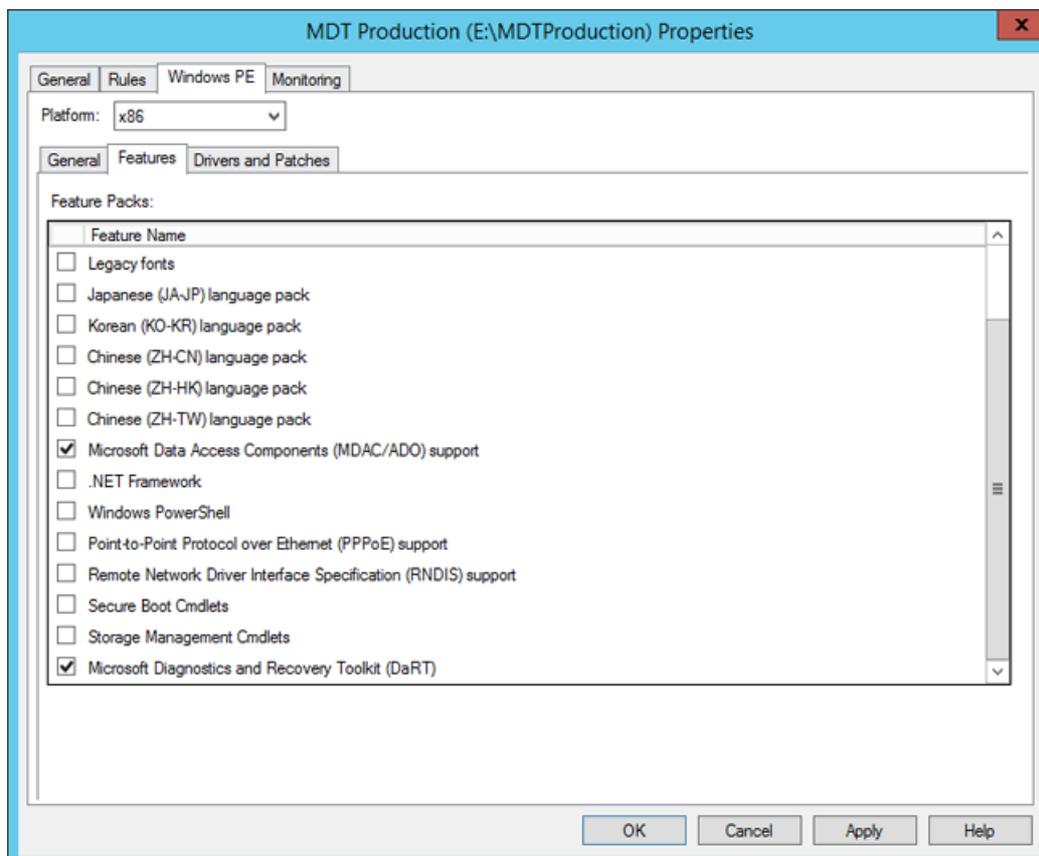


Figure 8. Selecting the DaRT 10 feature in the deployment share.

8. In the **Windows PE** tab, in the **Platform** drop-down list, select **x64**.
9. In the **Features** sub tab, in addition to the default selected feature pack, select the **Microsoft Diagnostics and Recovery Toolkit (DaRT)** check box.

10. Click **OK**.

Update the deployment share

Like the MDT Build Lab deployment share, the MDT Production deployment share needs to be updated after it has been configured. This is the process during which the Windows PE boot images are created.

1. Right-click the **MDT Production** deployment share and select **Update Deployment Share**.
2. Use the default options for the Update Deployment Share Wizard.

NOTE

The update process will take 5 to 10 minutes.

Step 8: Deploy the Windows 10 client image

These steps will walk you through the process of using task sequences to deploy Windows 10 images through a fully automated process. First, you need to add the boot image to Windows Deployment Services (WDS) and then start the deployment. In contrast with deploying images from the MDT Build Lab deployment share, we recommend using the Pre-Installation Execution Environment (PXE) to start the full deployments in the datacenter, even though you technically can use an ISO/CD or USB to start the process.

Configure Windows Deployment Services

You need to add the MDT Production Lite Touch x64 Boot image to WDS in preparation for the deployment. For the following steps, we assume that Windows Deployment Services has already been installed on MDT01.

1. Using the WDS console, right-click **Boot Images** and select **Add Boot Image**.
2. Browse to the E:\MDTProduction\Boot\LiteTouchPE_x64.wim file and add the image with the default settings.

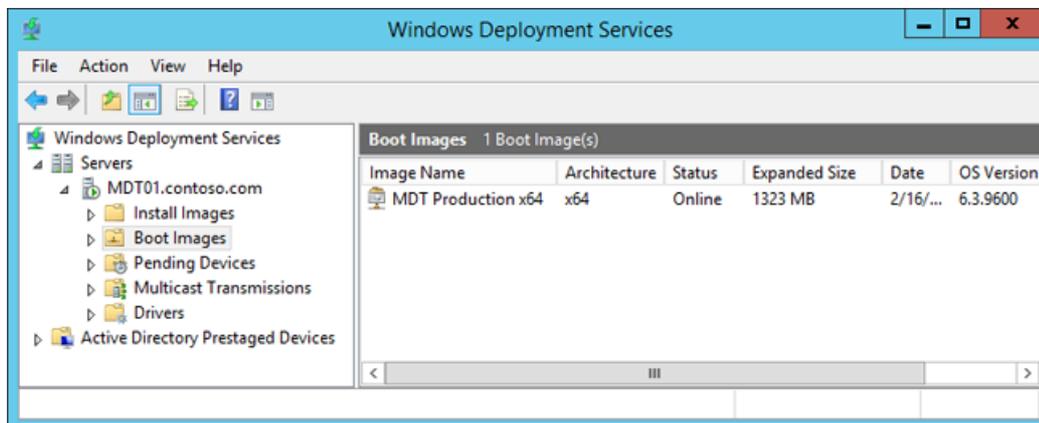


Figure 9. The boot image added to the WDS console.

Deploy the Windows 10 client

At this point, you should have a solution ready for deploying the Windows 10 client. We recommend starting by trying a few deployments at a time until you are confident that your configuration works as expected. We find it useful to try some initial tests on virtual machines before testing on physical hardware. This helps rule out hardware issues when testing or troubleshooting. Here are the steps to deploy your Windows 10 image to a virtual machine:

1. Create a virtual machine with the following settings:
 - a. Name: PC0005
 - b. Location: C:\VMs
 - c. Generation: 2

- d. Memory: 2048 MB
 - e. Hard disk: 60 GB (dynamic disk)
2. Start the PC0005 virtual machine, and press **Enter** to start the PXE boot. The machine will now load the Windows PE boot image from the WDS server.



Figure 10. The initial PXE boot process of PC0005.

3. After Windows PE has booted, complete the Windows Deployment Wizard using the following setting:
 - a. Password: P@ssw0rd
 - b. Select a task sequence to execute on this computer: Windows 10 Enterprise x64 RTM Custom Image
 - c. Computer Name: PC0005
 - d. Applications: Select the Install - Adobe Reader XI - x86 application.
4. The setup now starts and does the following:
 - a. Installs the Windows 10 Enterprise operating system.
 - b. Installs the added application.
 - c. Updates the operating system via your local Windows Server Update Services (WSUS) server.

Use the MDT monitoring feature

Now that you have enabled the monitoring on the MDT Production deployment share, you can follow your deployment of PC0005 via the monitoring node.

1. On MDT01, using Deployment Workbench, expand the **MDT Production** deployment share folder.
2. Select the **Monitoring** node, and wait until you see PC0005.
3. Double-click PC0005, and review the information.

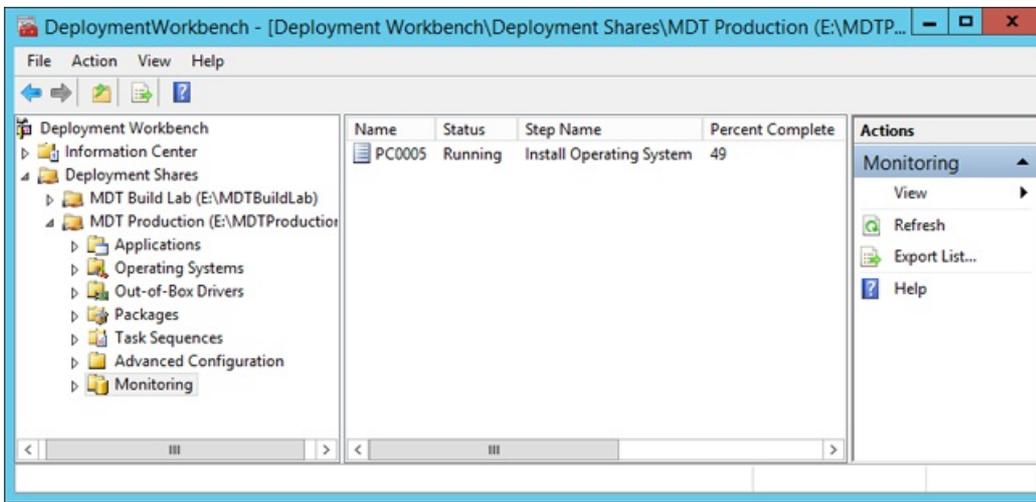


Figure 11. The Monitoring node, showing the deployment progress of PC0005.

Use information in the Event Viewer

When monitoring is enabled, MDT also writes information to the event viewer on MDT01. This information can be used to trigger notifications via scheduled tasks when deployment is completed. For example, you can configure scheduled tasks to send an email when a certain event is created in the event log.

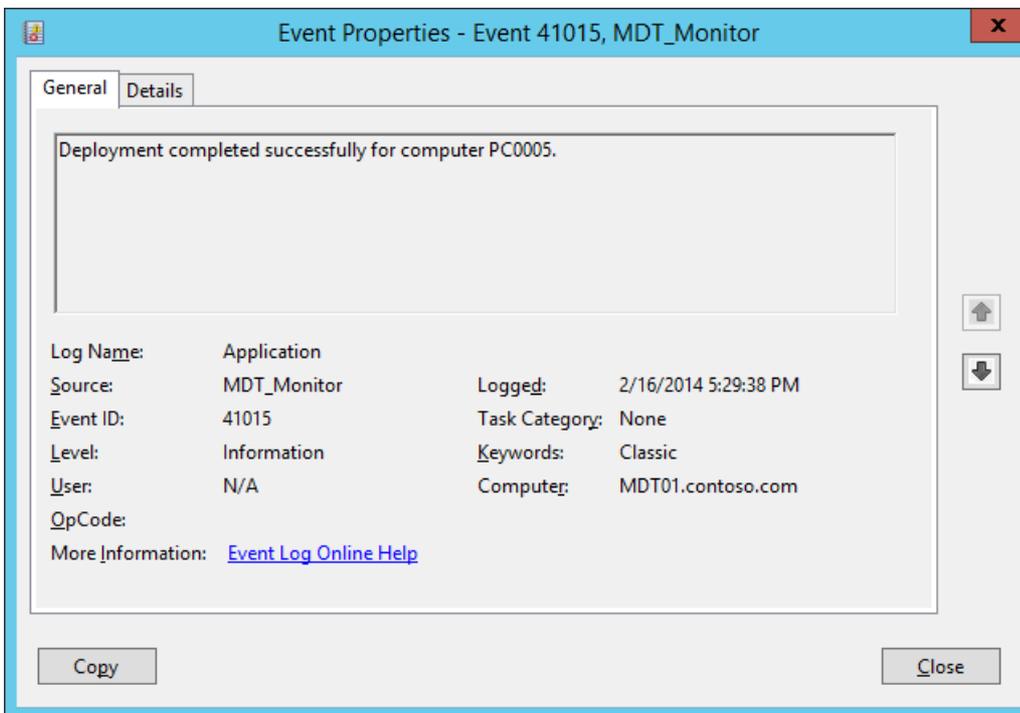


Figure 12. The Event Viewer showing a successful deployment of PC0005.

Multicast deployments

Multicast deployment allows for image deployment with reduced network load during simultaneous deployments. Multicast is a useful operating system deployment feature in MDT deployments, however it is important to ensure that your network supports it and is designed for it.

Requirements

Multicast requires that Windows Deployment Services (WDS) is running on Windows Server 2008 or later. In addition to the core MDT setup for multicast, the network needs to be configured to support multicast. In general, this means involving the organization networking team to make sure that Internet Group Management Protocol (IGMP) snooping is turned on and that the network is designed for multicast traffic. The multicast solution uses IGMPv3.

Set up MDT for multicast

Setting up MDT for multicast is straightforward. You enable multicast on the deployment share, and MDT takes care of the rest.

1. On MDT01, right-click the **MDT Production** deployment share folder and select **Properties**.
2. In the **General** tab, select the **Enable multicast for this deployment share (requires Windows Server 2008 R2 Windows Deployment Services)** check box, and click **OK**.
3. Right-click the **MDT Production** deployment share folder and select **Update Deployment Share**.
4. After updating the deployment share, use the Windows Deployment Services console to verify that the multicast namespace was created.

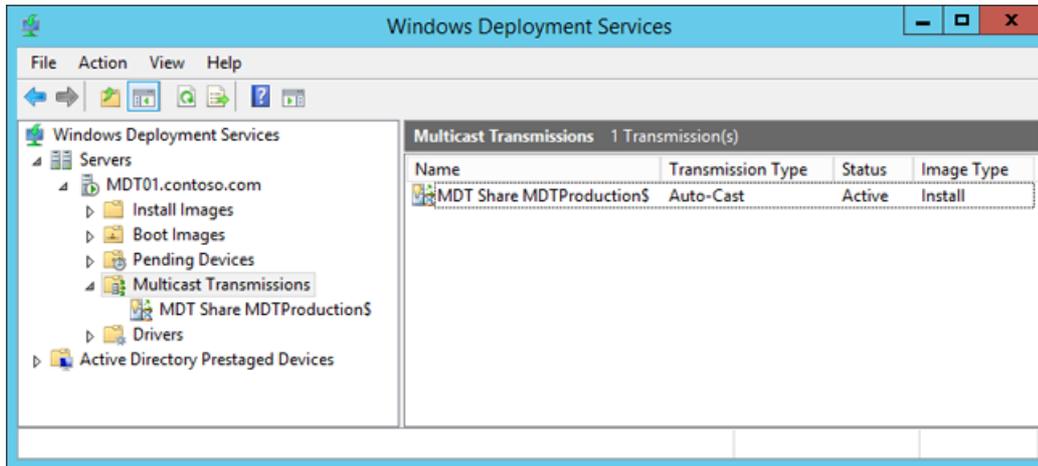


Figure 13. The newly created multicast namespace.

Use offline media to deploy Windows 10

In addition to network-based deployments, MDT supports the use of offline media-based deployments of Windows 10. You can very easily generate an offline version of your deployment share - either the full deployment share or a subset of it - by the use of selection profiles. The generated offline media can be burned to a DVD or copied to a USB stick for deployment.

Offline media are useful not only when you do not have network connectivity to the deployment share, but also when you have limited connection to the deployment share and do not want to copy 5 GB of data over the wire. Offline media can still join the domain, but you save the transfer of operating system images, drivers, and applications over the wire.

Create the offline media selection profile

To filter what is being added to the media, you create a selection profile. When creating selection profiles, you quickly realize the benefits of having created a good logical folder structure in the Deployment Workbench.

1. On MDT01, using Deployment Workbench, in the **MDT Production / Advanced Configuration** node, right-click **Selection Profile**, and select **New Selection Profile**.
2. Use the following settings for the New Selection Profile Wizard:
 - a. General Settings
 - Selection profile name: Windows 10 Offline Media
 - b. Folders
 - a. Applications / Adobe
 - b. Operating Systems / Windows 10
 - c. Out-Of-Box Drivers / WinPE x64
 - d. Out-Of-Box Drivers / Windows 10 x64
 - e. Task Sequences / Windows 10

Create the offline media

In these steps, you generate offline media from the MDT Production deployment share. To filter what is being added to the media, you use the previously created selection profile.

1. On MDT01, using File Explorer, create the **E:\MDTOfflineMedia** folder.

NOTE

When creating offline media, you need to create the target folder first. It is crucial that you do not create a subfolder inside the deployment share folder because it will break the offline media.

2. Using Deployment Workbench, in the **MDT Production / Advanced Configuration** node, right-click the **Media** node, and select **New Media**.
3. Use the following settings for the New Media Wizard:
 - General Settings
 - a. Media path: **E:\MDTOfflineMedia**
 - b. Selection profile: Windows 10 Offline Media

Configure the offline media

Offline media has its own rules, its own Bootstrap.ini and CustomSettings.ini files. These files are stored in the Control folder of the offline media; they also can be accessed via properties of the offline media in the Deployment Workbench.

1. On MDT01, using File Explorer, copy the CustomSettings.ini file from the **E:\MDTProduction\Control** folder to **E:\MDTOfflineMedia\Content\Deploy\Control**. Overwrite the existing files.
2. Using Deployment Workbench, in the **MDT Production / Advanced Configuration / Media** node, right-click the **MEDIA001** media, and select **Properties**.
3. In the **General** tab, configure the following:
 - a. Clear the Generate x86 boot image check box.
 - b. ISO file name: Windows 10 Offline Media.iso
4. Still in the **Windows PE** tab, in the **Platform** drop-down list, select **x64**.
5. In the **General** sub tab, configure the following settings:
 - a. In the **Lite Touch Boot Image Settings** area:
 - Image description: MDT Production x64
 - b. In the **Windows PE Customizations** area, set the Scratch space size to 128.
6. In the **Drivers and Patches** sub tab, select the **WinPE x64** selection profile and select the **Include all drivers from the selection profile** option.
7. Click **OK**.

Generate the offline media

You have now configured the offline media deployment share however the share has not yet been populated with the files required for deployment. Now everything is ready you populate the deployment share content folder and generate the offline media ISO.

1. On MDT01, using Deployment Workbench, navigate to the **MDT Production / Advanced Configuration / Media** node.
2. Right-click the **MEDIA001** media, and select **Update Media Content**. The Update Media Content process now generates the offline media in the **E:\MDTOfflineMedia\Content** folder.

Create a bootable USB stick

The ISO that you got when updating the offline media item can be burned to a DVD and used directly (it will be

bootable), but it is often more efficient to use USB sticks instead since they are faster and can hold more data. (A dual-layer DVD is limited to 8.5 GB.) Follow these steps to create a bootable USB stick from the offline media content:

1. On a physical machine running Windows 7 or later, insert the USB stick you want to use.
2. Copy the content of the **MDTOfflineMedia\Content** folder to the root of the USB stick.
3. Start an elevated command prompt (run as Administrator), and start the Diskpart utility by typing **Diskpart** and pressing **Enter**.
4. In the Diskpart utility, you can type **list volume** (or the shorter **list vol**) to list the volumes, but you really only need to remember the drive letter of the USB stick to which you copied the content. In our example, the USB stick had the drive letter F.
5. In the Diskpart utility, type **select volume F** (replace F with your USB stick drive letter).
6. In the Diskpart utility, type **active**, and then type **exit**.

Unified Extensible Firmware Interface (UEFI)-based deployments

As referenced in [Windows 10 deployment tools](#), Unified Extensible Firmware Interface (UEFI)-based deployments are becoming more common. In fact, when you create a generation 2 virtual machine in Hyper-V, you get a UEFI-based computer. During deployment, MDT automatically detects that you have an UEFI-based machine and creates the partitions UEFI requires. You do not need to update or change your task sequences in any way to accommodate UEFI.

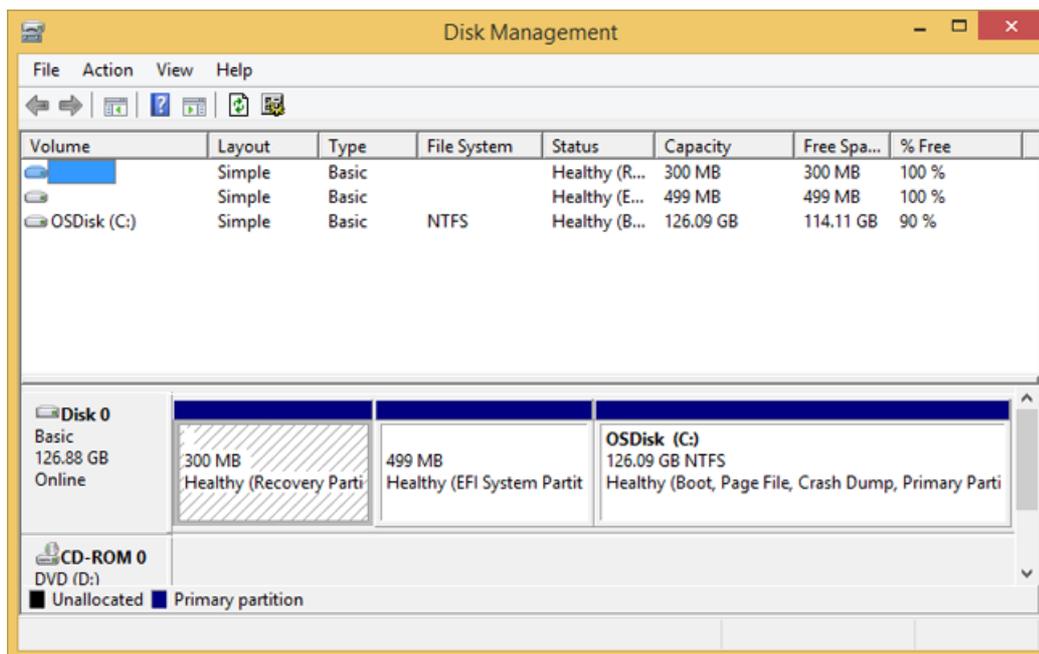


Figure 14. The partitions when deploying an UEFI-based machine.

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Create a Windows 10 reference image](#)

[Build a distributed environment for Windows 10 deployment](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

[Configure MDT settings](#)

Build a distributed environment for Windows 10 deployment

6/14/2019 • 9 minutes to read • [Edit Online](#)

Applies to

- Windows 10

In this topic, you will learn how to replicate your Windows 10 deployment shares to facilitate the deployment of Windows 10 in remote or branch locations. If you work in a distributed environment, replicating the deployment shares is an important part of the deployment solution. With images reaching 5 GB in size or more, you can't deploy machines in a remote office over the wire. You need to replicate the content, so that the clients can do local deployments.

We will use four machines for this topic: DC01, MDT01, MDT02, and PC0006. DC01 is a domain controller, MDT01 is a Windows Server 2012 R2 standard server, and PC0006 is a blank machine to which you will deploy Windows 10. You will configure a second deployment server (MDT02) for a remote site (Stockholm) by replicating the deployment share in the original site (New York). MDT01, MDT02, and PC0006 are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

Replicate deployment shares

Replicating the content between MDT01 (New York) and MDT02 (Stockholm) can be done in a number of different ways. The most common content replication solutions with Microsoft Deployment Toolkit (MDT) use either the Linked Deployment Shares (LDS) feature or Distributed File System Replication (DFS-R). Some organizations have used a simple robocopy script for replication of the content.

Note Robocopy has options that allow for synchronization between folders. It has a simple reporting function; it supports transmission retry; and, by default, it will only copy/remove files from the source that are newer than files on the target.

Linked deployment shares in MDT

LDS is a built-in feature in MDT for replicating content. However, LDS works best with strong connections such as LAN connections with low latency. For most WAN links, DFS-R is the better option.

Why DFS-R is a better option

DFS-R is not only very fast and reliable, but it also offers central monitoring, bandwidth control, and a great delta replication engine. DFS-R will work equally well whether you have 2 sites or 90. When using DFS-R for MDT, we recommend running your deployment servers on Windows Server 2008 R2 or higher. From that version on, you can configure the replication target(s) as read-only, which is exactly what you want for MDT. This way, you can have your master deployment share centralized and replicate out changes as they happen. DFS-R will quickly pick up changes at the central deployment share in MDT01 and replicate the delta changes to MDT02.

Set up Distributed File System Replication (DFS-R) for replication

Setting up DFS-R for replication is a quick and straightforward process. You prepare the deployment servers and then create a replication group. To complete the setup, you configure some replication settings.

Prepare MDT01 for replication

1. On MDT01, using Server Manager, click **Add roles and features**.
2. On the **Select installation type** page, select **Role-based or feature-based installation**.
3. On the **Select destination server** page, select **MDT01.contoso.com** and click **Next**.
4. On the **Select server roles** page, expand **File and Storage Services (Installed)** and expand **File and iSCSI Services (Installed)**.
5. In the **Roles** list, select **DFS Replication**. In the **Add Roles and Features Wizard** dialog box, select **Add Features**, and then click **Next**.

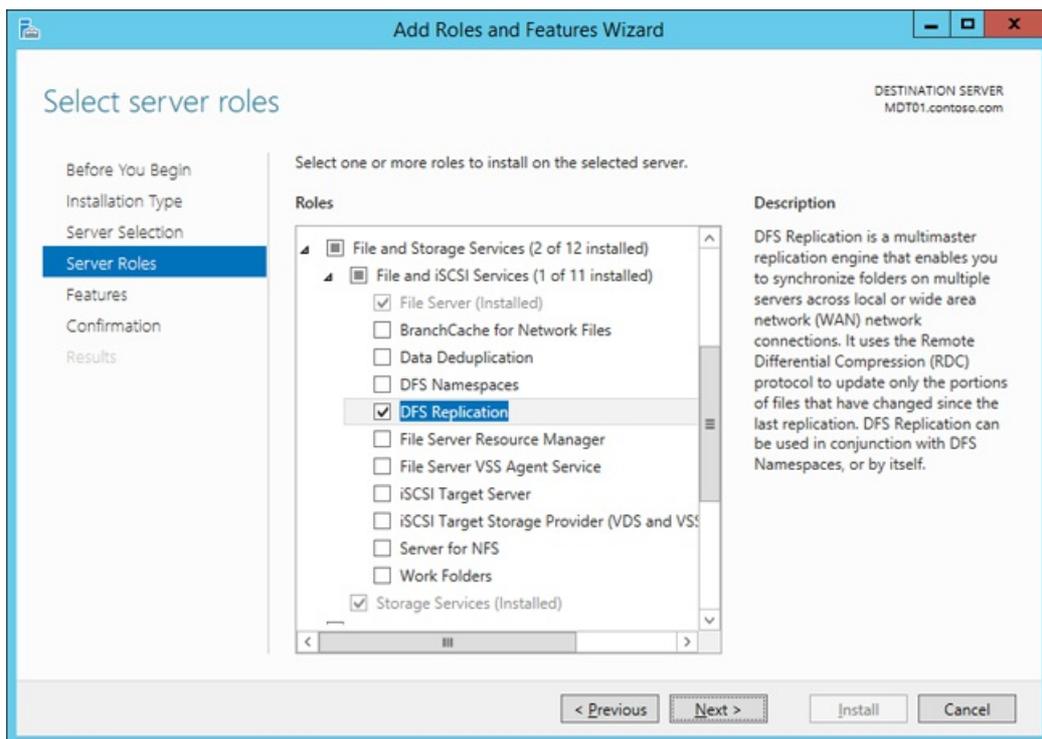


Figure 2. Adding the DFS Replication role to MDT01.

6. On the **Select features** page, accept the default settings, and click **Next**.
7. On the **Confirm installation selections** page, click **Install**.
8. On the **Installation progress** page, click **Close**.

Prepare MDT02 for replication

1. On MDT02, using Server Manager, click **Add roles and features**.
2. On the **Select installation type** page, select **Role-based or feature-based installation**.
3. On the **Select destination server** page, select **MDT02.contoso.com** and click **Next**.
4. On the **Select server roles** page, expand **File and Storage Services (Installed)** and expand **File and iSCSI Services (Installed)**.
5. In the **Roles** list, select **DFS Replication**. In the **Add Roles and Features Wizard** dialog box, select **Add Features**, and then click **Next**.
6. On the **Select features** page, accept the default settings, and click **Next**.
7. On the **Confirm installation selections** page, click **Install**.
8. On the **Installation progress** page, click **Close**.

Create the MDTProduction folder on MDT02

1. On MDT02, using File Explorer, create the **E:\MDTProduction** folder.
2. Share the **E:\MDTProduction** folder as **MDTProduction\$**. Use the default permissions.

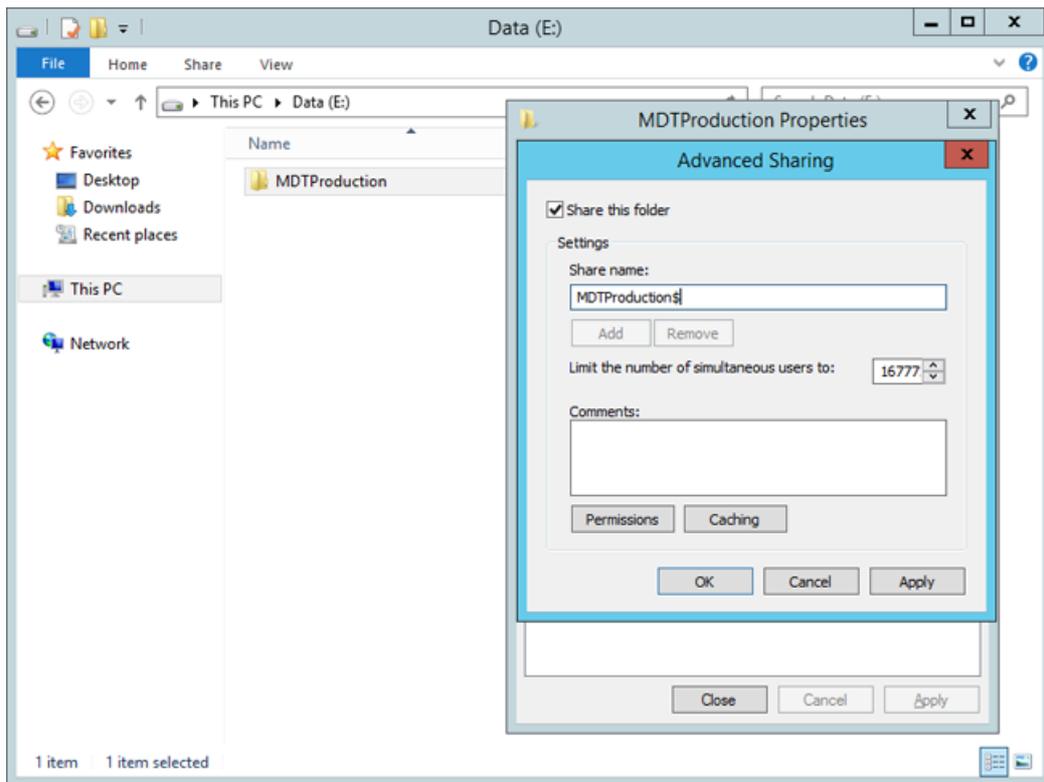


Figure 3. Sharing the **E:\MDTProduction** folder on MDT02.

Configure the deployment share

When you have multiple deployment servers sharing the same content, you need to configure the `Bootstrap.ini` file with information about which server to connect to based on where the client is located. In MDT, that can be done by using the `DefaultGateway` property.

1. On MDT01, using Notepad, navigate to the **E:\MDTProduction\Control** folder and modify the `Bootstrap.ini` file to look like this:

```
[Settings]
Priority=DefaultGateway, Default
[DefaultGateway]
192.168.1.1=NewYork
192.168.2.1=Stockholm
[NewYork]
DeployRoot=\\MDT01\MDTProduction$
[Stockholm]
DeployRoot=\\MDT02\MDTProduction$
[Default]
UserDomain=CONTOSO
UserID=MDT_BA
SkipBDDWelcome=YES
```

Note

The `DeployRoot` value needs to go into the `Bootstrap.ini` file, but you can use the same logic in the `CustomSettings.ini` file. For example, you can redirect the logs to the local deployment server (SLSHARE), or have the User State Migration Tool (USMT) migration store (UDDIR) local. To learn more about USMT, see [Refresh a Windows 7 computer with Windows 10](#) and [Replace a Windows 7 computer with a Windows 10 computer](#).

2. Save the Bootstrap.ini file.
3. Using the Deployment Workbench, right-click the **MDT Production** deployment share and select **Update Deployment Share**.

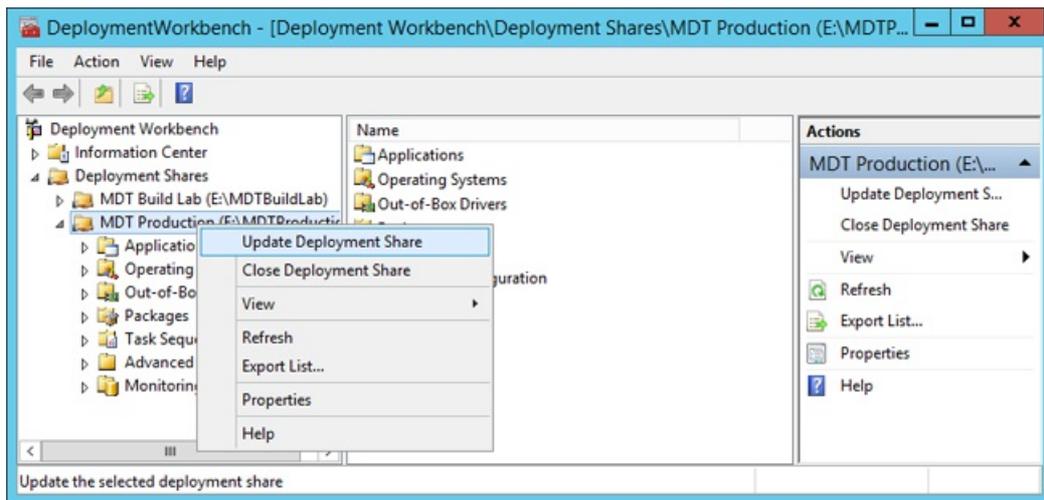


Figure 4. Updating the MDT Production deployment share.

4. Use the default settings for the Update Deployment Share Wizard.
5. After the update is complete, use the Windows Deployment Services console. In the **Boot Images** node, right-click the **MDT Production x64** boot image and select **Replace Image**.

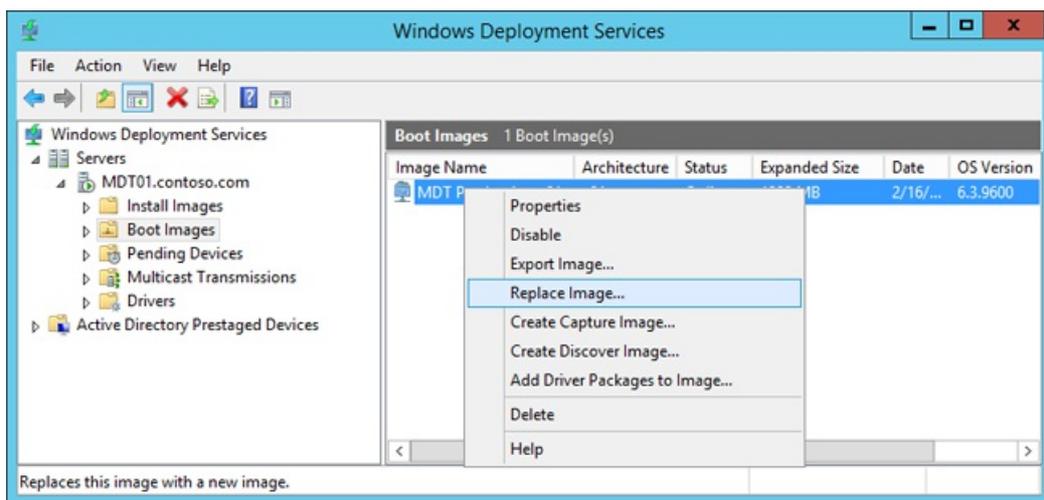


Figure 5. Replacing the updated boot image in WDS.

6. Browse and select the **E:\MDTProduction\Boot\LiteTouchPE_x64.wim** boot image, and then complete Replace Boot Image Wizard using the default settings.

Replicate the content

Once the MDT01 and MDT02 servers are prepared, you are ready to configure the actual replication.

Create the replication group

7. On MDT01, using DFS Management, right-click **Replication**, and select **New Replication Group**.
8. On the **Replication Group Type** page, select **Multipurpose replication group**, and click **Next**.
9. On the **Name and Domain** page, assign the **MDTProduction** name, and click **Next**.
10. On the **Replication Group Members** page, click **Add**, add **MDT01** and **MDT02**, and then click **Next**.

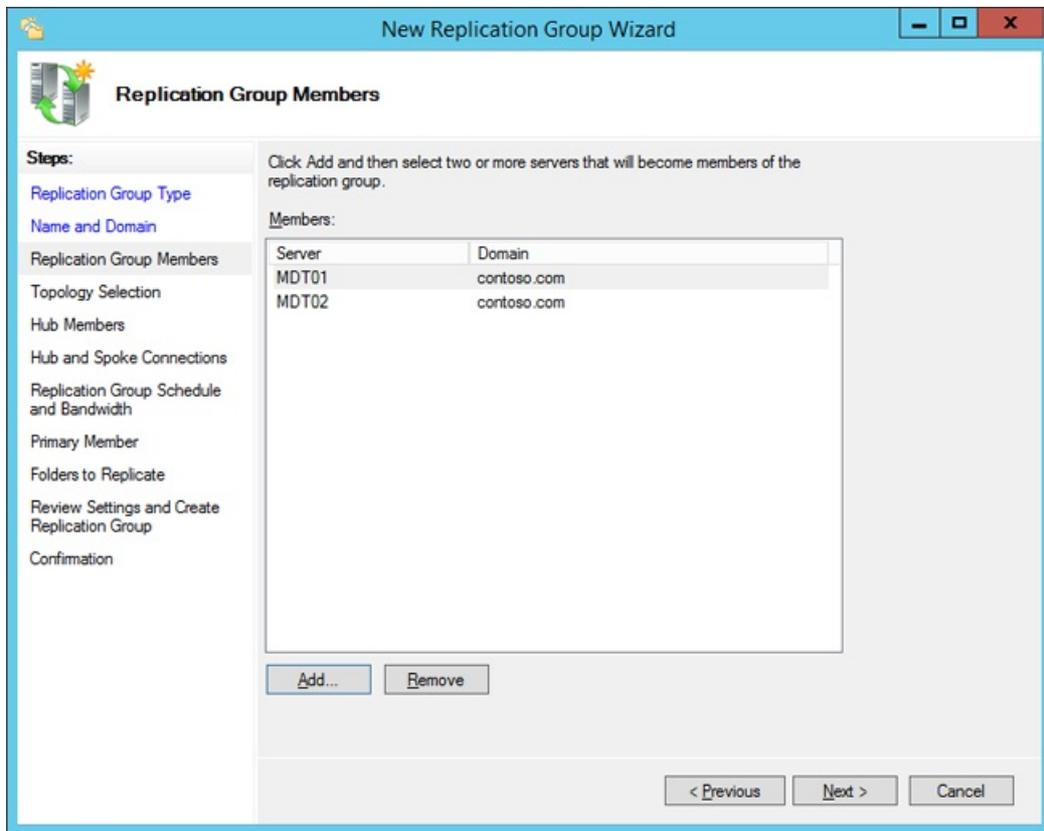


Figure 6. Adding the Replication Group Members.

11. On the **Topology Selection** page, select the **Full mesh** option and click **Next**.
12. On the **Replication Group Schedule and Bandwidth** page, accept the default settings and click **Next**.
13. On the **Primary Member** page, select **MDT01** and click **Next**.
14. On the **Folders to Replicate** page, click **Add**, type in **E:\MDTProduction** as the folder to replicate, click **OK**, and then click **Next**.
15. On the **Local Path of MDTProduction** on the **Other Members** page, select **MDT02**, and click **Edit**.
16. On the **Edit** page, select the **Enabled** option, type in **E:\MDTProduction** as the local path of folder, select the **Make the selected replicated folder on this member read-only** check box, click **OK**, and then click **Next**.

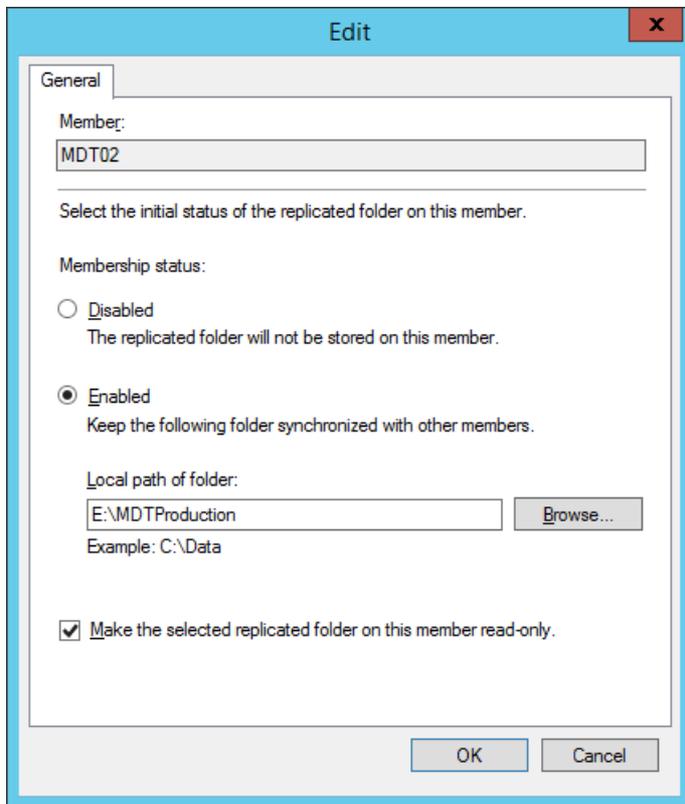


Figure 7. Configure the MDT02 member.

17. On the **Review Settings and Create Replication Group** page, click **Create**.

18. On the **Confirmation** page, click **Close**.

Configure replicated folders

19. On MDT01, using DFS Management, expand **Replication** and then select **MDTProduction**.

20. In the middle pane, right-click the **MDT01** member and select **Properties**.

21. On the **MDT01 (MDTProduction) Properties** page, configure the following and then click **OK**:

a. In the **Staging** tab, set the quota to **20480 MB**.

b. In the **Advanced** tab, set the quota to **8192 MB**. In this scenario the size of the deployment share is known, but you might need to change the values for your environment. A good rule of thumb is to get the size of the 16 largest files and make sure they fit in the staging area. Here is a Windows PowerShell example that calculates the size of the 16 largest files in the E:\MDTProduction deployment share:

```
(Get-ChildItem E:\MDTProduction -Recurse | Sort-Object Length -Descending | Select-Object -First
16 | Measure-Object -Property Length -Sum).Sum /1GB
```

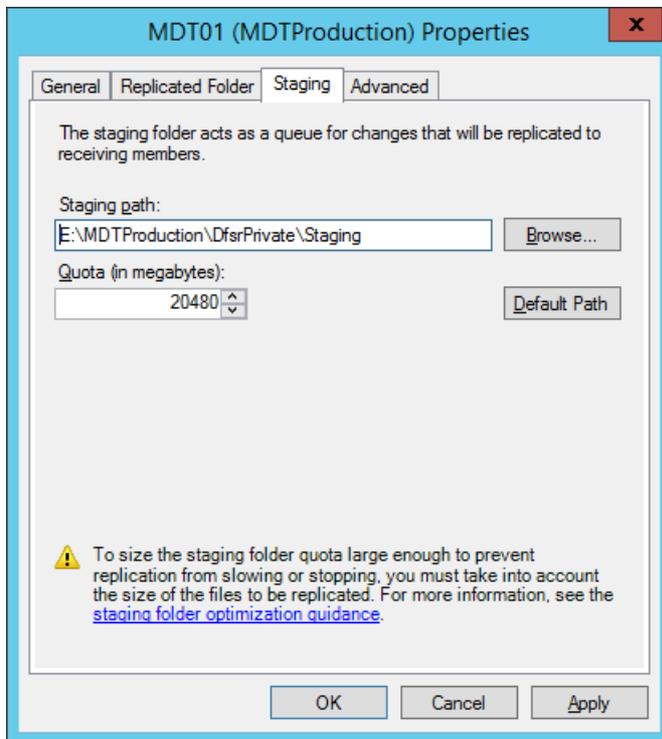


Figure 8. Configure the Staging settings.

22. In the middle pane, right-click the **MDT02** member and select **Properties**.
23. On the **MDT02 (MDTProduction) Properties** page, configure the following and then click **OK**:
 - a. In the **Staging** tab, set the quota to **20480 MB**.
 - b. In the **Advanced** tab, set the quota to **8192 MB**.

Note It will take some time for the replication configuration to be picked up by the replication members (MDT01 and MDT02). The time for the initial sync will depend on the WAN link speed between the sites. After that, delta changes are replicated quickly.

Verify replication

1. On MDT02, wait until you start to see content appear in the **E:\MDTProduction** folder.
2. Using DFS Management, expand **Replication**, right-click **MDTProduction**, and select **Create Diagnostics Report**.
3. In the Diagnostics Report Wizard, on the **Type of Diagnostics Report or Test** page, select **Health report** and click **Next**.
4. On the **Path and Name** page, accept the default settings and click **Next**.
5. On the **Members to Include** page, accept the default settings and click **Next**.
6. On the **Options** page, accept the default settings and click **Next**.
7. On the **Review Settings and Create Report** page, click **Create**.
8. Open the report in Internet Explorer, and if necessary, select the **Allow blocked content** option.

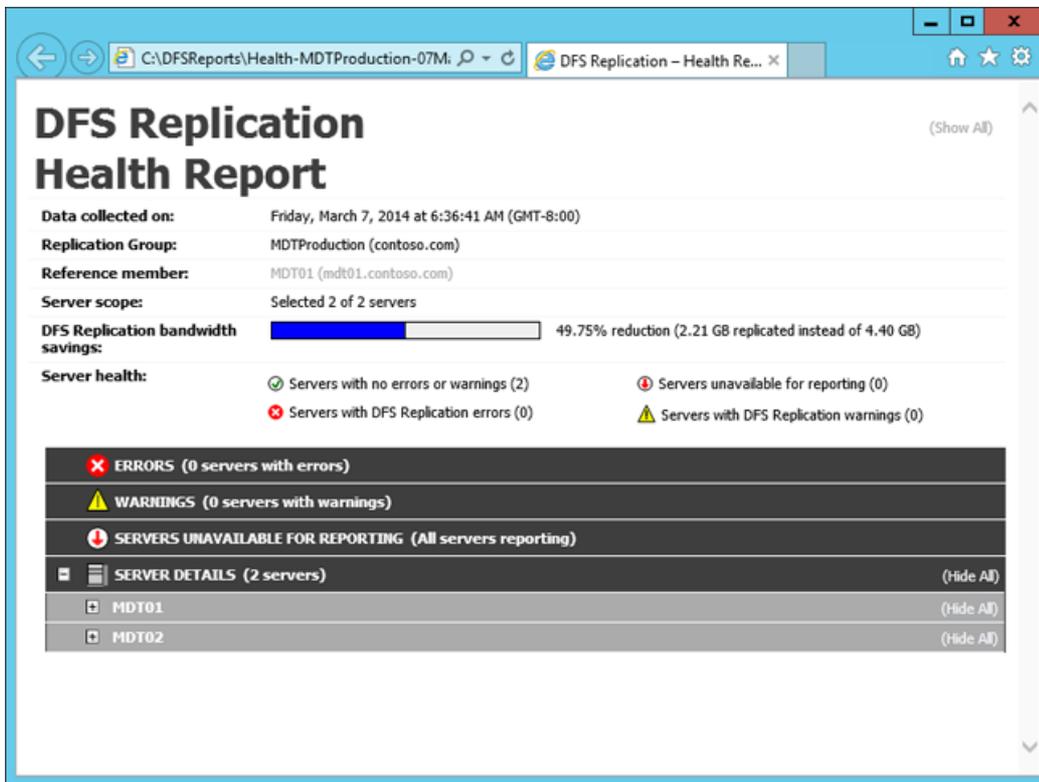


Figure 9. The DFS Replication Health Report.

Configure Windows Deployment Services (WDS) in a remote site

Like you did in the previous topic for MDT01, you need to add the MDT Production Lite Touch x64 Boot image to Windows Deployment Services on MDT02. For the following steps, we assume that WDS has already been installed on MDT02.

1. On MDT02, using the WDS console, right-click **Boot Images** and select **Add Boot Image**.
2. Browse to the E:\MDTProduction\Boot\LiteTouchPE_x64.wim file and add the image with the default settings.

Deploy the Windows 10 client to the remote site

Now you should have a solution ready for deploying the Windows 10 client to the remote site, Stockholm, connecting to the MDT Production deployment share replica on MDT02.

1. Create a virtual machine with the following settings:
 - a. Name: PC0006
 - b. Location: C:\VMs
 - c. Generation: 2
 - d. Memory: 2048 MB
 - e. Hard disk: 60 GB (dynamic disk)
2. Start the PC0006 virtual machine, and press **Enter** to start the Pre-Boot Execution Environment (PXE) boot. The machine will now load the Windows PE boot image from the WDS server.
3. After Windows Preinstallation Environment (Windows PE) has booted, complete the Windows Deployment Wizard using the following settings:
 - a. Password: P@ssw0rd
 - b. Select a task sequence to execute on this computer:
 - a. Windows 10 Enterprise x64 RTM Custom Image
 - b. Computer Name: PC0006
 - c. Applications: Select the Install - Adobe Reader XI - x86 application

4. The setup will now start and do the following:
 - a. Install the Windows 10 Enterprise operating system.
 - b. Install the added application.
 - c. Update the operating system via your local Windows Server Update Services (WSUS) server.

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Create a Windows 10 reference image](#)

[Deploy a Windows 10 image using MDT](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

[Configure MDT settings](#)

Refresh a Windows 7 computer with Windows 10

6/14/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic will show you how to use MDT Lite Touch Installation (LTI) to upgrade a Windows 7 computer to a Windows 10 computer using the computer refresh process. The refresh scenario, or computer refresh, is a reinstallation of an operating system on the same machine. You can refresh the machine to the same operating system as it is currently running, or to a later version.

For the purposes of this topic, we will use three machines: DC01, MDT01, and PC0001. DC01 is a domain controller and MDT01 is a Windows Server 2012 R2 Standard server. PC0001 is a machine with Windows 7 Service Pack 1 (SP1) that is going to be refreshed into a Windows 10 machine, with data and settings restored. MDT01 and PC0001 are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

The computer refresh process

Even though a computer will appear, to the end user, to be upgraded, a computer refresh is not, technically, an in-place upgrade. A computer refresh also involves taking care of user data and settings from the old installation and making sure to restore those at the end of the installation. For a computer refresh with MDT, you use the User State Migration Tool (USMT), which is part of the Windows Assessment and Deployment Kit (ADK) for Windows 10, to migrate user data and settings. To complete a computer refresh you will:

1. Back up data and settings locally, in a backup folder.
2. Wipe the partition, except for the backup folder.
3. Apply the new operating system image.
4. Install other applications.
5. Restore data and settings.

During the computer refresh, USMT uses a feature called Hard-Link Migration Store. When you use this feature, the files are simply linked in the file system, which allows for fast migration, even when there is a lot of data.

NOTE

In addition to the USMT backup, you can enable an optional full Windows Imaging (WIM) backup of the machine by configuring the MDT rules. If you do this, a .wim file is created in addition to the USMT backup. The .wim file will contain the entire volume from the computer, and helpdesk personnel can extract content from it if needed. Please note that this is a data WIM backup only. Using this backup to restore the entire machine is not a supported scenario.

Multi-user migration

By default, ScanState in USMT backs up all profiles on the machine, including local computer profiles. If you have

a machine that has been in your environment for a while, it likely has several domain-based profiles on it, including those of former users. You can limit which profiles are backed up by configuring command-line switches to ScanState (added as rules in MDT).

As an example, the following line configures USMT to migrate only domain user profiles and not profiles from the local SAM account database: `ScanStateArgs=/ue:* \ /ui:CONTOSO*`

NOTE

You also can combine the preceding switches with the `/uel` switch, which excludes profiles that have not been accessed within a specific number of days. For example, adding `/uel:60` will configure ScanState (or LoadState) not to include profiles that haven't been accessed for more than 60 days.

Support for additional settings

In addition to the command-line switches that control which profiles to migrate, the XML templates control exactly what data is being migrated. You can control data within and outside the user profiles

Create a custom User State Migration Tool (USMT) template

In this section, you learn to migrate additional data using a custom template. You configure the environment to use a custom USMT XML template that will:

1. Back up the **C:\Data** folder (including all files and folders).
2. Scan the local disk for PDF documents (*.pdf files) and restore them into the **C:\Data\PDF Documents** folder on the destination machine. The custom USMT template is named `MigContosoData.xml`, and you can find it in the sample files for this documentation, which include:
 - [Gather script](#)
 - [Set-OUPermissions](#) script
 - [MDT Sample Web Service](#)

Add the custom XML template

In order to use the custom `MigContosoData.xml` USMT template, you need to copy it to the MDT Production deployment share and update the `CustomSettings.ini` file. In these steps, we assume you have downloaded the `MigContosoData.xml` file.

1. Using File Explorer, copy the `MigContosoData.xml` file to the **E:\MDTProduction\Tools\x64\USMT5** folder.
2. Using Notepad, edit the `E:\MDTProduction\Control\CustomSettings.ini` file. After the `USMTMigFiles002=MigUser.xml` line add the following line:

```
USMTMigFiles003=MigContosoData.xml
```

3. Save the `CustomSettings.ini` file.

Refresh a Windows 7 SP1 client

After adding the additional USMT template and configuring the `CustomSettings.ini` file to use it, you are now ready to refresh a Windows 7 SP1 client to Windows 10. In these steps, we assume you have a Windows 7 SP1 client named `PC0001` in your environment that is ready for a refresh to Windows 10.

NOTE

MDT also supports an offline computer refresh. For more info on that scenario, see the USMTOfflineMigration property in the [MDT resource page](#).

Upgrade (refresh) a Windows 7 SP1 client

1. On PC0001, log on as **CONTOSO\Administrator**. Start the Lite Touch Deploy Wizard by executing **\\MDT01\MDTProduction\$\Scripts\Litetouch.vbs**. Complete the deployment guide using the following settings:

- Select a task sequence to execute on this computer: Windows 10 Enterprise x64 RTM
- Computer name: <default>
- Specify where to save a complete computer backup: Do not back up the existing computer

NOTE

Skip this optional full WIM backup. The USMT backup will still run.

2. Select one or more applications to install: Install - Adobe Reader XI - x86

3. The setup now starts and does the following:

- Backs up user settings and data using USMT.
- Installs the Windows 10 Enterprise x64 operating system.
- Installs the added application(s).
- Updates the operating system via your local Windows Server Update Services (WSUS) server.
- Restores user settings and data using USMT.

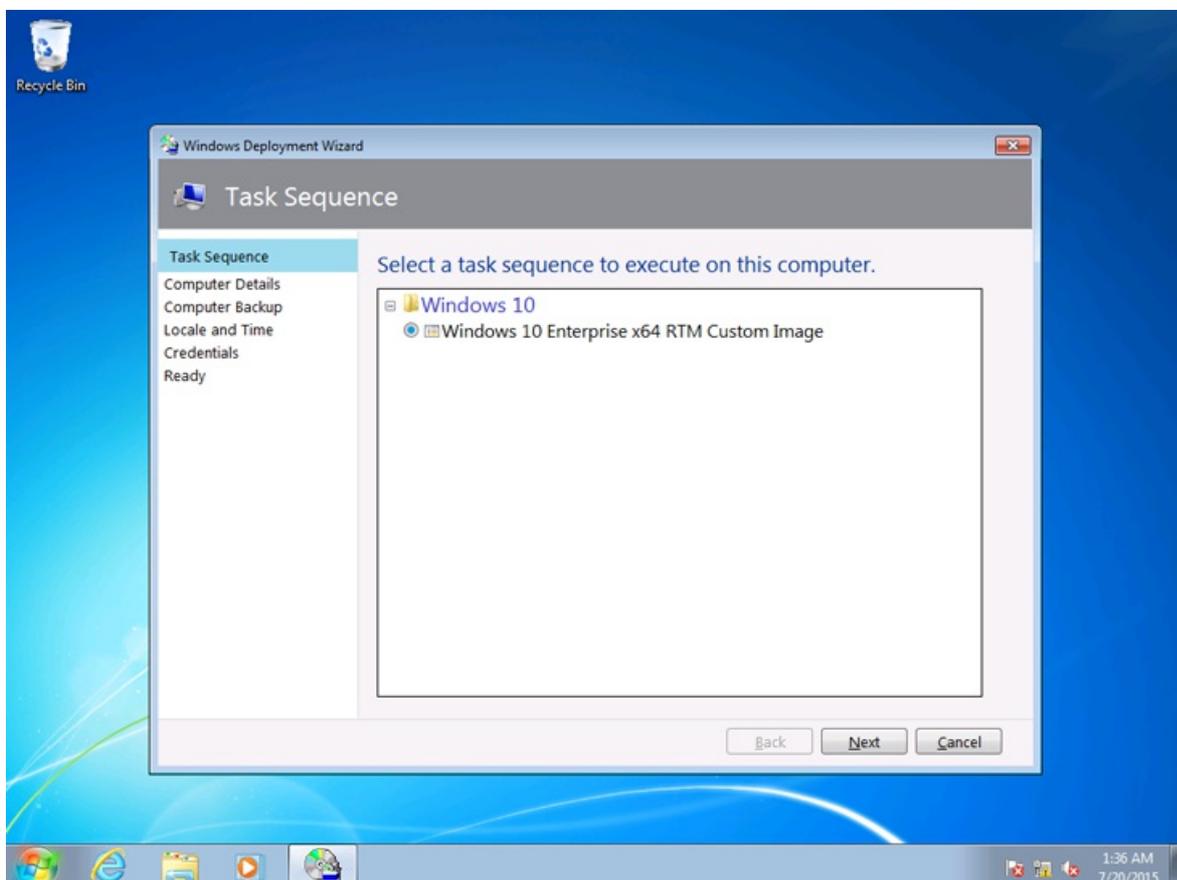


Figure 2. Starting the computer refresh from the running Windows 7 SP1 client.

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Create a Windows 10 reference image](#)

[Deploy a Windows 10 image using MDT](#)

[Build a distributed environment for Windows 10 deployment](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

[Configure MDT settings](#)

Replace a Windows 7 computer with a Windows 10 computer

6/14/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10

A computer replace scenario for Windows 10 is quite similar to a computer refresh for Windows 10; however, because you are replacing a machine, you cannot store the backup on the old computer. Instead you need to store the backup to a location where the new computer can read it. For the purposes of this topic, we will use four machines: DC01, MDT01, PC0002, and PC0007. DC01 is a domain controller and MDT01 is a Windows Server 2012 R2 standard server. PC0002 is an old machine running Windows 7 SP1. It is going to be replaced by a new Windows 10 machine, PC0007. User State Migration Tool (USMT) will be used to backup and restore data and settings. MDT01, PC0002, and PC0007 are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

Prepare for the computer replace

When preparing for the computer replace, you need to create a folder in which to store the backup, and a backup only task sequence that you run on the old computer.

Configure the rules on the Microsoft Deployment Toolkit (MDT) Production share

1. On MDT01, using the Deployment Workbench, update the MDT Production deployment share rules.
2. Change the **SkipUserData=YES** option to **NO**, and click **OK**.

Create and share the MigData folder

1. On MDT01, log on as **CONTOSO\Administrator**.
2. Create and share the **E:\MigData** folder by running the following three commands in an elevated Windows PowerShell prompt:

```
New-Item -Path E:\MigData -ItemType directory
New-SmbShare ?Name MigData$ ?Path E:\MigData
-ChangeAccess EVERYONE
icacls E:\MigData /grant '"MDT_BA":(OI)(CI)(M)'
```

Create a backup only (replace) task sequence

3. On MDT01, using the Deployment Workbench, in the MDT Production deployment share, select the **Task Sequences** node and create a new folder named **Other**.
4. Right-click the **Other** folder and select **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:

- Task sequence ID: REPLACE-001
 - Task sequence name: Backup Only Task Sequence
 - Task sequence comments: Run USMT to backup user data and settings
 - Template: Standard Client Replace Task Sequence
5. In the **Other** folder, double-click **Backup Only Task Sequence**, and then in the **Task Sequence** tab, review the sequence. Notice that it only contains a subset of the normal client task sequence actions.

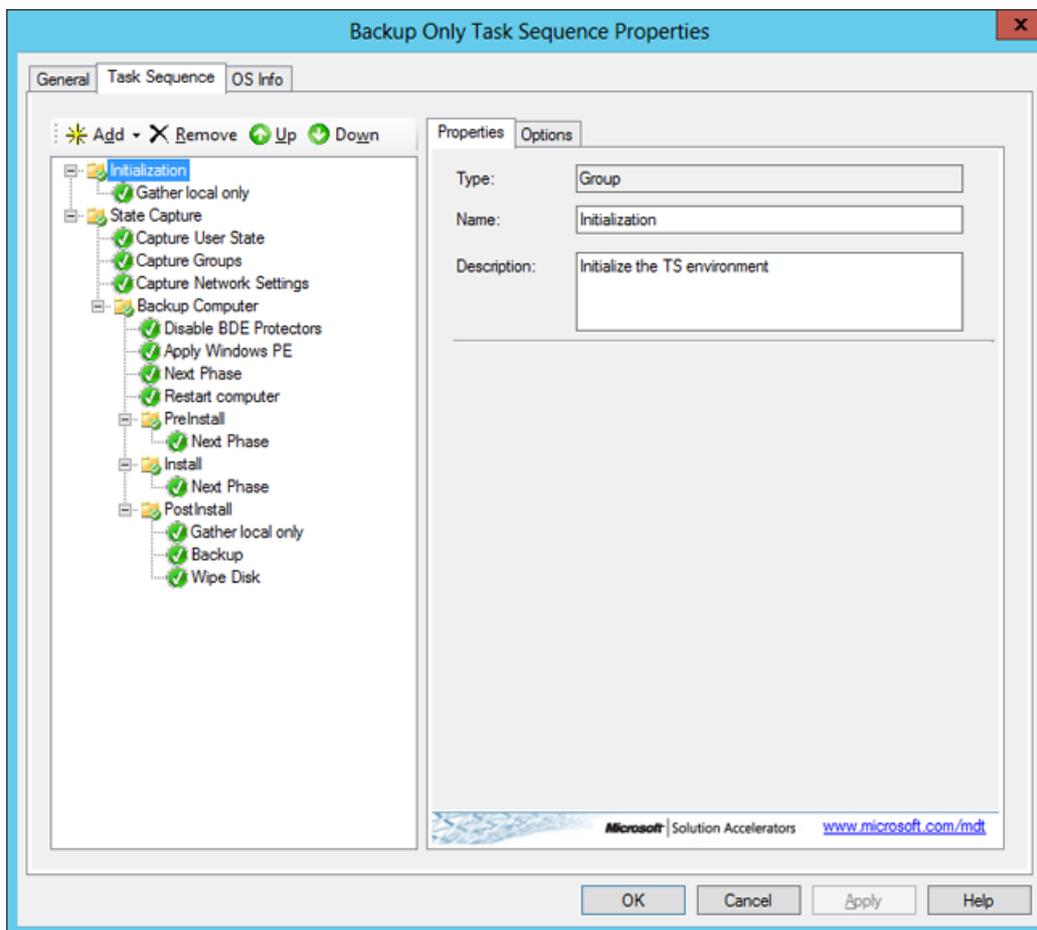


Figure 2. The Backup Only Task Sequence action list.

Perform the computer replace

During a computer replace, these are the high-level steps that occur:

1. On the computer you are replacing, a special replace task sequence runs the USMT backup and, if you configured it, runs the optional full Window Imaging (WIM) backup.
2. On the new machine, you perform a standard bare-metal deployment. At the end of the bare-metal deployment, the USMT backup from the old computer is restored.

Execute the replace task sequence

1. On PC0002, log on as **CONTOSO\Administrator**.
2. Verify that you have write access to the **\\MDT01\MigData\$** share.
3. Execute **\\MDT01\MDTProduction\$\Scripts\LiteTouch.vbs**.
4. Complete the Windows Deployment Wizard using the following settings:
 - a. Select a task sequence to execute on this computer: Backup Only Task Sequence
 - Specify where to save your data and settings: Specify a location

- Location: \\MDT01\MigData\$\PC0002

NOTE

If you are replacing the computer at a remote site you should create the MigData folder on MDT02 and use that share instead.

b. Specify where to save a complete computer backup: Do not back up the existing computer

c. Password: P@ssw0rd

The task sequence will now run USMT (Scanstate.exe) to capture user data and settings of the machine.

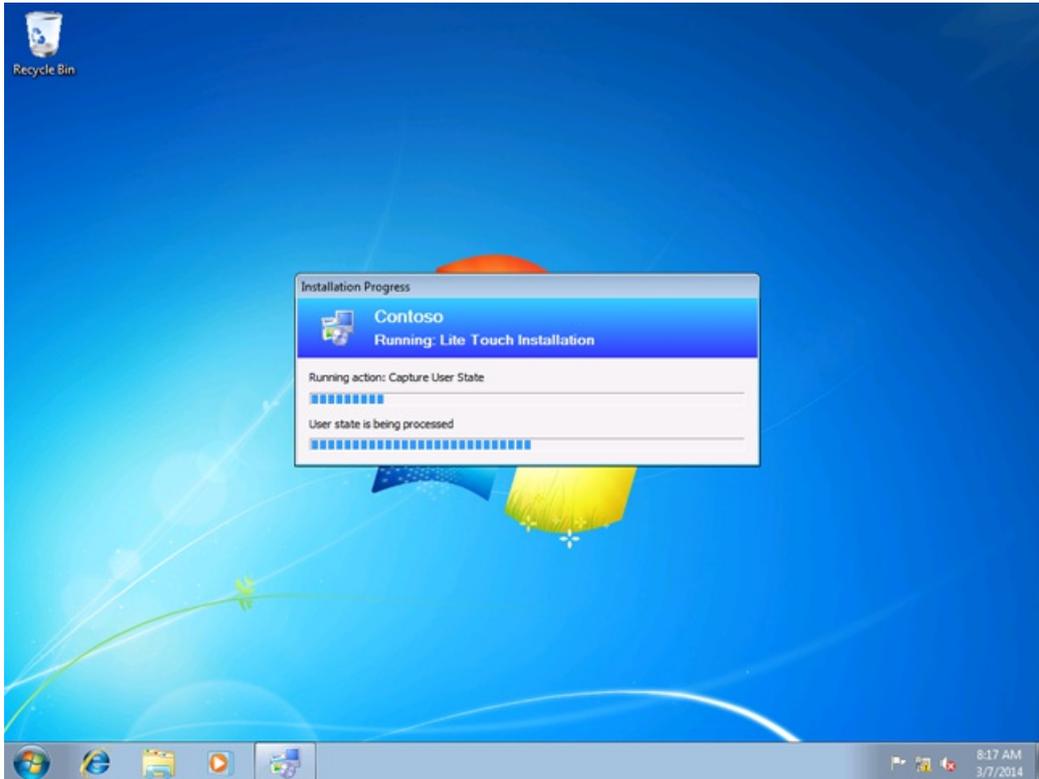


Figure 3. The new task sequence running the Capture User State action on PC0002.

5. On MDT01, verify that you have an USMT.MIG compressed backup file in the **E:\MigData\PC0002\USMT** folder.

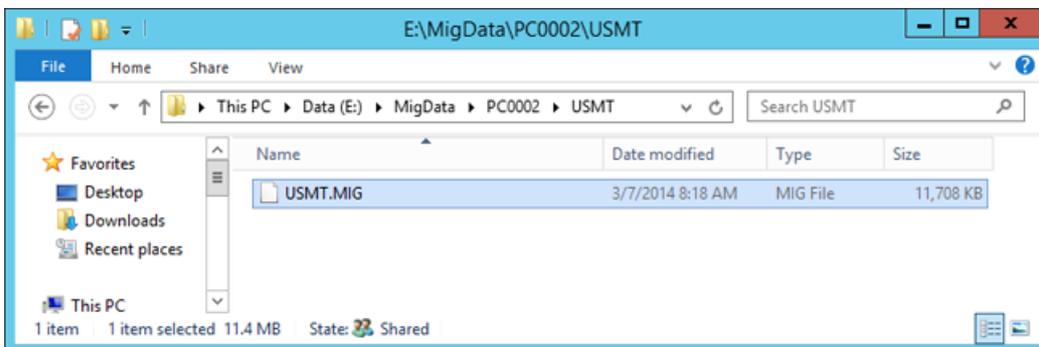


Figure 4. The USMT backup of PC0002.

Deploy the PC0007 virtual machine

1. Create a virtual machine with the following settings:
 - Name: PC0007
 - Location: C:\VMs

- Generation: 2
 - Memory: 2048 MB
 - Hard disk: 60 GB (dynamic disk)
2. Start the PC0007 virtual machine, and press **Enter** to start the Pre-Boot Execution Environment (PXE) boot. The machine will now load the Windows PE boot image from the WDS server.



```
WDS Boot Manager version 0800
Client IP: 192.168.1.100
Server IP: 192.168.1.210
Server Name: MDT01.contoso.com

Press ENTER for network boot service.
```

Figure 5. The initial PXE boot process of PC0005.

3. After Windows Preinstallation Environment (Windows PE) has booted, complete the Windows Deployment Wizard using the following settings:
 - Password: P@ssw0rd
 - Select a task sequence to execute on this computer:
 - Windows 10 Enterprise x64 RTM Custom Image
 - Computer Name: PC0007
 - Applications: Select the Install - Adobe Reader XI - x86 application.
4. The setup now starts and does the following:
 - Installs the Windows 10 Enterprise operating system.
 - Installs the added application.
 - Updates the operating system via your local Windows Server Update Services (WSUS) server.
 - Restores the USMT backup from PC0002.

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Create a Windows 10 reference image](#)

[Deploy a Windows 10 image using MDT](#)

[Build a distributed environment for Windows 10 deployment](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Configure MDT settings](#)

Perform an in-place upgrade to Windows 10 with MDT

6/14/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10

The simplest path to upgrade PCs that are currently running Windows 7, Windows 8, or Windows 8.1 to Windows 10 is through an in-place upgrade. You can use a Microsoft Deployment Toolkit (MDT) 2013 Update 2 task sequence to completely automate the process.

Proof-of-concept environment

For the purposes of this topic, we will use four machines: DC01, MDT01, and PC0001. DC01 is a domain controller and MDT01 is a Windows Server 2012 R2 standard machine, fully patched with the latest security updates, and configured as a member server in the fictional contoso.com domain. PC0001 is a machine with Windows 7 SP1, targeted for the Windows 10 upgrade. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

Set up the upgrade task sequence

MDT adds support for Windows 10 deployment, including a new in-place upgrade task sequence template that makes the process really simple.

Create the MDT production deployment share

The steps to create the deployment share for production are the same as when you created the deployment share to create the custom reference image:

1. On MDT01, log on as Administrator in the CONTOSO domain with a password of **P@ssw0rd**.
2. Using the Deployment Workbench, right-click **Deployment Shares** and select **New Deployment Share**.
3. On the **Path** page, in the **Deployment share path** text box, type **E:\MDTProduction**, and then click **Next**.
4. On the **Share** page, in the **Share name** text box, type **MDTProduction\$**, and then click **Next**.
5. On the **Descriptive Name** page, in the **Deployment share** description text box, type **MDT Production**, and then click **Next**.
6. On the **Options** page, accept the default settings and click **Next** twice, and then click **Finish**.
7. Using File Explorer, verify that you can access the **\\MDT01\MDTProduction\$** share.

Add Windows 10 Enterprise x64 (full source)

In these steps we assume that you have copied the content of a Windows 10 Enterprise x64 ISO to the **E:\Downloads\Windows 10 Enterprise x64** folder.

1. Using the Deployment Workbench, expand the **Deployment Shares** node, and then expand **MDT Production**.
2. Right-click the **Operating Systems** node, and create a new folder named **Windows 10**.
3. Expand the **Operating Systems** node, right-click the **Windows 10** folder, and select **Import Operating System**. Use the following settings for the Import Operating System Wizard:
 - Full set of source files
 - Source directory: E:\Downloads\Windows 10 Enterprise x64
 - Destination directory name: W10EX64RTM
4. After you add the operating system, in the **Operating Systems / Windows 10** folder, double-click the added operating system name in the **Operating System** node and change the name to the following: **Windows 10 Enterprise x64 RTM Default Image**

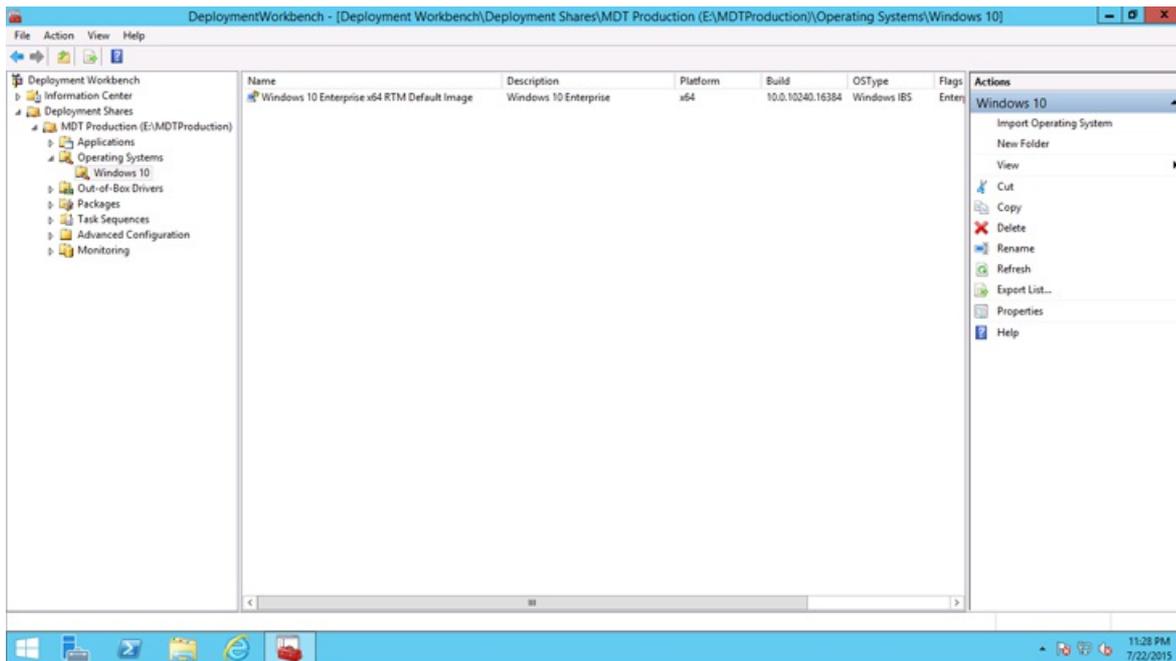


Figure 2. The imported Windows 10 operating system after you rename it.

Create a task sequence to upgrade to Windows 10 Enterprise

1. Using the Deployment Workbench, select **Task Sequences** in the **MDT Production** node, and create a folder named **Windows 10**.
2. Right-click the new **Windows 10** folder and select **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - Task sequence ID: W10-X64-UPG
 - Task sequence name: Windows 10 Enterprise x64 RTM Upgrade
 - Template: Standard Client Upgrade Task Sequence
 - Select OS: Windows 10 Enterprise x64 RTM Default Image
 - Specify Product Key: Do not specify a product key at this time
 - Full Name: Contoso
 - Organization: Contoso
 - Internet Explorer home page: about:blank
 - Admin Password: Do not specify an Administrator Password at this time

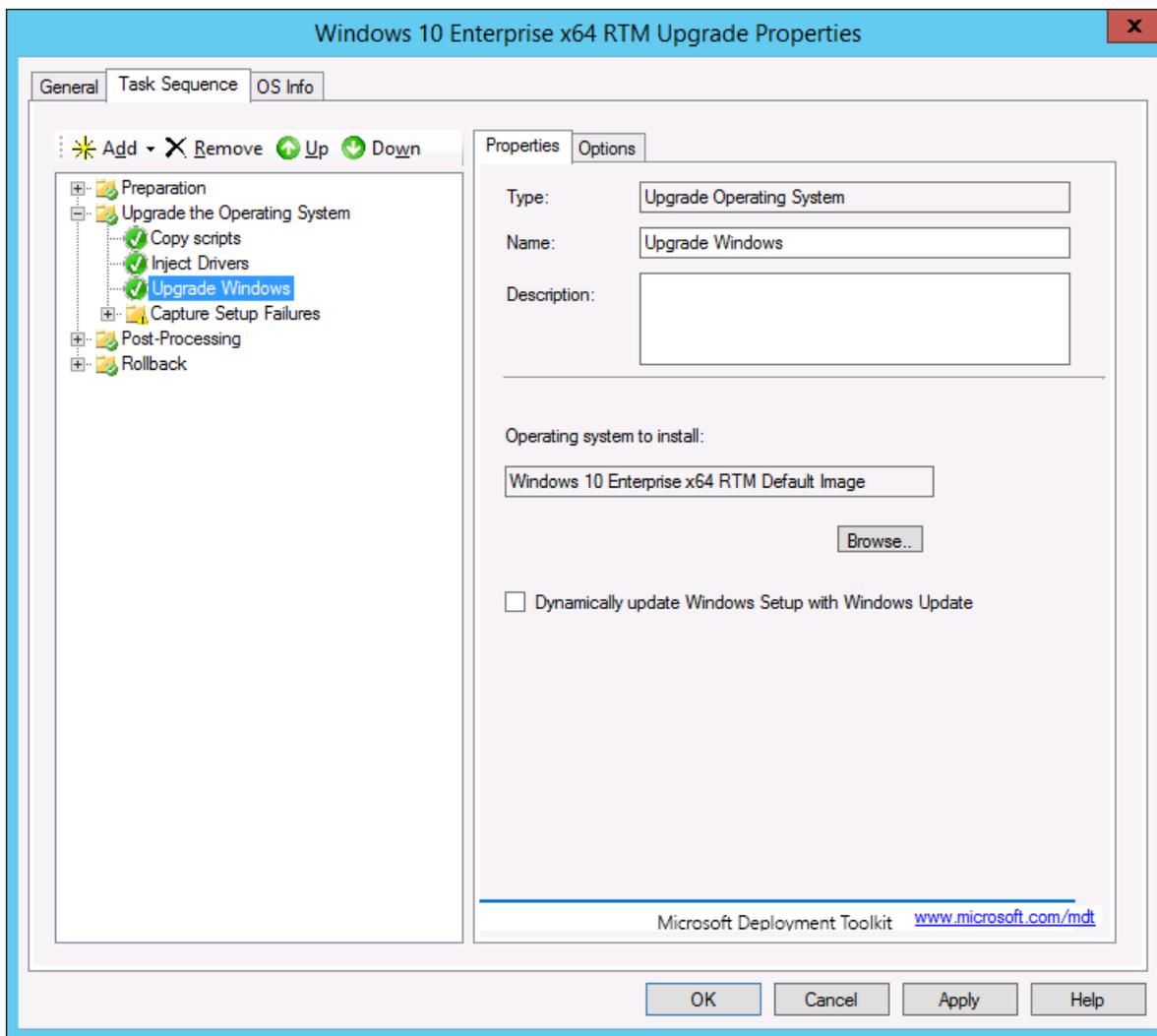


Figure 3. The task sequence to upgrade to Windows 10.

Perform the Windows 10 upgrade

To initiate the in-place upgrade, perform the following steps on PC0003 (currently running Windows 7 SP1).

1. Start the MDT deployment wizard by running the following command:
\\MDT01\MDTProduction\$\Scripts\LiteTouch.vbs
2. Select the **Windows 10 Enterprise x64 RTM Upgrade** task sequence, and then click **Next**.

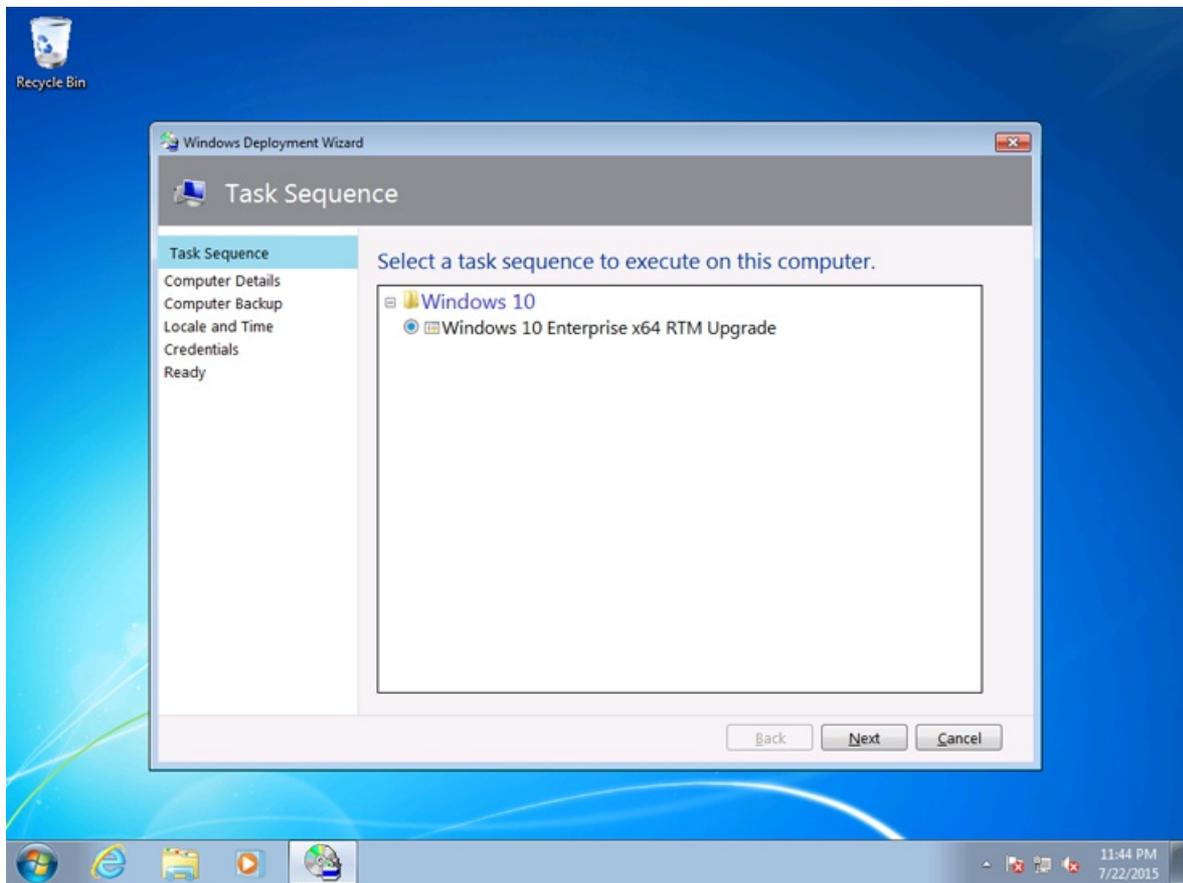


Figure 4. Upgrade task sequence.

3. On the **Credentials** tab, specify the **MDT_BA** account, **P@ssw0rd** password, and **CONTOSO** for the domain. (Some or all of these values can be specified in Bootstrap.ini so they are automatically populated.)
4. On the **Ready** tab, click **Begin** to start the task sequence. When the task sequence begins, it automatically initiates the in-place upgrade process by invoking the Windows setup program (Setup.exe) with the necessary command-line parameters to perform an automated upgrade, which preserves all data, settings, apps, and drivers.

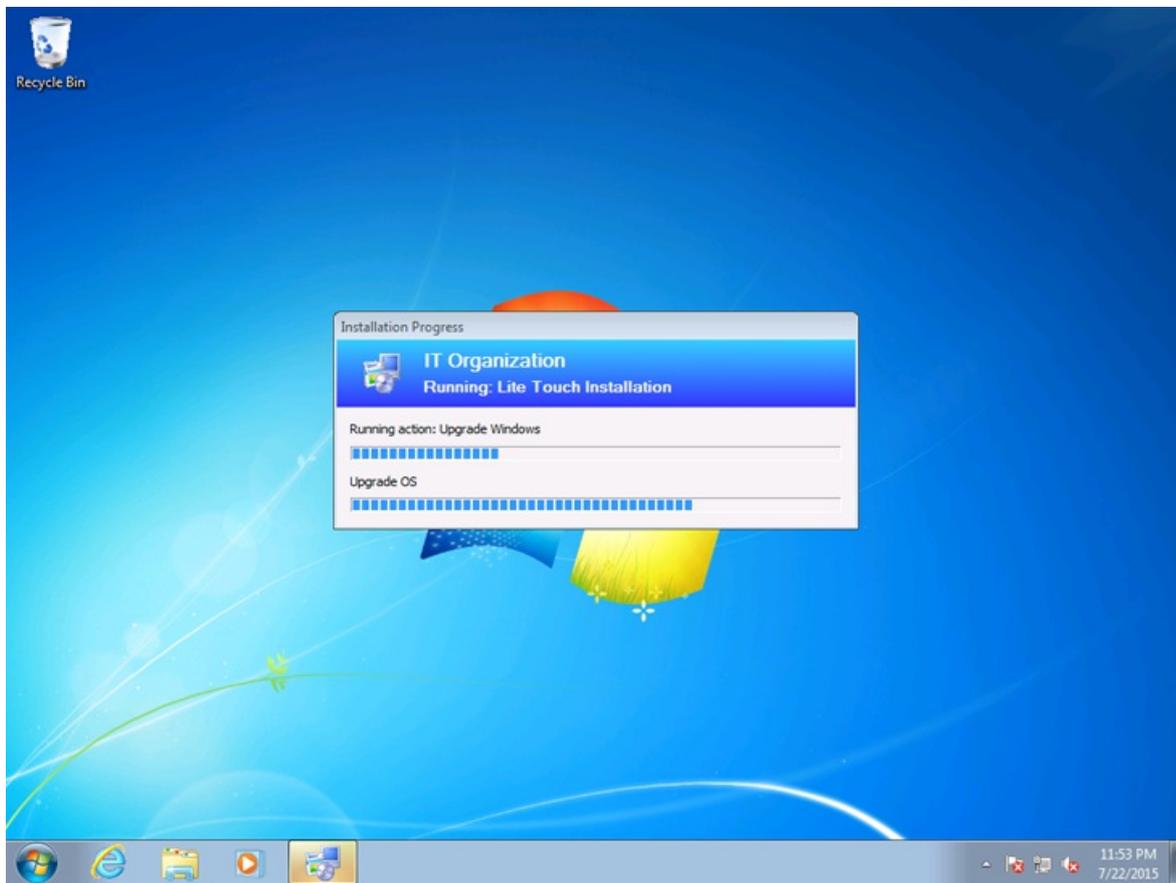


Figure 5. Upgrade from Windows 7 to Windows 10 Enterprise x64 with a task sequence.

After the task sequence completes, the computer will be fully upgraded to Windows 10.

Related topics

[Windows 10 deployment scenarios](#)

[Microsoft Deployment Toolkit downloads and resources](#)

Configure MDT settings

6/14/2019 • 2 minutes to read • [Edit Online](#)

One of the most powerful features in Microsoft Deployment Toolkit (MDT) is its extension capabilities; there is virtually no limitation to what you can do in terms of customization. In this topic, you learn about configuring customizations for your environment. For the purposes of this topic, we will use four machines: DC01, MDT01, HV01, and PC0001. DC01 is a domain controller, MDT01 is a Windows Server 2012 R2 Standard server, and PC0001 is a Windows 10 Enterprise x64 client used for the MDT simulation environment. OR01 has Microsoft System Center 2012 R2 Orchestrator installed. MDT01, OR01, and PC0001 are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

In this section

- [Set up MDT for BitLocker](#)
- [Configure MDT deployment share rules](#)
- [Configure MDT for UserExit scripts](#)
- [Simulate a Windows 10 deployment in a test environment](#)
- [Use the MDT database to stage Windows 10 deployment information](#)
- [Assign applications using roles in MDT](#)
- [Use web services in MDT](#)
- [Use Orchestrator runbooks with MDT](#)

Related topics

[Get started with the Microsoft Deployment Toolkit \(MDT\)](#)

[Create a Windows 10 reference image](#)

[Deploy a Windows 10 image using MDT](#)

[Build a distributed environment for Windows 10 deployment](#)

[Refresh a Windows 7 computer with Windows 10](#)

[Replace a Windows 7 computer with a Windows 10 computer](#)

Set up MDT for BitLocker

6/14/2019 • 6 minutes to read • [Edit Online](#)

This topic will show you how to configure your environment for BitLocker, the disk volume encryption built into Windows 10 Enterprise and Windows 10 Pro, using MDT. BitLocker in Windows 10 has two requirements in regard to an operating system deployment:

- A protector, which can either be stored in the Trusted Platform Module (TPM) chip, or stored as a password. Technically, you also can use a USB stick to store the protector, but it's not a practical approach as the USB stick can be lost or stolen. We, therefore, recommend that you instead use a TPM chip and/or a password.
- Multiple partitions on the hard drive.

To configure your environment for BitLocker, you will need to do the following:

1. Configure Active Directory for BitLocker.
2. Download the various BitLocker scripts and tools.
3. Configure the operating system deployment task sequence for BitLocker.
4. Configure the rules (CustomSettings.ini) for BitLocker.

NOTE

Even though it is not a BitLocker requirement, we recommend configuring BitLocker to store the recovery key and TPM owner information in Active Directory. For additional information about these features, see [Backing Up BitLocker and TPM Recovery Information to AD DS](#). If you have access to Microsoft BitLocker Administration and Monitoring (MBAM), which is part of Microsoft Desktop Optimization Pack (MDOP), you have additional management features for BitLocker.

For the purposes of this topic, we will use DC01, a domain controller that is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Configure Active Directory for BitLocker

To enable BitLocker to store the recovery key and TPM information in Active Directory, you need to create a Group Policy for it in Active Directory. For this section, we are running Windows Server 2012 R2, so you do not need to extend the Schema. You do, however, need to set the appropriate permissions in Active Directory.

NOTE

Depending on the Active Directory Schema version, you might need to update the Schema before you can store BitLocker information in Active Directory.

In Windows Server 2012 R2 (as well as in Windows Server 2008 R2 and Windows Server 2012), you have access to the BitLocker Drive Encryption Administration Utilities features, which will help you manage BitLocker. When you install the features, the BitLocker Active Directory Recovery Password Viewer is included, and it extends Active Directory Users and Computers with BitLocker Recovery information.

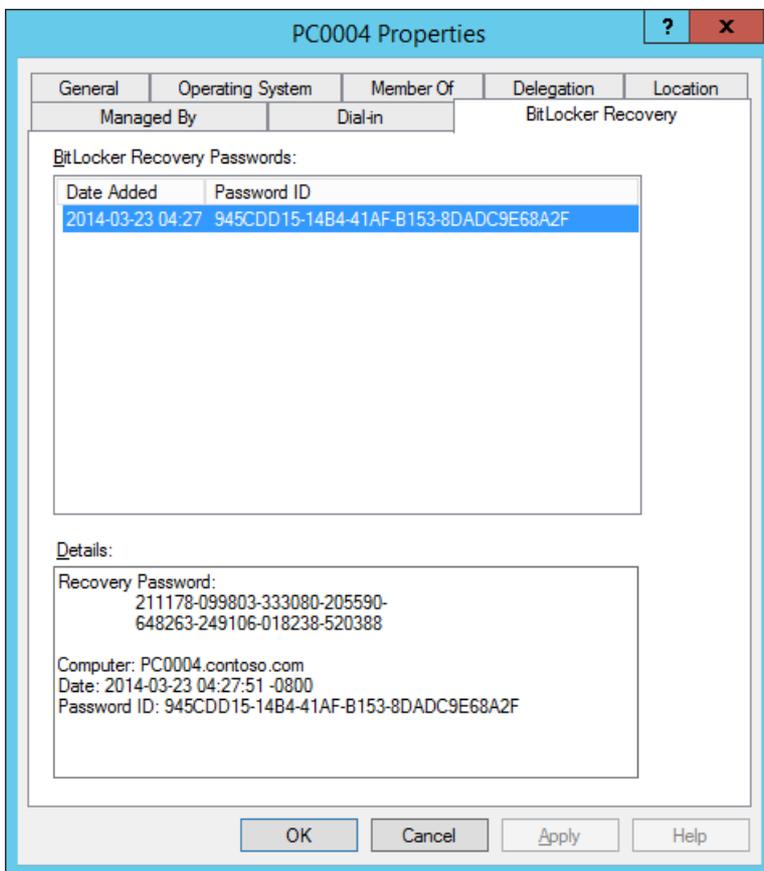


Figure 2. The BitLocker Recovery information on a computer object in the contoso.com domain.

Add the BitLocker Drive Encryption Administration Utilities

The BitLocker Drive Encryption Administration Utilities are added as features via Server Manager (or Windows PowerShell):

1. On DC01, log on as **CONTOSO\Administrator**, and, using Server Manager, click **Add roles and features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**, and click **Next**.
4. On the **Select destination server** page, select **DC01.contoso.com** and click **Next**.
5. On the **Select server roles** page, click **Next**.
6. On the **Select features** page, expand **Remote Server Administration Tools**, expand **Feature Administration Tools**, select the following features, and then click **Next**:
 - a. BitLocker Drive Encryption Administration Utilities
 - b. BitLocker Drive Encryption Tools
 - c. BitLocker Recovery Password Viewer
7. On the **Confirm installation selections** page, click **Install** and then click **Close**.

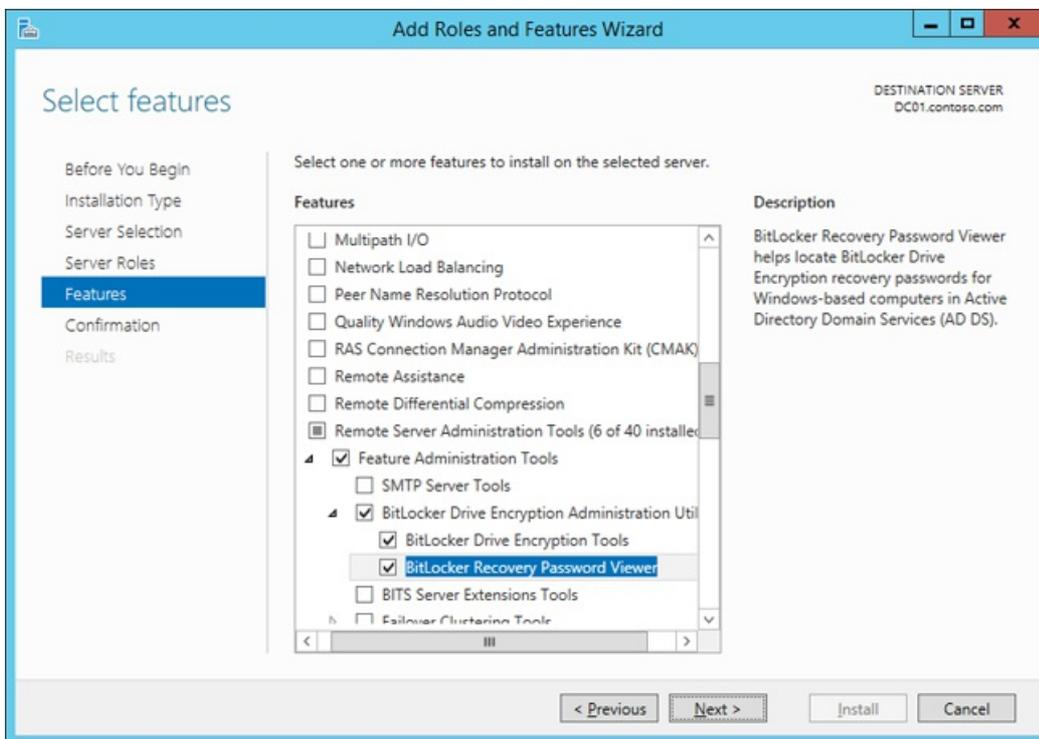


Figure 3. Selecting the BitLocker Drive Encryption Administration Utilities.

Create the BitLocker Group Policy

Following these steps, you enable the backup of BitLocker and TPM recovery information to Active Directory. You also enable the policy for the TPM validation profile.

1. On DC01, using Group Policy Management, right-click the **Contoso** organizational unit (OU), and select **Create a GPO in this domain, and Link it here**.
2. Assign the name **BitLocker Policy** to the new Group Policy.
3. Expand the **Contoso** OU, right-click the **BitLocker Policy**, and select **Edit**. Configure the following policy settings: Computer Configuration / Policies / Administrative Templates / Windows Components / BitLocker Drive Encryption / Operating System Drives
 - a. Enable the **Choose how BitLocker-protected operating system drives can be recovered** policy, and configure the following settings:
 - a. Allow data recovery agent (default)
 - b. Save BitLocker recovery information to Active Directory Domain Services (default)
 - c. Do not enable BitLocker until recovery information is stored in AD DS for operating system drives
 - b. Enable the **Configure TPM platform validation profile for BIOS-based firmware configurations** policy.
 - c. Enable the **Configure TPM platform validation profile for native UEFI firmware configurations** policy. Computer Configuration / Policies / Administrative Templates / System / Trusted Platform Module Services
 - d. Enable the **Turn on TPM backup to Active Directory Domain Services** policy.

NOTE

If you consistently get the error "Windows BitLocker Drive Encryption Information. The system boot information has changed since BitLocker was enabled. You must supply a BitLocker recovery password to start this system." after encrypting a computer with BitLocker, you might have to change the various "Configure TPM platform validation profile" Group Policies, as well. Whether or not you need to do this will depend on the hardware you are using.

Set permissions in Active Directory for BitLocker

In addition to the Group Policy created previously, you need to configure permissions in Active Directory to be able to store the TPM recovery information. In these steps, we assume you have downloaded the [Add-TPMSelfWriteACE.vbs script](#) from Microsoft to C:\Setup\Scripts on DC01.

1. On DC01, start an elevated PowerShell prompt (run as Administrator).
2. Configure the permissions by running the following command:

```
cscript C:\Setup\Scripts\Add-TPMSelfWriteACE.vbs
```

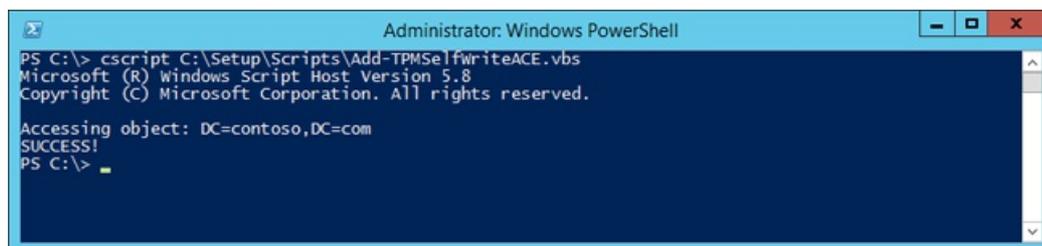


Figure 4. Running the Add-TPMSelfWriteACE.vbs script on DC01.

Add BIOS configuration tools from Dell, HP, and Lenovo

If you want to automate enabling the TPM chip as part of the deployment process, you need to download the vendor tools and add them to your task sequences, either directly or in a script wrapper.

Add tools from Dell

The Dell tools are available via the Dell Client Configuration Toolkit (CCTK). The executable file from Dell is named cctk.exe. Here is a sample command to enable TPM and set a BIOS password using the cctk.exe tool:

```
cctk.exe --tpm=on --valsetuppwd=Password1234
```

Add tools from HP

The HP tools are part of HP System Software Manager. The executable file from HP is named BiosConfigUtility.exe. This utility uses a configuration file for the BIOS settings. Here is a sample command to enable TPM and set a BIOS password using the BiosConfigUtility.exe tool:

```
BIOSConfigUtility.EXE /SetConfig:TPMEnable.REPSET /NewAdminPassword:Password1234
```

And the sample content of the TPMEnable.REPSET file:

```
English
Activate Embedded Security On Next Boot
*Enable
Embedded Security Activation Policy
*No prompts
F1 to Boot
Allow user to reject
Embedded Security Device Availability
*Available
```

Add tools from Lenovo

The Lenovo tools are a set of VBScripts available as part of the Lenovo BIOS Setup using Windows Management Instrumentation Deployment Guide. Lenovo also provides a separate download of the scripts. Here is a sample

command to enable TPM using the Lenovo tools:

```
cscript.exe SetConfig.vbs SecurityChip Active
```

Configure the Windows 10 task sequence to enable BitLocker

When configuring a task sequence to run any BitLocker tool, either directly or using a custom script, it is helpful if you also add some logic to detect whether the BIOS is already configured on the machine. In this task sequence, we are using a sample script (ZTICheckforTPM.wsf) from the Deployment Guys web page to check the status on the TPM chip. You can download this script from the Deployment Guys Blog post, [Check to see if the TPM is enabled](#). In the following task sequence, we have added five actions:

- **Check TPM Status.** Runs the ZTICheckforTPM.wsf script to determine if TPM is enabled. Depending on the status, the script will set the TPMEnabled and TPMActivated properties to either true or false.
- **Configure BIOS for TPM.** Runs the vendor tools (in this case, HP, Dell, and Lenovo). To ensure this action is run only when necessary, add a condition so the action is run only when the TPM chip is not already activated. Use the properties from the ZTICheckforTPM.wsf. **Note** It is common for organizations wrapping these tools in scripts to get additional logging and error handling.
- **Restart computer.** Self-explanatory, reboots the computer.
- **Check TPM Status.** Runs the ZTICheckforTPM.wsf script one more time.
- **Enable BitLocker.** Runs the built-in action to activate BitLocker.

Related topics

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows 10 deployment in a test environment](#)

[Use the MDT database to stage Windows 10 deployment information](#)

[Assign applications using roles in MDT](#)

[Use web services in MDT](#)

[Use Orchestrator runbooks with MDT](#)

Configure MDT deployment share rules

6/14/2019 • 3 minutes to read • [Edit Online](#)

In this topic, you will learn how to configure the MDT rules engine to reach out to other resources, including external scripts, databases, and web services, for additional information instead of storing settings directly in the rules engine. The rules engine in MDT is powerful: most of the settings used for operating system deployments are retrieved and assigned via the rules engine. In its simplest form, the rules engine is the CustomSettings.ini text file.

Assign settings

When using MDT, you can assign setting in three distinct ways:

- You can pre-stage the information before deployment.
- You can prompt the user or technician for information.
- You can have MDT generate the settings automatically.

In order illustrate these three options, let's look at some sample configurations.

Sample configurations

Before adding the more advanced components like scripts, databases, and web services, consider the commonly used configurations below; they demonstrate the power of the rules engine.

Set computer name by MAC Address

If you have a small test environment, or simply want to assign settings to a very limited number of machines, you can edit the rules to assign settings directly for a given MAC Address. If you have many machines, it makes sense to use the database instead.

```
[Settings]
Priority=MacAddress, Default
[Default]
OSInstall=YES
[00:15:5D:85:6B:00]
OSDComputerName=PC00075
```

In the preceding sample, you set the PC00075 computer name for a machine with a MAC Address of 00:15:5D:85:6B:00.

Set computer name by serial number

Another way to assign a computer name is to identify the machine via its serial number.

```
[Settings]
Priority=SerialNumber, Default
[Default]
OSInstall=YES
[CND0370RJ7]
OSDComputerName=PC00075
```

In this sample, you set the PC00075 computer name for a machine with a serial number of CND0370RJ7.

Generate a computer name based on a serial number

You also can configure the rules engine to use a known property, like a serial number, to generate a computer name on the fly.

```
[Settings]
Priority=Default
[Default]
OSInstall=YES
OSDComputerName=PC-%SerialNumber%
```

In this sample, you configure the rules to set the computer name to a prefix (PC-) and then the serial number. If the serial number of the machine is CND0370RJ7, the preceding configuration sets the computer name to PC-CND0370RJ7. **Note**

Be careful when using the serial number to assign computer names. A serial number can contain more than 15 characters, but the Windows setup limits a computer name to 15 characters.

Generate a limited computer name based on a serial number

To avoid assigning a computer name longer than 15 characters, you can configure the rules in more detail by adding VBScript functions, as follows:

```
[Settings]
Priority=Default
[Default]
OSInstall=YES
OSDComputerName=PC-#Left("%SerialNumber%",12)#
```

In the preceding sample, you still configure the rules to set the computer name to a prefix (PC-) followed by the serial number. However, by adding the Left VBScript function, you configure the rule to use only the first 12 serial-number characters for the name.

Add laptops to a different organizational unit (OU) in Active Directory

In the rules, you find built-in properties that use a Windows Management Instrumentation (WMI) query to determine whether the machine you are deploying is a laptop, desktop, or server. In this sample, we assume you want to add laptops to different OUs in Active Directory. Note that ByLaptopType is not a reserved word; rather, it is the name of the section to read.

```
[Settings]
Priority=ByLaptopType, Default
[Default]
MachineObjectOU=OU=Workstations,OU=Contoso,DC=contoso,DC=com
[ByLaptopType]
Subsection=Laptop-%IsLaptop%
[Laptop-True]
MachineObjectOU=OU=Laptops,OU=Contoso,DC=contoso,DC=com
```

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows 10 deployment in a test environment](#)

[Use the MDT database to stage Windows 10 deployment information](#)

[Assign applications using roles in MDT](#)

[Use web services in MDT](#)

[Use Orchestrator runbooks with MDT](#)

Configure MDT for UserExit scripts

6/14/2019 • 2 minutes to read • [Edit Online](#)

In this topic, you will learn how to configure the MDT rules engine to use a UserExit script to generate computer names based on a prefix and the computer MAC Address. MDT supports calling external VBScripts as part of the Gather process; these scripts are referred to as UserExit scripts. The script also removes the colons in the MAC Address.

Configure the rules to call a UserExit script

You can call a UserExit by referencing the script in your rules. Then you can configure a property to be set to the result of a function of the VBScript. In this example, we have a VBScript named Setname.vbs (provided in the book sample files, in the UserExit folder).

```
[Settings]
Priority=Default
[Default]
OSINSTALL=YES
UserExit=Setname.vbs
OSDComputerName=#SetName("%MACADDRESS%")#
```

The UserExit=Setname.vbs calls the script and then assigns the computer name to what the SetName function in the script returns. In this sample the %MACADDRESS% variable is passed to the script

The Setname.vbs UserExit script

The Setname.vbs script takes the MAC Address passed from the rules. The script then does some string manipulation to add a prefix (PC) and remove the semicolons from the MAC Address.

```
Function UserExit(sType, sWhen, sDetail, bSkip)
    UserExit = Success
End Function
Function SetName(sMac)
    Dim re
    Set re = new RegExp
    re.IgnoreCase = true
    re.Global = true
    re.Pattern = ":"
    SetName = "PC" & re.Replace(sMac, "")
End Function
```

The first three lines of the script make up a header that all UserExit scripts have. The interesting part is the lines between Function and End Function. Those lines add a prefix (PC), remove the colons from the MAC Address, and return the value to the rules by setting the SetName value.

Note The purpose of this sample is not to recommend that you use the MAC Address as a base for computer naming, but to show you how to take a variable from MDT, pass it to an external script, make some changes to it, and then return the new value to the deployment process.

Related topics

[Set up MDT for BitLocker](#)

Configure MDT deployment share rules

Simulate a Windows 10 deployment in a test environment

Use the MDT database to stage Windows 10 deployment information

Assign applications using roles in MDT

Use web services in MDT

Use Orchestrator runbooks with MDT

Simulate a Windows 10 deployment in a test environment

6/14/2019 • 2 minutes to read • [Edit Online](#)

This topic will walk you through the process of creating a simulated environment on which to test your Windows 10 deployment using MDT. When working with advanced settings and rules, especially those like database calls, it is most efficient to be able to test the settings without having to run through a complete deployment. Luckily, MDT enables you to perform a simulated deployment by running the Gather process by itself. The simulation works best when you are using a domain-joined machine (client or server). In the following example, you use the PC0001 Windows 10 client. For the purposes of this topic, you already will have either downloaded and installed the free Microsoft System Center 2012 R2 Configuration Manager Toolkit, or copied Configuration Manager Trace (CMTrace) if you have access to the System Center 2012 R2 Configuration Manager media. We also assume that you have downloaded the [sample Gather.ps1 script](#) from the TechNet gallery.

1. On PC0001, log on as **CONTOSO\Administrator** using the password **P@ssw0rd**.
2. Using Computer Management, add the **CONTOSO\MDT_BA** user account to the local **Administrators** group.
3. Log off, and then log on to PC0001 as **CONTOSO\MDT_BA**.
4. Using File Explorer, create a folder named **C:\MDT**.
5. Copy the downloaded Gather.ps1 script to the **C:\MDT** folder.
6. From the **\\MDT01\MDTProduction\$\Scripts** folder, copy the following files to **C:\MDT**:
 - a. ZTIDataAccess.vbs
 - b. ZTIGather.wsf
 - c. ZTIGather.xml
 - d. ZTIUtility.vbs
7. From the **\\MDT01\MDTProduction\$\Control** folder, copy the CustomSettings.ini file to **C:\MDT**.
8. In the **C:\MDT** folder, create a subfolder named **X64**.
9. From the **\\MDT01\MDTProduction\$\Tools\X64** folder, copy the Microsoft.BDD.Utility.dll file to **C:\MDT\X64**.

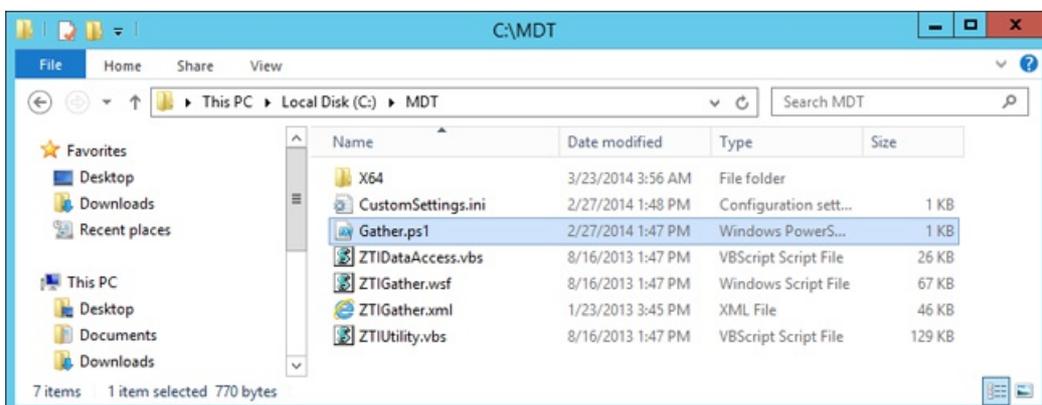


Figure 6. The C:\MDT folder with the files added for the simulation environment.

10. Using an elevated Windows PowerShell prompt (run as Administrator), run the following commands.

Press Enter after each command:

```
Set-Location C:\MDT
.\Gather.ps1
```

11. Review the ZTIGather.log in the **C:\MININT\SMSOSD\OSDLOGS** folder. **Note** Warnings or errors with regard to the Wizard.hta are expected. If the log file looks okay, you are ready to try a real deployment.

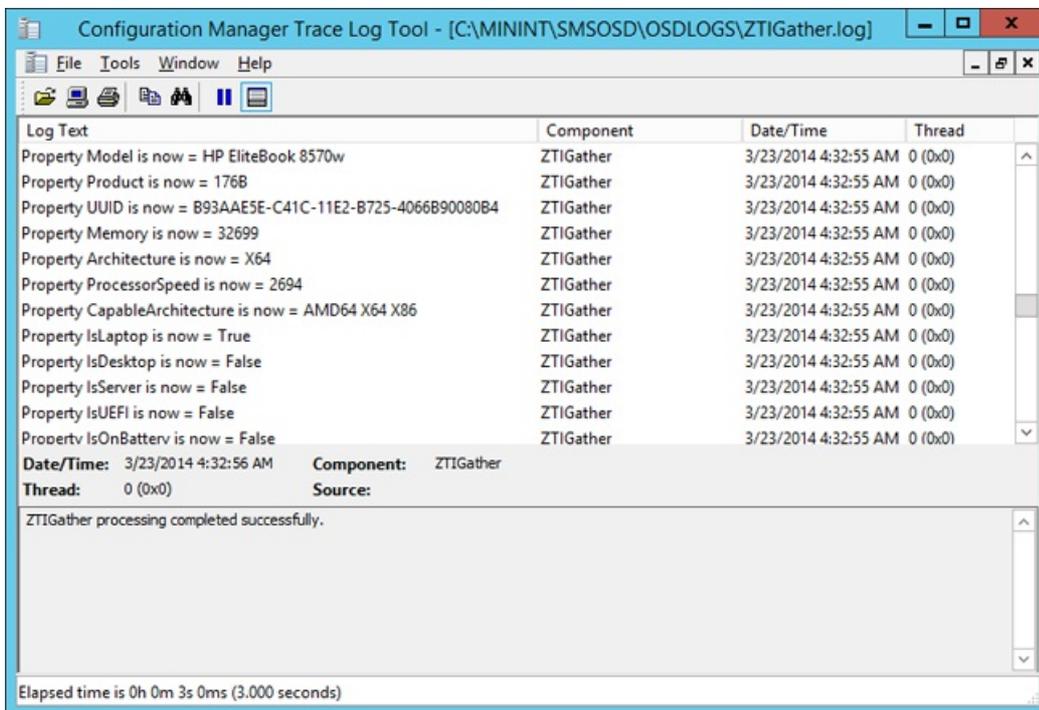


Figure 7. The ZTIGather.log file from PC0001, displaying some of its hardware capabilities.

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Use the MDT database to stage Windows 10 deployment information](#)

[Assign applications using roles in MDT](#)

[Use web services in MDT](#)

[Use Orchestrator runbooks with MDT](#)

Use the MDT database to stage Windows 10 deployment information

6/14/2019 • 3 minutes to read • [Edit Online](#)

This topic is designed to teach you how to use the MDT database to pre-stage information on your Windows 10 deployment in a Microsoft SQL Server 2012 SP1 Express database, rather than include the information in a text file (CustomSettings.ini). You can use this process, for example, to add the client machines you want to deploy, specify their computer names and IP addresses, indicate applications to be deployed, and determine many additional settings for the machines.

Database prerequisites

MDT can use either SQL Server Express or full SQL Server, but since the deployment database isn't big, even in large enterprise environments, we recommend using the free SQL Server 2012 SP1 Express database in your environment.

NOTE

Be sure to enable Named Pipes when configuring the SQL Server 2012 SP1 Express database. Although it is a legacy protocol, Named Pipes has proven to work well when connecting from Windows Preinstallation Environment (Windows PE) to the SQL Server database.

Create the deployment database

The MDT database is by default created and managed from the Deployment Workbench. In these steps, we assume you have installed SQL Server 2012 SP1 Express on MDT01.

NOTE

Since SQL Server 2012 SP1 Express runs by default on a separate instance (SQLEXPRESS), the SQL Server Browser service must be running, and the firewall configured to allow traffic to it. Port 1433 TCP and port 1434 UDP need to be opened for inbound traffic on MDT01.

1. On MDT01, using Deployment Workbench, expand the MDT Production deployment share, expand **Advanced Configuration**, right-click **Database**, and select **New Database**.
2. In the New DB Wizard, on the **SQL Server Details** page, enter the following settings and click **Next**:
 - a. SQL Server Name: MDT01
 - b. Instance: SQLEXPRESS
 - c. Port: <blank>
 - d. Network Library: Named Pipes
3. On the **Database** page, select **Create a new database**; in the **Database** field, type **MDT** and click **Next**.
4. On the **SQL Share** page, in the **SQL Share** field, type **Logs\$** and click **Next**. Click **Next** again and then click **Finish**.

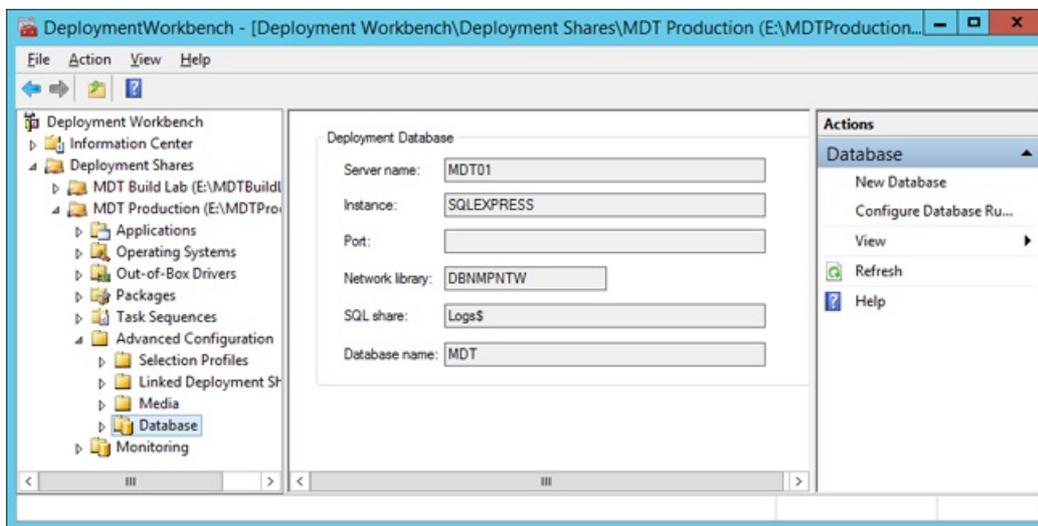


Figure 8. The MDT database added to MDT01.

Configure database permissions

After creating the database, you need to assign permissions to it. In MDT, the account you used to run the deployment is used to access the database. In this environment, the network access account is MDT_BA.

1. On MDT01, start SQL Server Management Studio.
2. In the **Connect to Server** dialog box, in the **Server name** list, select **MDT01\SQLEXPRESS** and click **Connect**.
3. In the **Object Explorer** pane, expand the top-level **Security** node, right-click **Logins**, and select **New Login**.

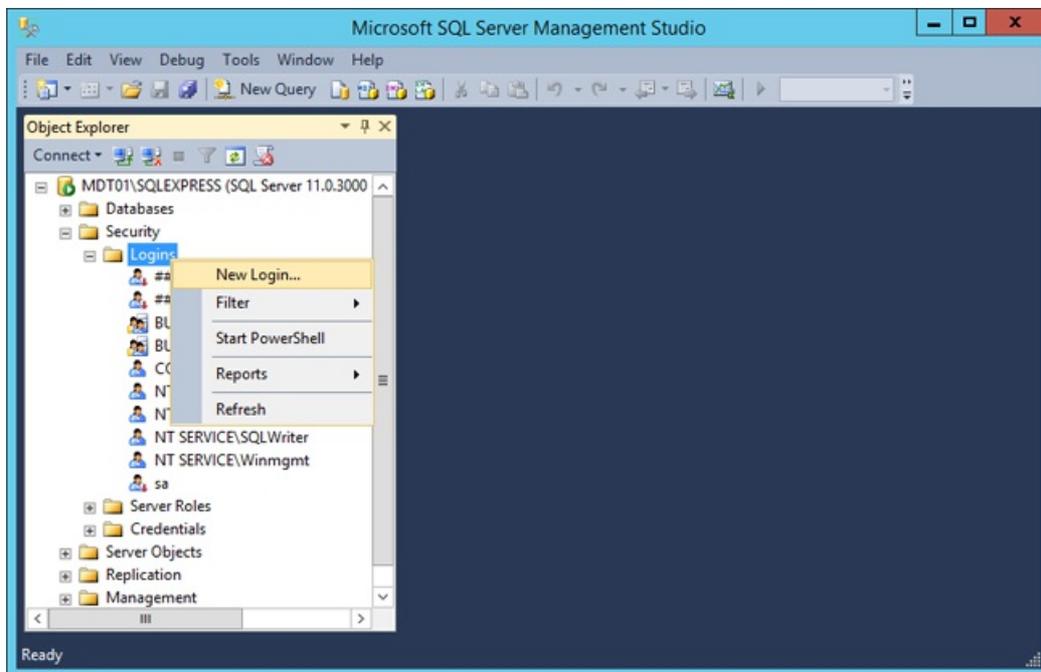


Figure 9. The top-level Security node.

4. On the **Login - New** page, next to the **Login** name field, click **Search**, and search for **CONTOSO\MDT_BA**. Then in the left pane, select **User Mapping**. Select the **MDT** database, and assign the following roles:
 - a. db_datareader
 - b. public (default)

5. Click **OK**, and close SQL Server Management Studio.

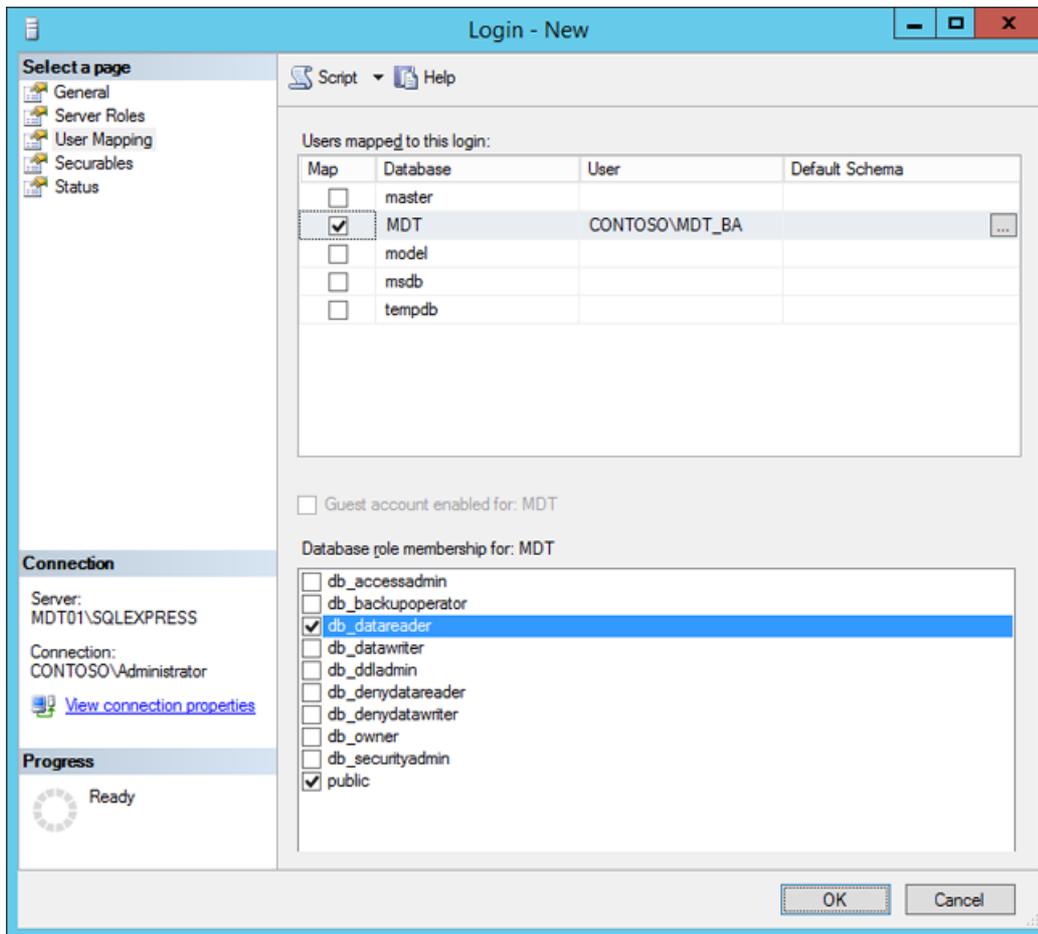


Figure 10. Creating the login and settings permissions to the MDT database.

Create an entry in the database

To start using the database, you add a computer entry and assign a description and computer name. Use the computer's MAC Address as the identifier.

1. On MDT01, using the Deployment Workbench, in the MDT Production deployment share, expand **Advanced Configuration**, and expand **Database**.
2. Right-click **Computers**, select **New**, and add a computer entry with the following settings:
 - a. Description: New York Site - PC00075
 - b. MacAddress: <PC00075 MAC Address in the 00:00:00:00:00:00 format>
 - c. Details Tab / OSDComputerName: PC00075

The screenshot shows a 'Properties' dialog box with the following fields and values:

| Field | Value |
|---------------|------------------------------------|
| ID | |
| Description | New York Site - PC00075 (optional) |
| Asset tag | |
| UUID | |
| Serial number | |
| MAC address | 00:15:5D:83:40:3A |

Figure 11. Adding the PC00075 computer to the database.

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows 10 deployment in a test environment](#)

[Assign applications using roles in MDT](#)

[Use web services in MDT](#)

[Use Orchestrator runbooks with MDT](#)

Assign applications using roles in MDT

6/14/2019 • 2 minutes to read • [Edit Online](#)

This topic will show you how to add applications to a role in the MDT database and then assign that role to a computer. For the purposes of this topic, the application we are adding is Adobe Reader XI. In addition to using computer-specific entries in the database, you can use roles in MDT to group settings together.

Create and assign a role entry in the database

1. On MDT01, using Deployment Workbench, in the MDT Production deployment share, expand **Advanced Configuration** and then expand **Database**.
2. In the **Database** node, right-click **Role**, select **New**, and create a role entry with the following settings:
 - a. Role name: Standard PC
 - b. Applications / Lite Touch Applications:
 - c. Install - Adobe Reader XI - x86

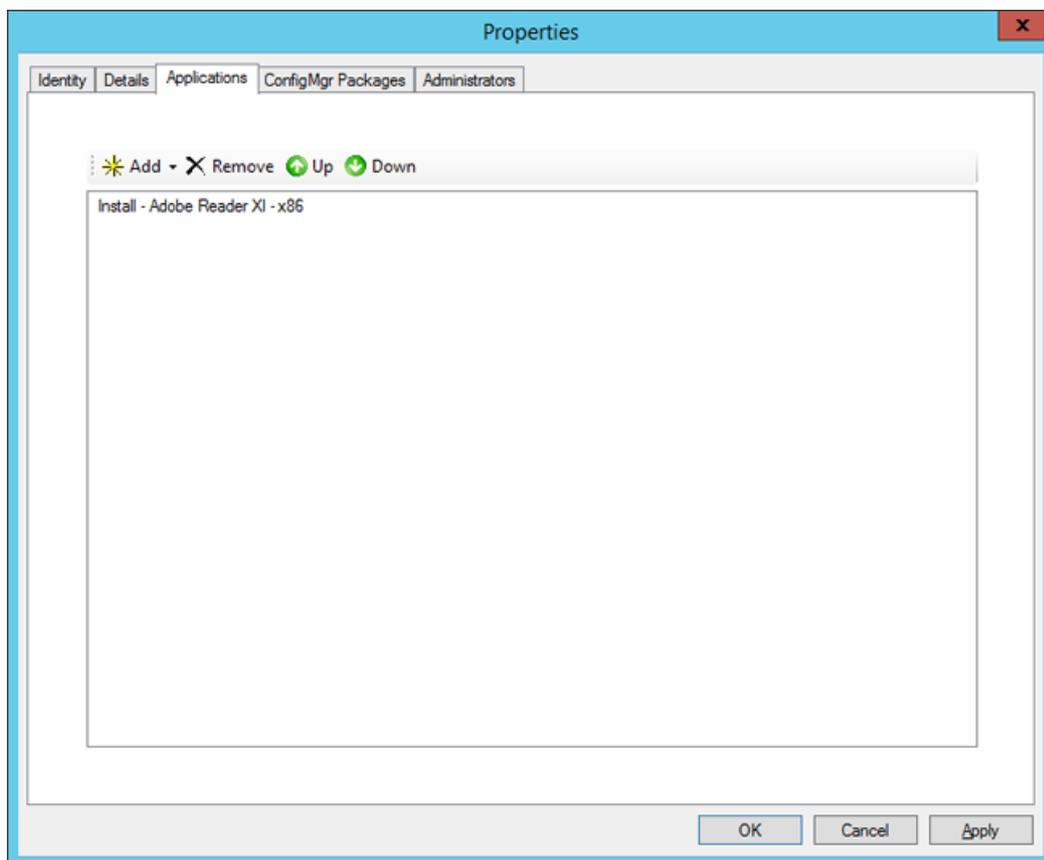


Figure 12. The Standard PC role with the application added

Associate the role with a computer in the database

After creating the role, you can associate it with one or more computer entries.

1. Using Deployment Workbench, expand **MDT Production**, expand **Advanced Configuration**, expand **Database**, and select **Computers**.
2. In the **Computers** node, double-click the **PC00075** entry, and add the following setting:
 - Roles: Standard PC

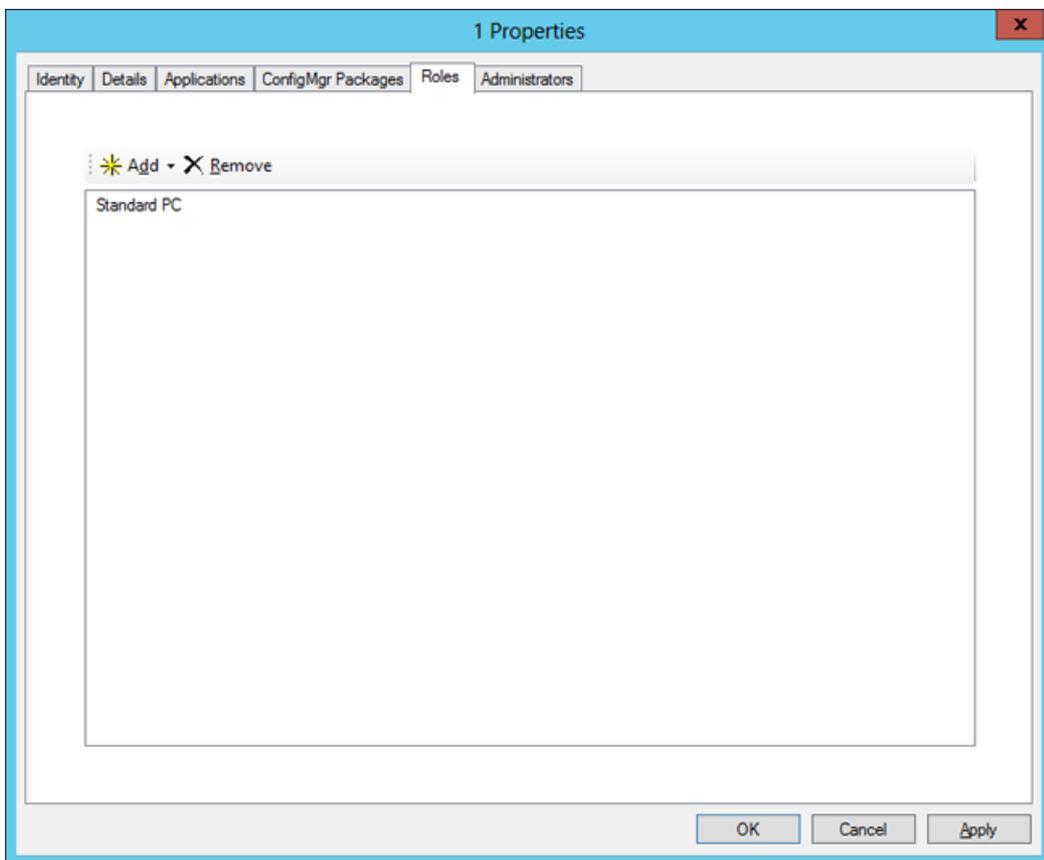


Figure 13. The Standard PC role added to PC00075 (having ID 1 in the database).

Verify database access in the MDT simulation environment

When the database is populated, you can use the MDT simulation environment to simulate a deployment. The applications are not installed, but you can see which applications would be installed if you did a full deployment of the computer.

1. On PC0001, log on as **CONTOSO\MDT_BA**.
2. Modify the C:\MDT\CustomSettings.ini file to look like the following:

```

[Settings]
Priority=CSettings, CRoles, RApplications, Default
[Default]
_SMSTSORGNAME=Contoso
OSInstall=Y
UserDataLocation=AUTO
TimeZoneName=Pacific Standard Time
AdminPassword=P@ssw0rd
JoinDomain=contoso.com
DomainAdmin=CONTOSO\MDT_JD
DomainAdminPassword=P@ssw0rd
MachineObjectOU=OU=Workstations,OU=Computers,OU=Contoso,DC=contoso,DC=com
SLShare=\\MDT01\Logs$
ScanStateArgs=/ue:* \* /ui:CONTOSO\*
USMTMigFiles001=MigApp.xml
USMTMigFiles002=MigUser.xml
HideShell=YES
ApplyGPOPack=NO
SkipAppsOnUpgrade=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerName=NO
SkipDomainMembership=YES
SkipUserData=NO
SkipLocaleSelection=YES
SkipTaskSequence=NO
SkipTimeZone=YES
SkipApplications=NO
SkipBitLocker=YES
SkipSummary=YES
SkipCapture=YES
SkipFinalSummary=NO
EventService=http://MDT01:9800
[CSettings]
SQLServer=MDT01
Instance=SQLEXPRESS
Database=MDT
Netlib=DBNMPNTW
SQLShare=Logs$
Table=ComputerSettings
Parameters=UUID, AssetTag, SerialNumber, MacAddress
ParameterCondition=OR
[CRoles]
SQLServer=MDT01
Instance=SQLEXPRESS
Database=MDT
Netlib=DBNMPNTW
SQLShare=Logs$
Table=ComputerRoles
Parameters=UUID, AssetTag, SerialNumber, MacAddress
ParameterCondition=OR
[RApplications]
SQLServer=MDT01
Instance=SQLEXPRESS
Database=MDT
Netlib=DBNMPNTW
SQLShare=Logs$
Table=RoleApplications
Parameters=Role
Order=Sequence

```

- Using an elevated Windows PowerShell prompt (run as Administrator), run the following commands. Press **Enter** after each command:

```
Set-Location C:\MDT
.\Gather.ps1
```

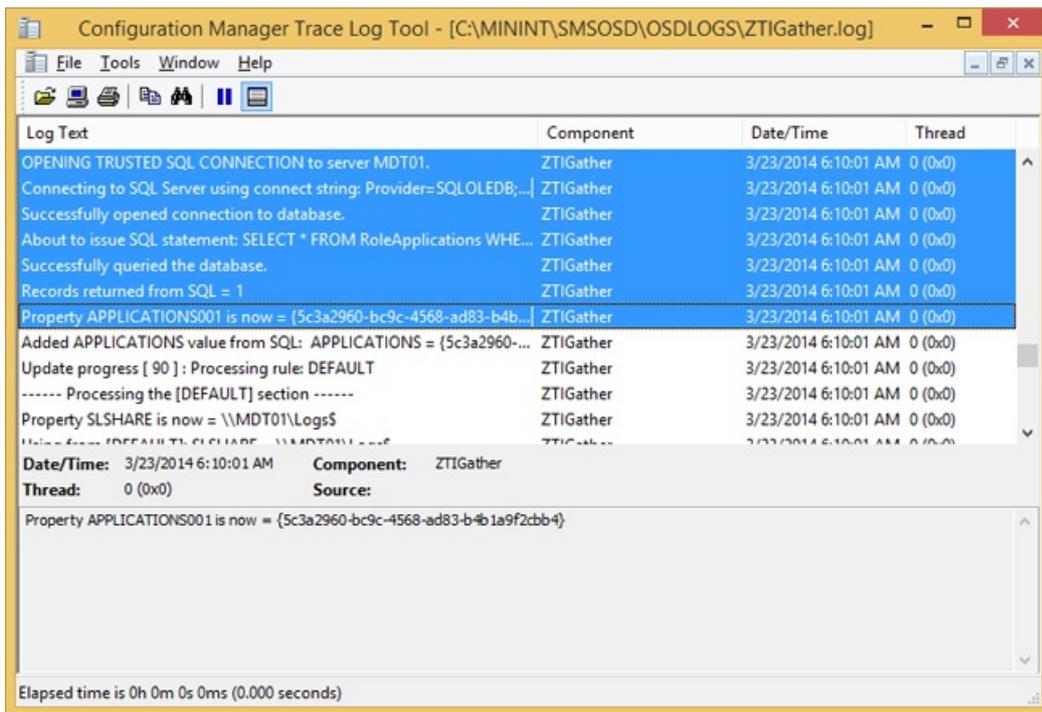


Figure 14. ZTIGather.log displaying the application GUID belonging to the Adobe Reader XI application that would have been installed if you deployed this machine.

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows 10 deployment in a test environment](#)

[Use the MDT database to stage Windows 10 deployment information](#)

[Use web services in MDT](#)

[Use Orchestrator runbooks with MDT](#)

Use web services in MDT

6/14/2019 • 3 minutes to read • [Edit Online](#)

In this topic, you will learn how to create a simple web service that generates computer names and then configure MDT to use that service during your Windows 10 deployment. Web services provide a powerful way to assign settings during a deployment. Simply put, web services are web applications that run code on the server side, and MDT has built-in functions to call these web services. Using a web service in MDT is straightforward, but it does require that you have enabled the Web Server (IIS) role on the server. Developing web services involves a little bit of coding, but for most web services used with MDT, you can use the free Microsoft Visual Studio Express 2013 for Web.

Create a sample web service

In these steps we assume you have installed Microsoft Visual Studio Express 2013 for Web on PC0001 (the Windows 10 client) and downloaded the [MDT Sample Web Service](#) from the Microsoft Download Center and extracted it to C:\Projects.

1. On PC0001, using Visual Studio Express 2013 for Web, open the C:\Projects\MDTSample\MDTSample.sln solution file.
2. On the ribbon bar, verify that Release is selected.
3. In the **Debug** menu, select the **Build MDTSample** action.
4. On MDT01, create a folder structure for **E:\MDTSample\bin**.
5. From PC0001, copy the C:\Projects\MDTSample\obj\Release\MDTSample.dll file to the **E:\MDTSample\bin** folder on MDT01.
6. From PC0001, copy the following files from C:\Projects\MDTSample file to the **E:\MDTSample** folder on MDT01:
 - a. Web.config
 - b. mdtsample.asmx

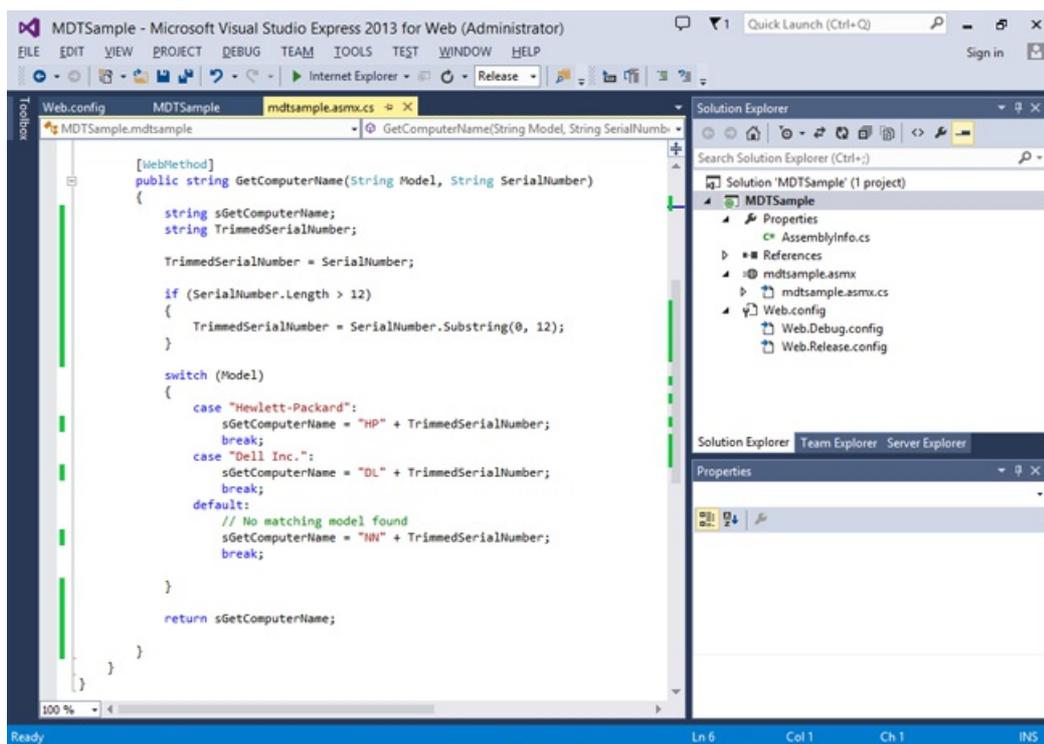


Figure 15. The sample project in Microsoft Visual Studio Express 2013 for Web.

Create an application pool for the web service

This section assumes that you have enabled the Web Server (IIS) role on MDT01.

1. On MDT01, using Server Manager, install the **IIS Management Console** role (available under Web Server (IIS) / Management Tools).
2. Using Internet Information Services (IIS) Manager, expand the **MDT01 (CONTOSO\Administrator)** node. If prompted with the "Do you want to get started with Microsoft Web Platform?" question, select the **Do not show this message** check box and then click **No**.
3. Right-click **Application Pools**, select **Add Application Pool**, and configure the new application pool with the following settings:
 - a. Name: MDTSample
 - b. .NET Framework version: .NET Framework 4.0.30319
 - c. Manage pipeline mode: Integrated
 - d. Select the **Start application pool immediately** check box.
 - e. Click **OK**.

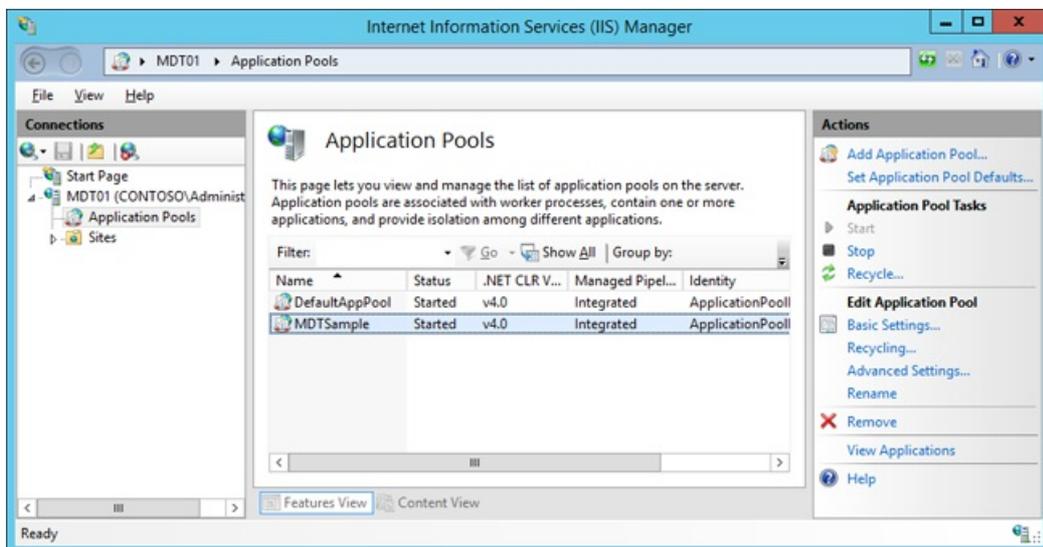


Figure 16. The new MDTSample application.

Install the web service

1. On MDT01, using Internet Information Services (IIS) Manager, expand **Sites**, right-click **Default Web Site**, and select **Add Application**. Use the following settings for the application:
 - a. Alias: MDTSample
 - b. Application pool: MDTSample
 - c. Physical Path: E:\MDTSample

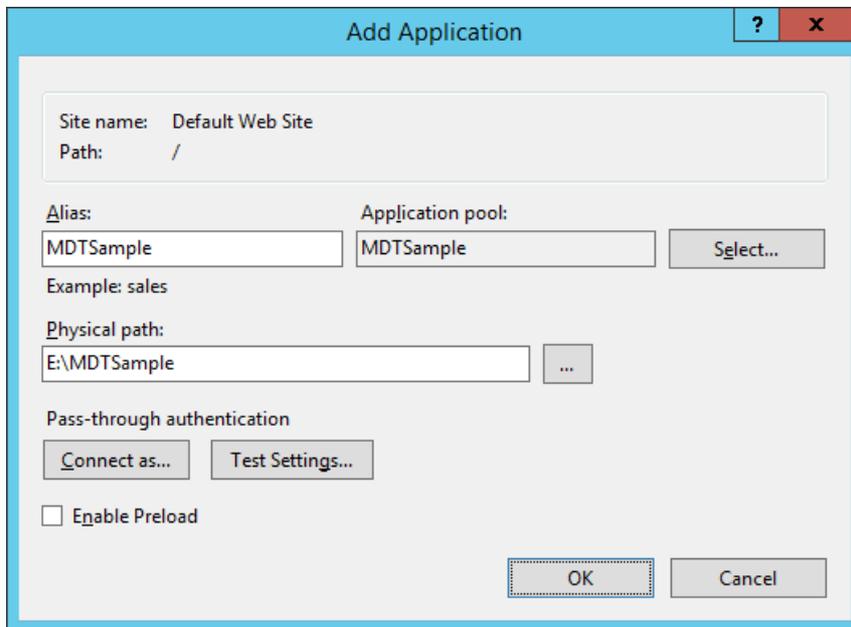


Figure 17. Adding the MDTSample web application.

2. In the **Default Web Site** node, select the MDTSample web application, and in the right pane, double-click **Authentication**. Use the following settings for the **Authentication** dialog box:
 - a. Anonymous Authentication: Enabled
 - b. ASP.NET Impersonation: Disabled



Figure 18. Configuring Authentication for the MDTSample web service.

Test the web service in Internet Explorer

1. On PC0001, using Internet Explorer, navigate to: **http://MDT01/MDTSample/mdtsample.asmx**.
2. Click the **GetComputerName** link.

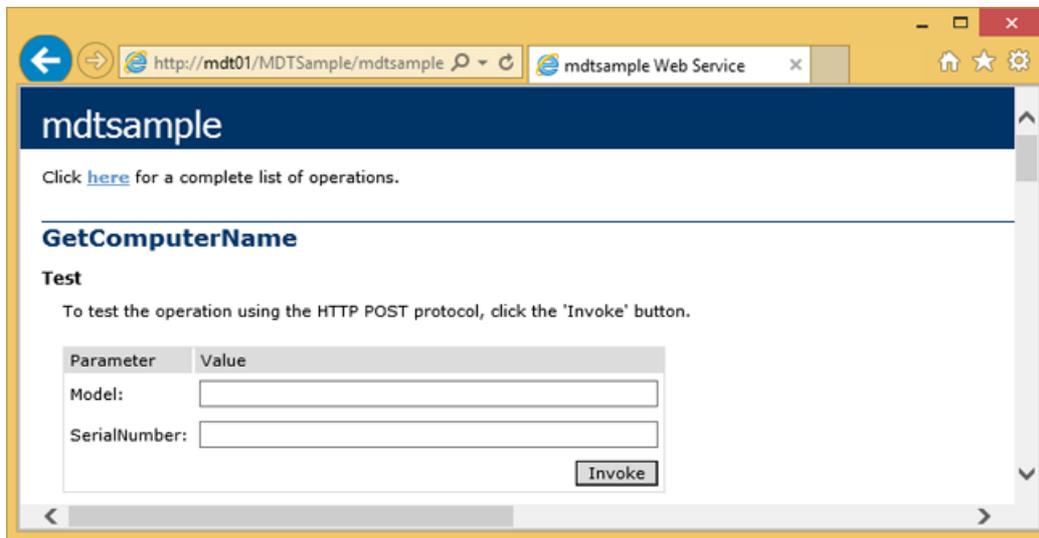


Figure 19. The MDT Sample web service.

3. On the **GetComputerName** page, type in the following settings, and click **Invoke**:
 - a. Model: Hewlett-Packard
 - b. SerialNumber: 123456789



Figure 20. The result from the MDT Sample web service.

Test the web service in the MDT simulation environment

After verifying the web service using Internet Explorer, you are ready to do the same test in the MDT simulation environment.

1. On PC0001, edit the CustomSettings.ini file in the **C:\MDT** folder to look like the following:

```
[Settings]
Priority=Default, GetComputerName
[Default]
OSInstall=YES
[GetComputerName]
WebService=http://mdt01/MDTSample/mdtsample.asmx/GetComputerName
Parameters=Model,SerialNumber
OSDComputerName=string
```

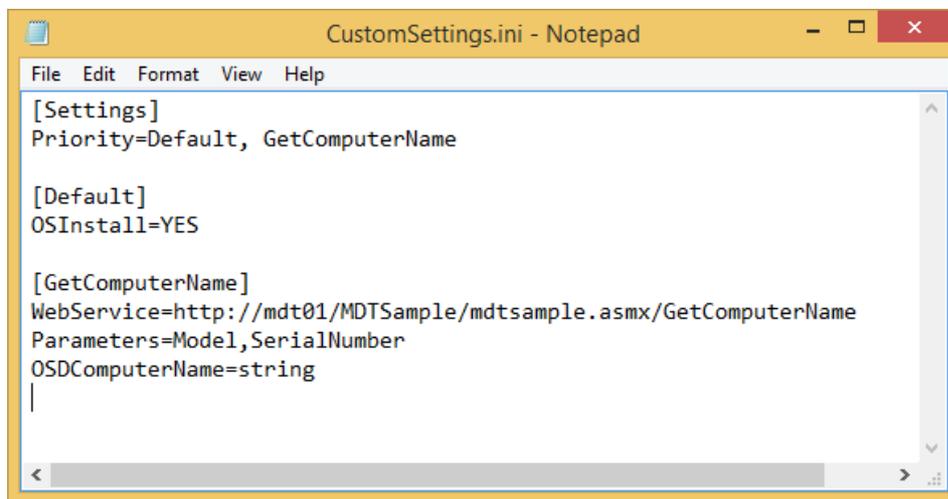


Figure 21. The updated CustomSettings.ini file.

2. Save the CustomSettings.ini file.
3. Using an elevated Windows PowerShell prompt (run as Administrator), run the following commands. Press **Enter** after each command:

```

Set-Location C:\MDT
.\Gather.ps1
  
```

4. Review the ZTIGather.log in the **C:\MININT\SMSOSD\OSDLOGS** folder.

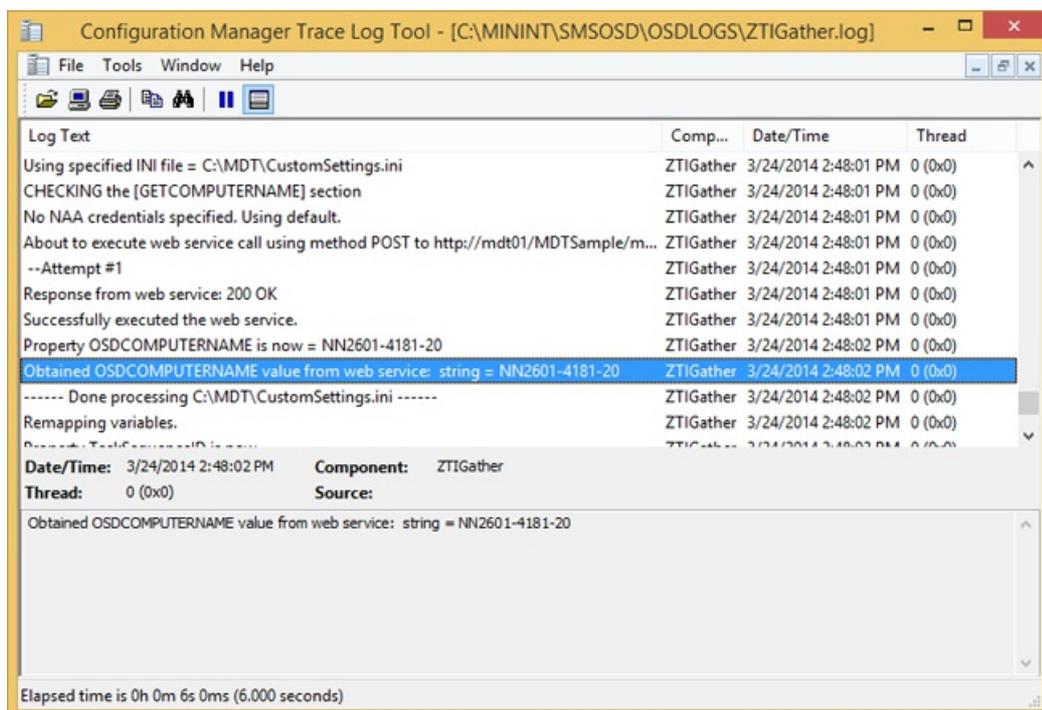


Figure 22. The OSDCOMPUTERNAME value obtained from the web service.

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows 10 deployment in a test environment](#)

Use the MDT database to stage Windows 10 deployment information

Assign applications using roles in MDT

Use Orchestrator runbooks with MDT

Use Orchestrator runbooks with MDT

6/14/2019 • 5 minutes to read • [Edit Online](#)

This topic will show you how to integrate Microsoft System Center 2012 R2 Orchestrator with MDT to replace the existing web services that are used in deployment solutions. MDT can integrate with System Center 2012 R2 Orchestrator, which is a component that ties the Microsoft System Center products together, as well as other products from both Microsoft and third-party vendors. The difference between using Orchestrator and "normal" web services, is that with Orchestrator you have a rich drag-and-drop style interface when building the solution, and little or no coding is required.

Note If you are licensed to use Orchestrator, we highly recommend that you start using it. To find out more about licensing options for System Center 2012 R2 and Orchestrator, visit the [System Center 2012 R2](#) website.

Orchestrator terminology

Before diving into the core details, here is a quick course in Orchestrator terminology:

- **Orchestrator Server.** This is a server that executes runbooks.
- **Runbooks.** A runbook is similar to a task sequence; it is a series of instructions based on conditions. Runbooks consist of workflow activities; an activity could be Copy File, Get User from Active Directory, or even Write to Database.
- **Orchestrator Designer.** This is where you build the runbooks. In brief, you do that by creating an empty runbook, dragging in the activities you need, and then connecting them in a workflow with conditions and subscriptions.
- **Subscriptions.** These are variables that come from an earlier activity in the runbook. So if you first execute an activity in which you type in a computer name, you can then subscribe to that value in the next activity. All these variables are accumulated during the execution of the runbook.
- **Orchestrator Console.** This is the Microsoft Silverlight-based web page you can use interactively to execute runbooks. The console listens to TCP port 81 by default.
- **Orchestrator web services.** These are the web services you use in the Microsoft Deployment Toolkit to execute runbooks during deployment. The web services listen to TCP port 82 by default.
- **Integration packs.** These provide additional workflow activities you can import to integrate with other products or solutions, like the rest of Active Directory, other System Center 2012 R2 products, or Microsoft Exchange Server, to name a few.

Note To find and download additional integration packs, see [Integration Packs for System Center 2012 - Orchestrator](#).

Create a sample runbook

This section assumes you have Orchestrator 2012 R2 installed on a server named OR01. In this section, you create a sample runbook, which is used to log some of the MDT deployment information into a text file on OR01.

1. On OR01, using File Explorer, create the **E:\Logfile** folder, and grant Users modify permissions (NTFS).
2. In the **E:\Logfile** folder, create the DeployLog.txt file. **Note** Make sure File Explorer is configured to show known file extensions so the file is not named DeployLog.txt.txt.

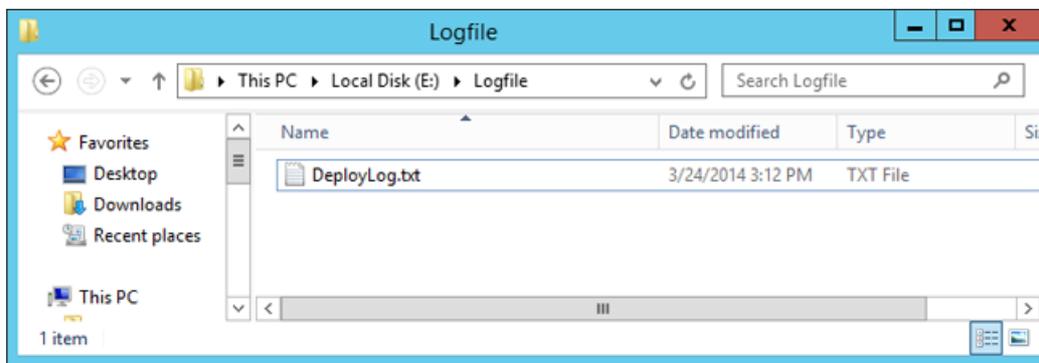


Figure 23. The DeployLog.txt file.

3. Using System Center 2012 R2 Orchestrator Runbook Designer, in the **Runbooks** node, create the **1.0 MDT** folder.

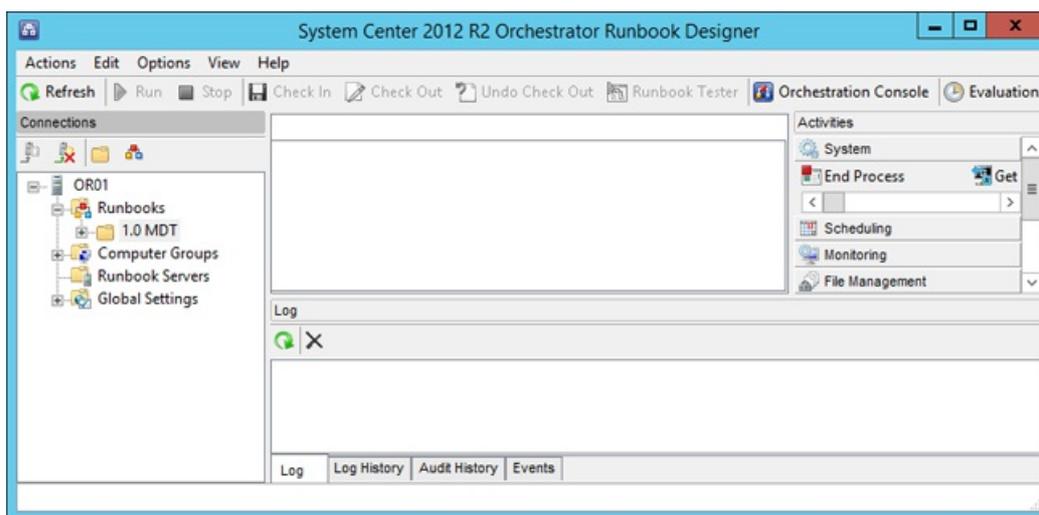


Figure 24. Folder created in the Runbooks node.

4. In the **Runbooks** node, right-click the **1.0 MDT** folder, and select **New / Runbook**.
5. On the ribbon bar, click **Check Out**.
6. Right-click the **New Runbook** label, select **Rename**, and assign the name **MDT Sample**.
7. Add (using a drag-and-drop operation) the following items from the **Activities** list to the middle pane:
 - a. Runbook Control / Initialize Data
 - b. Text File Management / Append Line
8. Connect **Initialize Data** to **Append Line**.

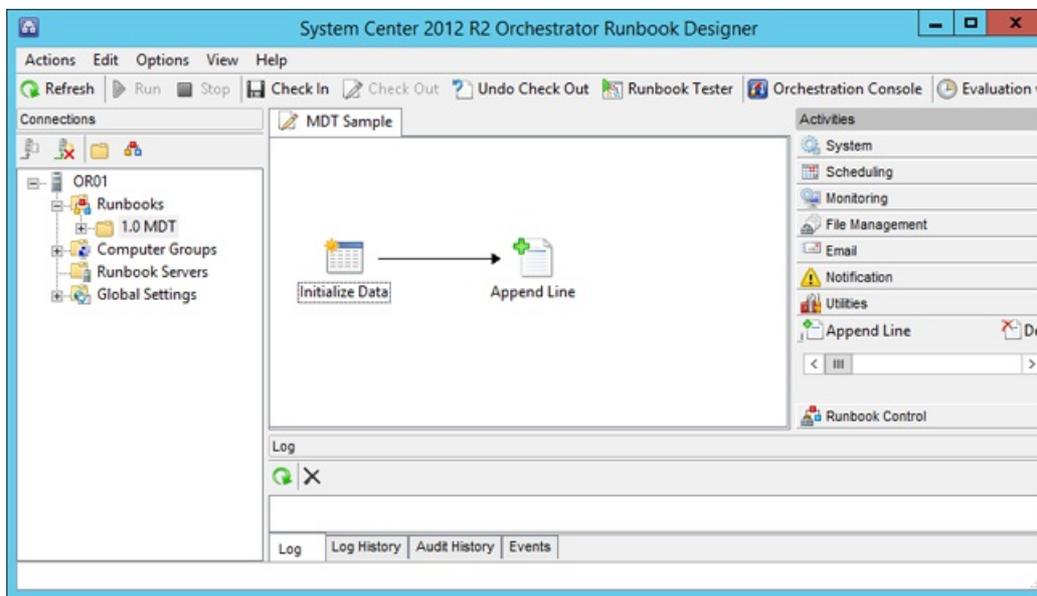


Figure 25. Activities added and connected.

9. Right-click the **Initialize Data** activity, and select **Properties**
10. On **the Initialize Data Properties** page, click **Add**, change **Parameter 1** to **OSDComputerName**, and then click **Finish**.

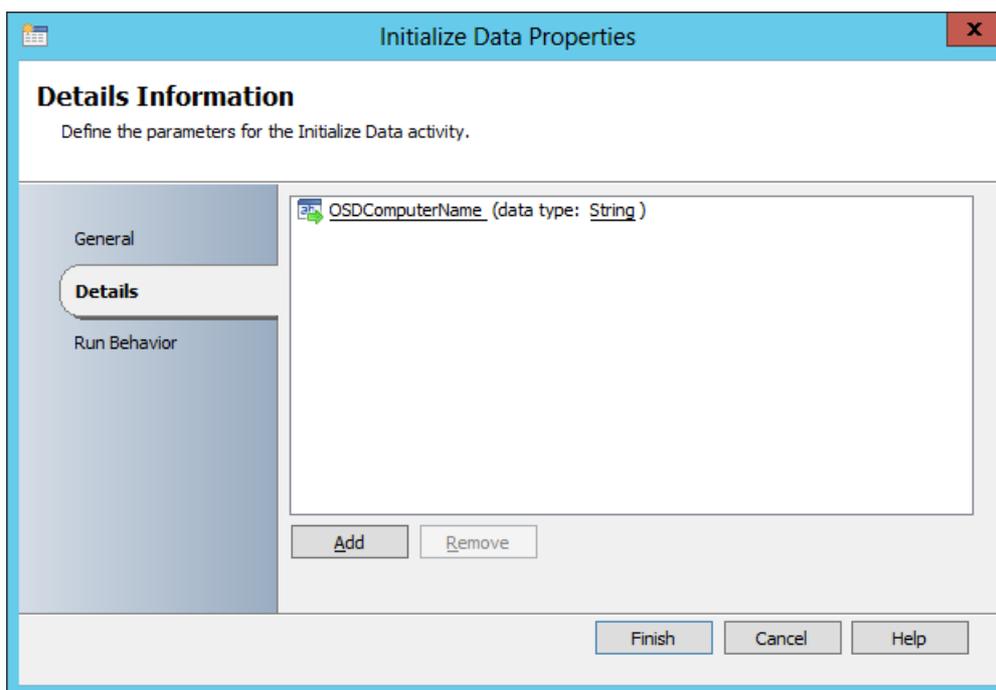


Figure 26. The Initialize Data Properties window.

11. Right-click the **Append Line** activity, and select **Properties**.
12. On the **Append Line Properties** page, in the **File** text box, type **E:\Logfile\DeployLog.txt**.
13. In the **File** encoding drop-down list, select **ASCII**.
14. In the **Append** area, right-click inside the **Text** text box and select **Expand**.

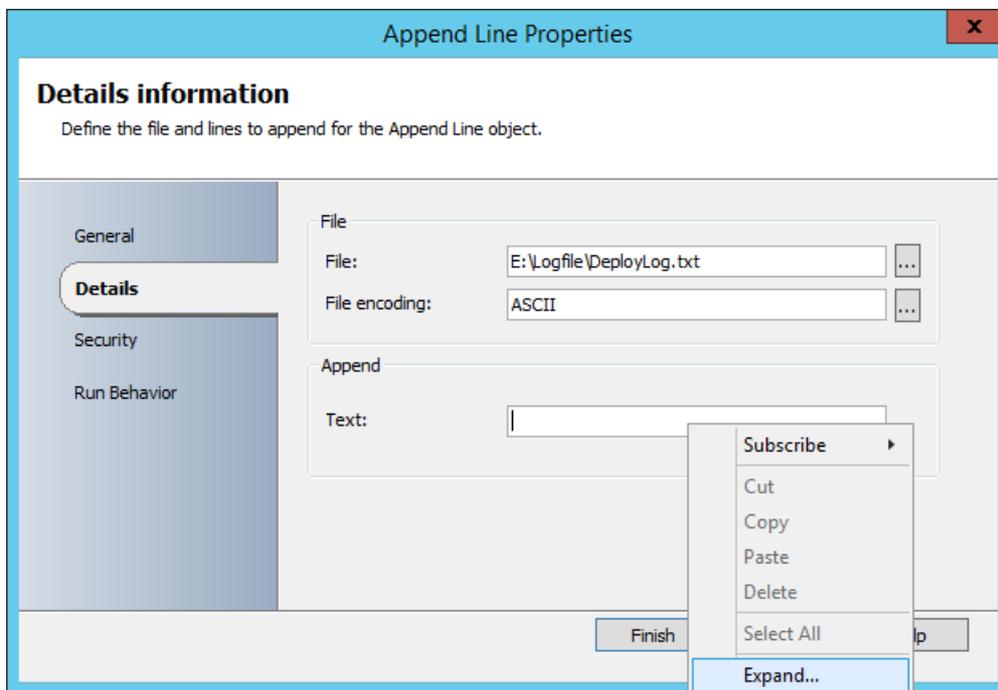


Figure 27. Expanding the Text area.

15. In the blank text box, right-click and select **Subscribe / Published Data**.

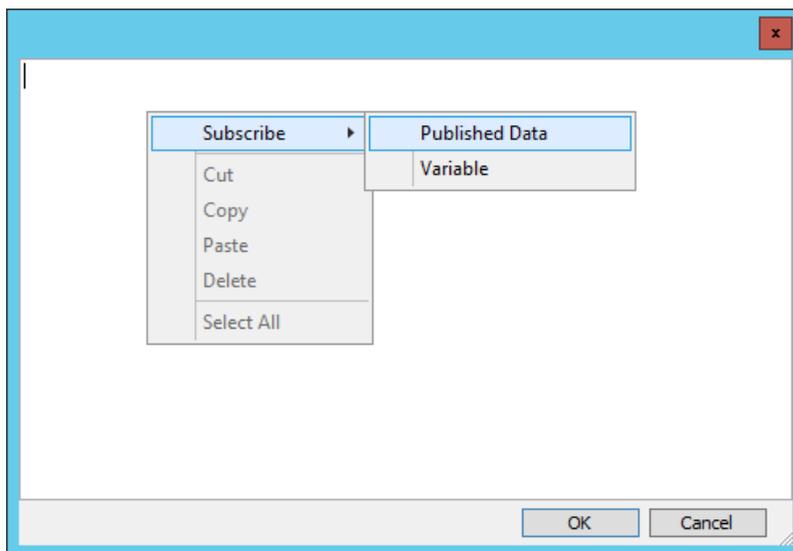


Figure 28. Subscribing to data.

16. In the **Published Data** window, select the **OSDComputerName** item, and click **OK**.
17. After the **{OSDComputerName from "Initialize Data"}** text, type in **has been deployed at** and, once again, right-click and select **Subscribe / Published Data**.
18. In the **Published Data** window, select the **Show common Published Data** check box, select the **Activity end time** item, and click **OK**.

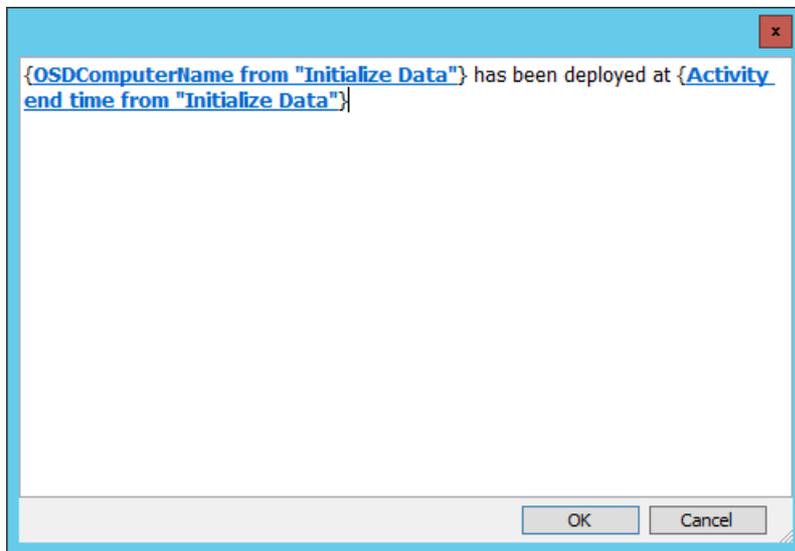


Figure 29. The expanded text box after all subscriptions have been added.

19. On the **Append Line Properties** page, click **Finish**.

Test the demo MDT runbook

After the runbook is created, you are ready to test it.

20. On the ribbon bar, click **Runbook Tester**.
21. Click **Run**, and in the **Initialize Data Parameters** dialog box, use the following setting and then click **OK**:
 - OSDComputerName: PC0010
22. Verify that all activities are green (for additional information, see each target).
23. Close the **Runbook Tester**.
24. On the ribbon bar, click **Check In**.

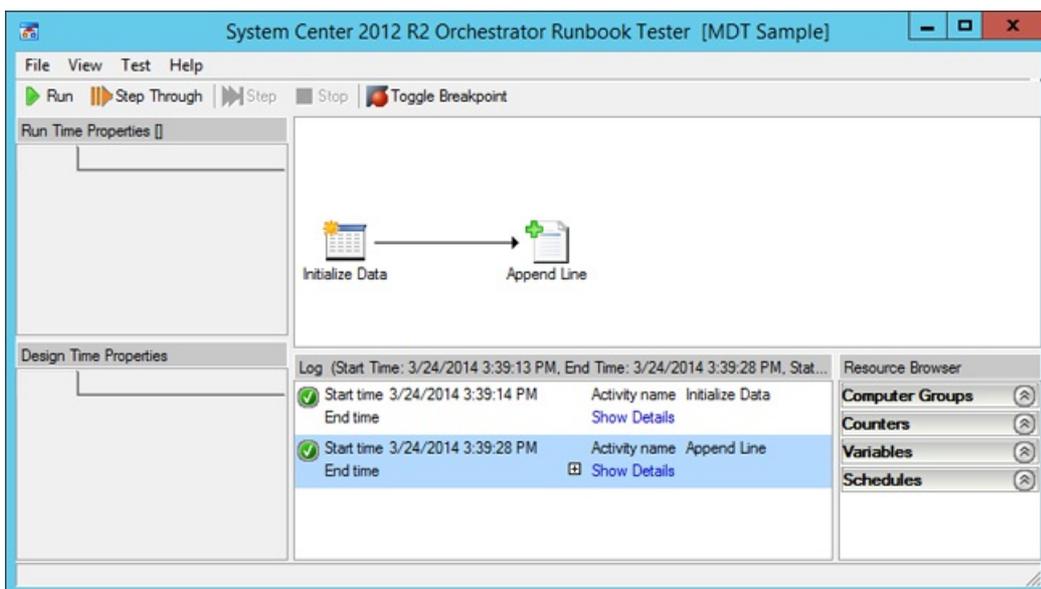


Figure 30. All tests completed.

Use the MDT demo runbook from MDT

1. On MDT01, using the Deployment Workbench, in the MDT Production deployment share, select the **Task Sequences** node, and create a folder named **Orchestrator**.

2. Right-click the **Orchestrator** node, and select **New Task Sequence**. Use the following settings for the New Task Sequence Wizard:
 - a. Task sequence ID: OR001
 - b. Task sequence name: Orchestrator Sample
 - c. Task sequence comments: <blank>
 - d. Template: Custom Task Sequence
3. In the **Orchestrator** node, double-click the **Orchestrator Sample** task sequence, and then select the **Task Sequence** tab.
4. Remove the default **Application Install** action.
5. Add a **Gather** action and select the **Gather only local data (do not process rules)** option.
6. After the **Gather** action, add a **Set Task Sequence Variable** action with the following settings:
 - a. Name: Set Task Sequence Variable
 - b. Task Sequence Variable: OSDComputerName
 - c. Value: %hostname%
7. After the **Set Task Sequence Variable** action, add a new **Execute Orchestrator Runbook** action with the following settings:
 - a. Orchestrator Server: OR01.contoso.com
 - b. Use Browse to select **1.0 MDT / MDT Sample**.
8. Click **OK**.

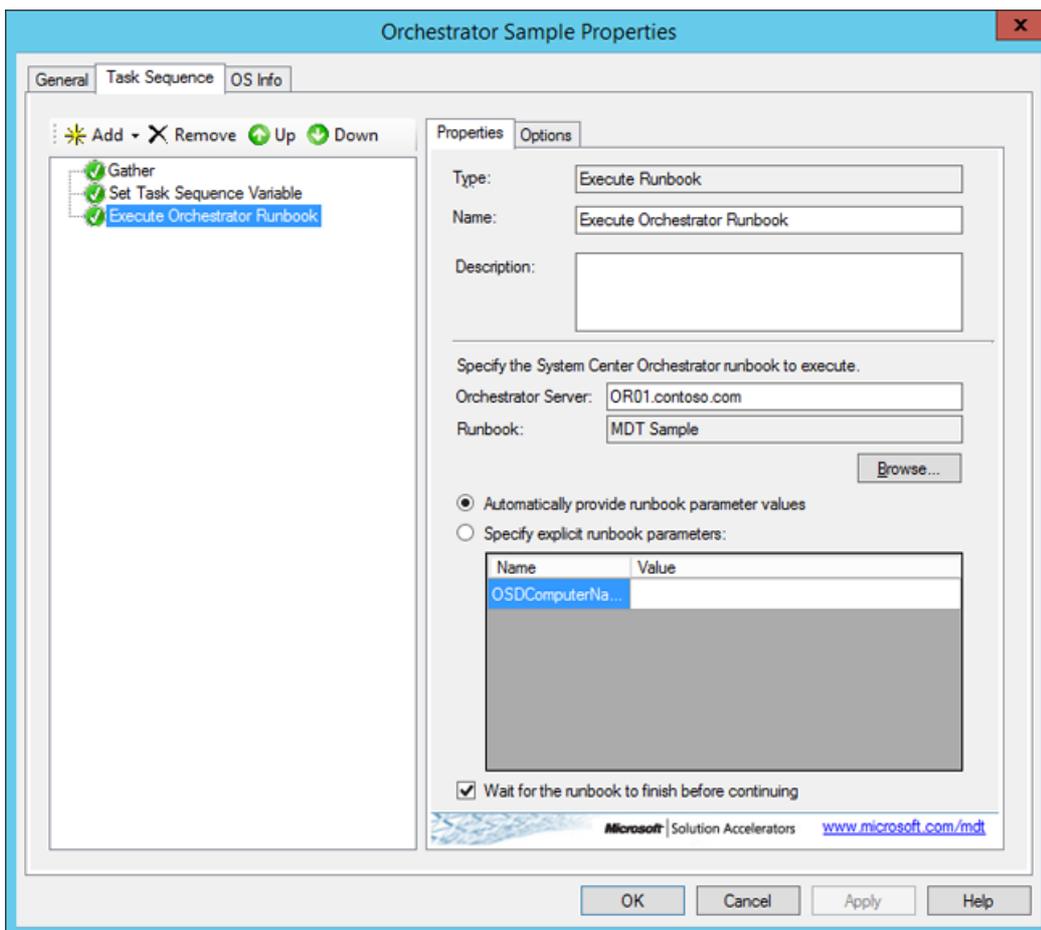


Figure 31. The ready-made task sequence.

Run the orchestrator sample task sequence

Since this task sequence just starts a runbook, you can test this on the PC0001 client that you used for the MDT simulation environment. **Note** Make sure the account you are using has permissions to run runbooks on the Orchestrator server. For more information about runbook permissions, see [Runbook Permissions](#).

1. On PC0001, log on as **CONTOSO\MDT_BA**.
2. Using an elevated command prompt (run as Administrator), type the following command:

```
cscript \\MDT01\MDTProduction$\Scripts\Litetouch.vbs
```

3. Complete the Windows Deployment Wizard using the following information:
 - a. Task Sequence: Orchestrator Sample
 - b. Credentials:
 - a. User Name: MDT_BA
 - b. Password: P@ssw0rd
 - c. Domain: CONTOSO
4. Wait until the task sequence is completed and then verify that the DeployLog.txt file in the E:\Logfile folder on OR01 was updated.

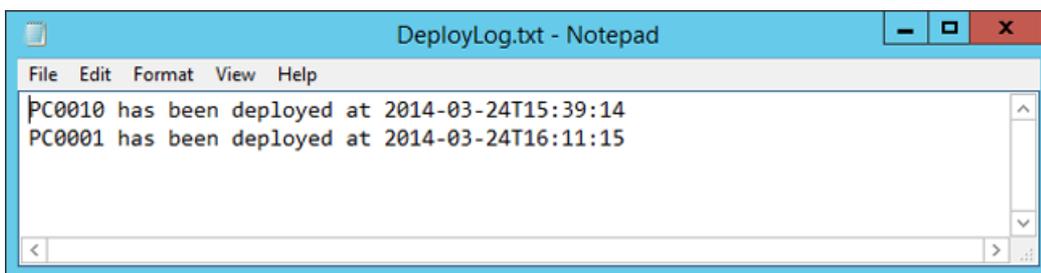


Figure 32. The ready-made task sequence.

Related topics

[Set up MDT for BitLocker](#)

[Configure MDT deployment share rules](#)

[Configure MDT for UserExit scripts](#)

[Simulate a Windows10 deployment in a test environment](#)

[Use the MDT database to stage Windows 10 deployment information](#)

[Assign applications using roles in MDT](#)

[Use web services in MDT](#)

Deploy Windows 10 with System Center 2012 R2 Configuration Manager

5/31/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

If you have Microsoft System Center 2012 R2 Configuration Manager in your environment, you will most likely want to use it to deploy Windows 10. This topic will show you how to set up Configuration Manager for operating system deployment and how to integrate Configuration Manager with the Microsoft Deployment Toolkit (MDT).

For the purposes of this topic, we will use four machines: DC01, CM01, PC0003, and PC0004. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 standard. PC0003 and PC0004 are machines with Windows 7 SP1, on which Windows 10 will be deployed via both refresh and replace scenarios. In addition to these four ready-made machines, you could also include a few blank virtual machines to be used for bare-metal deployments. DC01, CM01, PC003, and PC0004 are all members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

In this section

- [Integrate Configuration Manager with MDT](#)
- [Prepare for Zero Touch Installation of Windows with Configuration Manager](#)
- [Create a custom Windows PE boot image with Configuration Manager](#)
- [Add a Windows 10 operating system image using Configuration Manager](#)
- [Create an application to deploy with Windows 10 using Configuration Manager](#)
- [Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)
- [Create a task sequence with Configuration Manager and MDT](#)
- [Finalize the operating system configuration for Windows 10 deployment with Configuration Manager](#)
- [Deploy Windows 10 using PXE and Configuration Manager](#)

- [Monitor the Windows 10 deployment with Configuration Manager](#)
- [Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)
- [Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Components of Configuration Manager operating system deployment

Operating system deployment with Configuration Manager is part of the normal software distribution infrastructure, but there are additional components. For example, operating system deployment in Configuration Manager may use the State Migration Point role, which is not used by normal application deployment in Configuration Manager. This section describes the Configuration Manager components involved with the deployment of an operating system, such as Windows 10.

- **State migration point (SMP).** The state migration point is used to store user state migration data during computer replace scenarios.
- **Distribution point (DP).** The distribution point is used to store all packages in Configuration Manager, including the operating system deployment-related packages.
- **Software update point (SUP).** The software update point, which is normally used to deploy updates to existing machines, also can be used to update an operating system as part of the deployment process. You also can use offline servicing to update the image directly on the Configuration Manager server.
- **Reporting services point.** The reporting services point can be used to monitor the operating system deployment process.
- **Boot images.** Boot images are the Windows Preinstallation Environment (Windows PE) images Configuration Manager uses to start the deployment.
- **Operating system images.** The operating system image package contains only one file, the custom .wim image. This is typically the production deployment image.
- **Operating system installers.** The operating system installers were originally added to create reference images using Configuration Manager. Instead, we recommend that you use MDT Lite Touch to create your reference images. For more information on how to create a reference image, see [Create a Windows 10 reference image](#).
- **Drivers.** Like MDT Lite Touch, Configuration Manager also provides a repository (catalog) of managed device drivers.
- **Task sequences.** The task sequences in Configuration Manager look and feel pretty much like the sequences in MDT Lite Touch, and they are used for the same purpose. However, in Configuration Manager the task sequence is delivered to the clients as a policy via the Management Point (MP). MDT provides additional task sequence templates to Configuration Manager.

Note Configuration Manager SP1 along with the Windows Assessment and Deployment Kit (ADK) for Windows 10 are required to support management and deployment of Windows 10.

See also

- [Microsoft Deployment Toolkit downloads and resources](#)
- [Windows deployment tools](#)
- [Deploy Windows 10 with the Microsoft Deployment Toolkit](#)

- [Upgrade to Windows 10 with the Microsoft Deployment Toolkit](#)
- [Deploy Windows To Go in your organization](#)
- [Sideload Windows Store apps](#)
- [Windows ADK for Windows 10](#)

Integrate Configuration Manager with MDT

6/14/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic will help you understand the benefits of integrating the Microsoft Deployment Toolkit with Microsoft System Center 2012 R2 Configuration Manager SP1 when you deploy a new or updated version of the Windows operating system. MDT is a free, supported download from Microsoft that adds approximately 280 enhancements to Windows operating system deployment with System Center 2012 R2 Configuration Manager SP1. It is, therefore, recommended that you utilize MDT when deploying the Windows operating system with Configuration Manager SP1. In addition to integrating MDT with Configuration Manager, we also recommend using MDT Lite Touch to create the Windows 10 reference images used in Configuration Manager. For more information on how to create a reference image, see [Create a Windows 10 reference image](#).

Why integrate MDT with Configuration Manager

As noted above, MDT adds many enhancements to Configuration Manager. While these enhancements are called Zero Touch, that name does not reflect how deployment is conducted. The following sections provide a few samples of the 280 enhancements that MDT adds to Configuration Manager.

MDT enables dynamic deployment

When MDT is integrated with Configuration Manager, the task sequence takes additional instructions from the MDT rules. In its most simple form, these settings are stored in a text file, the CustomSettings.ini file, but you can store the settings in Microsoft SQL Server databases, or have Microsoft Visual Basic Scripting Edition (VBScripts) or web services provide the settings used.

The task sequence uses instructions that allow you to reduce the number of task sequences in Configuration Manager and instead store settings outside the task sequence. Here are a few examples:

- The following settings instruct the task sequence to install the HP Hotkeys package, but only if the hardware is a HP EliteBook 8570w. Note that you don't have to add the package to the task sequence.

```
[Settings]
Priority=Model
[HP EliteBook 8570w]
Packages001=PS100010:Install HP Hotkeys
```

- The following settings instruct the task sequence to put laptops and desktops in different organizational units (OUs) during deployment, assign different computer names, and finally have the task sequence install the Cisco VPN client, but only if the machine is a laptop.

```

[Settings]
Priority= ByLaptopType, ByDesktopType
[ByLaptopType]
Subsection=Laptop-%IsLaptop%
[ByDesktopType]
Subsection=Desktop-%IsDesktop%
[Laptop-True]
Packages001=PS100012:Install Cisco VPN Client
OSDComputerName=LT-%SerialNumber%
MachineObjectOU=ou=laptops,ou=Contoso,dc=contoso,dc=com
[Desktop-True]
OSDComputerName=DT-%SerialNumber%
MachineObjectOU=ou=desktops,ou=Contoso,dc=contoso,dc=com

```

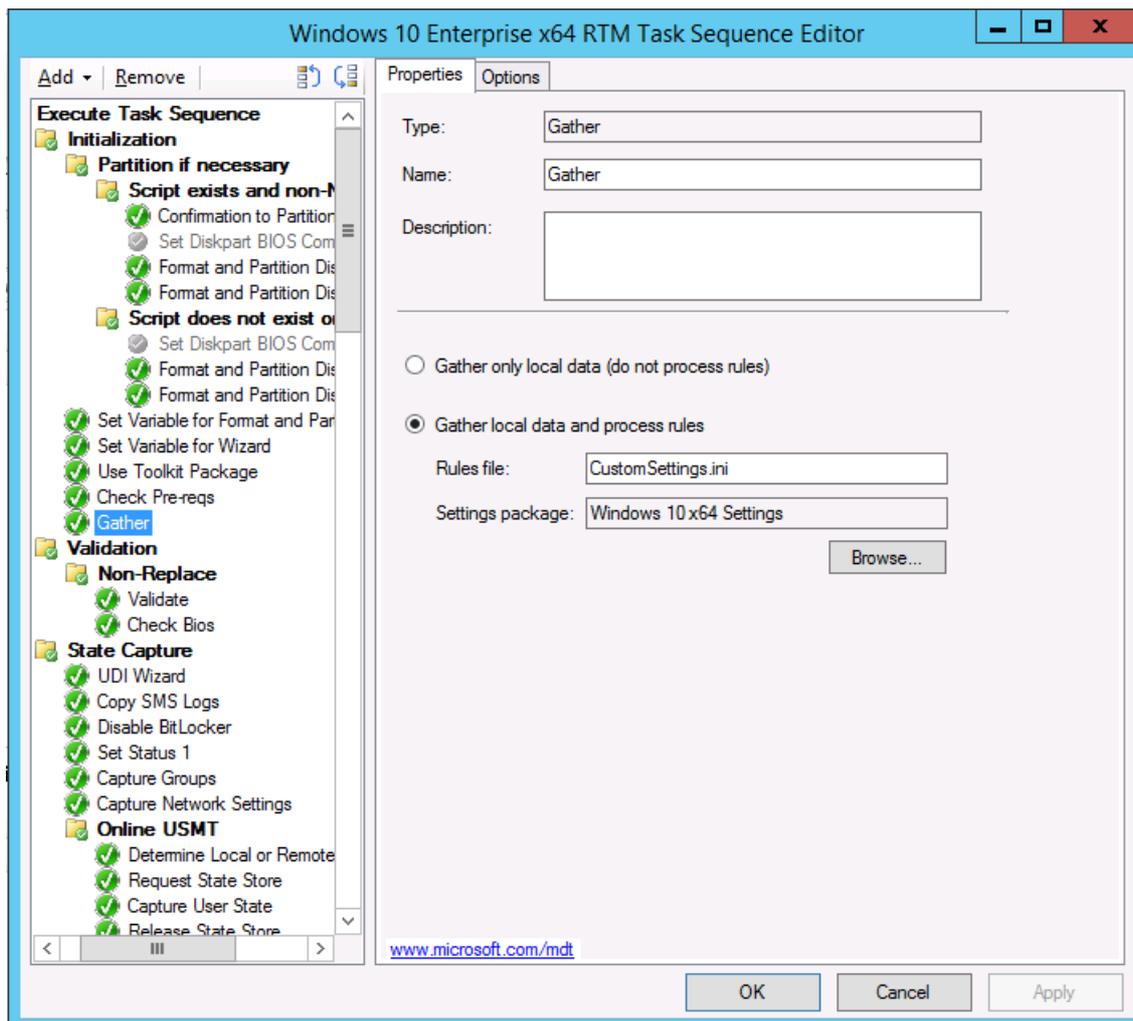


Figure 2. The Gather action in the task sequence is reading the rules.

MDT adds an operating system deployment simulation environment

When testing a deployment, it is important to be able to quickly test any changes you make to the deployment without needing to run through an entire deployment. MDT rules can be tested very quickly, saving significant testing time in a deployment project. For more information, see [Configure MDT settings](#).

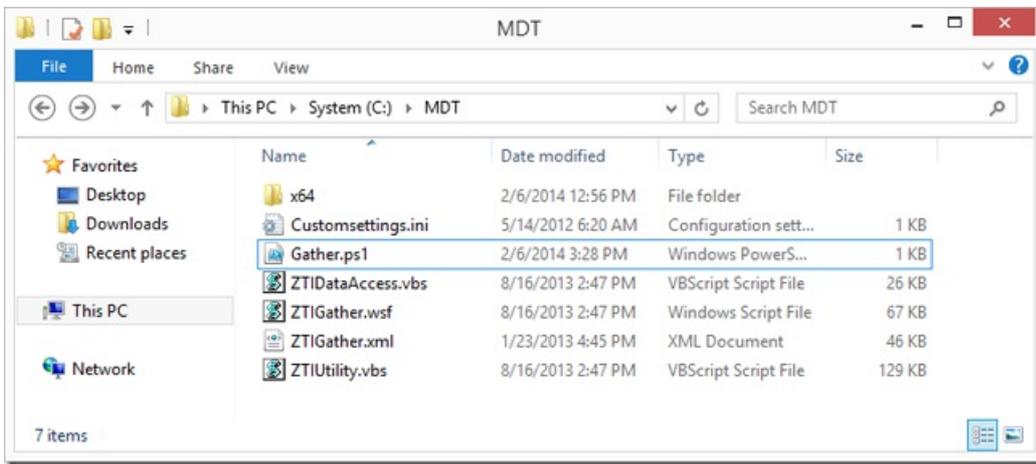


Figure 3. The folder that contains the rules, a few scripts from MDT, and a custom script (Gather.ps1).

MDT adds real-time monitoring

With MDT integration, you can follow your deployments in real time, and if you have access to Microsoft Diagnostics and Recovery Toolkit (DaRT), you can even remote into Windows Preinstallation Environment (Windows PE) during deployment. The real-time monitoring data can be viewed from within the MDT Deployment Workbench, via a web browser, Windows PowerShell, the Event Viewer, or Microsoft Excel 2013. In fact, any script or app that can read an Open Data (OData) feed can read the information.

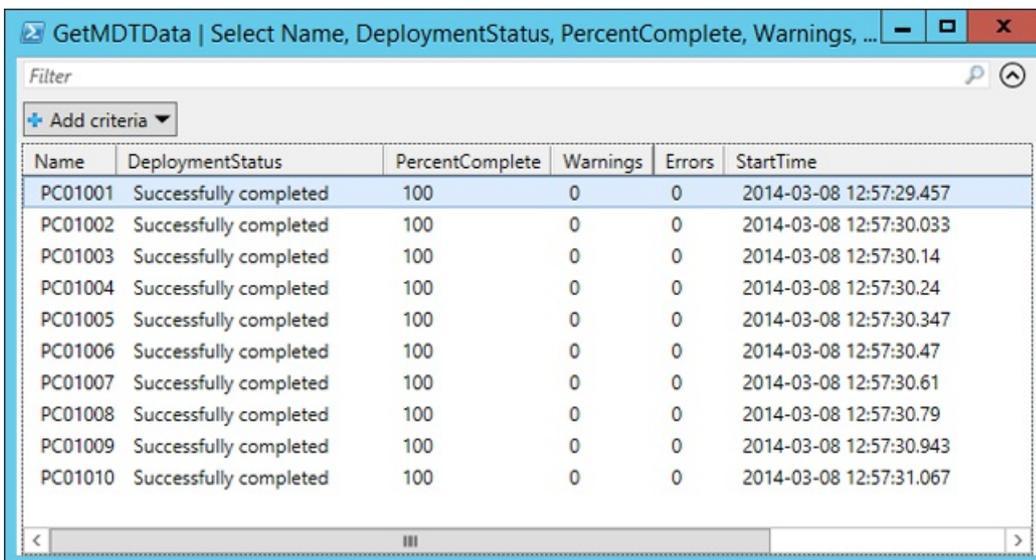


Figure 4. View the real-time monitoring data with PowerShell.

MDT adds an optional deployment wizard

For some deployment scenarios, you may need to prompt the user for information during deployment such as the computer name, the correct organizational unit (OU) for the computer, or which applications should be installed by the task sequence. With MDT integration, you can enable the User-Driven Installation (UDI) wizard to gather the required information, and customize the wizard using the UDI Wizard Designer.

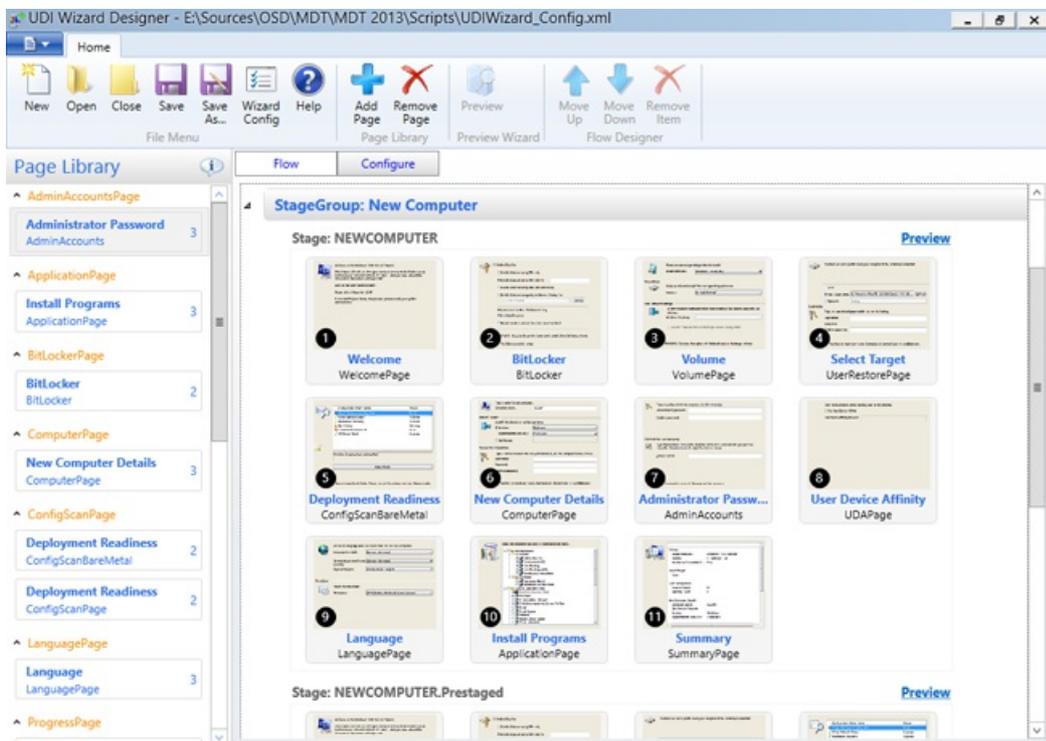


Figure 5. The optional UDI wizard open in the UDI Wizard Designer.

MDT Zero Touch simply extends Configuration Manager with many useful built-in operating system deployment components. By providing well-established, supported solutions, MDT reduces the complexity of deployment in Configuration Manager.

Why use MDT Lite Touch to create reference images

You can create reference images for Configuration Manager in Configuration Manager, but in general we recommend creating them in MDT Lite Touch for the following reasons:

- In a deployment project, it is typically much faster to create a reference image using MDT Lite Touch than Configuration Manager.
- You can use the same image for every type of operating system deployment - Microsoft Virtual Desktop Infrastructure (VDI), Microsoft System Center 2012 R2 Virtual Machine Manager (SCVMM), MDT, Configuration Manager, Windows Deployment Services (WDS), and more.
- Microsoft System Center 2012 R2 performs deployment in the LocalSystem context. This means that you cannot configure the Administrator account with all of the settings that you would like to be included in the image. MDT runs in the context of the Local Administrator, which means you can configure the look and feel of the configuration and then use the CopyProfile functionality to copy these changes to the default user during deployment.
- The Configuration Manager task sequence does not suppress user interface interaction.
- MDT Lite Touch supports a Suspend action that allows for reboots, which is useful when you need to perform a manual installation or check the reference image before it is automatically captured.
- MDT Lite Touch does not require any infrastructure and is easy to delegate.

Related topics

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager

Create a task sequence with Configuration Manager and MDT

Deploy Windows 10 using PXE and Configuration Manager

Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager

Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager

Prepare for Zero Touch Installation of Windows 10 with Configuration Manager

6/10/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

This topic will walk you through the process of integrating Microsoft System Center 2012 R2 Configuration Manager SP1 with Microsoft Deployment Toolkit (MDT) 2013 Update 2, as well as the other preparations needed to deploying Windows 10 via Zero Touch Installation. Additional preparations include the installation of hotfixes as well as activities that speed up the Pre-Boot Execution Environment (PXE).

Prerequisites

In this topic, you will use an existing Configuration Manager server structure to prepare for operating system deployment. In addition to the base setup, the following configurations should be made in the Configuration Manager environment:

- Active Directory Schema has been extended and System Management container created.
- Active Directory Forest Discovery and Active Directory System Discovery have been enabled.
- IP range boundaries and a boundary group for content and site assignment have been created.
- The Configuration Manager reporting services point role has been added and configured
- A file system folder structure for packages has been created.
- A Configuration Manager console folder structure for packages has been created.
- System Center 2012 R2 Configuration Manager SP1 and any additional Windows 10 prerequisites are installed.

For the purposes of this topic, we will use two machines: DC01 and CM01. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. DC01 and CM01 are both members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Create the Configuration Manager service accounts

To configure permissions for the various service accounts needed for operating system deployment in Configuration Manager, you use a role-based model. To create the Configuration Manager Join Domain account as well as the Configuration Manager Network Access account, follow these steps:

1. On DC01, using Active Directory User and Computers, browse to **contoso.com / Contoso / Service Accounts**.
2. Select the Service Accounts OU and create the CM_JD account using the following settings:
 - Name: CM_JD
 - User logon name: CM_JD
 - Password: P@ssw0rd
 - User must change password at next logon: Clear
 - User cannot change password: Select
 - Password never expires: Select
3. Repeat the step, but for the CM_NAA account.
4. After creating the accounts, assign the following descriptions:
 - CM_JD: Configuration Manager Join Domain Account
 - CM_NAA: Configuration Manager Network Access Account

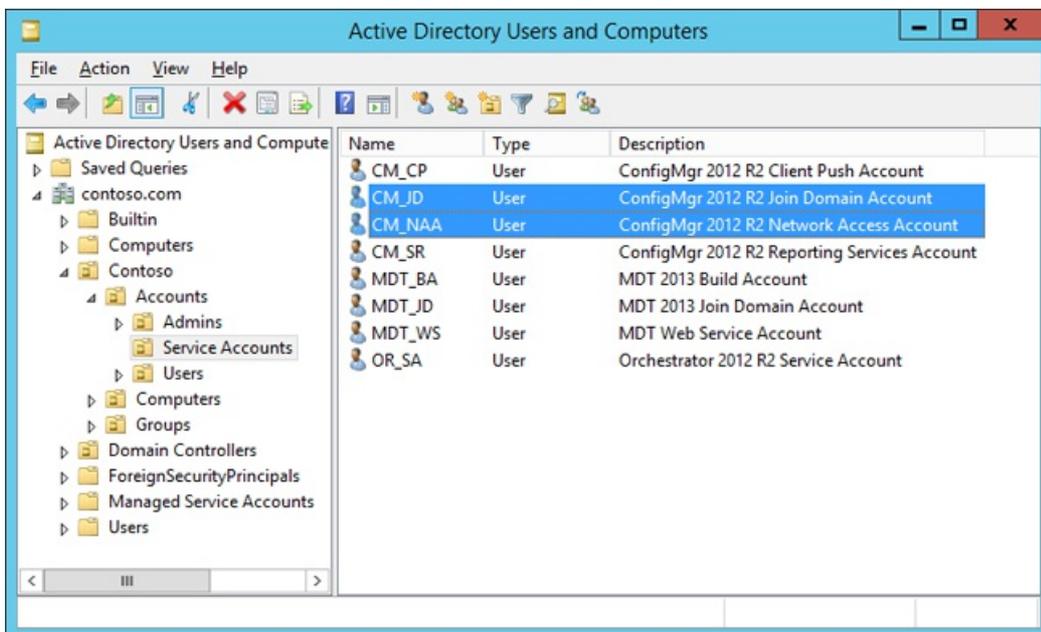


Figure 6. The Configuration Manager service accounts used for operating system deployment.

Configure Active Directory permissions

In order for the Configuration Manager Join Domain Account (CM_JD) to join machines into the contoso.com domain you need to configure permissions in Active Directory. These steps assume you have downloaded the sample [Set-OUPermissions.ps1](#) script and copied it to C:\Setup\Scripts on DC01.

1. On DC01, log on as Administrator in the CONTOSO domain using the password **P@ssw0rd**.
2. In an elevated Windows PowerShell prompt (run as Administrator), run the following commands, pressing **Enter** after each command:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Force
```

```
Set-Location C:\Setup\Scripts
```

```
.\Set-OUPermissions.ps1 -Account CM_JD  
-TargetOU "OU=Workstations,OU=Computers,OU=Contoso"
```

3. The Set-OUPermissions.ps1 script allows the CM_JD user account permissions to manage computer accounts in the Contoso / Computers / Workstations OU. The following is a list of the permissions being granted:

- Scope: This object and all descendant objects
- Create Computer objects
- Delete Computer objects
- Scope: Descendant Computer objects
- Read All Properties
- Write All Properties
- Read Permissions
- Modify Permissions
- Change Password
- Reset Password
- Validated write to DNS host name
- Validated write to service principal name

Review the Sources folder structure

To support the packages you create in this section, the following folder structure should be created on the Configuration Manager primary site server (CM01):

NOTE

In most production environments, the packages are stored on a Distributed File System (DFS) share or a "normal" server share, but in a lab environment you can store them on the site server.

- E:\Sources
- E:\Sources\OSD
- E:\Sources\OSD\Boot
- E:\Sources\OSD\DriverPackages
- E:\Sources\OSD\DriverSources
- E:\Sources\OSD\MDT
- E:\Sources\OSD\OS
- E:\Sources\OSD\Settings

- E:\Sources\Software
- E:\Sources\Software\Adobe
- E:\Sources\Software\Microsoft

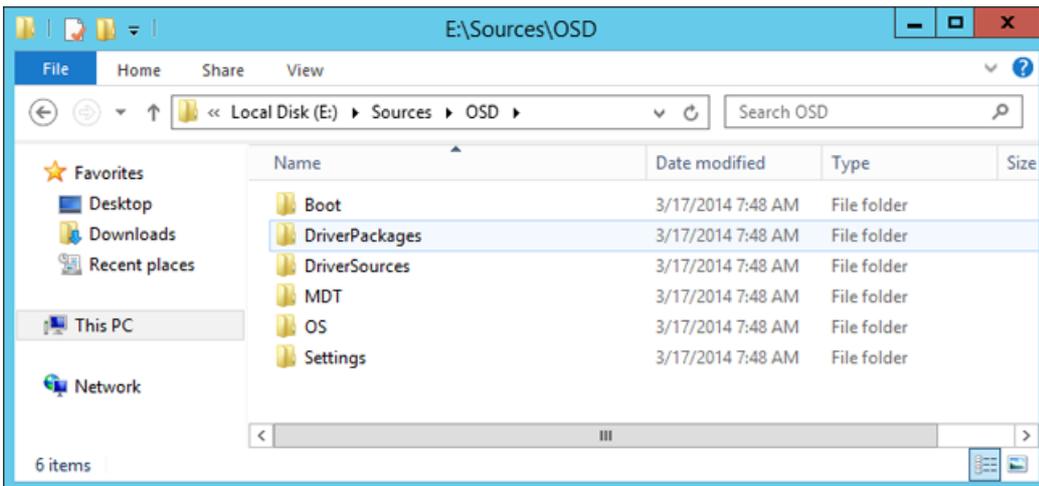


Figure 7. The E:\Sources\OSD folder structure.

Integrate Configuration Manager with MDT

To extend the Configuration Manager console with MDT wizards and templates, you install MDT in the default location and run the integration setup. In these steps, we assume you have downloaded MDT to the C:\Setup\MDT2013 folder on CM01.

1. On CM01, log on as Administrator in the CONTOSO domain using the password **P@ssw0rd**.
2. Make sure the Configuration Manager Console is closed before continuing.
3. Using File Explorer, navigate to the **C:\Setup\MDT** folder.
4. Run the MDT setup (MicrosoftDeploymentToolkit2013_x64.msi), and use the default options in the setup wizard.
5. From the Start screen, run Configure ConfigManager Integration with the following settings:
 - Site Server Name: CM01.contoso.com
 - Site code: PS1

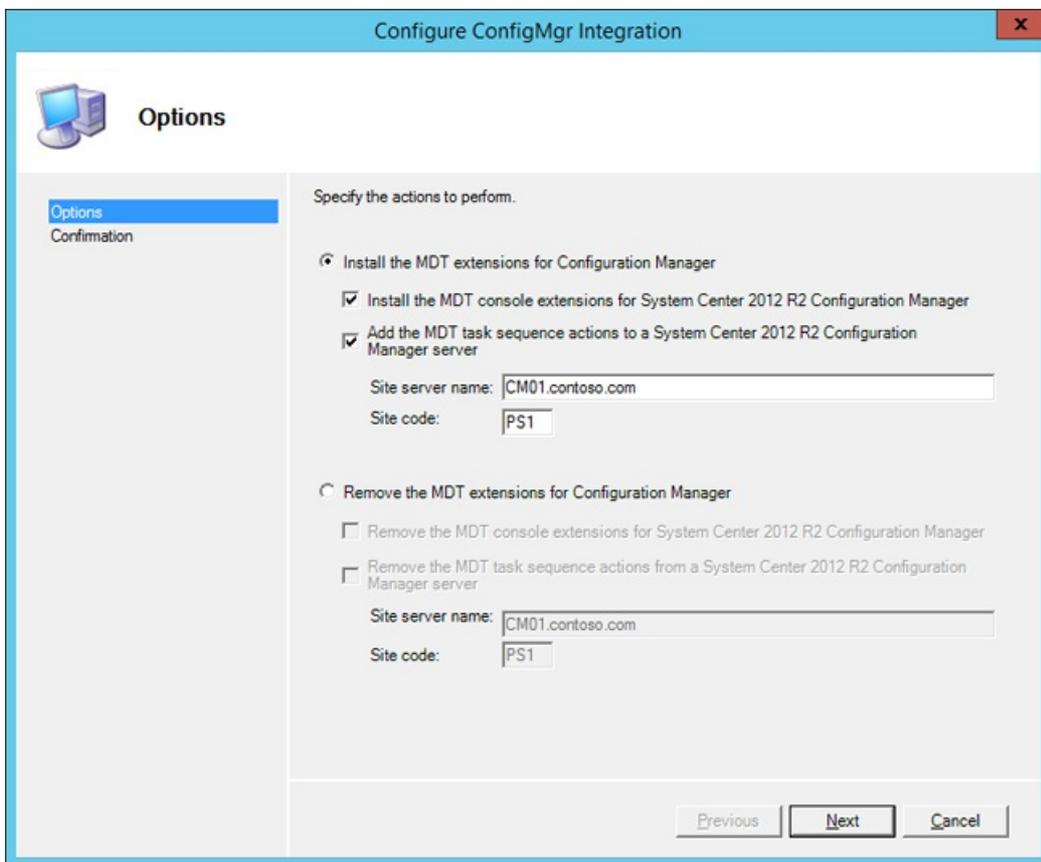


Figure 8. Set up the MDT integration with Configuration Manager.

Configure the client settings

Most organizations want to display their name during deployment. In this section, you configure the default Configuration Manager client settings with the Contoso organization name.

1. On CM01, using the Configuration Manager Console, in the Administration workspace, select **Client Settings**.
2. In the right pane, right-click **Default Client Settings**, and select **Properties**.
3. In the **Computer Agent** node, in the **Organization name displayed in Software Center** text box, type in **Contoso** and click **OK**.

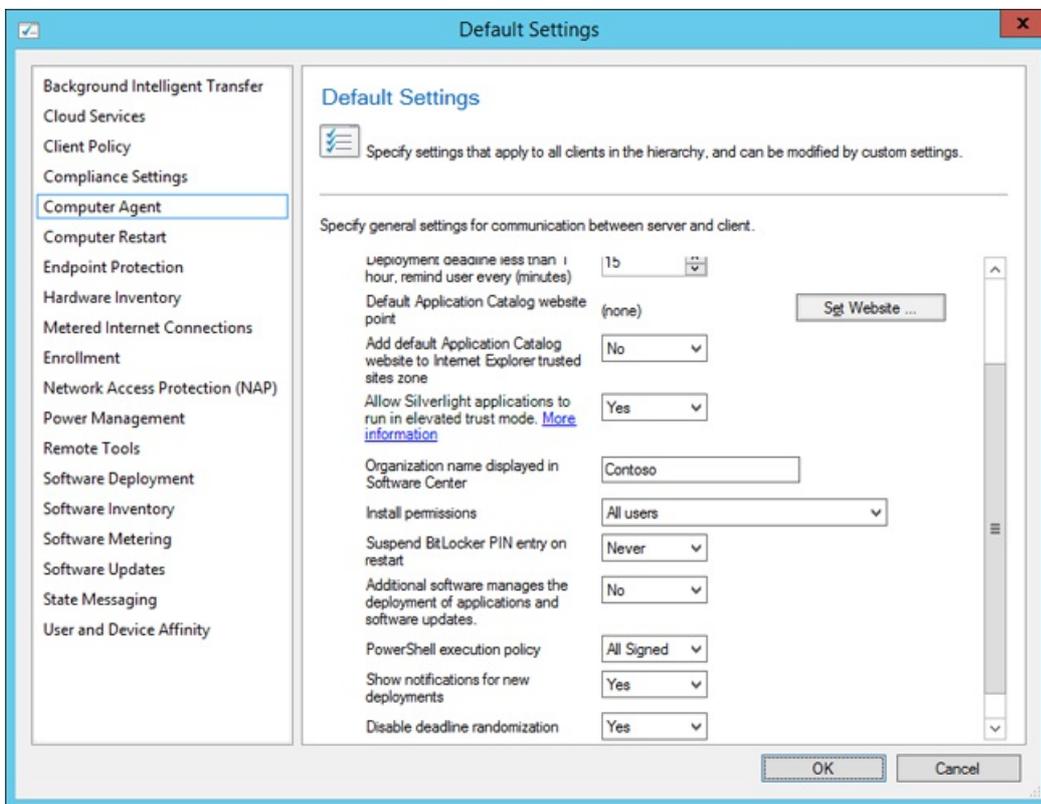


Figure 9. Configure the organization name in client settings.

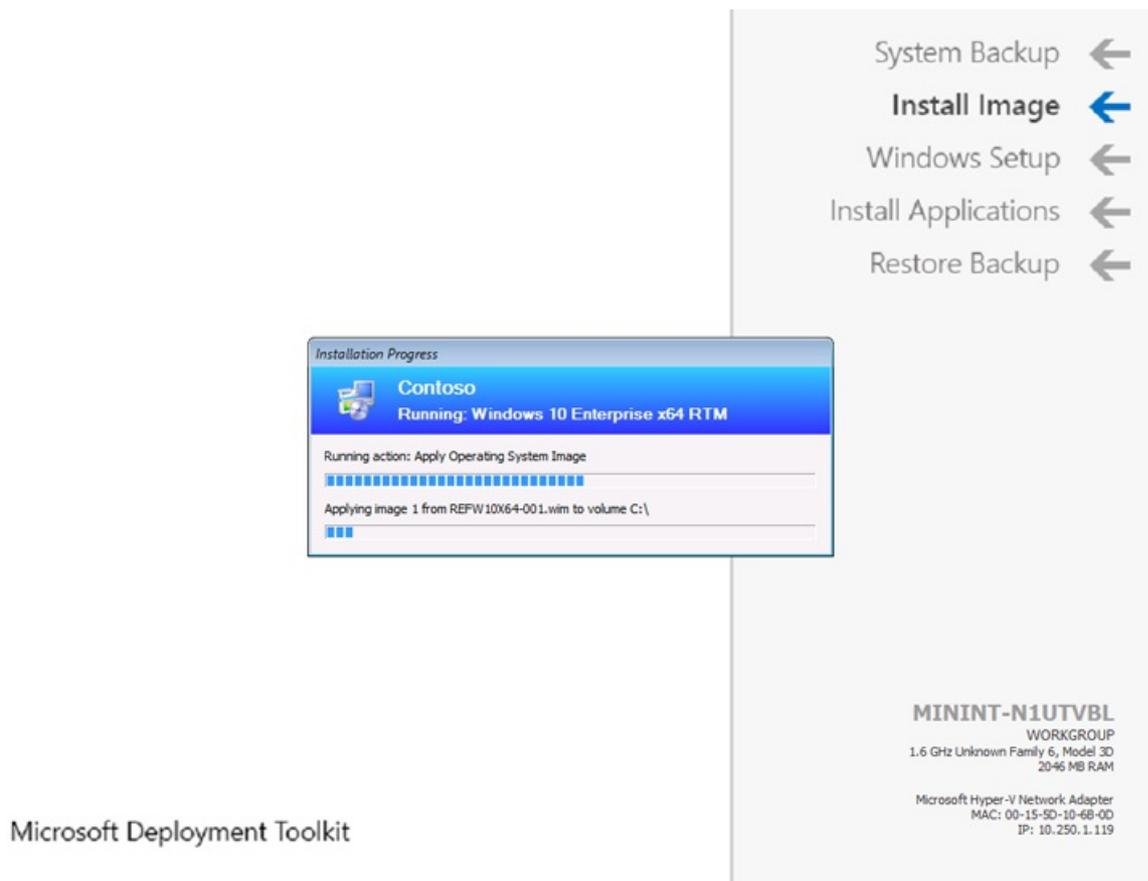


Figure 10. The Contoso organization name displayed during deployment.

Configure the Network Access account

Configuration Manager uses the Network Access account during the Windows 10 deployment process to access content on the distribution point(s). In this section, you configure the Network Access account.

1. Using the Configuration Manager Console, in the Administration workspace, expand **Site Configuration** and select **Sites**.
2. Right-click **PS1 - Primary Site 1**, select **Configure Site Components**, and then select **Software Distribution**.
3. In the **Network Access Account** tab, configure the **CONTOSO\CM_NAA** user account (select New Account) as the Network Access account. Use the new **Verify** option to verify that the account can connect to the **\\DC01\sysvol** network share.

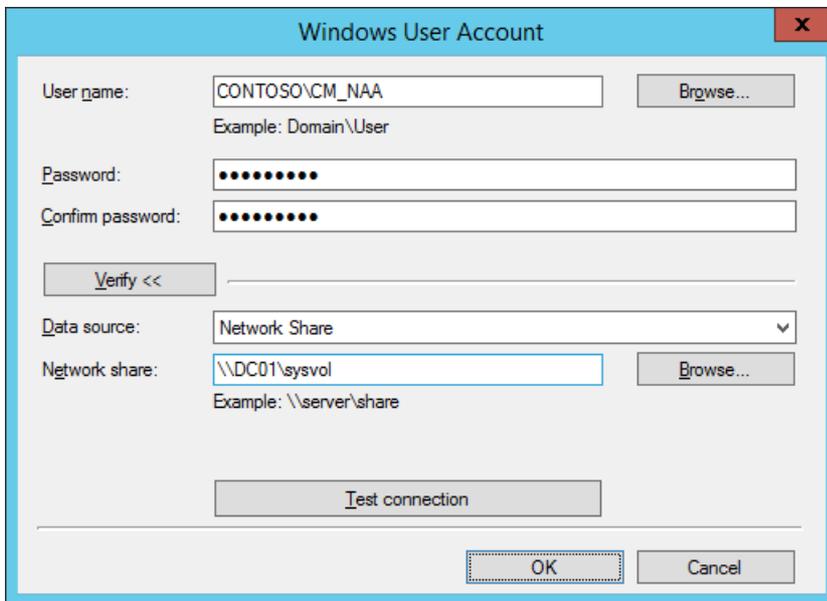


Figure 11. Test the connection for the Network Access account.

Enable PXE on the CM01 distribution point

Configuration Manager has many options for starting a deployment, but starting via PXE is certainly the most flexible in a large environment. In this section, you enable PXE on the CM01 distribution point.

1. In the Configuration Manager Console, in the Administration workspace, select **Distribution Points**.
2. Right-click the **\\CM01.CONTOSO.COM distribution point** and select **Properties**.
3. In the **PXE** tab, select the following settings:
 - Enable PXE support for clients
 - Allow this distribution point to respond to incoming PXE requests
 - Enable unknown computer support
 - Require a password when computers use PXE
 - Password and Confirm password: Passw0rd!

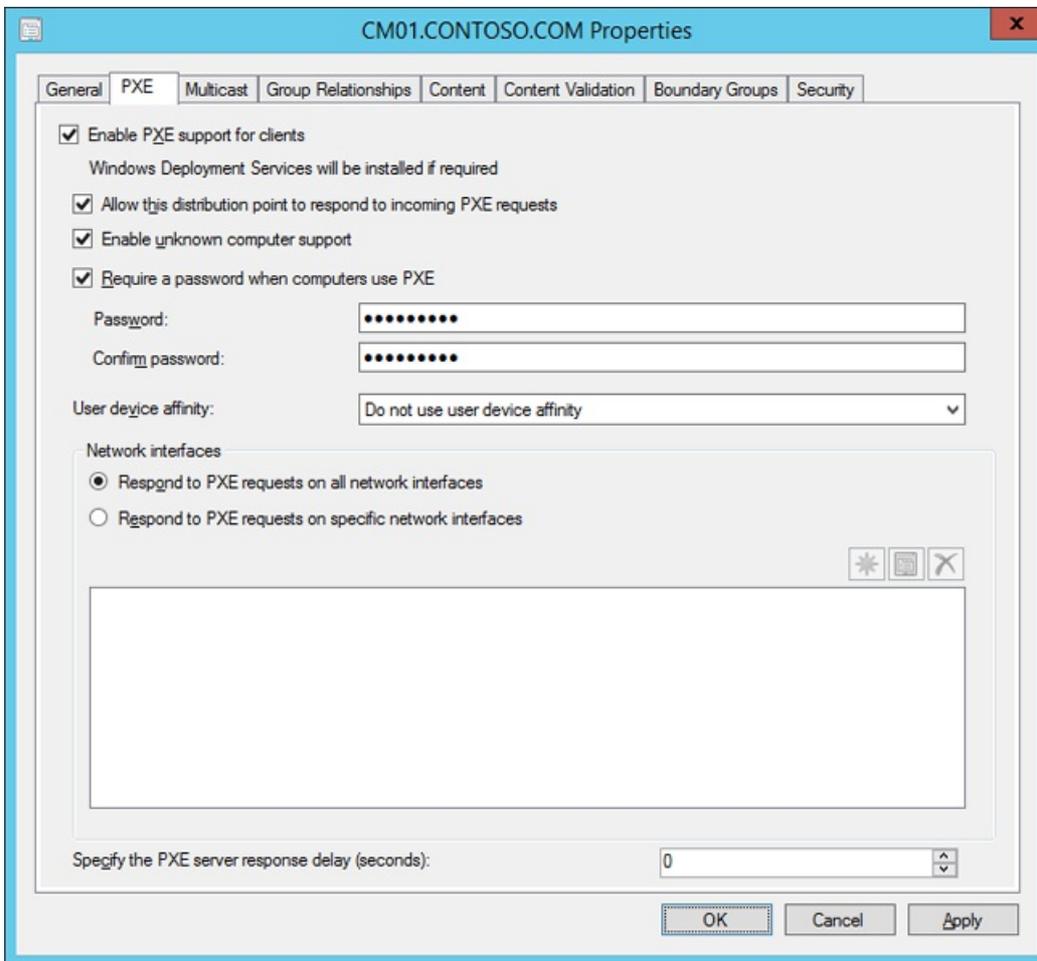


Figure 12. Configure the CM01 distribution point for PXE.

- Using the Configuration Manager Trace Log Tool, review the E:\Program Files\Microsoft Configuration Manager\Logs\distmgr.log file. Look for ConfigurePXE and CcmInstallPXE lines.

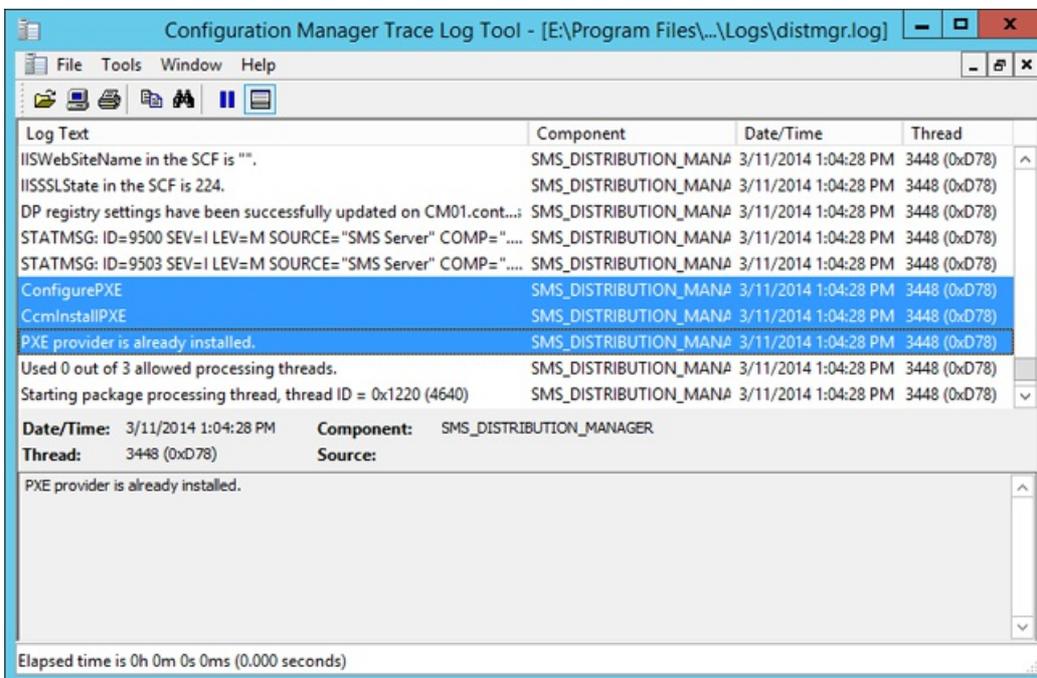


Figure 13. The distmgr.log displays a successful configuration of PXE on the distribution point.

- Verify that you have seven files in each of the folders **E:\RemoteInstall\SMSBoot\x86** and **E:\RemoteInstall\SMSBoot\x64**.

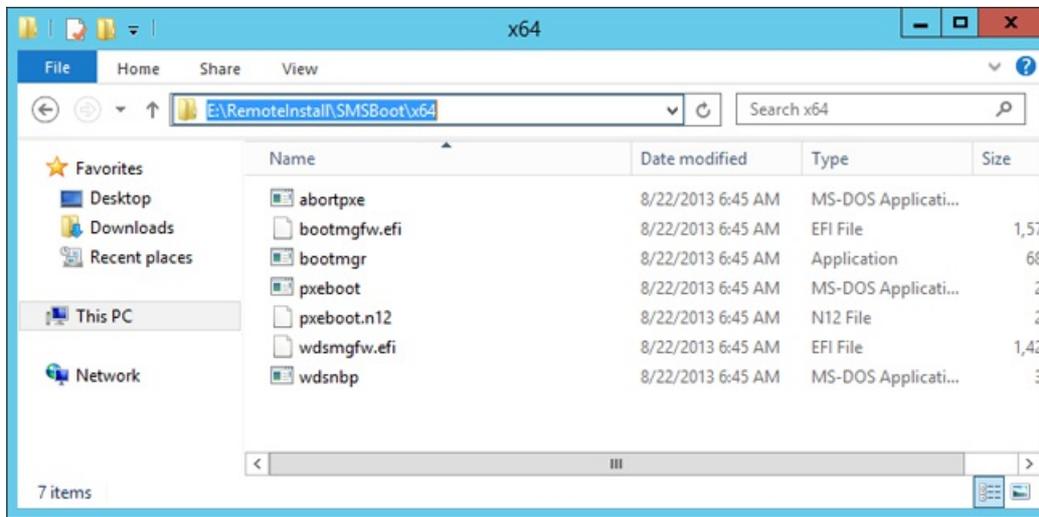


Figure 14. The contents of the E:\RemoteInstall\SMSBoot\x64 folder after you enable PXE.

Related topics

[Integrate Configuration Manager with MDT](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Create a custom Windows PE boot image with Configuration Manager

5/31/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

In Microsoft System Center 2012 R2 Configuration Manager, you can create custom Windows Preinstallation Environment (Windows PE) boot images that include extra components and features. This topic shows you how to create a custom Windows PE 5.0 boot image with the Microsoft Deployment Toolkit (MDT) wizard. You can also add the Microsoft Diagnostics and Recovery Toolset (DaRT) 10 to the boot image as part of the boot image creation process.

For the purposes of this topic, we will use two machines: DC01 and CM01. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. Both are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Add DaRT 10 files and prepare to brand the boot image

The steps below outline the process for adding DaRT 10 installation files to the MDT installation directory. You also copy a custom background image to be used later. We assume you have downloaded Microsoft Desktop Optimization Pack (MDOP) 2015 and copied the x64 version of MSDaRT10.msi to the C:\Setup\DaRT 10 folder. We also assume you have created a custom background image and saved it in C:\Setup\Branding on CM01. In this section, we use a custom background image named ContosoBackground.bmp.

1. Install DaRT 10 (C:\Setup\DaRT 10\MSDaRT10.msi) using the default settings.
2. Using File Explorer, navigate to the **C:\Program Files\Microsoft DaRT\v10** folder.
3. Copy the Toolsx64.cab file to the **C:\Program Files\Microsoft Deployment Toolkit\Templates\Distribution\Tools\x64** folder.
4. Copy the Toolsx86.cab file to the **C:\Program Files\Microsoft Deployment Toolkit\Templates\Distribution\Tools\x86** folder.
5. Using File Explorer, navigate to the **C:\Setup** folder.
6. Copy the **Branding** folder to **E:\Sources\OSD**.

Create a boot image for Configuration Manager using the MDT wizard

By using the MDT wizard to create the boot image in Configuration Manager, you gain additional options for

adding components and features to the boot image. In this section, you create a boot image for Configuration Manager using the MDT wizard.

1. Using the Configuration Manager Console, in the Software Library workspace, expand **Operating Systems**, right-click **Boot Images**, and select **Create Boot Image using MDT**.
2. On the **Package Source** page, in the **Package source folder to be created (UNC Path):** text box, type `\\CM01\Sources$\OSD\Boot\Zero Touch WinPE x64` and click **Next**.

NOTE

The Zero Touch WinPE x64 folder does not yet exist. The folder will be created later by the wizard.

3. On the **General Settings** page, assign the name **Zero Touch WinPE x64** and click **Next**.
4. On the **Options** page, select the **x64** platform, and click **Next**.
5. On the **Components** page, in addition to the default selected **Microsoft Data Access Components (MDAC/ADO)** support, select the **Microsoft Diagnostics and Recovery Toolkit (DaRT)** check box.

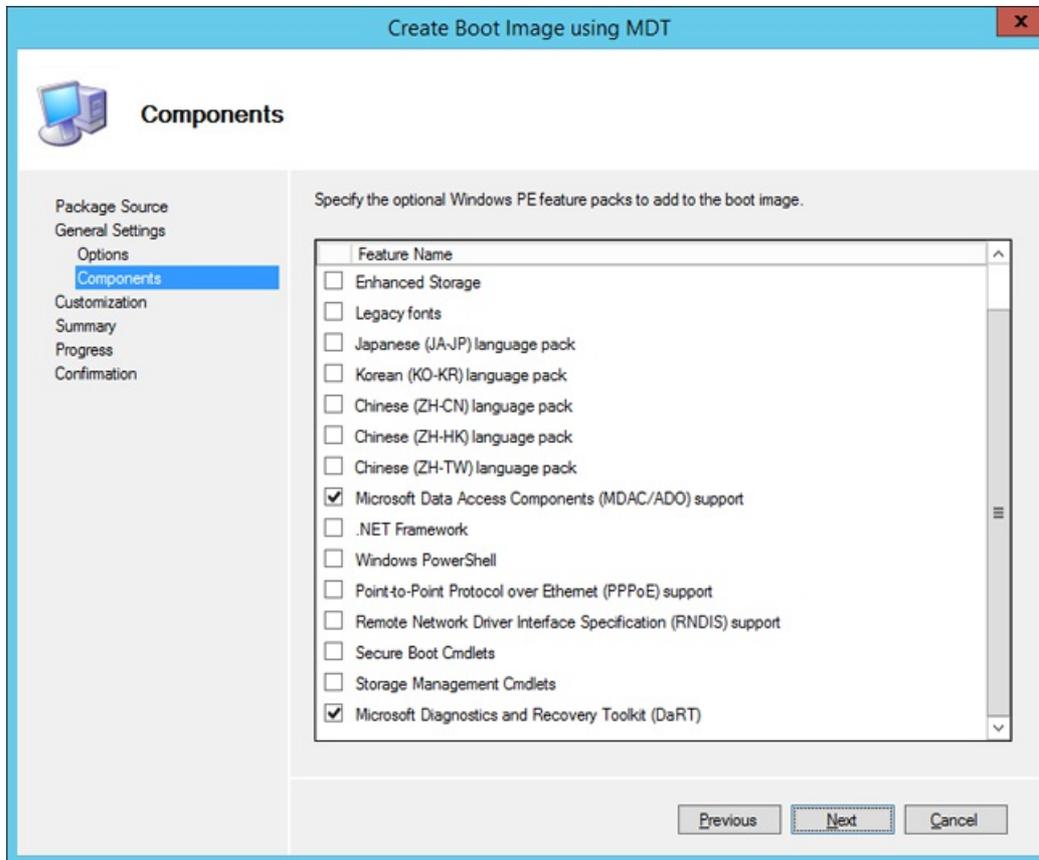


Figure 15. Add the DaRT component to the Configuration Manager boot image.

6. On the **Customization** page, select the **Use a custom background bitmap file** check box, and in the **UNC path:** text box, browse to `\\CM01\Sources$\OSD\Branding\ContosoBackground.bmp`. Then click **Next** twice.

NOTE

It will take a few minutes to generate the boot image.

7. Distribute the boot image to the CM01 distribution point by selecting the **Boot images** node, right-clicking the **Zero Touch WinPE x64** boot image, and selecting **Distribute Content**.

8. In the Distribute Content Wizard, add the CM01 distribution point, and complete the wizard.
9. Using Configuration Manager Trace, review the E:\Program Files\Microsoft Configuration Manager\Logs\distmgr.log file. Do not continue until you can see that the boot image is distributed. Look for the line that reads STATMSG: ID=2301. You also can view Content Status in the Configuration Manager Console by selecting **the Zero Touch WinPE x86** boot image.

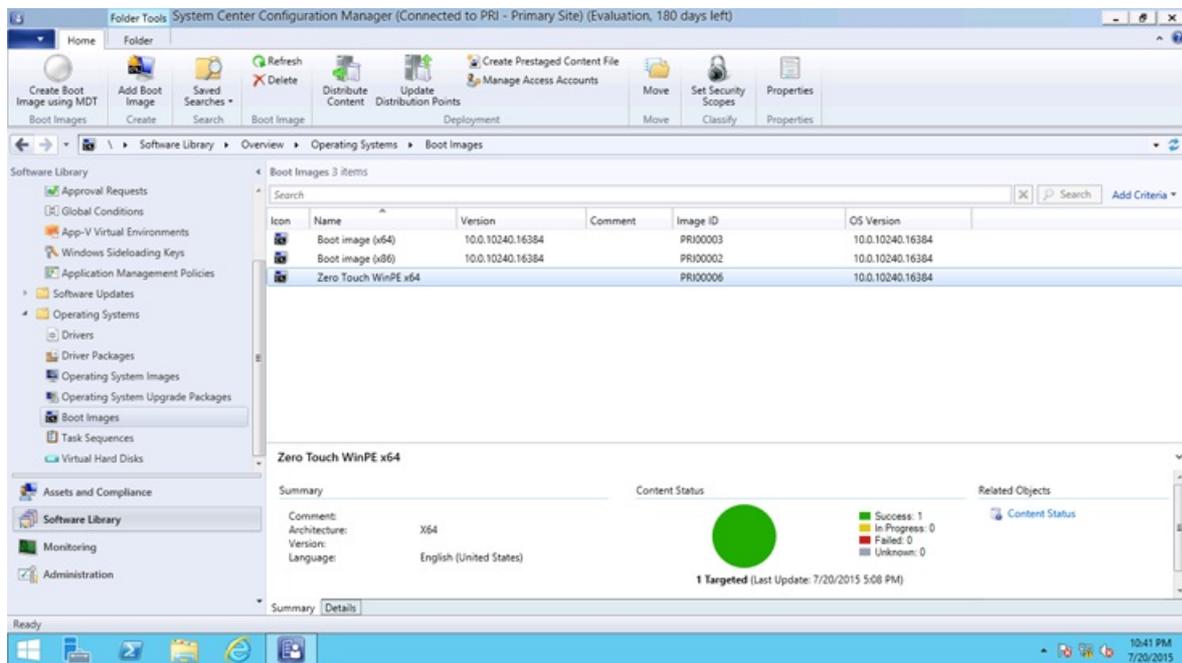


Figure 16. Content status for the Zero Touch WinPE x64 boot image

10. Using the Configuration Manager Console, right-click the **Zero Touch WinPE x64** boot image and select **Properties**.
11. In the **Data Source** tab, select the **Deploy this boot image from the PXE-enabled distribution point** check box, and click **OK**.
12. Using Configuration Manager Trace, review the E:\Program Files\Microsoft Configuration Manager\Logs\distmgr.log file and look for this text: Expanding PS10000B to E:\RemoteInstall\SMSImages.
13. Review the **E:\RemoteInstall\SMSImages** folder. You should see three folders containing boot images. Two are from the default boot images, and the third folder (PS10000B) is from your new boot image with DaRT.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager

Add a Windows 10 operating system image using Configuration Manager

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

Operating system images are typically the production image used for deployment throughout the organization. This topic shows you how to add a Windows 10 operating system image created with Microsoft System Center 2012 R2 Configuration Manager, and how to distribute the image to a distribution point.

For the purposes of this topic, we will use CM01, a machine running Windows Server 2012 R2 Standard, as the distribution point. CM01 is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#). Our image is named REFW10-X64-001.wim. For details on building this image, please see [Create a Windows 10 reference image](#).

1. Using File Explorer, in the **E:\Sources\OSD\OS** folder, create a subfolder named **Windows 10 Enterprise x64 RTM**.
2. Copy the REFW10-X64-001.wim file to the **E:\Sources\OSD\OS\Windows 10 Enterprise x64 RTM** folder.

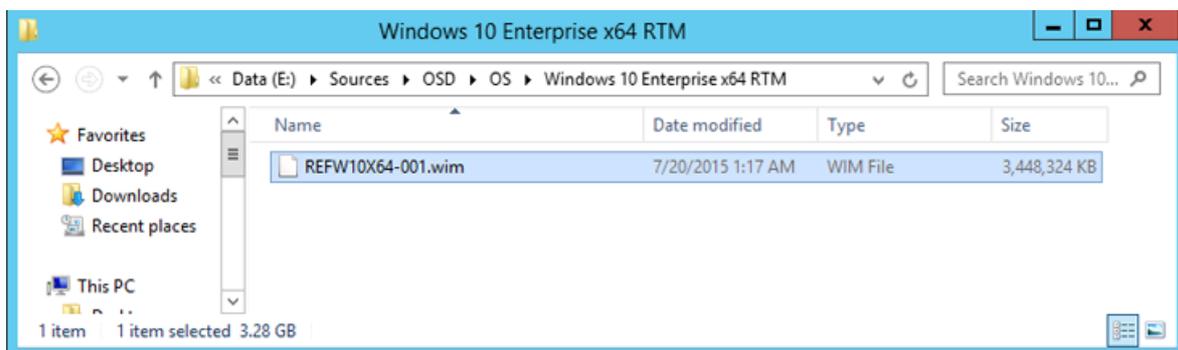


Figure 17. The Windows 10 image copied to the Sources folder structure.

3. Using the Configuration Manager Console, in the Software Library workspace, right-click **Operating System Images**, and select **Add Operating System Image**.
4. On the **Data Source** page, in the **Path:** text box, browse to \\CM01\Sources\$\OSD\OS\Windows 10 Enterprise x64 RTM\REFW10-X64-001.wim and click **Next**.
5. On the **General** page, assign the name Windows 10 Enterprise x64 RTM and click **Next** twice, and then click **Close**.

6. Distribute the operating system image to the CM01 distribution point by right-clicking the Windows 10 Enterprise x64 RTM operating system image and selecting **Distribute Content**.
7. In the Distribute Content Wizard, add the CM01 distribution point.
8. View the content status for the Windows 10 Enterprise x64 RTM package. Do not continue until the distribution is completed. You also can review the E:\Program Files\Microsoft Configuration Manager\Logs\distmgr.log file and look for the **STATMSG: ID=2301** line.

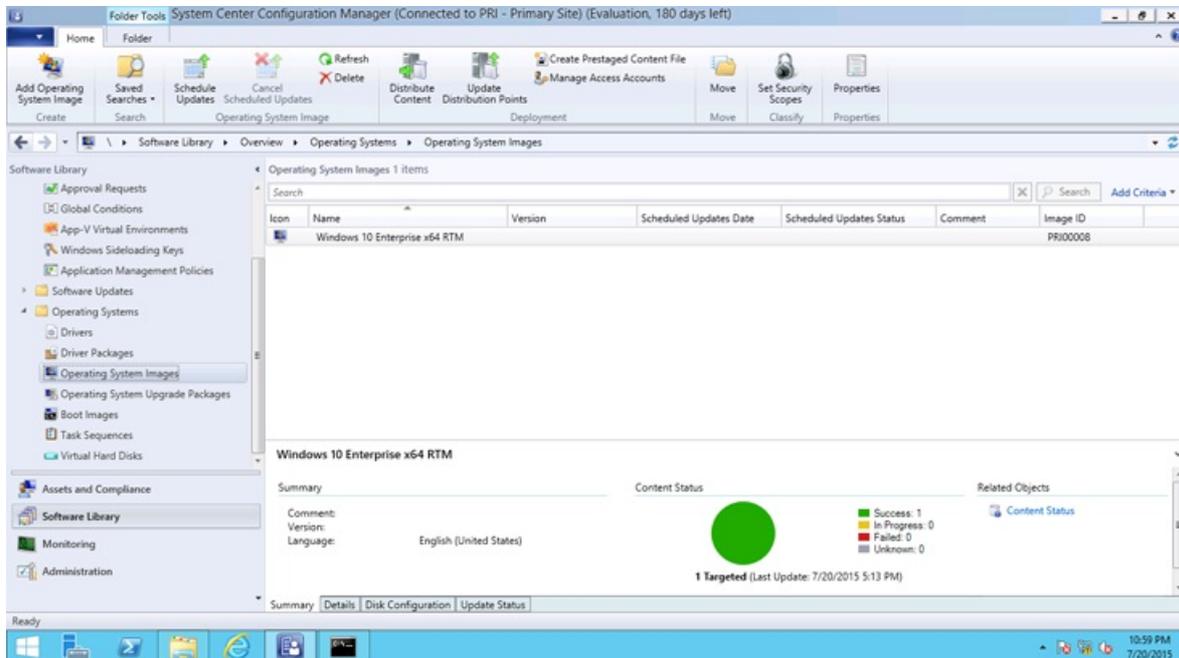


Figure 18. The distributed Windows 10 Enterprise x64 RTM package.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Create an application to deploy with Windows 10 using Configuration Manager

6/10/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

Microsoft System Center 2012 R2 Configuration Manager supports deploying applications as part of the Windows 10 deployment process. In this section, you create an application in System Center 2012 R2 Configuration Manager that you later configure the task sequence to use.

For the purposes of this topic, we will use CM01, a machine running Windows Server 2012 R2 Standard that is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

NOTE

Even though the new application model is fully supported to deploy via the task sequence, the most reliable way to deploy software via the task sequence is still the legacy packages, especially if you deploy many applications.

Example: Create the Adobe Reader XI application

The following steps show you how to create the Adobe Reader XI application. This section assumes that you have downloaded the MSI version of Adobe Reader XI to the C:\Setup\Adobe Reader XI folder on CM01.

1. On CM01, using File Explorer, copy the **C:\Setup\Adobe Reader XI** folder to the **E:\Sources\Software\Adobe** folder.
2. Using the Configuration Manager Console, in the Software Library workspace, expand **Application Management**.
3. Right-click **Applications** and select **Folder / Create Folder**. Assign the name **OSD**.
4. Right-click the **OSD** folder, and select **Create Application**.
5. In the Create Application Wizard, on the **General** page, use the following settings:
 - Automatically detect information about this application from installation files
 - Type: Windows Installer (*.msi file)
 - Location: \\CM01\Sources\$\Software\Adobe\Adobe Reader XI

- \\AdbeRdr11000_en_US.msi

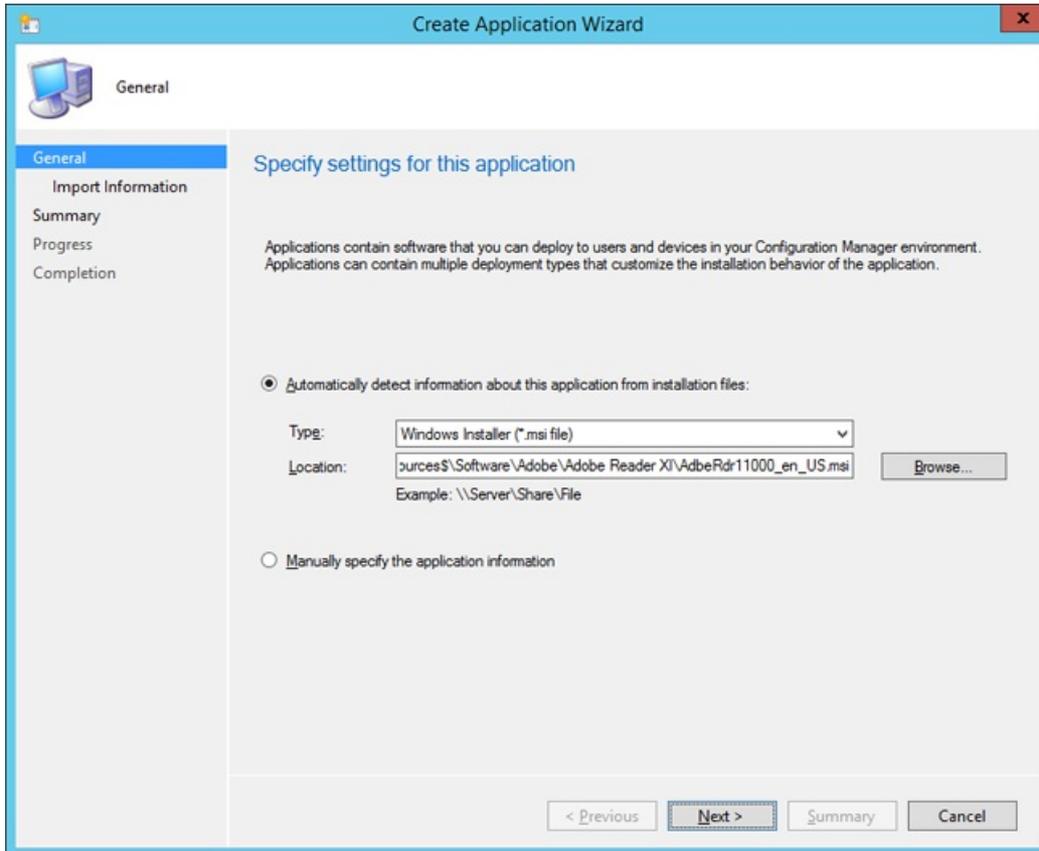


Figure 19. The Create Application Wizard

6. Click **Next**, and wait while Configuration Manager parses the MSI file.
7. On the **Import Information** page, review the information and then click **Next**.
8. On the **General Information** page, name the application Adobe Reader XI - OSD Install, click **Next** twice, and then click **Close**.

NOTE

Because it is not possible to reference an application deployment type in the task sequence, you should have a single deployment type for applications deployed by the task sequence. If you are deploying applications via both the task sequence and normal application deployment, and you have multiple deployment types, you should have two applications of the same software. In this section, you add the "OSD Install" suffix to applications that are deployed via the task sequence. If using packages, you can still reference both package and program in the task sequence.

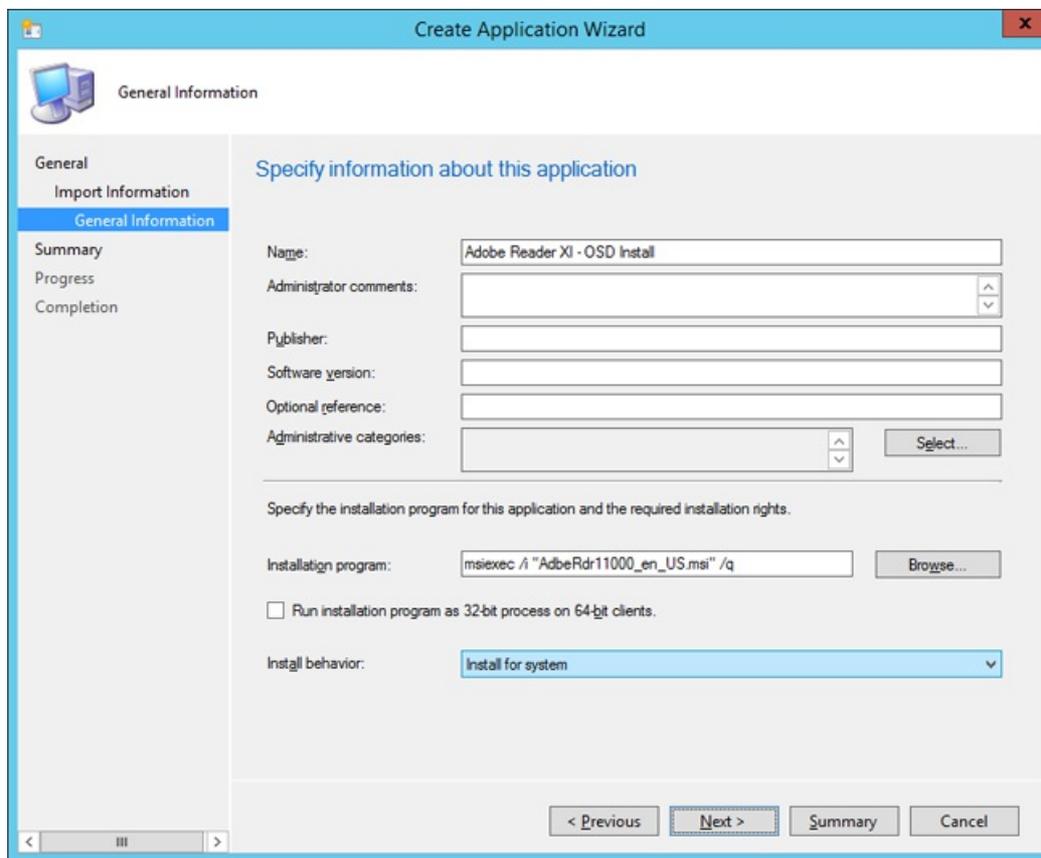


Figure 20. Add the "OSD Install" suffix to the application name

9. In the **Applications** node, select the Adobe Reader XI - OSD Install application, and click **Properties** on the ribbon bar.
10. In the **General Information** tab, select the **Allow this application to be installed from the Install Application task sequence action without being deployed** check box, and click **OK**.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager

6/10/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

In this topic, you will learn how to configure the Windows Preinstallation Environment (Windows PE) to include the network drivers required to connect to the deployment share and the storage drivers required to see the local storage on machines. Even though the Windows PE boot image and the Windows 10 operating system contain many out-of-the-box drivers, it is likely you will have to add new or updated drivers to support all your hardware. In this section, you import drivers for both Windows PE and the full Windows 10 operating system.

For the purposes of this topic, we will use CM01, a machine running Windows Server 2012 R2 Standard that is a member of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Add drivers for Windows PE

This section will show you how to import some network and storage drivers for Windows PE. This section assumes you have downloaded some drivers to the E:\Sources\OSD\DriverSources\WinPE x64 folder on CM01.

1. On CM01, using the Configuration Manager Console, in the Software Library workspace, right-click the **Drivers** node and select **Import Driver**.
2. In the Import New Driver Wizard, on the **Specify a location to import driver** page, below the Import all drivers in the following network path (UNC) option, browse to the **\\CM01\Sources\$\OSD\DriverSources\WinPE x64** folder and click **Next**.
3. On the **Specify the details for the imported driver** page, click **Categories**, create a category named **WinPE x64**, and then click **Next**.
4. On the **Select the packages to add the imported driver** page, click **Next**.
5. On the **Select drivers to include in the boot image** page, select the **Zero Touch WinPE x64** boot image. Also select the **Update distribution points when finished** check box, and click **Next** twice.

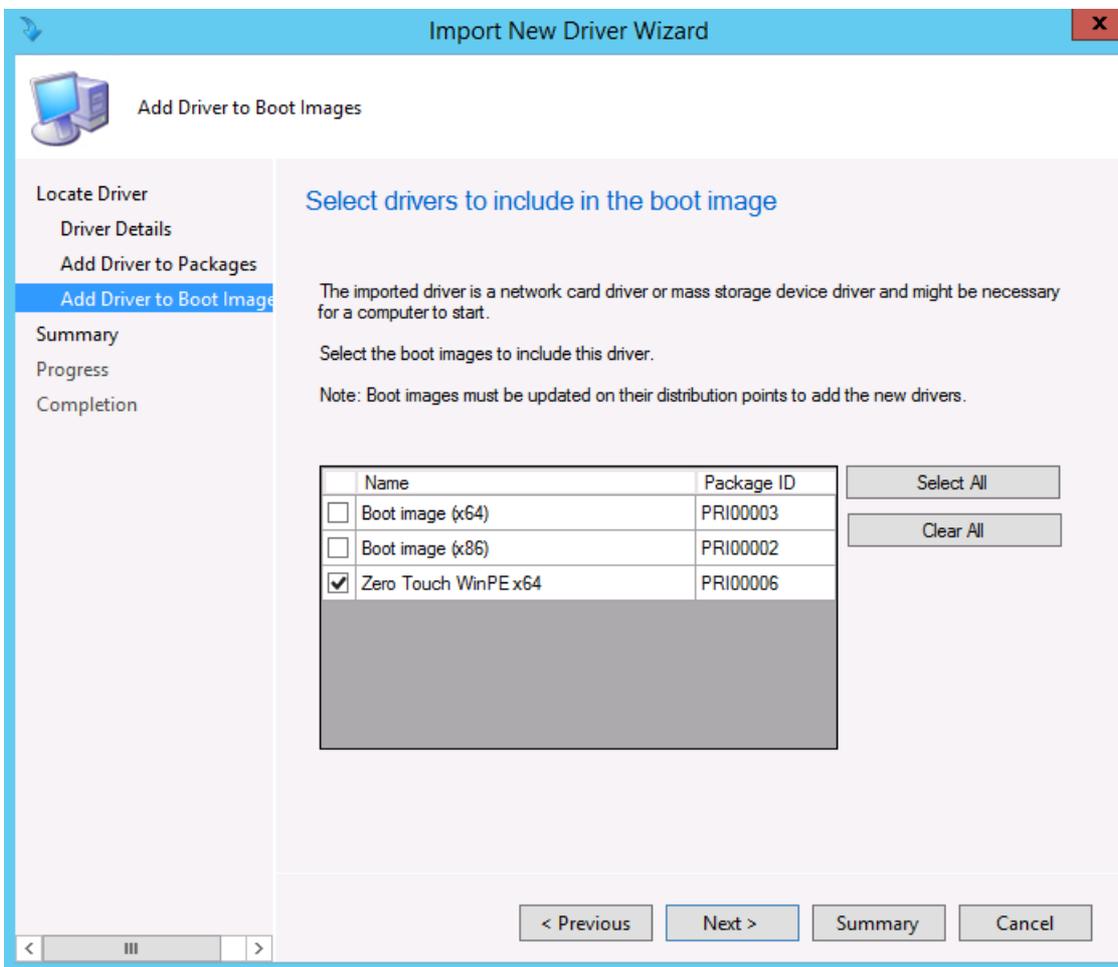


Figure 21. Add drivers to Windows PE

NOTE

The Updating Boot Image part of the wizard will appear to hang when displaying Done. It will complete in a minute or two.

Add drivers for Windows 10

This section illustrates how to add drivers for Windows 10 through an example in which you want to import Windows 10 drivers for the HP EliteBook 8560w model. For the purposes of this section, we assume that you have downloaded the Windows 10 drivers for the HP EliteBook 8560w model and copied them to the E:\Sources\OSD\DriverSources\Windows 10 x64\HP EliteBook 8560w folder on CM01.

1. On CM01, using the Configuration Manager Console, right-click the **Drivers** folder and select **Import Driver**.
2. In the Import New Driver Wizard, on the **Specify a location to import driver** page, below the Import all drivers in the following network path (UNC) option, browse to the **\\CM01\Sources\OSD\DriverSources\Windows 10 x64\HP EliteBook 8560w** folder and click **Next**.
3. On the **Specify the details for the imported driver** page, click **Categories**, create a category named Windows 10 x64 - HP EliteBook 8560w, and then click **Next**.

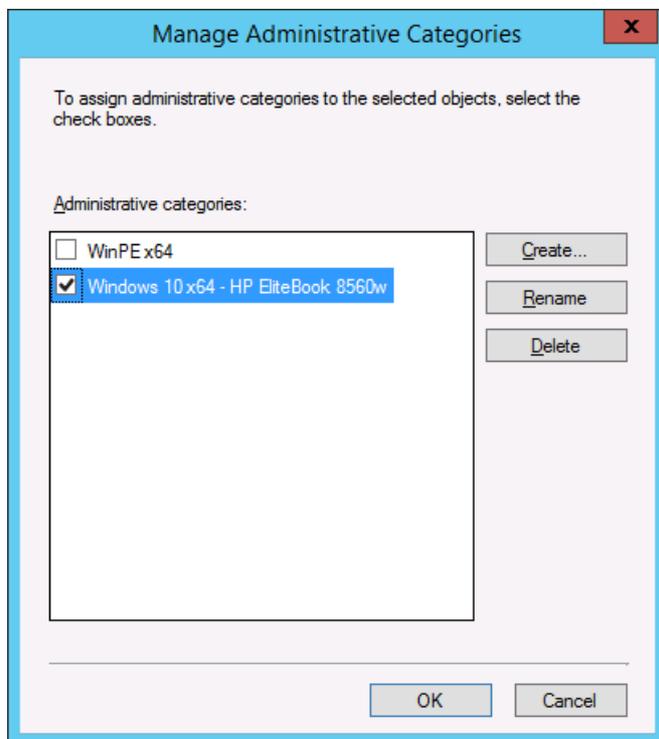


Figure 22. Create driver categories

4. On the **Select the packages to add the imported driver** page, click **New Package**, use the following settings for the package, and then click **Next**:
 - Name: Windows 10 x64 - HP EliteBook 8560w
 - Path: \\CM01\Sources\$\OSD\DriverPackages\Windows 10 x64\HP EliteBook 8560w

NOTE

The package path does not yet exist, so you have to type it in. The wizard will create the new package in that folder.

5. On the **Select drivers to include in the boot image** page, do not select anything, and click **Next** twice. After the package has been created, click **Close**.

NOTE

If you want to monitor the driver import process more closely, you can open the SMSProv.log file during driver import.

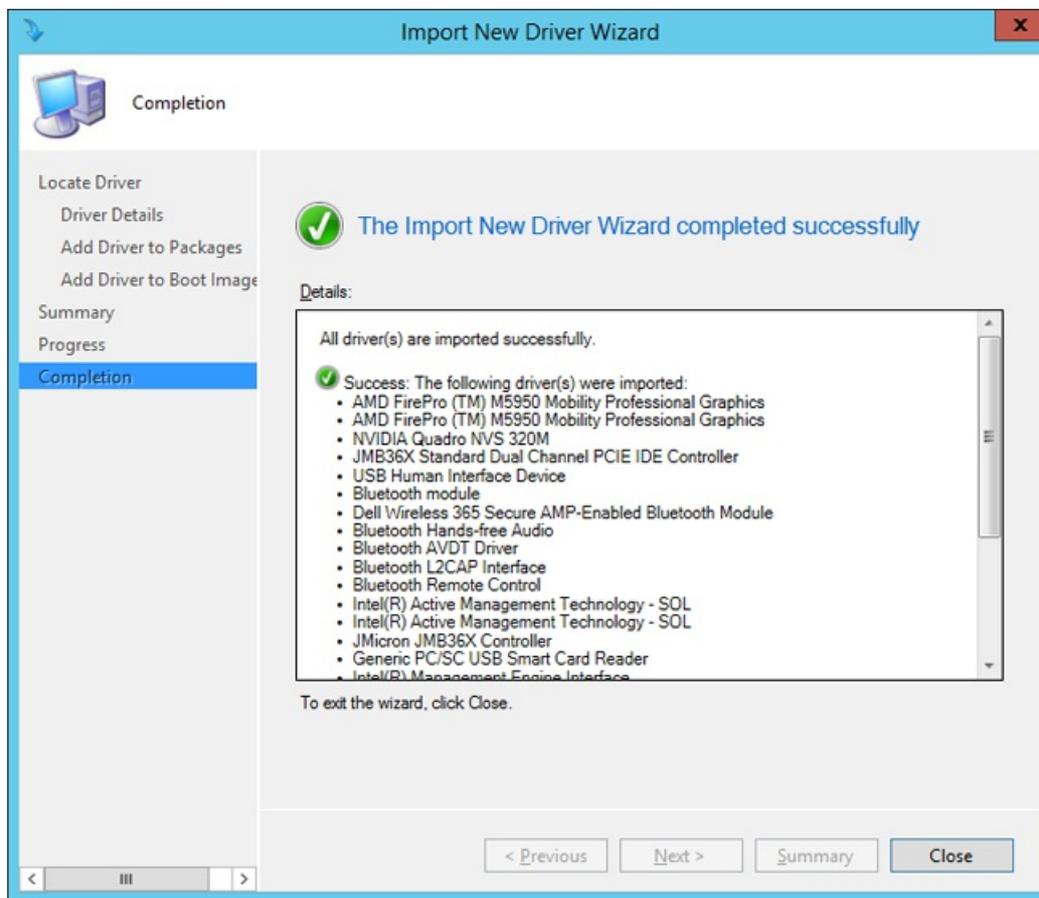


Figure 23. Drivers imported and a new driver package created

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Create a task sequence with Configuration Manager and MDT

6/14/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

In this topic, you will learn how to create a Microsoft System Center 2012 R2 Configuration Manager task sequence with Microsoft Deployment Toolkit (MDT) integration using the MDT wizard. Creating task sequences in System Center 2012 R2 Configuration Manager requires many more steps than creating task sequences for MDT Lite Touch installation. Luckily, the MDT wizard helps you through the process and also guides you through creating the needed packages.

For the purposes of this topic, we will use two machines: DC01 and CM01. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard, both of which are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Create a task sequence using the MDT Integration Wizard

This section walks you through the process of creating a System Center 2012 R2 Configuration Manager task sequence for production use.

1. On CM01, using the Configuration Manager Console, in the Software Library workspace, expand **Operating Systems**, right-click **Task Sequences**, and select **Create MDT Task Sequence**.
2. On the **Choose Template** page, select the **Client Task Sequence** template and click **Next**.
3. On the **General** page, assign the following settings and then click **Next**:
 - Task sequence name: Windows 10 Enterprise x64 RTM
 - Task sequence comments: Production image with Office 2013
4. On the **Details** page, assign the following settings and then click **Next**:
 - Join a Domain
 - Domain: contoso.com
 - Account: CONTOSO\CM_JD
 - Password: Passw0rd!
 - Windows Settings
 - User name: Contoso
 - Organization name: Contoso
 - Product key: <blank>
5. On the **Capture Settings** page, accept the default settings, and click **Next**.
6. On the **Boot Image** page, browse and select the **Zero Touch WinPE x64** boot image package. Then

click **Next**.

7. On the **MDT Package** page, select **Create a new Microsoft Deployment Toolkit Files package**, and in the **Package source folder to be created (UNC Path)**: text box, type `\\CM01\Sources$\OSD\MDT\MDT`. Then click **Next**.
8. On the **MDT Details** page, assign the name **MDT** and click **Next**.
9. On the **OS Image** page, browse and select the **Windows 10 Enterprise x64 RTM** package. Then click **Next**.
10. On the **Deployment Method** page, accept the default settings and click **Next**.
11. On the **Client Package** page, browse and select the **OSD / Configuration Manager Client** package. Then click **Next**.
12. On the **USMT Package** page, browse and select **the OSD / Microsoft Corporation User State Migration Tool for Windows 8 10.0.10240.16384** package. Then click **Next**.
13. On the **Settings Package** page, select the **Create a new settings package** option, and in the **Package source folder to be created (UNC Path)**: text box, type `\\CM01\Sources$\OSD\Settings\Windows 10 x64 Settings`. Then click **Next**.
14. On the **Settings Details** page, assign the name **Windows 10 x64 Settings** and click **Next**.
15. On the **Sysprep Package** page, click **Next** twice.
16. On the **Confirmation** page, click **Finish**.

Edit the task sequence

After you create the task sequence, we recommend that you configure the task sequence for an optimal deployment experience. The configurations include enabling support for Unified Extensible Firmware Interface (UEFI), dynamic organizational unit (OU) allocation, computer replace scenarios, and more.

1. On CM01, using the Configuration Manager Console, select **Task Sequences**, right-click **Windows 10 Enterprise x64 RTM** task sequence, and select **Edit**.
2. In the **Install** group, select the **Set Variable for Drive Letter** action and configure the following:
 - OSDPreserveDriveLetter: True

NOTE

If you don't change this value, your Windows installation will end up in E:\Windows.

3. In the **Post Install** group, select **Apply Network Settings**, and configure the Domain OU value to use the **Contoso / Workstations** OU (browse for values).
4. In the **Post Install** group, disable the **Auto Apply Drivers** action. (Disabling is done by selecting the action and, in the **Options** tab, selecting the **Disable this step** check box.)
5. After the disabled **Post Install / Auto Apply Drivers** action, add a new group name: **Drivers**.
6. After the **Post Install / Drivers** group, add an **Apply Driver Package** action with the following settings:
 - Name: HP EliteBook 8560w
 - Driver Package: Windows 10 x64 - HP EliteBook 8560w
 - Options: Task Sequence Variable: Model equals HP EliteBook 8560w

NOTE

You also can add a Query WMI condition with the following query: `SELECT * FROM Win32_ComputerSystem WHERE Model LIKE '%HP EliteBook 8560w%'`

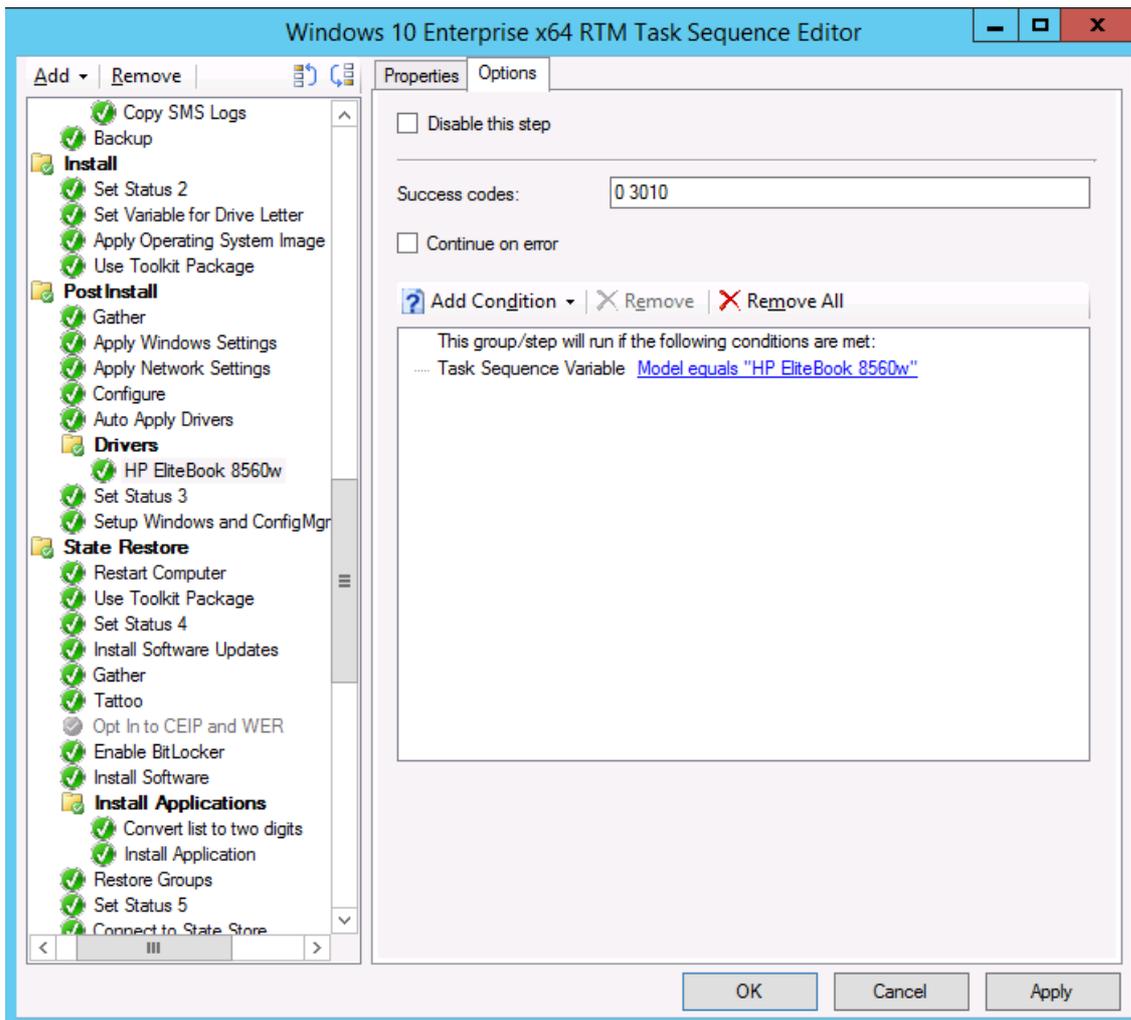


Figure 24. The driver package options

7. In the **State Restore / Install Applications** group, select the **Install Application** action.
8. Select the **Install the following applications** option, and add the OSD / Adobe Reader XI - OSD Install application to the list.

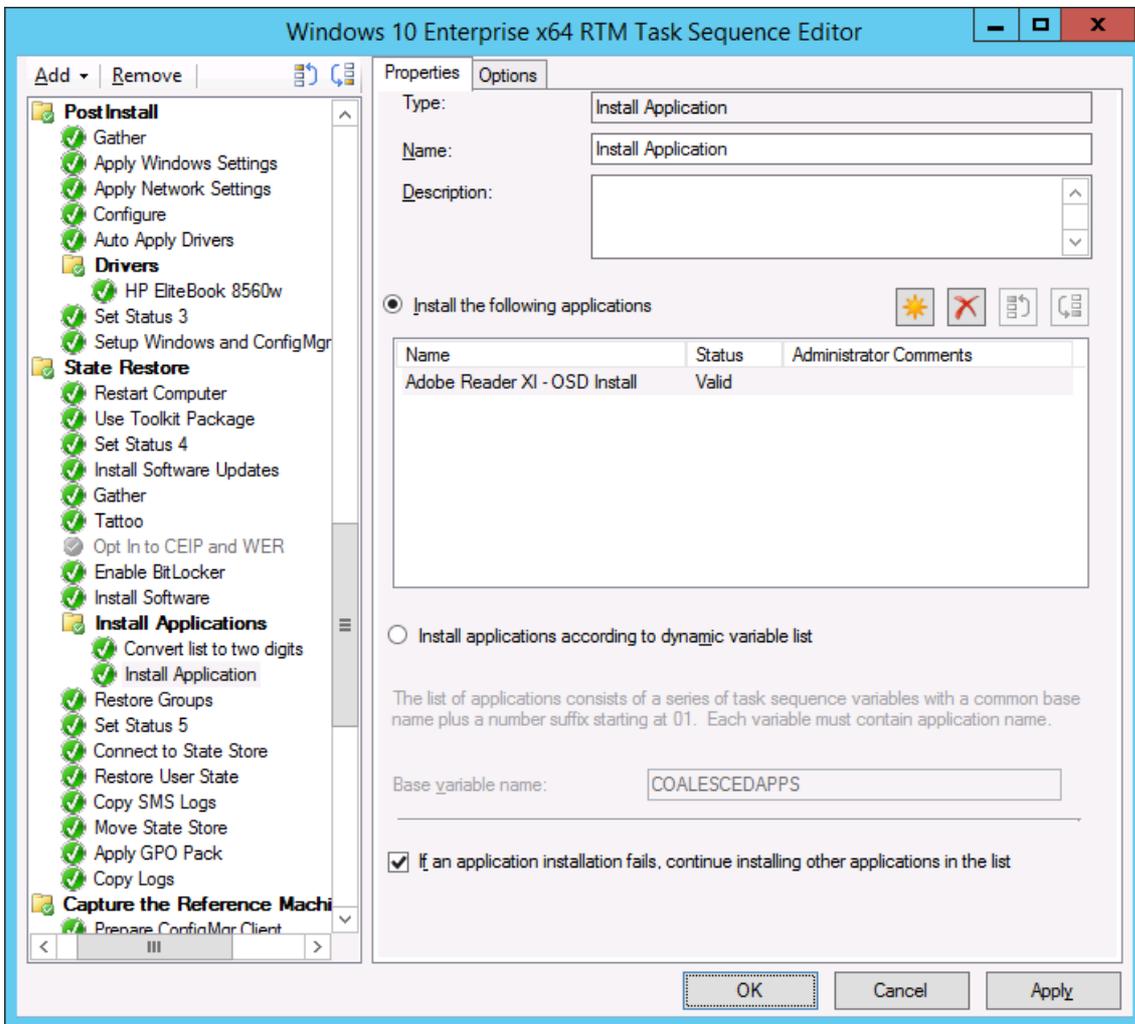


Figure 25. Add an application to the Configuration Manager task sequence

9. In the **State Restore** group, after the **Set Status 5** action, add a **Request State Store** action with the following settings:
 - Restore state from another computer
 - If computer account fails to connect to state store, use the Network Access account
 - Options: Continue on error
 - Options / Condition:
 - Task Sequence Variable
 - USMTLOCAL not equals True
10. In the **State Restore** group, after the **Restore User State** action, add a **Release State Store** action with the following settings:
 - Options: Continue on error
 - Options / Condition:
 - Task Sequence Variable
 - USMTLOCAL not equals True
11. Click **OK**.

NOTE

The Request State Store and Release State Store actions need to be added for common computer replace scenarios.

Move the packages

While creating the task sequence with the MDT wizard, a few operating system deployment packages were created. To move these packages to the OSD folder, take the following steps.

1. On CM01, using the Configuration Manager Console, in the Software Library workspace, expand **Application Management**, and then select **Packages**.
2. Select the **MDT** and **Windows 10 x64 Settings** packages, right-click and select **Move**.
3. In the **Move Selected Items** dialog box, select the **OSD** folder, and click **OK**.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Finalize the operating system configuration for Windows 10 deployment with Configuration Manager

6/10/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

This topic walks you through the steps to finalize the configuration of your Windows 10 operating deployment, which includes enablement of the optional Microsoft Deployment Toolkit (MDT) monitoring for Microsoft System Center 2012 R2 Configuration Manager, logs folder creation, rules configuration, content distribution, and deployment of the previously created task sequence.

For the purposes of this topic, we will use two machines: DC01 and CM01. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. Both are members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

Enable MDT monitoring

This section will walk you through the process of creating the E:\MDTProduction deployment share using the MDT Deployment Workbench to enable monitoring for Configuration Manager.

1. On CM01, using the Deployment Workbench, right-click **Deployment Shares** and select **New Deployment Share**. Use the following settings for the New Deployment Share Wizard:
 - Deployment share path: E:\MDTProduction
 - Share name: MDTProduction\$
 - Deployment share description: MDT Production
 - Options: <default settings>
2. Right-click the **MDT Production** deployment share, and select **Properties**. In the **Monitoring** tab, select the **Enable monitoring for this deployment share** check box, and click **OK**.

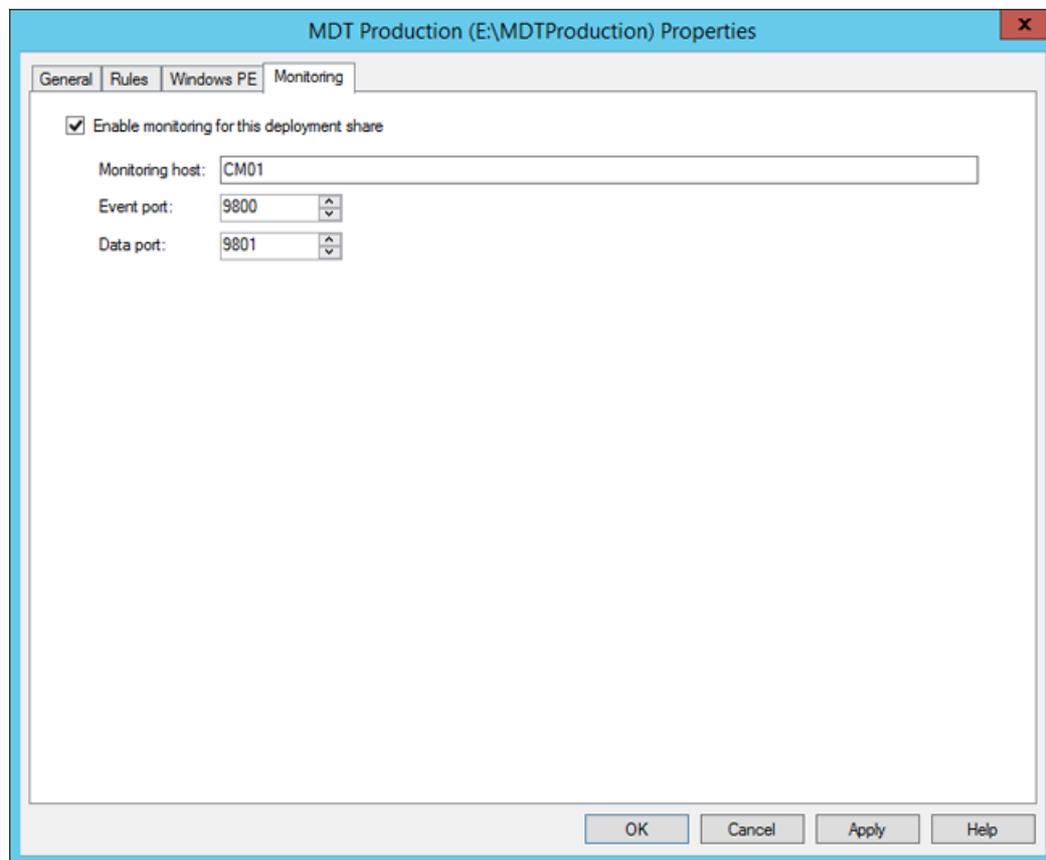


Figure 26. Enable MDT monitoring for Configuration Manager

Create and share the Logs folder

To support additional server-side logging in Configuration Manager, you create and share the E:\Logs folder on CM01 using Windows PowerShell. Then in the next step, you enable server-side logging by modifying the CustomSettings.ini file used by the Configuration Manager task sequence.

1. On CM01, start an elevated Windows PowerShell prompt (run as Administrator).
2. Type the following commands, pressing **Enter** after each one:

```
New-Item -Path E:\Logs -ItemType directory
New-SmbShare -Name Logs$ -Path E:\Logs -ChangeAccess EVERYONE
icacls E:\Logs /grant '"CM_NAA":(OI)(CI)(M)'
```

Configure the rules (Windows 10 x64 Settings package)

This section will show you how to configure the rules (the Windows 10 x64 Settings package) to support the Contoso environment.

1. On CM01, using File Explorer, navigate to the **E:\Sources\OSD\Settings\Windows 10 x64 Settings** folder.
2. Using Notepad, edit the CustomSetting.ini file with the following settings:

```

[Settings]
Priority=Default
Properties=OSDMigrateConfigFiles,OSDMigrateMode
[Default]
DoCapture=NO
ComputerBackupLocation=NONE
MachineObjectOU=ou=Workstations,ou=Computers,ou=Contoso,dc=contoso,dc=com
OSDMigrateMode=Advanced
OSDMigrateAdditionalCaptureOptions=/ue:*\* /ui:CONTOSO\*
OSDMigrateConfigFiles=Miguser.xml,Migapp.xml
SLSHARE=\\CM01\Logs$
EventService=http://CM01:9800
ApplyGPOPack=NO

```

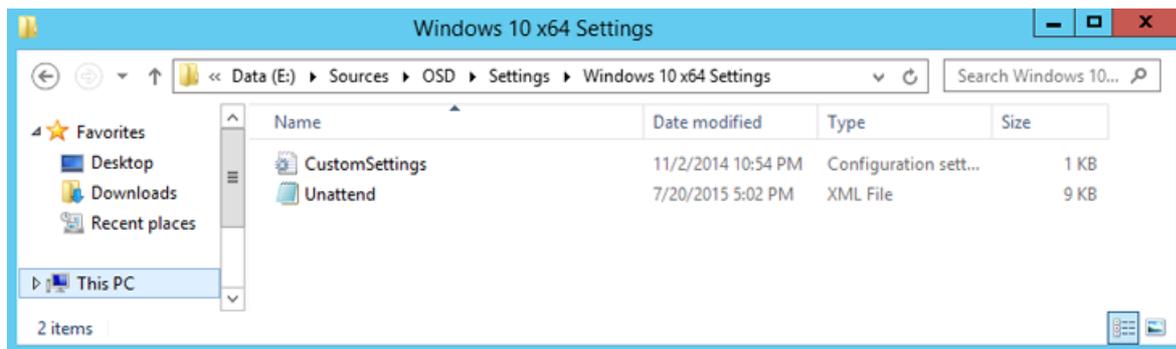


Figure 27. The Settings package, holding the rules and the Unattend.xml template used during deployment

- Update the distribution point for the **Windows 10 x64 Settings** package by right-clicking the **Windows 10 x64 Settings** package and selecting **Update Distribution Points**.

NOTE

Although you have not yet added a distribution point, you still need to select Update Distribution Points. That process also updates the Configuration Manager 2012 content library with changes.

Distribute content to the CM01 distribution portal

In Configuration Manager, you can distribute all packages needed by a task sequence in a single task. In this section, you distribute packages that have not yet been distributed to the CM01 distribution point.

- On CM01, using the Configuration Manager Console**, select **Task Sequences**, right-click the **Windows 10 Enterprise x64 RTM** task sequence, and select **Distribute Content**.
- In the Distribute Content Wizard, add the CM01 distribution point, and complete the wizard.
- Using Configuration Manager Trace, verify the distribution to the CM01 distribution point by reviewing the distmgr.log file, or use the Distribution Status / Content Status option in the Monitoring workspace. Do not continue until you see all the new packages being distributed successfully.

Create a deployment for the task sequence

This section provides steps to help you create a deployment for the task sequence.

- On CM01, using the Configuration Manager Console, select **Task Sequences**, right-click **Windows 10 Enterprise x64 RTM**, and then select **Deploy**.
- On the **General** page, select the **All Unknown Computers** collection and click **Next**.
- On the **Deployment Settings** page, use the following settings and then click **Next**:

- Purpose: Available
- Make available to the following: Only media and PXE

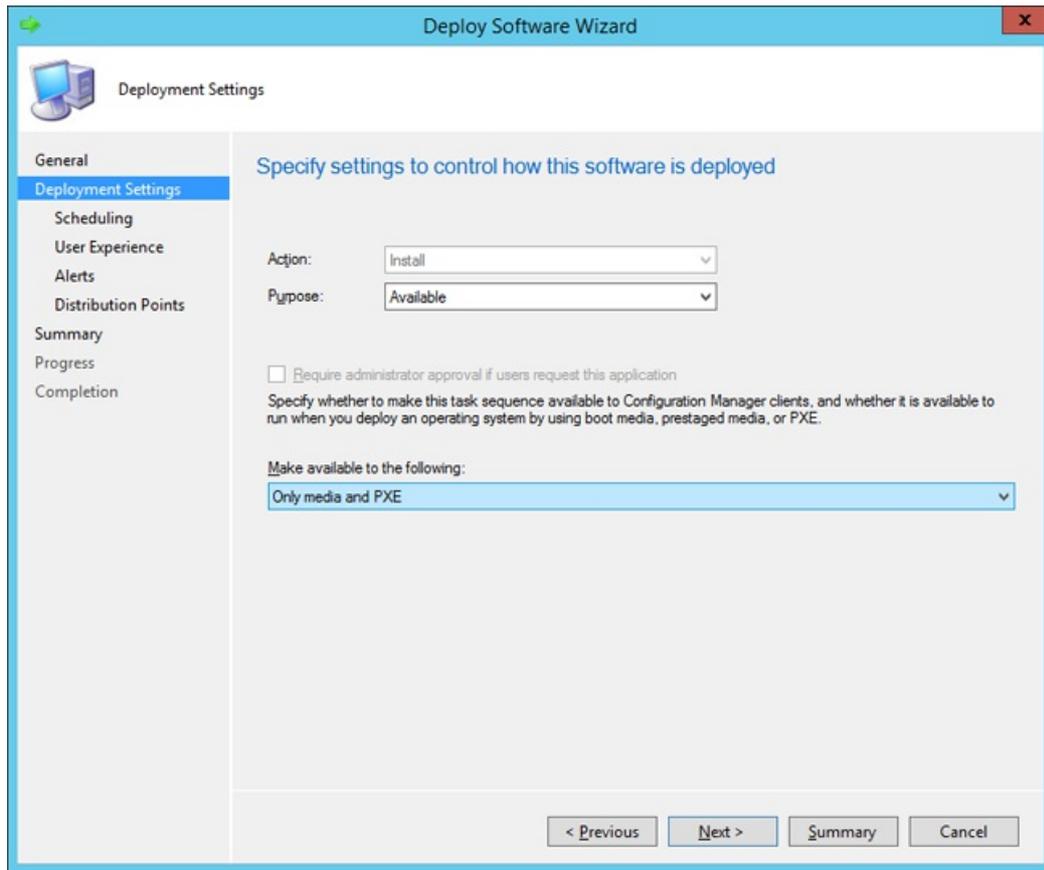


Figure 28. Configure the deployment settings

4. On the **Scheduling** page, accept the default settings and click **Next**.
5. On the **User Experience** page, accept the default settings and click **Next**.
6. On the **Alerts** page, accept the default settings and click **Next**.
7. On the **Distribution Points** page, accept the default settings, click **Next** twice, and then click **Close**.

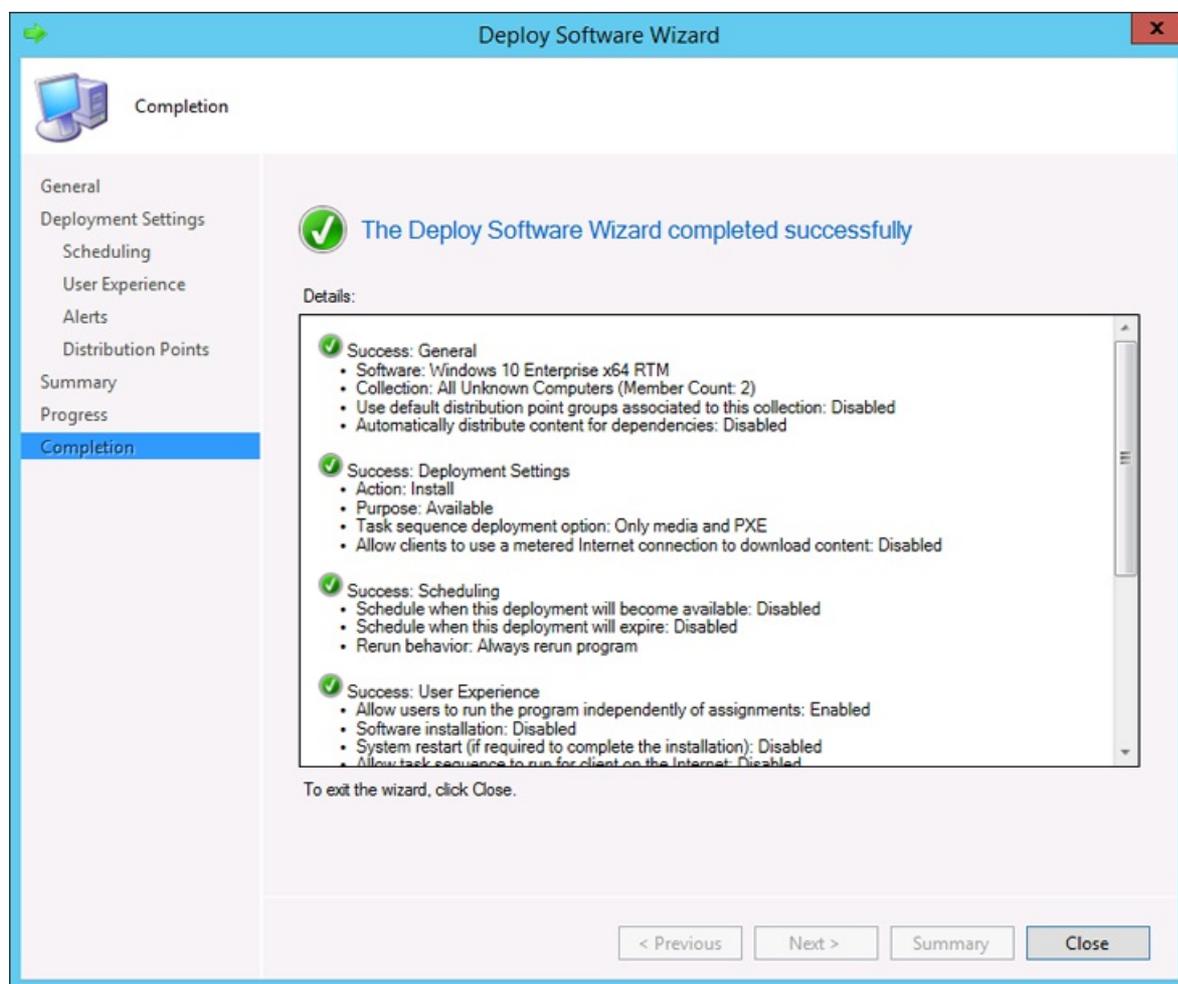


Figure 29. The Windows 10 Enterprise x64 RTM task sequence deployed to the All Unknown Computers collections available for media and PXE

Configure Configuration Manager to prompt for the computer name during deployment (optional)

You can have Configuration Manager prompt you for a computer name or you can use rules to generate a computer name. For more details on how to do this, see [Configure MDT settings](#).

This section provides steps to help you configure the All Unknown Computers collection to have Configuration Manager prompt for computer names.

1. Using the Configuration Manager Console, in the Asset and Compliance workspace, select **Device Collections**, right-click **All Unknown Computers**, and select **Properties**.
2. In the **Collection Variables** tab, create a new variable with the following settings:
 - Name: OSDComputerName
 - Clear the **Do not display this value in the Configuration Manager console** check box.
3. Click **OK**.

NOTE

Configuration Manager can prompt for information in many ways. Using a collection variable with an empty value is just one of them. Another option is the User-Driven Installation (UDI) wizard.

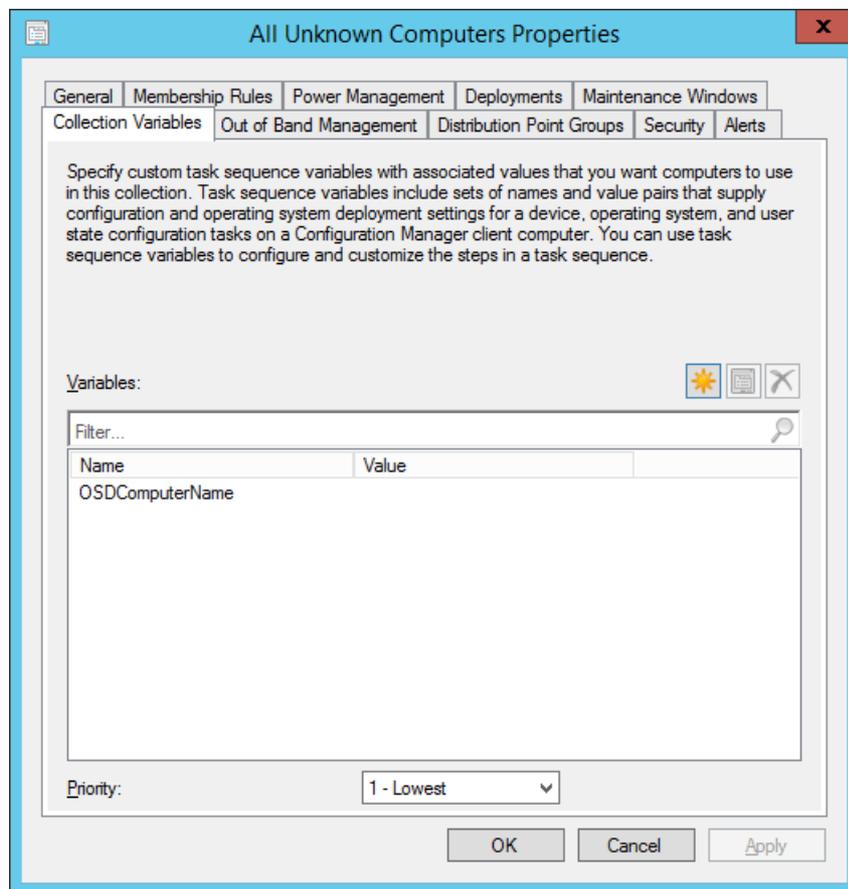


Figure 30. Configure a collection variable

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Deploy Windows 10 using PXE and Configuration Manager

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

In this topic, you will learn how to deploy Windows 10 using Microsoft System Center 2012 R2 Configuration Manager deployment packages and task sequences. This topic will walk you through the process of deploying the Windows 10 Enterprise image to a Unified Extensible Firmware Interface (UEFI) machine named PC0001.

For the purposes of this topic, we will use two additional machines: DC01 and CM01. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. DC01, CM01, and PC0001 are all members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

1. Start the PC0001 machine. At the Pre-Boot Execution Environment (PXE) boot menu, press **Enter** to allow it to PXE boot.

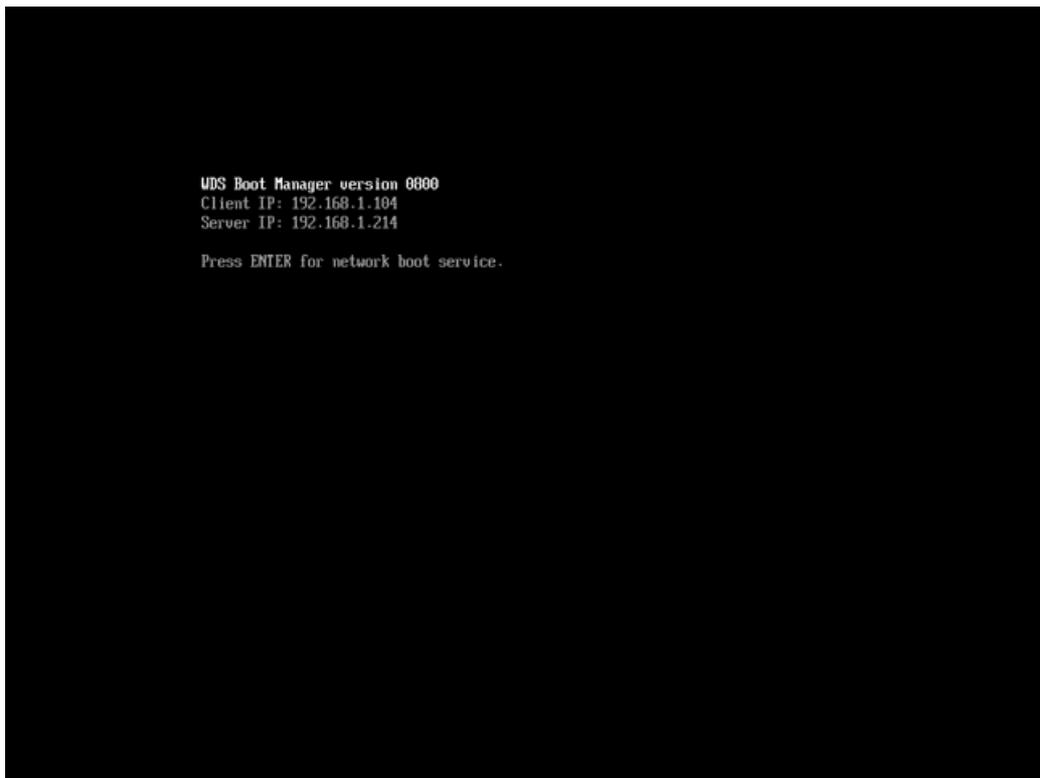


Figure 31. PXE booting PC0001.

2. On the **Welcome to the Task Sequence Wizard** page, type in the password **Passw0rd!** and click **Next**.
3. On the **Select a task sequence to run** page, select **Windows 10 Enterprise x64 RTM** and click **Next**.
4. On the **Edit Task Sequence Variables** page, double-click the **OSDComputerName** variable, and in the **Value** field, type **PC0001** and click **OK**. Then click **Next**.

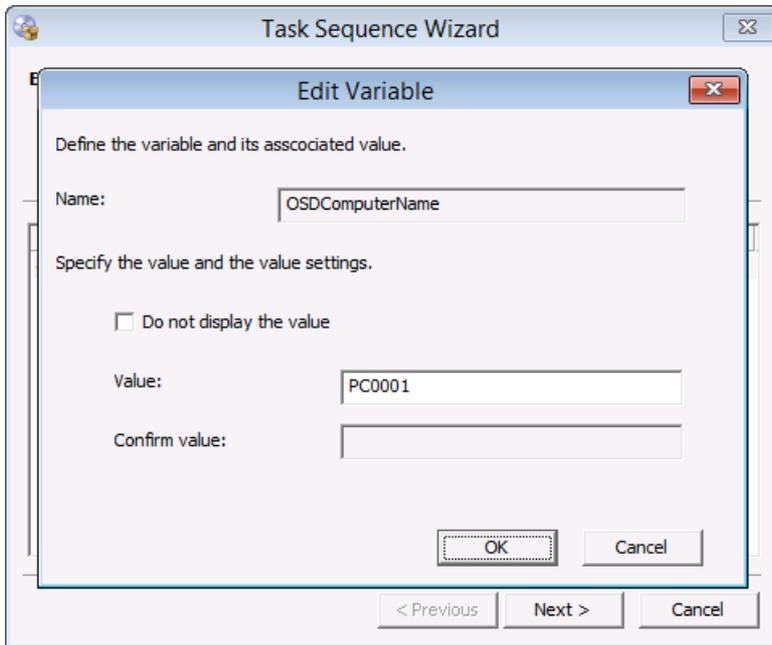


Figure 32. Typing in the computer name.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Monitor the Windows 10 deployment with Configuration Manager

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

In this topic, you will learn how to monitor a Windows 10 deployment that was started previously using Microsoft System Center 2012 R2 Configuration Manager and the Microsoft Deployment Toolkit (MDT) Deployment Workbench. You will also use the Deployment Workbench to access the computer remotely via the Microsoft Diagnostics and Recovery Toolkit (DaRT) Remote Connection feature.

For the purposes of this topic, we will use four machines: DC01, CM01, and PC0001. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. PC0001 is a Unified Extensible Firmware Interface (UEFI) machine to which Windows 10 Enterprise has been deployed. DC01, CM01, and PC0001 are all members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

To monitor an operating system deployment conducted through System Center 2012 R2 Configuration Manager, you will use the Deployment Workbench in MDT as follows:

1. On CM01, using the Deployment Workbench, expand **MDT Production**, and use the **Monitoring** node to view the deployment process (press **F5** to refresh).

NOTE

It takes a little while for the task sequence to start reporting monitor information, so if PC0001 does not appear when you press F5 the first time, wait 20 seconds and try again.

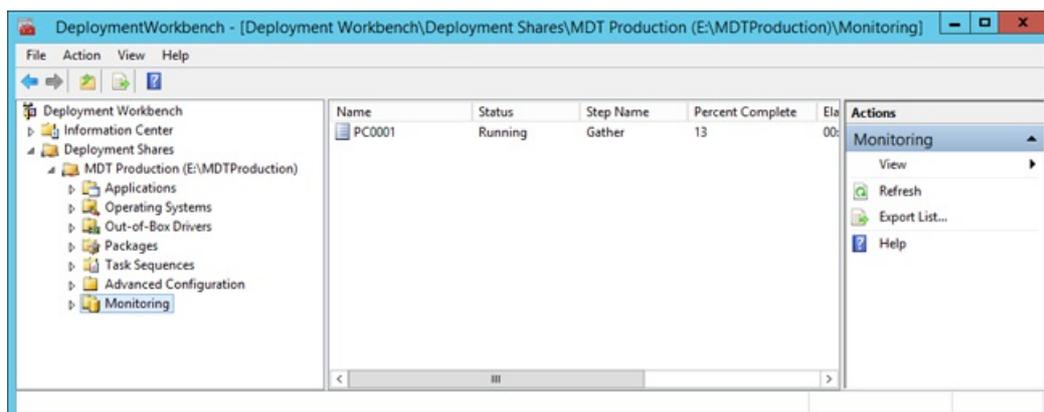


Figure 33. PC0001 being deployed by Configuration Manager

2. When you see the PC0001 entry, double-click **PC0001**, and then click **DaRT Remote Control** and review the **Remote Control** option.
3. The task sequence will now run and do the following:
 - Install the Windows 10 operating system.
 - Install the Configuration Manager client and the client hotfix.
 - Join the machine to the domain.
 - Install the application added to the task sequence.

NOTE

You also can use the built-in reports to get information about ongoing deployments. For example, a task sequence report gives you a quick overview of the task sequence progress.

4. If time permits, allow the deployment of PC0001 to complete. Then log in as Administrator in the CONTOSO domain and verify that Adobe Reader XI was installed.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager

6/10/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

This topic will show you how to use a previously created task sequence to refresh a Windows 7 SP1 client with Windows 10 using Microsoft System Center 2012 R2 Configuration Manager and Microsoft Deployment Toolkit (MDT) 2013 Update 2. When refreshing a machine to a later version, it appears as an upgrade to the end user, but technically it is not an in-place upgrade. A computer refresh also involves taking care of user data and settings from the old installation and making sure to restore those at the end of the installation. For more information, see [Refresh a Windows 7 computer with Windows 10](#).

A computer refresh with System Center 2012 R2 Configuration Manager works the same as it does with MDT Lite Touch installation. Configuration Manager also uses the User State Migration Tool (USMT) from the Windows Assessment and Deployment Kit (Windows ADK) 10 in the background. A computer refresh with Configuration Manager involves the following steps:

1. Data and settings are backed up locally in a backup folder.
2. The partition is wiped, except for the backup folder.
3. The new operating system image is applied.
4. Other applications are installed.
5. Data and settings are restored.

For the purposes of this topic, we will use three machines: DC01, CM01, and PC0003. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. PC0003 is a machine with Windows 7 SP1, on which Windows 10 will be deployed. DC01, CM01, and PC003 are all members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

In this topic, we assume that you have a Windows 7 SP1 client named PC0003 with the Configuration Manager client installed.

Create a device collection and add the PC0003 computer

1. On CM01, using the Configuration Manager console, in the Asset and Compliance workspace, right-click **Device Collections**, and then select **Create Device Collection**. Use the following settings:

- General
- Name: Install Windows 10 Enterprise x64
- Limited Collection: All Systems
- Membership rules:
- Direct rule
- Resource Class: System Resource
- Attribute Name: Name
- Value: PC0003
- Select **Resources**
- Select **PC0003**

2. Review the Install Windows 10 Enterprise x64 collection. Do not continue until you see the PC0003 machine in the collection.

NOTE

It may take a short while for the collection to refresh; you can view progress via the Collevel.log file. If you want to speed up the process, you can manually update membership on the Install Windows 10 Enterprise x64 collection by right-clicking the collection and selecting Update Membership.

Create a new deployment

Using the Configuration Manager console, in the Software Library workspace, select **Task Sequences**, right-click **Windows 10 Enterprise x64 RTM**, and then select **Deploy**. Use the following settings:

- General
 - Collection: Install Windows 10 Enterprise x64
- Deployment Settings
 - Purpose: Available
 - Make available to the following: Configuration Manager clients, media and PXE

NOTE

It is not necessary to make the deployment available to media and Pre-Boot Execution Environment (PXE) for a computer refresh, but you will use the same deployment for bare-metal deployments later on and you will need it at that point.

- Scheduling
 - <default>
- User Experience
 - <default>
- Alerts
 - <default>
- Distribution Points

- o <default>

Initiate a computer refresh

Now you can start the computer refresh on PC0003.

1. Using the Configuration Manager console, in the Asset and Compliance workspace, in the Install Windows 10 Enterprise x64 collection, right-click **PC0003** and select **Client Notification / Download Computer Policy**. Click **OK**.

NOTE

The Client Notification feature is new in Configuration Manager.

2. On PC0003, using the Software Center (begin using the Start screen, or click the **New software is available** balloon in the system tray), select the **Windows 10 Enterprise x64 RTM** deployment and click **INSTALL**.
3. In the **Software Center** warning dialog box, click **INSTALL OPERATING SYSTEM**.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Replace a Windows 7 SP1 client with Windows 10 using Configuration Manager

6/10/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10 versions 1507, 1511

IMPORTANT

For instructions to deploy the most recent version of Windows 10 with Configuration Manager, see [Scenarios to deploy enterprise operating systems with System Center Configuration Manager](#). Configuration Manager 2012 and 2012 R2 provide support for Windows 10 versions 1507 and 1511 only. Later versions of Windows 10 require an updated Configuration Manager release. For a list of Configuration Manager versions and the corresponding Windows 10 client versions that are supported, see [Support for Windows 10 for System Center Configuration Manager](#).

In this topic, you will learn how to replace a Windows 7 SP1 computer using Microsoft System Center 2012 R2 Configuration Manager. This process is similar to refreshing a computer, but since you are replacing the machine, you have to run the backup job separately from the deployment of Windows 10.

For the purposes of this topic, we will use three machines: DC01, CM01, and PC0004. DC01 is a domain controller and CM01 is a machine running Windows Server 2012 R2 Standard. PC0004 is a machine with Windows 7 SP1 that will be replaced with a new machine running Windows 10. DC01, CM01, and PC0004 are all members of the domain contoso.com for the fictitious Contoso Corporation. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).

In this topic, you will create a backup-only task sequence that you run on PC0004, the machine you are replacing. For more information, see [Replace a Windows 7 computer with a Windows 10 computer](#).

Create a replace task sequence

1. On CM01, using the Configuration Manager Console, in the Software Library workspace, expand **Operating Systems**, right-click **Task Sequences**, and select **Create MDT Task Sequence**.
2. On the **Choose Template** page, select the **Client Replace Task Sequence** template and click **Next**.
3. On the **General** page, assign the following settings and click **Next**:
 - Task sequence name: Replace Task Sequence
 - Task sequence comments: USMT backup only
4. On the **Boot Image** page, browse and select the **Zero Touch WinPE x64** boot image package. Then click **Next**.
5. On the **MDT Package** page, browse and select the **OSD / MDT** package. Then click **Next**.
6. On the **USMT Package** page, browse and select the **OSD / Microsoft Corporation User State Migration Tool for Windows 8 10.0.10240.16384** package. Then click **Next**.
7. On the **Settings Package** page, browse and select the **OSD / Windows 10 x64 Settings** package. Then click **Next**.

8. On the **Summary** page, review the details and then click **Next**.
9. On the **Confirmation** page, click **Finish**.
10. Review the Replace Task Sequence.

NOTE

This task sequence has many fewer actions than the normal client task sequence. If it doesn't seem different, make sure you selected the Client Replace Task Sequence template when creating the task sequence.

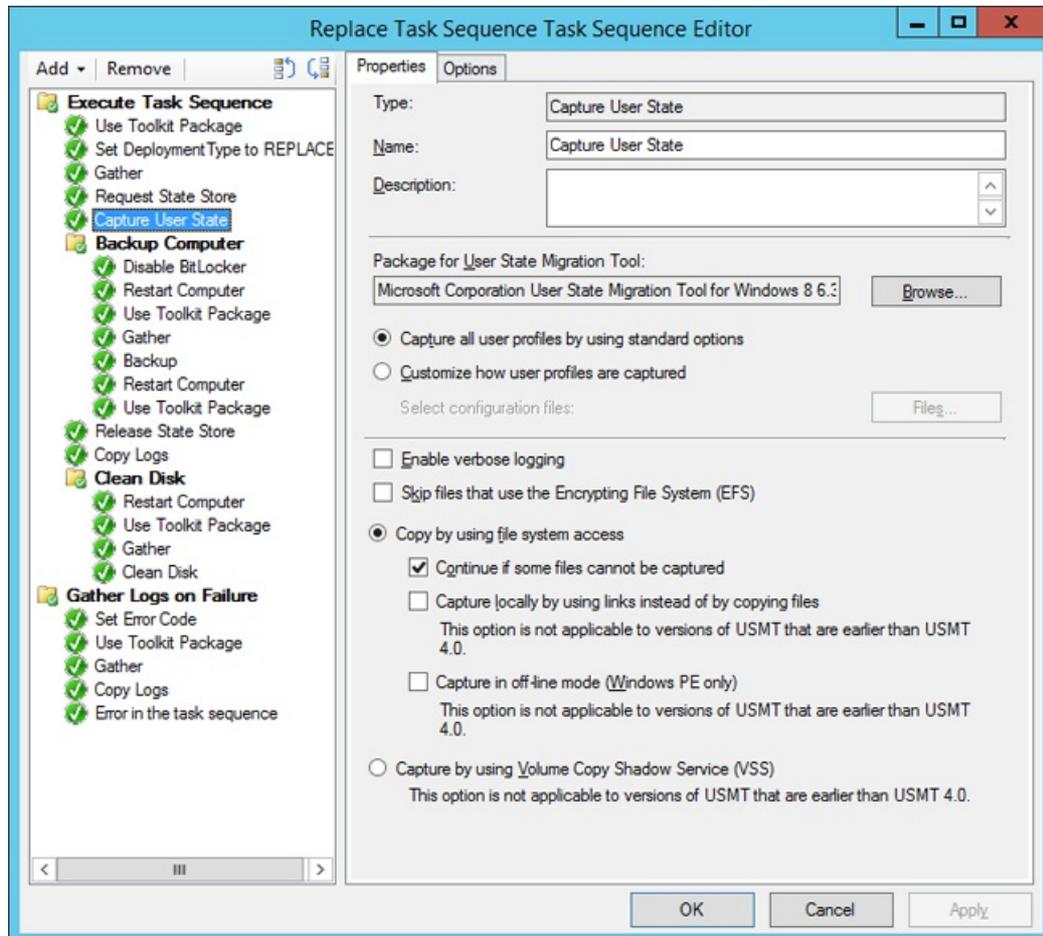


Figure 34. The backup-only task sequence (named Replace Task Sequence).

Associate the new machine with the old computer

This section walks you through the process of associating a blank machine, PC0006, with an old machine, PC0004, for the purpose of replacing PC0004 with PC0006. PC0006 can be either a physical or virtual machine.

1. Make a note of the PC0006 machine's MAC Address. (If PC0006 is a virtual machine, you can see the MAC Address in the virtual machine settings.) In our example, the PC0006 MAC Address is 00:15:5D:0A:6A:96.
2. Using the Configuration Manager console, in the Asset and Compliance workspace, right-click **Devices**, and then select **Import Computer Information**.
3. On the **Select Source** page, select **Import single computer** and click **Next**.
4. On the **Single Computer** page, use the following settings and then click **Next**:
 - Computer Name: PC0006

- MAC Address: <the mac address from step 1 >
- Source Computer: PC0004

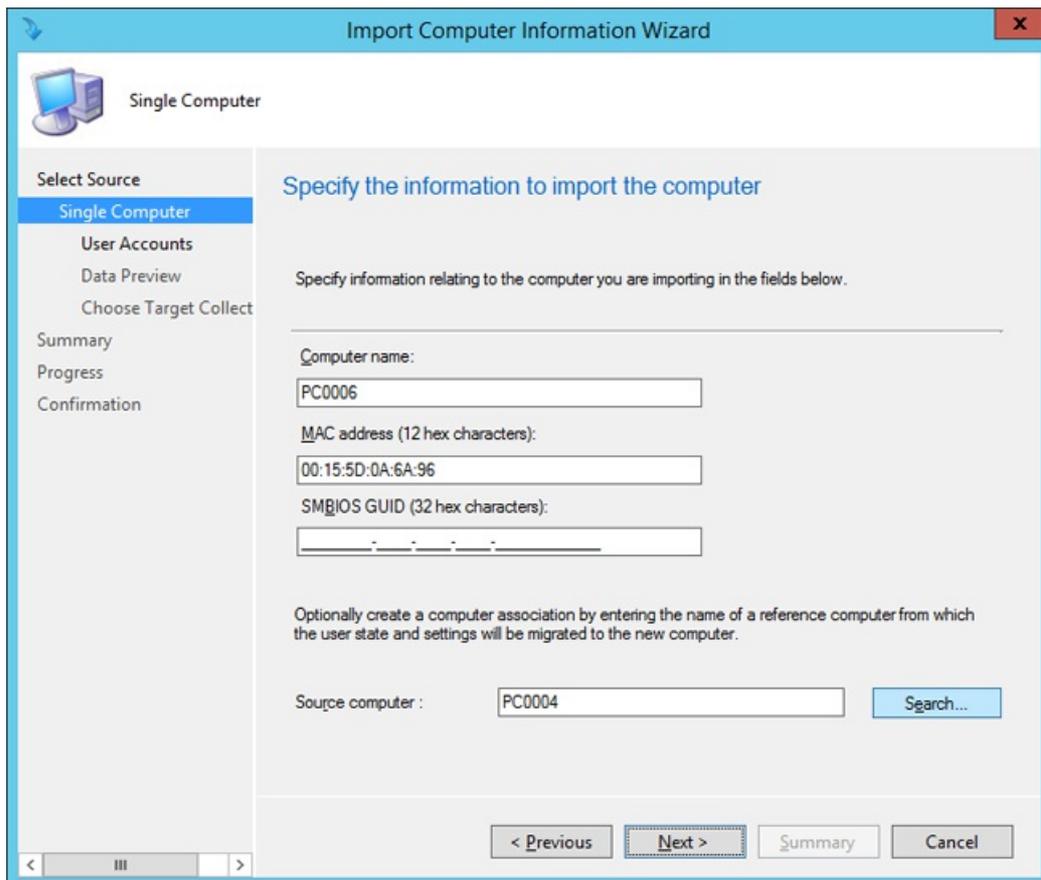


Figure 35. Creating the computer association between PC0004 and PC0006.

5. On the **User Accounts** page, select **Capture and restore all user accounts** and click **Next**.
6. On the **Data Preview** page, click **Next**.
7. On the **Choose Target Collection** page, select the **Install Windows 10 Enterprise x64** collection and click **Next**.
8. On the **Summary** page, click **Next**, and then click **Close**.
9. Select the **User State Migration** node and review the computer association in the right pane.
10. Right-click the **PC0004/PC0006** association and select **View Recovery Information**. Note that a recovery key has been assigned already, but a user state store location has not.
11. Review the Install Windows 10 Enterprise x64 collection. Do not continue until you see the PC0006 machine in the collection. You might have to update and refresh the collection again.

Create a device collection and add the PC0004 computer

1. On CM01, using the Configuration Manager console, in the Asset and Compliance workspace, right-click **Device Collections**, and then select **Create Device Collection**. Use the following settings.
 - General
 - Name: USMT Backup (Replace)
 - Limited Collection: All Systems
 - Membership rules:

- Direct rule
- Resource Class: System Resource
- Attribute Name: Name
- Value: PC0004
- Select **Resources**
- Select **PC0004**

2. Review the USMT Backup (Replace) collection. Do not continue until you see the PC0004 machine in the collection.

Create a new deployment

Using the Configuration Manager console, in the Software Library workspace, select **Task Sequences**, right-click **Replace Task Sequence**, and then select **Deploy**. Use the following settings:

- General
 - Collection: USMT Backup (Replace)
- Deployment Settings
 - Purpose: Available
 - Make available to the following: Only Configuration Manager Clients
- Scheduling
 - <default>
- User Experience
 - <default>
- Alerts
 - <default>
- Distribution Points
 - <default>

Verify the backup

This section assumes that you have a machine named PC0004 with the Configuration Manager 2012 client installed.

1. Start the PC0004 machine, and using the Control Panel, start the Configuration Manager applet.
2. In the **Actions** tab, select the **Machine Policy Retrieval & Evaluation Cycle**, select **Run Now**, and click **OK**.

NOTE

You also can use the Client Notification option in the Configuration Manager console, as shown in [Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#).

3. Using the Software Center, select the **Replace Task Sequence** deployment and click **INSTALL**.
4. In the **Software Center** dialog box, click **INSTALL OPERATING SYSTEM**.

5. Allow the Replace Task Sequence to complete. It should only take about five minutes.
6. On CM01, in the **D:\MigData** folder, verify that a folder was created containing the USMT backup.
7. Using the Configuration Manager console, in the Asset and Compliance workspace, select the **User State Migration** node, right-click the **PC0004/PC0006** association, and select **View Recovery Information**. Note that the object now also has a user state store location.

NOTE

It may take a few minutes for the user state store location to be populated.

Deploy the new computer

1. Start the PC0006 virtual machine, press **F12** to Pre-Boot Execution Environment (PXE) boot when prompted. Allow it to boot Windows Preinstallation Environment (Windows PE), and then complete the deployment wizard using the following settings:
 - Password: P@ssw0rd
 - Select a task sequence to execute on this computer: Windows 10 Enterprise x64 Custom Image
2. The setup now starts and does the following:
 - Installs the Windows 10 operating system
 - Installs the Configuration Manager client
 - Joins it to the domain
 - Installs the applications
 - Restores the PC0004 backup

When the process is complete, you will have a new Windows 10 machine in your domain with user data and settings restored.

Related topics

[Integrate Configuration Manager with MDT](#)

[Prepare for Zero Touch Installation of Windows 10 with Configuration Manager](#)

[Create a custom Windows PE boot image with Configuration Manager](#)

[Add a Windows 10 operating system image using Configuration Manager](#)

[Create an application to deploy with Windows 10 using Configuration Manager](#)

[Add drivers to a Windows 10 deployment with Windows PE using Configuration Manager](#)

[Create a task sequence with Configuration Manager and MDT](#)

[Deploy Windows 10 using PXE and Configuration Manager](#)

[Refresh a Windows 7 SP1 client with Windows 10 using Configuration Manager](#)

Perform an in-place upgrade to Windows 10 using Configuration Manager

6/14/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10

The simplest path to upgrade PCs currently running Windows 7, Windows 8, or Windows 8.1 to Windows 10 is through an in-place upgrade. You can use a System Center Configuration Manager task sequence to completely automate the process.

Proof-of-concept environment

For the purposes of this topic, we will use three machines: DC01, CM01, and PC0001. DC01 is a domain controller and CM01 is a Windows Server 2012 R2 standard machine, fully patched with the latest security updates, and configured as a member server in the fictional contoso.com domain. PC0001 is a machine with Windows 7 SP1, targeted for the Windows 10 upgrade. For more details on the setup for this topic, please see [Deploy Windows 10 with the Microsoft Deployment Toolkit](#).



Figure 1. The machines used in this topic.

Upgrade to Windows 10 with System Center 2012 R2 Configuration Manager

System Center 2012 R2 Configuration Manager SP1 adds support to manage and deploy Windows 10. Although it does not include built-in support to perform an in-place upgrade from Windows 7, Windows 8, or Windows 8.1 to Windows 10, you can build a custom task sequence to perform the necessary tasks.

Create the task sequence

To help with this process, the Configuration Manager team has published [a blog](#) that provides a sample task sequence, as well as the [original blog that includes the instructions for setting up the task sequence](#). To summarize, here are the tasks you need to perform:

1. Download the [Windows10Upgrade1506.zip](#) file that contains the sample task sequence and related scripts. Extract the contents onto a network share.
2. Copy the Windows 10 Enterprise RTM x64 media into the extracted but empty **Windows vNext Upgrade Media** folder.
3. Using the Configuration Manager Console, right-click the **Task Sequences** node, and then choose **Import Task Sequence**. Select the **Windows-vNextUpgradeExport.zip** file that you extracted in Step 1.
4. Distribute the two created packages (one contains the Windows 10 Enterprise x64 media, the other contains the related scripts) to the Configuration Manager distribution point.

For full details and an explanation of the task sequence steps, review the full details of the two blogs that are referenced above.

Create a device collection

After you create the upgrade task sequence, you can create a collection to test a deployment. In this section, we assume you have the PC0001 machine running Windows 7 SP1, with the Configuration Manager client installed.

1. On CM01, using the Configuration Manager console, in the Asset and Compliance workspace, right-click **Device Collections**, and then select **Create Device Collection**. Use the following settings:
 - General
 - Name: Windows 10 Enterprise x64 Upgrade
 - Limited Collection: All Systems
 - Membership rules:
 - Direct rule
 - Resource Class: System Resource
 - Attribute Name: Name
 - Value: PC0001
 - Select Resources
 - Select PC0001
2. Review the Windows 10 Enterprise x64 Upgrade collection. Do not continue until you see the PC0001 machine in the collection.

Deploy the Windows 10 upgrade

In this section, you create a deployment for the Windows 10 Enterprise x64 Update application.

1. On CM01, using the Configuration Manager console, in the Software Library workspace, right-click the **Windows vNext Upgrade** task sequence, and then select **Deploy**.
2. On the **General** page, select the **Windows 10 Enterprise x64 Upgrade** collection, and then click **Next**.
3. On the **Content** page, click **Next**.
4. On the **Deployment Settings** page, select the following settings, and then click **Next**:
 - Action: Install
 - Purpose: Available
5. On the **Scheduling** page, accept the default settings, and then click **Next**.
6. On the **User Experience** page, accept the default settings, and then click **Next**.
7. On the **Alerts** page, accept the default settings, and then click **Next**.
8. On the **Summary** page, click **Next**, and then click **Close**.

Start the Windows 10 upgrade

In this section, you start the Windows 10 Upgrade task sequence on PC0001 (currently running Windows 7 SP1).

1. On PC0001, start the **Software Center**.
2. Select the **Windows vNext Upgrade** task sequence, and then click **Install**.

When the task sequence begins, it will automatically initiate the in-place upgrade process by invoking the Windows setup program (Setup.exe) with the necessary command-line parameters to perform an automated upgrade, which preserves all data, settings, apps, and drivers.

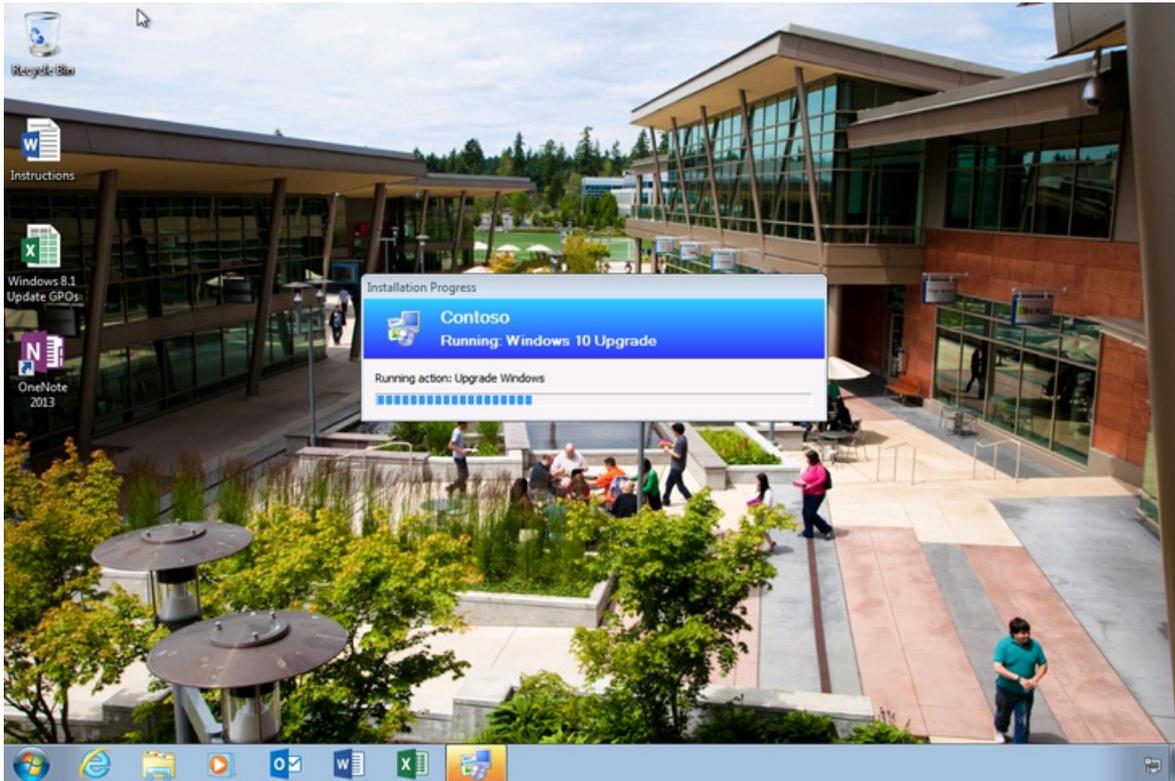


Figure 2. Upgrade from Windows 7 to Windows 10 Enterprise x64 with a task sequence.

After the task sequence finishes, the computer will be fully upgraded to Windows 10.

Upgrade to Windows 10 with System Center Configuration Manager Current Branch

With System Center Configuration Manager Current Branch, new built-in functionality makes it easier to upgrade to Windows 10.

Note For more details about Configuration Manager Current Branch, see the [Configuration Manager Team blog](#). An [evaluation version is currently available](#) for you to try. The instructions below are specific to the Technical Preview 2 release and may change after the next version of Configuration Manager is released.

Create the OS upgrade package

First, you need to create an operating system upgrade package that contains the full Windows 10 Enterprise x64 installation media.

1. On CM01, using the Configuration Manager console, in the Software Library workspace, right-click the **Operating System Upgrade Packages** node, then select **Add Operating System Upgrade Package**.
2. On the **Data Source** page, specify the UNC path to the Windows 10 Enterprise x64 media, and then click **Next**.
3. On the **General** page, specify Windows 10 Enterprise x64 Upgrade, and then click **Next**.
4. On the **Summary** page, click **Next**, and then click **Close**.
5. Right-click the created **Windows 10 Enterprise x64 Update** package, and then select **Distribute Content**. Choose the CM01 distribution point.

Create the task sequence

To create an upgrade task sequence, perform the following steps:

1. On CM01, using the Configuration Manager console, in the Software Library workspace, right-click the **Task Sequences** node, and then select **Create Task Sequence**.
2. On the **Create a new task sequence** page, select **Upgrade an operating system from upgrade package**, and then click **Next**.
3. On the **Task Sequence Information** page, specify **Windows 10 Enterprise x64 Upgrade**, and then click **Next**.
4. On the **Upgrade the Windows operating system** page, select the **Windows 10 Enterprise x64 Upgrade operating system upgrade** package, and then click **Next**.
5. Click **Next** through the remaining wizard pages, and then click **Close**.

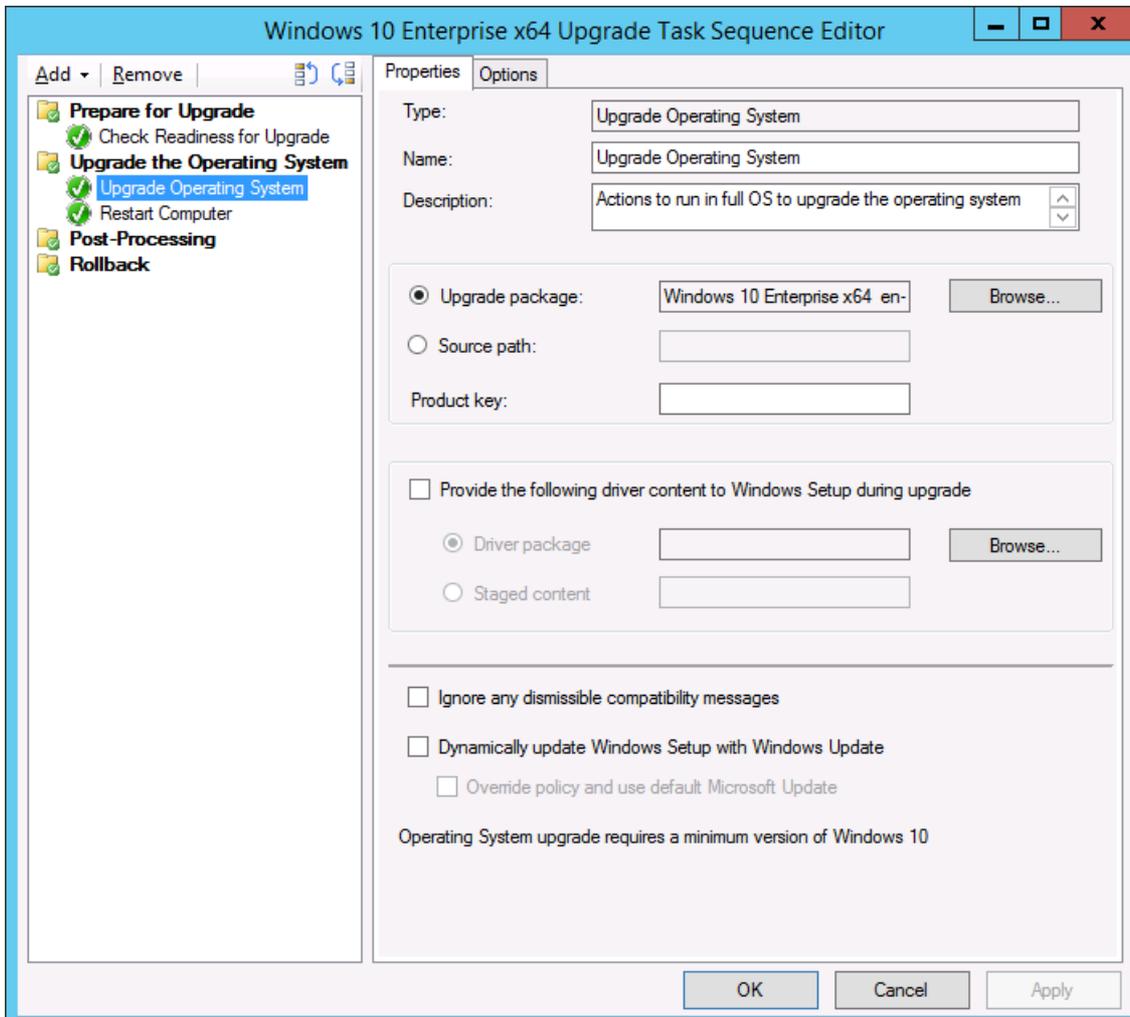


Figure 3. The Configuration Manager upgrade task sequence.

Create a device collection

After you create the upgrade task sequence, you can create a collection to test a deployment. In this section, we assume you have the PC0001 machine running Windows 7 SP1, with the next version of System Center Configuration Manager client installed.

1. On CM01, using the Configuration Manager console, in the Asset and Compliance workspace, right-click **Device Collections**, and then select **Create Device Collection**. Use the following settings:
 - General
 - Name: Windows 10 Enterprise x64 Upgrade
 - Limited Collection: All Systems
 - Membership rules:

- Direct rule
 - Resource Class: System Resource
 - Attribute Name: Name
 - Value: PC0001
- Select Resources
- Select PC0001

2. Review the Windows 10 Enterprise x64 Upgrade collection. Do not continue until you see the PC0001 machine in the collection.

Deploy the Windows 10 upgrade

In this section, you create a deployment for the Windows 10 Enterprise x64 Update application.

1. On CM01, using the Configuration Manager console, in the Software Library workspace, right-click the **Windows vNext Upgrade** task sequence, and then select **Deploy**.
2. On the **General** page, select the **Windows 10 Enterprise x64 Upgrade** collection, and then click **Next**.
3. On the **Content** page, click **Next**.
4. On the **Deployment Settings** page, select the following settings and click **Next**:
 - Action: Install
 - Purpose: Available
5. On the **Scheduling** page, accept the default settings, and then click **Next**.
6. On the **User Experience** page, accept the default settings, and then click **Next**.
7. On the **Alerts** page, accept the default settings, and then click **Next**.
8. On the **Summary** page, click **Next**, and then click **Close**.

Start the Windows 10 upgrade

In this section, you start the Windows 10 Upgrade task sequence on PC0001 (currently running Windows 7 SP1).

1. On PC0001, start the **Software Center**.
2. Select the **Windows 10 Enterprise x64 Upgrade** task sequence, and then click **Install**.

When the task sequence begins, it automatically initiates the in-place upgrade process by invoking the Windows setup program (Setup.exe) with the necessary command-line parameters to perform an automated upgrade, which preserves all data, settings, apps, and drivers.

After the task sequence completes, the computer will be fully upgraded to Windows 10.

Related topics

[Windows 10 deployment scenarios](#)

[Configuration Manager Team blog](#)

Windows 10 deployment tools

6/18/2019 • 2 minutes to read • [Edit Online](#)

Learn about the tools available to deploy Windows 10.

| TOPIC | DESCRIPTION |
|--|--|
| Windows 10 deployment scenarios and tools | To successfully deploy the Windows 10 operating system and applications for your organization, it is essential that you know about the available tools to help with the process. In this topic, you will learn about the most commonly used tools for Windows 10 deployment. |
| Convert MBR partition to GPT | This topic provides detailed instructions for using the MBR2GPT partition conversion tool. |
| Configure a PXE server to load Windows PE | This guide describes how to configure a PXE server to load Windows PE by booting a client computer from the network. |
| Windows ADK for Windows 10 scenarios for IT Pros | The Windows Assessment and Deployment Kit (Windows ADK) contains tools that can be used by IT Pros to deploy Windows. |
| Deploy Windows To Go in your organization | This topic helps you to deploy Windows To Go in your organization. Before you begin deployment, make sure that you have reviewed the topics Windows To Go: feature overview and Prepare your organization for Windows To Go to ensure that you have the correct hardware and are prepared to complete the deployment. You can then use the steps in this topic to start your Windows To Go deployment. |
| Volume Activation Management Tool (VAMT) Technical Reference | The Volume Activation Management Tool (VAMT) enables network administrators and other IT professionals to automate and centrally manage the Windows®, Microsoft® Office, and select other Microsoft products volume and retail-activation process. |
| User State Migration Tool (USMT) Technical Reference | The User State Migration Tool (USMT) 10.0 is included with the Windows Assessment and Deployment Kit (Windows ADK) for Windows 10. USMT provides a highly customizable user-profile migration experience for IT professionals |

Windows 10 deployment scenarios and tools

6/18/2019 • 16 minutes to read • [Edit Online](#)

To successfully deploy the Windows 10 operating system and applications for your organization, it is essential that you know about the available tools to help with the process. In this topic, you will learn about the most commonly used tools for Windows 10 deployment.

Microsoft provides many tools, services, and solutions. These tools include Windows Deployment Services (WDS), the Volume Activation Management Tool (VAMT), the User State Migration Tool (USMT), Windows System Image Manager (Windows SIM), Windows Preinstallation Environment (Windows PE), and Windows Recovery Environment (Windows RE). Keep in mind that these are just tools and not a complete solution on their own. It's when you combine these tools with solutions like [Microsoft Deployment Toolkit \(MDT\)](#) or [Microsoft System Center 2012 R2 Configuration Manager](#) that you get the complete deployment solution.

In this topic, you also learn about different types of reference images that you can build, and why reference images are beneficial for most organizations

Windows Assessment and Deployment Kit

Windows ADK contains core assessment and deployment tools and technologies, including Deployment Image Servicing and Management (DISM), Windows Imaging and Configuration Designer (Windows ICD), Windows System Image Manager (Windows SIM), User State Migration Tool (USMT), Volume Activation Management Tool (VAMT), Windows Preinstallation Environment (Windows PE), Windows Assessment Services, Windows Performance Toolkit (WPT), Application Compatibility Toolkit (ACT), and Microsoft SQL Server 2012 Express. For more details, see [Windows ADK for Windows 10](#) or [Windows ADK for Windows 10 scenarios for IT Pros](#).

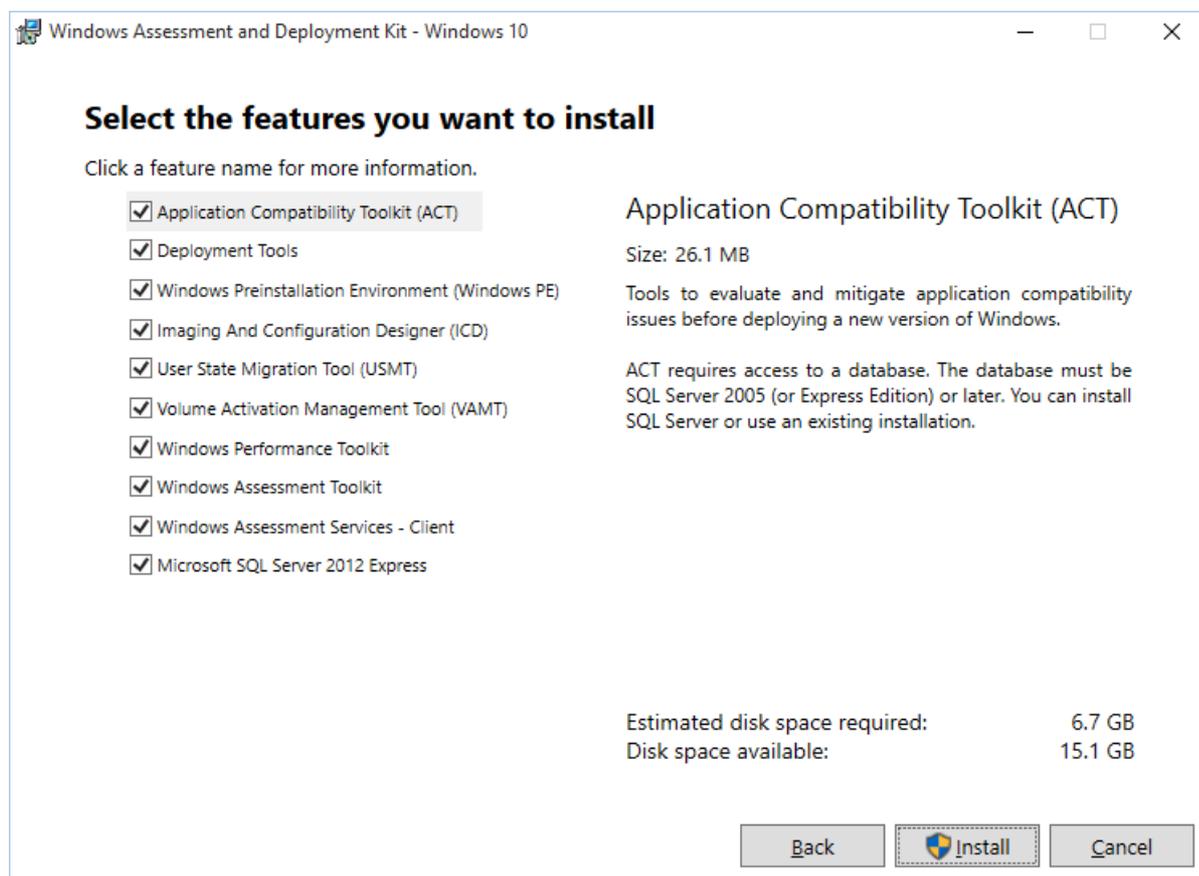


Figure 1. The Windows 10 ADK feature selection page.

Deployment Image Servicing and Management (DISM)

DISM is one of the deployment tools included in the Windows ADK and is used for capturing, servicing, and deploying boot images and operating system images.

DISM services online and offline images. For example, with DISM you can install the Microsoft .NET Framework 3.5.1 in Windows 10 online, which means that you can start the installation in the running operating system, not that you get the software online. The `/LimitAccess` switch configures DISM to get the files only from a local source:

```
Dism.exe /Online /Enable-Feature /FeatureName:NetFx3 /All /Source:D:\Sources\SxS /LimitAccess
```

In Windows 10, you can use Windows PowerShell for many of the functions performed by DISM.exe. The equivalent command in Windows 10 using PowerShell is:

```
Enable-WindowsOptionalFeature -Online -FeatureName NetFx3 -All  
-Source D:\Sources\SxS -LimitAccess
```

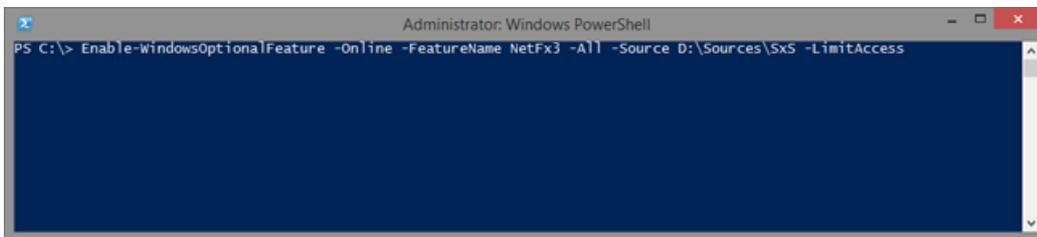


Figure 2. Using DISM functions in PowerShell.

For more information on DISM, see [DISM technical reference](#).

User State Migration Tool (USMT)

USMT is a backup and restore tool that allows you to migrate user state, data, and settings from one installation to another. Microsoft Deployment Toolkit (MDT) and System Center 2012 R2 Configuration Manager use USMT as part of the operating system deployment process.

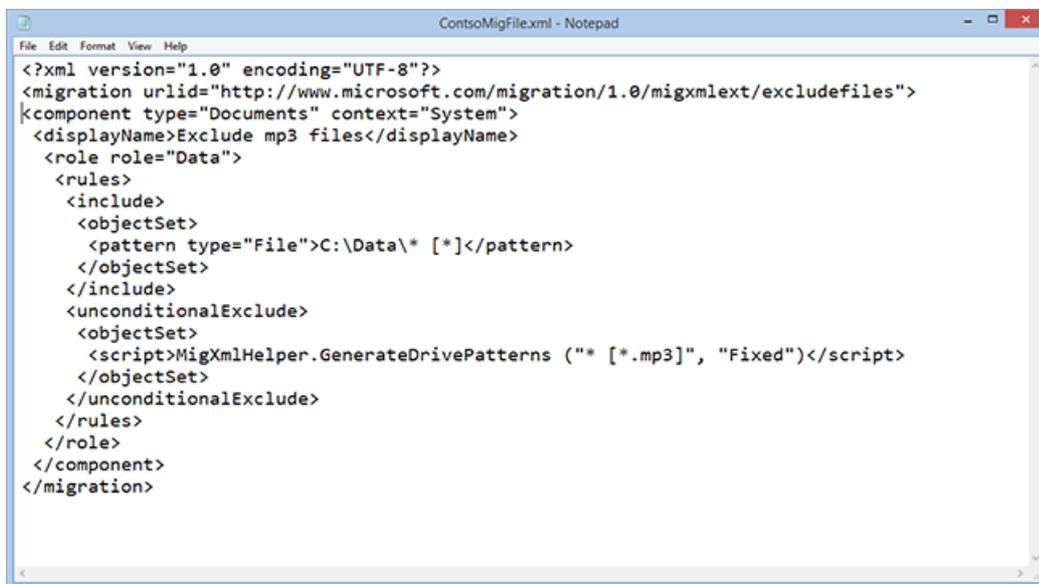
Note Occasionally, we find that customers are wary of USMT because they believe it requires significant configuration, but, as you will learn below, using USMT is not difficult. If you use MDT and Lite Touch to deploy your machines, the USMT feature is automatically configured and extended so that it is easy to use. With MDT, you do nothing at all and USMT just works.

USMT includes several command-line tools, the most important of which are ScanState and LoadState:

- **ScanState.exe.** This performs the user-state backup.
- **LoadState.exe.** This performs the user-state restore.
- **UsmtUtils.exe.** This supplements the functionality in ScanState.exe and LoadState.exe.

In addition to these tools, there are also XML templates that manage which data is migrated. You can customize the templates, or create new ones, to manage the backup process at a high level of detail. USMT uses the following terms for its templates:

- **Migration templates.** The default templates in USMT.
- **Custom templates.** Custom templates that you create.
- **Config template.** An optional template, called Config.xml, which you can use to exclude or include components in a migration without modifying the other standard XML templates.



```
<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/excludefiles">
  <component type="Documents" context="System">
    <displayName>Exclude mp3 files</displayName>
    <role role="Data">
      <rules>
        <include>
          <objectSet>
            <pattern type="File">C:\Data\* [*]</pattern>
          </objectSet>
        </include>
        <unconditionalExclude>
          <objectSet>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*].mp3", "Fixed")</script>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

Figure 3. A sample USMT migration file that will exclude .MP3 files on all local drives and include the folder C:\Data and all its files, including its subdirectories and their files.

USMT supports capturing data and settings from Windows Vista and later, and restoring the data and settings to Windows 7 and later (including Windows 10 in both cases). It also supports migrating from a 32-bit operating system to a 64-bit operating system, but not the other way around. For example, you can use USMT to migrate from Windows 7 x86 to Windows 10 x64.

By default USMT migrates many settings, most of which are related to the user profile but also to Control Panel configurations, file types, and more. The default templates that are used in Windows 10 deployments are MigUser.xml and MigApp.xml. These two default templates migrate the following data and settings:

- Folders from each profile, including those from user profiles as well as shared and public profiles. For example, the My Documents, My Video, My Music, My Pictures, desktop files, Start menu, Quick Launch settings, and Favorites folders are migrated.
- Specific file types. USMT templates migrate the following file types: .accdb, .ch3, .csv, .dif, .doc*, .dot*, .dqy, .iqy, .mcw, .mdb*, .mpp, .one*, .oqy, .or6, .pot*, .ppa, .pps*, .ppt*, .pre, .pst, .pub, .qdf, .qel, .qph, .qsd, .rqy, .rtf, .scd, .sh3, .slk, .txt, .vl*, .vsd, .wk*, .wpd, .wps, .wq1, .wri, .xl*, .xla, .xlb, .xls*.

Note The OpenDocument extensions (*.odt, *.odp, *.ods, etc.) that Microsoft Office applications can use are not migrated by default.

- Operating system component settings
- Application settings

These are the settings migrated by the default MigUser.xml and MigApp.xml templates. For more details on what USMT migrates, see [What does USMT migrate?](#) For more information on the USMT overall, see the [USMT technical reference](#).

Windows Imaging and Configuration Designer

Windows Imaging and Configuration Designer (Windows ICD) is a tool designed to assist with the creation of provisioning packages that can be used to dynamically configure a Windows device (PCs, tablets, and phones). This is particularly useful for setting up new devices, without the need for re-imaging the device with a custom image.

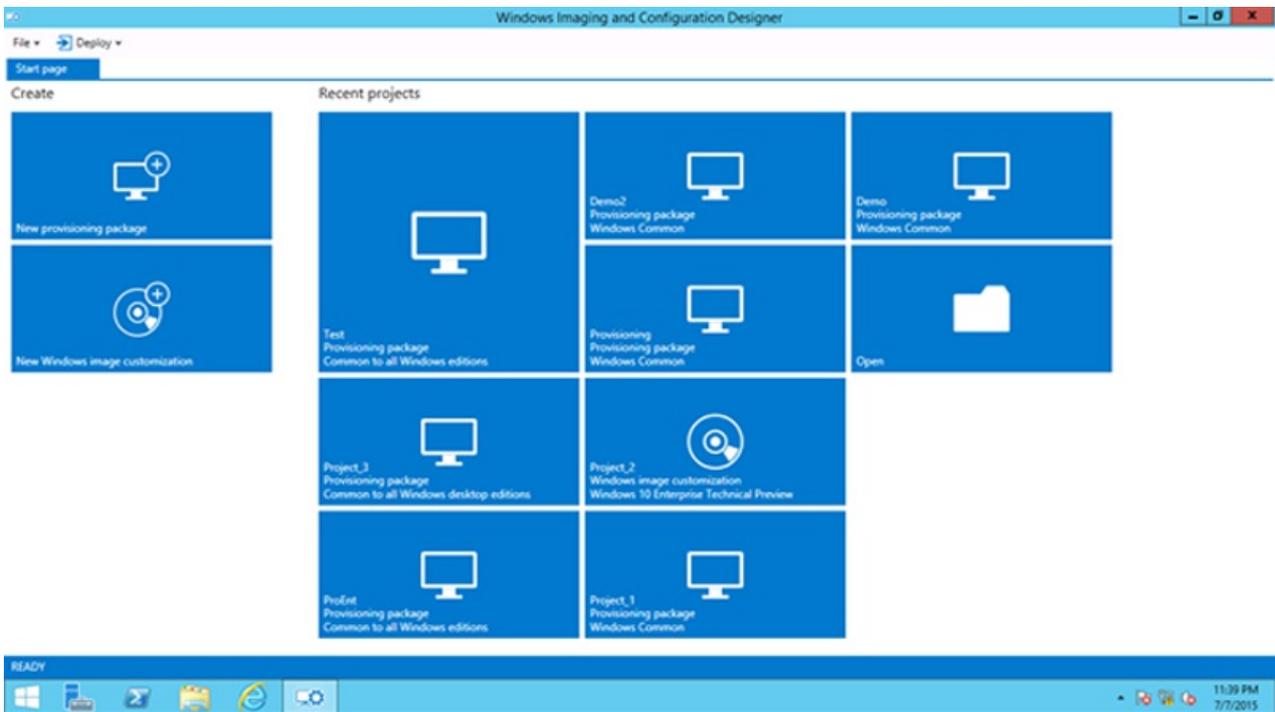


Figure 4. Windows Imaging and Configuration Designer.

For more information, see [Windows Imaging and Configuration Designer](#).

Windows System Image Manager (Windows SIM)

Windows SIM is an authoring tool for Unattend.xml files. When using MDT and/or Configuration Manager, you don't need Windows SIM very often because those systems automatically update the Unattend.xml file during the deployment, greatly simplifying the process overall.

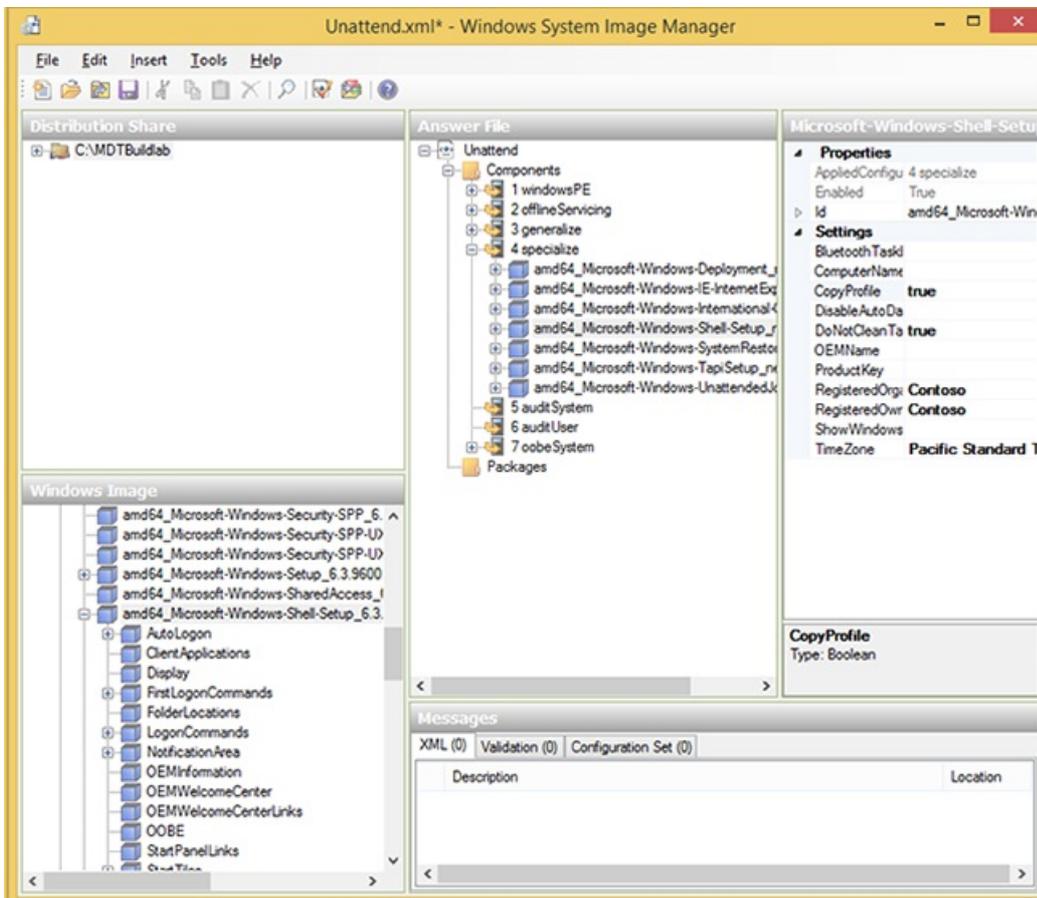


Figure 5. Windows answer file opened in Windows SIM.

For more information, see [Windows System Image Manager Technical Reference](#).

Volume Activation Management Tool (VAMT)

If you don't use KMS, you can still manage your MAKs centrally with the Volume Activation Management Tool (VAMT). With this tool, you can install and manage product keys throughout the organization. VAMT also can activate on behalf of clients without Internet access, acting as a MAK proxy.

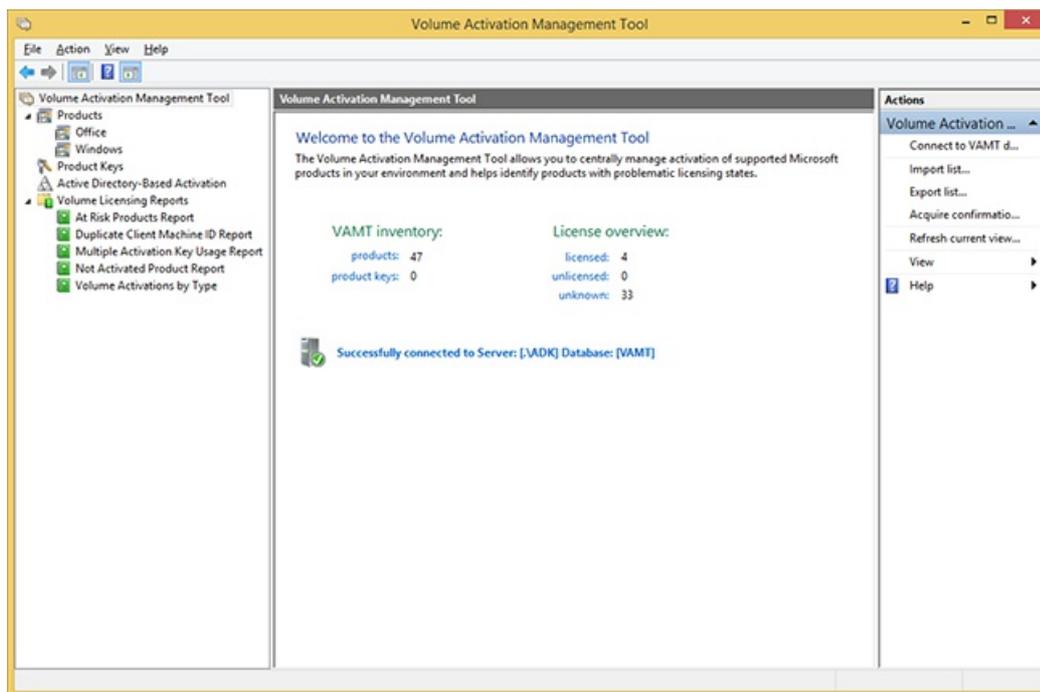


Figure 6. The updated Volume Activation Management Tool.

VAMT also can be used to create reports, switch from MAK to KMS, manage Active Directory-based activation, and manage Office 2010 and Office 2013 volume activation. VAMT also supports PowerShell (instead of the old command-line tool). For example, if you want to get information from the VAMT database, you can type:

```
Get-VamtProduct
```

For more information on the VAMT, see [VAMT technical reference](#).

Windows Preinstallation Environment (Windows PE)

Windows PE is a "Lite" version of Windows 10 and was created to act as a deployment platform. Windows PE replaces the DOS or Linux boot disks that ruled the deployment solutions of the last decade.

The key thing to know about Windows PE is that, like the operating system, it needs drivers for at least network and storage devices in each PC. Luckily Windows PE includes the same drivers as the full Windows 10 operating system, which means much of your hardware will work out of the box.

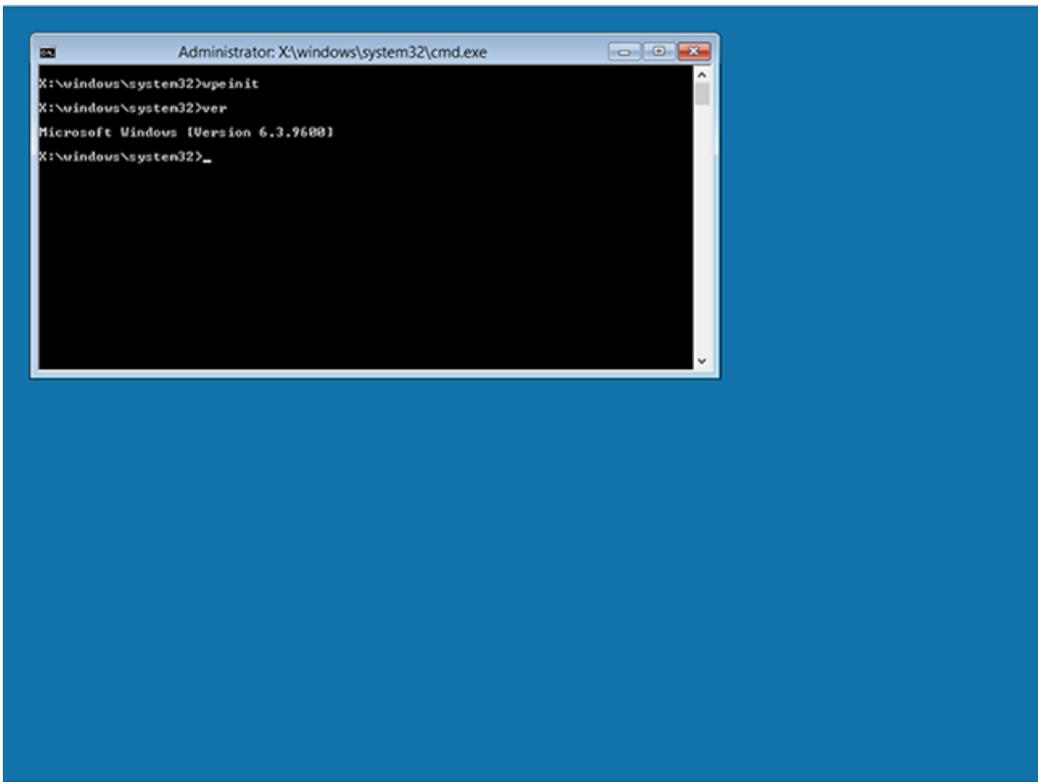


Figure 7. A machine booted with the Windows ADK default Windows PE boot image.

For more details on Windows PE, see [Windows PE \(WinPE\)](#).

Windows Recovery Environment

Windows Recovery Environment (Windows RE) is a diagnostics and recovery toolset included in Windows Vista and later operating systems. The latest version of Windows RE is based on Windows PE. You can also extend Windows RE and add your own tools if needed. If a Windows installation fails to start and Windows RE is installed, you will see an automatic failover into Windows RE.

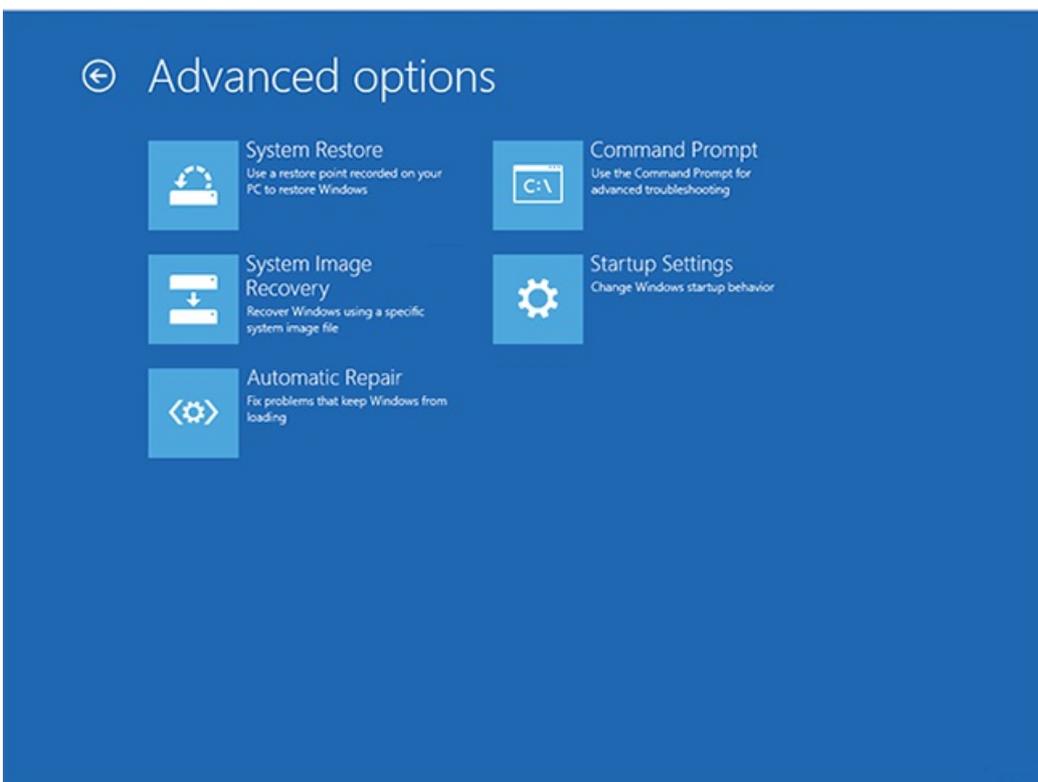


Figure 8. A Windows 10 client booted into Windows RE, showing Advanced options.

For more information on Windows RE, see [Windows Recovery Environment](#).

Windows Deployment Services

Windows Deployment Services (WDS) has been updated and improved in several ways starting with Windows 8. Remember that the two main functions you will use are the PXE boot support and multicast. Most of the changes are related to management and increased performance. In Windows Server 2012 R2, WDS also can be used for the Network Unlock feature in BitLocker.

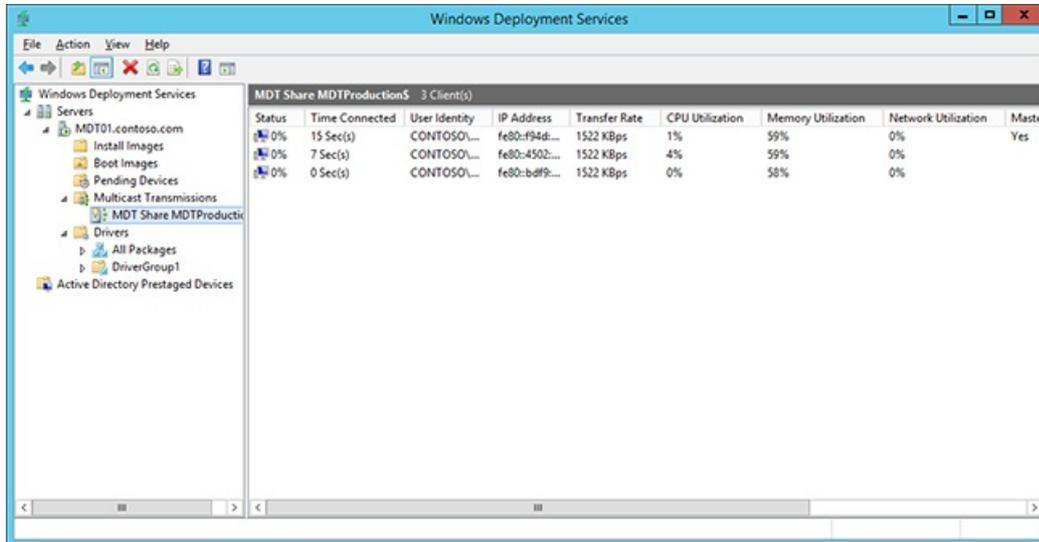


Figure 9. Windows Deployment Services using multicast to deploy three machines.

In Windows Server 2012 R2, [Windows Deployment Services](#) can be configured for stand-alone mode or for Active Directory integration. In most scenarios, the Active Directory integration mode is the best option. WDS also has the capability to manage drivers; however, driver management through MDT and Configuration Manager is more suitable for deployment due to the flexibility offered by both solutions, so you will use them instead. In WDS, it is possible to pre-stage devices in Active Directory, but here, too, Configuration Manager has that capability built in, and MDT has the ability to use a SQL Server database for pre-staging. In most scenarios, those solutions are better than the built-in pre-staging function as they allow greater control and management.

Trivial File Transfer Protocol (TFTP) configuration

In some cases, you need to modify TFTP Maximum Block Size settings for performance tuning reasons, especially when PXE traffic travels through routers and such. In the previous version of WDS, it was possible to change that, but the method of do so—editing the registry—was not user friendly. In Windows Server 2012, this has become much easier to do as it can be configured as a setting.

Also, there are a few new features related to TFTP performance:

- **Scalable buffer management.** Allows buffering an entire file instead of a fixed-size buffer for each client, enabling different sessions to read from the same shared buffer.
- **Scalable port management.** Provides the capability to service clients with shared UDP port allocation, increasing scalability.
- **Variable-size transmission window (Variable Windows Extension).** Improves TFTP performance by allowing the client and server to determine the largest workable window size.

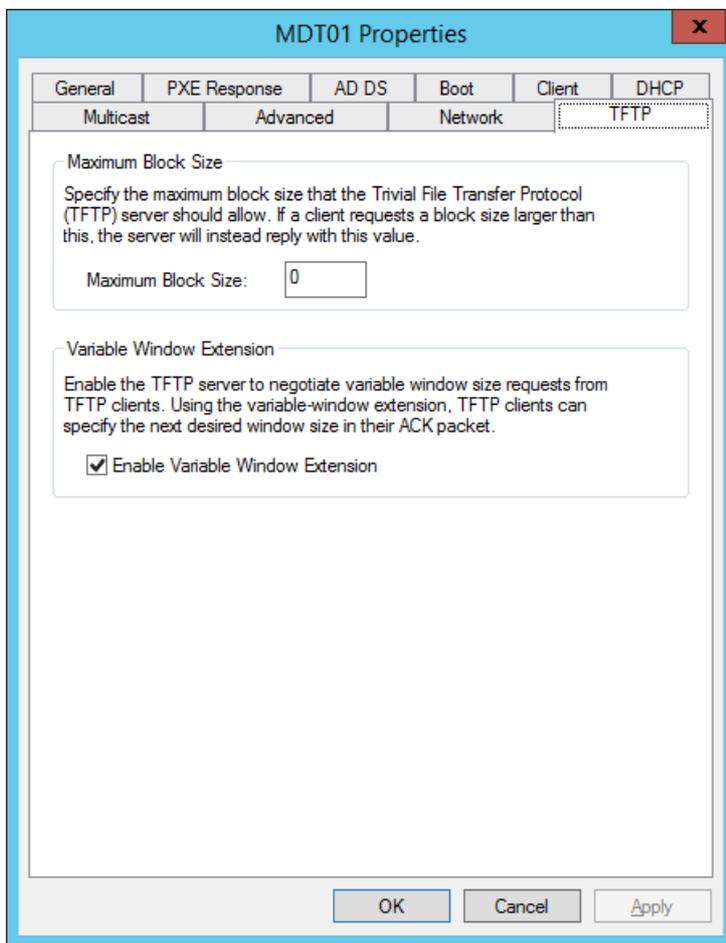


Figure 10. TFTP changes are now easy to perform.

Microsoft Deployment Toolkit

MDT is a free deployment solution from Microsoft. It provides end-to-end guidance, best practices, and tools for planning, building, and deploying Windows operating systems. MDT builds on top of the core deployment tools in the Windows ADK by contributing guidance, reducing complexity, and adding critical features for an enterprise-ready deployment solution.

MDT has two main parts: the first is Lite Touch, which is a stand-alone deployment solution; the second is Zero Touch, which is an extension to System Center 2012 R2 Configuration Manager.

Note Lite Touch and Zero Touch are marketing names for the two solutions that MDT supports, and the naming has nothing to do with automation. You can fully automate the stand-alone MDT solution (Lite Touch), and you can configure the solution integration with Configuration Manager to prompt for information.

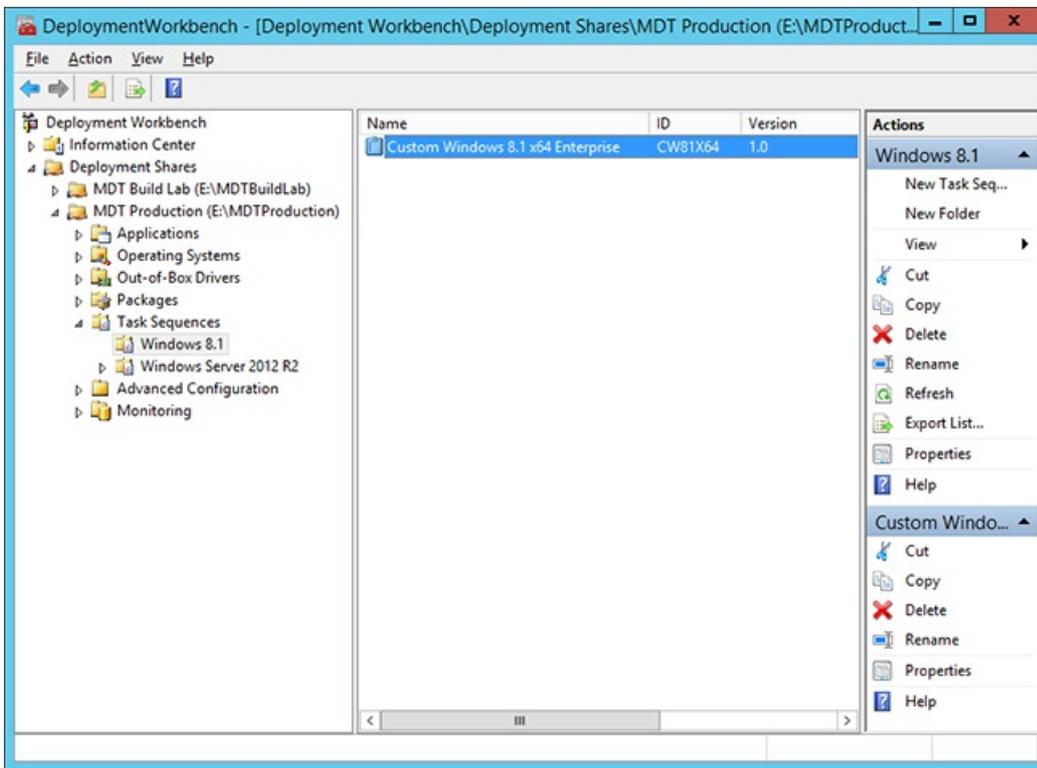


Figure 11. The Deployment Workbench in, showing a task sequence.

For more information on MDT, see the [Microsoft Deployment Toolkit](#) resource center.

Microsoft Security Compliance Manager 2013

Microsoft SCM is a free utility used to create baseline security settings for the Windows client and server environment. The baselines can be exported and then deployed via Group Policy, local policies, MDT, or Configuration Manager. The current version of Security Compliance Manager includes baselines for Windows 8.1 and several earlier versions of Windows, Windows Server, and Internet Explorer.

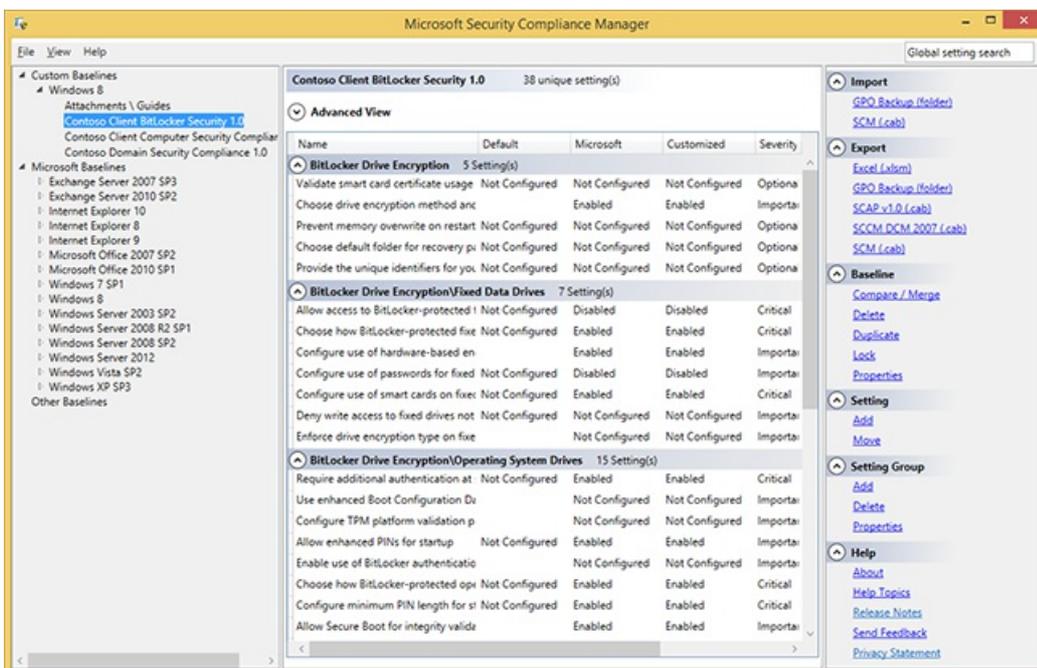


Figure 12. The SCM console showing a baseline configuration for a fictional client's computer security compliance.

Microsoft Desktop Optimization Pack

MDOP is a suite of technologies available to Software Assurance customers through an additional subscription.

The following components are included in the MDOP suite:

- **Microsoft Application Virtualization (App-V).** App-V 5.0 provides an integrated platform, more flexible virtualization, and powerful management for virtualized applications. With the release of App-V 5.0 SP3, you have support to run virtual applications on Windows 10.
- **Microsoft User Experience Virtualization (UE-V).** UE-V monitors the changes that are made by users to application settings and Windows operating system settings. The user settings are captured and centralized to a settings storage location. These settings can then be applied to the different computers that are accessed by the user, including desktop computers, laptop computers, and virtual desktop infrastructure (VDI) sessions.
- **Microsoft Advanced Group Policy Management (AGPM).** AGPM enables advanced management of Group Policy objects by providing change control, offline editing, and role-based delegation.
- **Microsoft Diagnostics and Recovery Toolset (DaRT).** DaRT provides additional tools that extend Windows RE to help you troubleshoot and repair your machines.
- **Microsoft BitLocker Administration and Monitoring (MBAM).** MBAM is an administrator interface used to manage BitLocker drive encryption. It allows you to configure your enterprise with the correct BitLocker encryption policy options, as well as monitor compliance with these policies.

For more information on the benefits of an MDOP subscription, see [Microsoft Desktop Optimization Pack](#).

Internet Explorer Administration Kit 11

There has been a version of IEAK for every version of Internet Explorer since 3.0. It gives you the capability to customize Internet Explorer as you would like. The end result of using IEAK is an Internet Explorer package that can be deployed unattended. The wizard creates one .exe file and one .msi file.

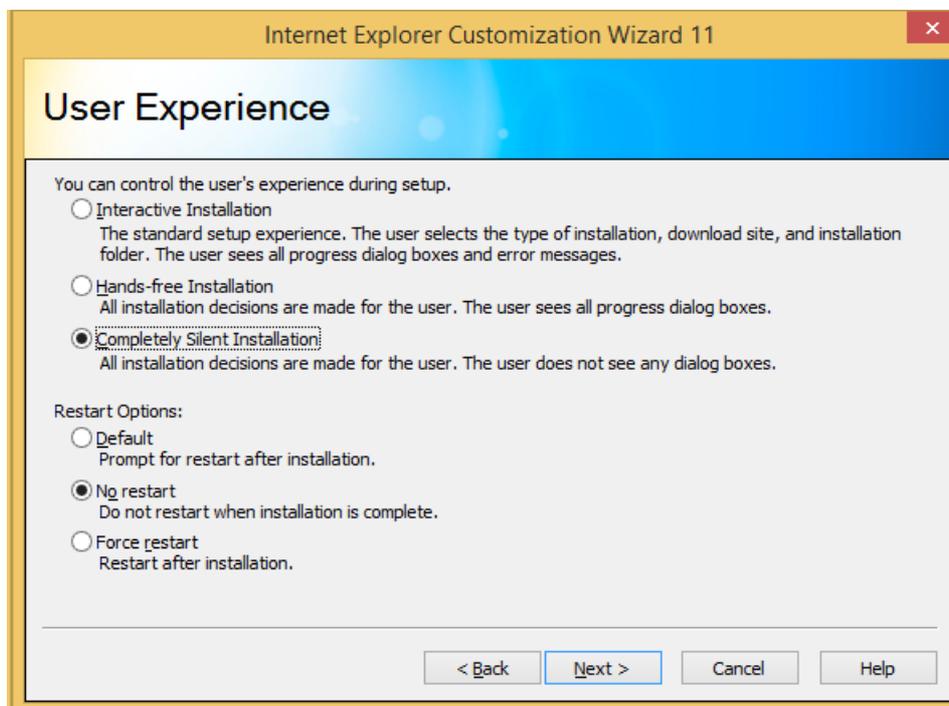


Figure 13. The User Experience selection screen in IEAK 11.

To download IEAK 11, see the [Internet Explorer Administration Kit \(IEAK\) Information and Downloads](#) page.

Windows Server Update Services

WSUS is a server role in Windows Server 2012 R2 that enables you to maintain a local repository of Microsoft updates and then distribute them to machines on your network. WSUS offers approval control and reporting of update status in your environment.

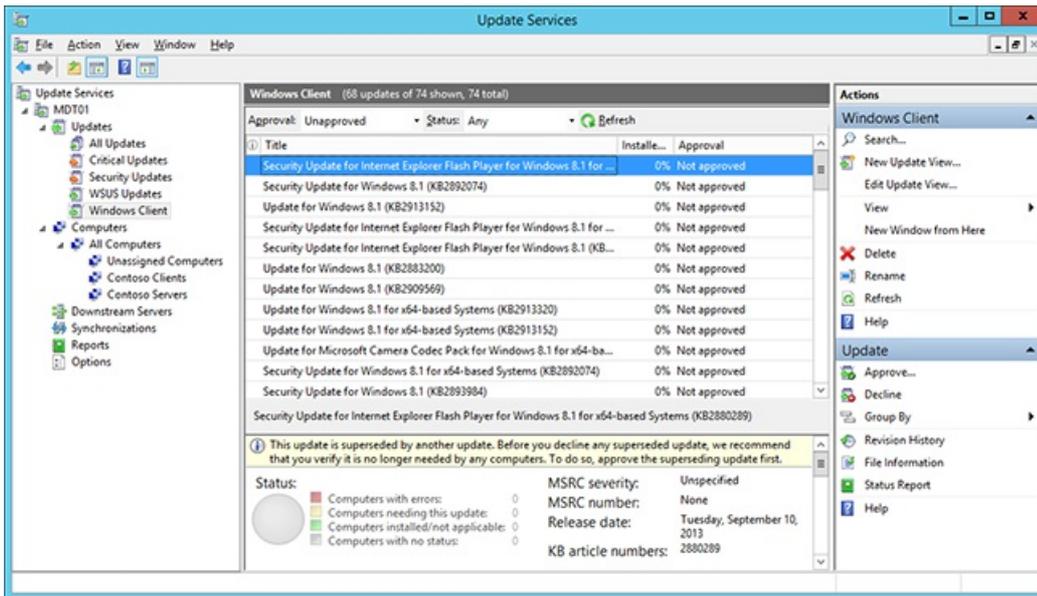


Figure 14. The Windows Server Update Services console.

For more information on WSUS, see the [Windows Server Update Services Overview](#).

Unified Extensible Firmware Interface

For many years BIOS has been the industry standard for booting a PC. BIOS has served us well, but it is time to replace it with something better. **UEFI** is the replacement for BIOS, so it is important to understand the differences between BIOS and UEFI. In this section, you learn the major differences between the two and how they affect operating system deployment.

Introduction to UEFI

BIOS has been in use for approximately 30 years. Even though it clearly has proven to work, it has some limitations, including:

- 16-bit code
- 1 MB address space
- Poor performance on ROM initialization
- MBR maximum bootable disk size of 2.2 TB

As the replacement to BIOS, UEFI has many features that Windows can and will use.

With UEFI, you can benefit from:

- **Support for large disks.** UEFI requires a GUID Partition Table (GPT) based disk, which means a limitation of roughly 16.8 million TB in disk size and more than 100 primary disks.
- **Faster boot time.** UEFI does not use INT 13, and that improves boot time, especially when it comes to resuming from hibernate.
- **Multicast deployment.** UEFI firmware can use multicast directly when it boots up. In WDS, MDT, and Configuration Manager scenarios, you need to first boot up a normal Windows PE in unicast and then switch into multicast. With UEFI, you can run multicast from the start.
- **Compatibility with earlier BIOS.** Most of the UEFI implementations include a compatibility support

module (CSM) that emulates BIOS.

- **CPU-independent architecture.** Even if BIOS can run both 32- and 64-bit versions of firmware, all firmware device drivers on BIOS systems must also be 16-bit, and this affects performance. One of the reasons is the limitation in addressable memory, which is only 64 KB with BIOS.
- **CPU-independent drivers.** On BIOS systems, PCI add-on cards must include a ROM that contains a separate driver for all supported CPU architectures. That is not needed for UEFI because UEFI has the ability to use EFI Byte Code (EBC) images, which allow for a processor-independent device driver environment.
- **Flexible pre-operating system environment.** UEFI can perform many functions for you. You just need an UEFI application, and you can perform diagnostics and automatic repairs, and call home to report errors.
- **Secure boot.** Windows 8 and later can use the UEFI firmware validation process, called secure boot, which is defined in UEFI 2.3.1. Using this process, you can ensure that UEFI launches only a verified operating system loader and that malware cannot switch the boot loader.

Versions

UEFI Version 2.3.1B is the version required for Windows 8 and later logo compliance. Later versions have been released to address issues; a small number of machines may need to upgrade their firmware to fully support the UEFI implementation in Windows 8 and later.

Hardware support for UEFI

In regard to UEFI, hardware is divided into four device classes:

- **Class 0 devices.** This is the UEFI definition for a BIOS, or non-UEFI, device.
- **Class 1 devices.** These devices behave like a standard BIOS machine, but they run EFI internally. They should be treated as normal BIOS-based machines. Class 1 devices use a CSM to emulate BIOS. These older devices are no longer manufactured.
- **Class 2 devices.** These devices have the capability to behave as a BIOS- or a UEFI-based machine, and the boot process or the configuration in the firmware/BIOS determines the mode. Class 2 devices use a CSM to emulate BIOS. These are the most common type of devices currently available.
- **Class 3 devices.** These are UEFI-only devices, which means you must run an operating system that supports only UEFI. Those operating systems include Windows 8, Windows 8.1, Windows Server 2012, and Windows Server 2012 R2. Windows 7 is not supported on these class 3 devices. Class 3 devices do not have a CSM to emulate BIOS.

Windows support for UEFI

Microsoft started with support for EFI 1.10 on servers and then added support for UEFI on both clients and servers.

With UEFI 2.3.1, there are both x86 and x64 versions of UEFI. Windows 10 supports both. However, UEFI does not support cross-platform boot. This means that a computer that has UEFI x64 can run only a 64-bit operating system, and a computer that has UEFI x86 can run only a 32-bit operating system.

How UEFI is changing operating system deployment

There are many things that affect operating system deployment as soon as you run on UEFI/EFI-based hardware. Here are considerations to keep in mind when working with UEFI devices:

- Switching from BIOS to UEFI in the hardware is easy, but you also need to reinstall the operating system because you need to switch from MBR/NTFS to GPT/FAT32 and NTFS.
- When you deploy to a Class 2 device, make sure the boot option you select matches the setting you want to

have. It is common for old machines to have several boot options for BIOS but only a few for UEFI, or vice versa.

- When deploying from media, remember the media has to be FAT32 for UEFI, and FAT32 has a file-size limitation of 4GB.
- UEFI does not support cross-platform booting; therefore, you need to have the correct boot media (32- or 64-bit).

For more information on UEFI, see the [UEFI firmware](#) overview and related resources.

Related topics

[Deploy Windows To Go](#)

[Sideload apps in Windows 10](#)

[Windows ADK for Windows 10 scenarios for IT pros](#)

MBR2GPT.EXE

6/18/2019 • 13 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Summary

MBR2GPT.EXE converts a disk from the Master Boot Record (MBR) to the GUID Partition Table (GPT) partition style without modifying or deleting data on the disk. The tool is designed to be run from a Windows Preinstallation Environment (Windows PE) command prompt, but can also be run from the full Windows 10 operating system (OS) by using the **/allowFullOS** option.

MBR2GPT.EXE is located in the **Windows\System32** directory on a computer running Windows 10 version 1703 (also known as the Creator's Update) or later. The tool is available in both the full OS environment and Windows PE. To use this tool in a deployment task sequence with Configuration Manager or Microsoft Deployment Toolkit (MDT), you must first update the Windows PE image (winpe.wim, boot.wim) with the [Windows ADK 1703](#), or a later version.

See the following video for a detailed description and demonstration of MBR2GPT.

<https://www.youtube-nocookie.com/embed/hfJep4hmg9o>

You can use MBR2GPT to:

- Convert any attached MBR-formatted system disk to the GPT partition format. You cannot use the tool to convert non-system disks from MBR to GPT.
- Convert an MBR disk with BitLocker-encrypted volumes as long as protection has been suspended. To resume BitLocker after conversion, you will need to delete the existing protectors and recreate them.
- Convert operating system disks that have earlier versions of Windows 10 installed, such as versions 1507, 1511, and 1607. However, you must run the tool while booted into Windows 10 version 1703 or later, and perform an offline conversion.
- Convert an operating system disk from MBR to GPT using Configuration Manager or MDT provided that your task sequence uses Windows PE version 1703 or later.

Offline conversion of system disks with earlier versions of Windows installed, such as Windows 7, 8, or 8.1 are not officially supported. The recommended method to convert these disks is to upgrade the operating system to Windows 10 first, then perform the MBR to GPT conversion.

IMPORTANT

After the disk has been converted to GPT partition style, the firmware must be reconfigured to boot in UEFI mode. Make sure that your device supports UEFI before attempting to convert the disk.

Disk Prerequisites

Before any change to the disk is made, MBR2GPT validates the layout and geometry of the selected disk to ensure that:

- The disk is currently using MBR

- There is enough space not occupied by partitions to store the primary and secondary GPTs:
 - 16KB + 2 sectors at the front of the disk
 - 16KB + 1 sector at the end of the disk
- There are at most 3 primary partitions in the MBR partition table
- One of the partitions is set as active and is the system partition
- The disk does not have any extended/logical partition
- The BCD store on the system partition contains a default OS entry pointing to an OS partition
- The volume IDs can be retrieved for each volume which has a drive letter assigned
- All partitions on the disk are of MBR types recognized by Windows or has a mapping specified using the /map command-line option

If any of these checks fails, the conversion will not proceed and an error will be returned.

Syntax

```
MBR2GPT /validate|convert [/disk:<diskNumber>] [/logs:<logDirectory>] [/map:<source>=<destination>]
[/allowFullIOS]
```

Options

OPTION	DESCRIPTION
/validate	Instructs MBR2GPT.exe to perform only the disk validation steps and report whether the disk is eligible for conversion.
/convert	Instructs MBR2GPT.exe to perform the disk validation and to proceed with the conversion if all validation tests pass.
/disk:<diskNumber>	Specifies the disk number of the disk to be converted to GPT. If not specified, the system disk is used. The mechanism used is the same as that used by the diskpart.exe tool SELECT DISK SYSTEM command.
/logs:<logDirectory>	Specifies the directory where MBR2GPT.exe logs should be written. If not specified, %windir% is used. If specified, the directory must already exist, it will not be automatically created or overwritten.
/map:<source>=<destination>	Specifies additional partition type mappings between MBR and GPT. The MBR partition number is specified in decimal notation, not hexadecimal. The GPT GUID can contain brackets, for example: /map:42={af9b60a0-1431-4f62-bc68-3311714a69ad} . Multiple /map options can be specified if multiple mappings are required.
/allowFullIOS	By default, MBR2GPT.exe is blocked unless it is run from Windows PE. This option overrides this block and enables disk conversion while running in the full Windows environment. Note: Since the existing MBR system partition is in use while running the full Windows environment, it cannot be reused. In this case, a new ESP is created by shrinking the OS partition.

Examples

Validation example

In the following example, disk 0 is validated for conversion. Errors and warnings are logged to the default location, **%windir%**.

```
X:\>mbr2gpt /validate /disk:0
MBR2GPT: Attempting to validate disk 0
MBR2GPT: Retrieving layout of disk
MBR2GPT: Validating layout, disk sector size is: 512
MBR2GPT: Validation completed successfully
```

Conversion example

In the following example:

1. Using DiskPart, the current disk partition layout is displayed prior to conversion - three partitions are present on the MBR disk (disk 0): a system reserved partition, a Windows partition, and a recovery partition. A DVD-ROM is also present as volume 0.
2. The OS volume is selected, partitions are listed, and partition details are displayed for the OS partition. The **MBR partition type** is **07** corresponding to the installable file system (IFS) type.
3. The MBR2GPT tool is used to convert disk 0.
4. The DiskPart tool displays that disk 0 is now using the GPT format.
5. The new disk layout is displayed - four partitions are present on the GPT disk: three are identical to the previous partitions and one is the new EFI system partition (volume 3).
6. The OS volume is selected again, and detail displays that it has been converted to the **GPT partition type** of **ebd0a0a2-b9e5-4433-87c0-68b6b72699c7** corresponding to the **PARTITION_BASIC_DATA_GUID** type.

As noted in the output from the MBR2GPT tool, you must make changes to the computer firmware so that the new EFI system partition will boot properly.

```
X:\>DiskPart

Microsoft DiskPart version 10.0.15048.0

Copyright (C) Microsoft Corporation.
On computer: MININT-K71F13N

DISKPART> list volume

   Volume ###  Ltr  Label          Fs          Type          Size         Status       Info
   -----  ---  -----  ---          -
Volume 0      F    CENA_X64FRE  UDF         DVD-ROM       4027 MB      Healthy
Volume 1      C    System Rese  NTFS        Partition     499 MB       Healthy
Volume 2      D    Windows     NTFS        Partition     58 GB        Healthy
Volume 3      E    Recovery    NTFS        Partition     612 MB       Healthy  Hidden

DISKPART> select volume 2

Volume 2 is the selected volume.

DISKPART> list partition

   Partition ###  Type          Size         Offset
   -----  ---
   Partition 1    Primary       499 MB       1024 KB
   * Partition 2    Primary       58 GB        500 MB
   Partition 3    Recovery      612 MB        59 GB

DISKPART> detail partition

Partition 2
Type : 07
```

Hidden: No
Active: No
Offset in Bytes: 524288000

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
* Volume 2	D	Windows	NTFS	Partition	58 GB	Healthy	

DISKPART> exit

Leaving DiskPart...

X:\>mbr2gpt /convert /disk:0

MBR2GPT will now attempt to convert disk 0.
If conversion is successful the disk can only be booted in GPT mode.
These changes cannot be undone!

MBR2GPT: Attempting to convert disk 0
MBR2GPT: Retrieving layout of disk
MBR2GPT: Validating layout, disk sector size is: 512 bytes
MBR2GPT: Trying to shrink the system partition
MBR2GPT: Trying to shrink the OS partition
MBR2GPT: Creating the EFI system partition
MBR2GPT: Installing the new boot files
MBR2GPT: Performing the layout conversion
MBR2GPT: Migrating default boot entry
MBR2GPT: Adding recovery boot entry
MBR2GPT: Fixing drive letter mapping
MBR2GPT: Conversion completed successfully
MBR2GPT: Before the new system can boot properly you need to switch the firmware to boot to UEFI mode!

X:\>DiskPart

Microsoft DiskPart version 10.0.15048.0

Copyright (C) Microsoft Corporation.
On computer: MININT-K71F13N

DISKPART> list disk

Disk ###	Status	Size	Free	Dyn	Gpt
Disk 0	Online	60 GB	0 B		*

DISKPART> select disk 0

Disk 0 is now the selected disk.

DISKPART> list volume

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
Volume 0	F	CENA_X64FRE	UDF	DVD-ROM	4027 MB	Healthy	
Volume 1	D	Windows	NTFS	Partition	58 GB	Healthy	
Volume 2	C	System Rese	NTFS	Partition	499 MB	Healthy	Hidden
Volume 3			FAT32	Partition	100 MB	Healthy	Hidden
Volume 4	E	Recovery	NTFS	Partition	612 MB	Healthy	Hidden

DISKPART> select volume 1

Volume 1 is the selected volume.

DISKPART> list partition

Partition ###	Type	Size	Offset
Partition 1	Recovery	499 MB	1024 KB
* Partition 2	Primary	58 GB	500 MB

```
Partition 4  System          100 MB   59 GB
Partition 3  Recovery          612 MB   59 GB
```

```
DISKPART> detail partition
```

```
Partition 2
Type       : ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
Hidden    : No
Required  : No
Attrib    : 0000000000000000
Offset in Bytes: 524288000
```

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
* Volume 1	D	Windows	NTFS	Partition	58 GB	Healthy	

Specifications

Disk conversion workflow

The following steps illustrate high-level phases of the MBR-to-GPT conversion process:

1. Disk validation is performed.
2. The disk is repartitioned to create an EFI system partition (ESP) if one does not already exist.
3. UEFI boot files are installed to the ESP.
4. GPT metadata and layout information is applied.
5. The boot configuration data (BCD) store is updated.
6. Drive letter assignments are restored.

Creating an EFI system partition

For Windows to remain bootable after the conversion, an EFI system partition (ESP) must be in place. MBR2GPT creates the ESP using the following rules:

1. The existing MBR system partition is reused if it meets these requirements:
 - a. It is not also the OS or Windows Recovery Environment partition.
 - b. It is at least 100MB (or 260MB for 4K sector size disks) in size.
 - c. It is less than or equal to 1GB in size. This is a safety precaution to ensure it is not a data partition.
 - d. The conversion is not being performed from the full OS. In this case, the existing MBR system partition is in use and cannot be repurposed.
2. If the existing MBR system partition cannot be reused, a new ESP is created by shrinking the OS partition. This new partition has a size of 100MB (or 260MB for 4K sector size disks) and is formatted FAT32.

If the existing MBR system partition is not reused for the ESP, it is no longer used by the boot process after the conversion. Other partitions are not modified.

IMPORTANT

If the existing MBR system partition is not reused for the ESP, it might be assigned a drive letter. If you do not wish to use this small partition, you must manually hide the drive letter.

Partition type mapping and partition attributes

Since GPT partitions use a different set of type IDs than MBR partitions, each partition on the converted disk must be assigned a new type ID. The partition type mapping follows these rules:

1. The ESP is always set to partition type PARTITION_SYSTEM_GUID (c12a7328-f81f-11d2-ba4b-00a0c93ec93b).
2. If an MBR partition is of a type that matches one of the entries specified in the /map switch, the specified GPT

partition type ID is used.

3. If the MBR partition is of type 0x27, the partition is converted to a GPT partition of type PARTITION_MSFT_RECOVERY_GUID (de94bba4-06d1-4d40-a16a-bfd50179d6ac).
4. All other MBR partitions recognized by Windows are converted to GPT partitions of type PARTITION_BASIC_DATA_GUID (ebd0a0a2-b9e5-4433-87c0-68b6b72699c7).

In addition to applying the correct partition types, partitions of type PARTITION_MSFT_RECOVERY_GUID also have the following GPT attributes set:

- GPT_ATTRIBUTE_PLATFORM_REQUIRED (0x0000000000000001)
- GPT_BASIC_DATA_ATTRIBUTE_NO_DRIVE_LETTER (0x8000000000000000)

For more information about partition types, see:

- [GPT partition types](#)
- [MBR partition types](#)

Persisting drive letter assignments

The conversion tool will attempt to remap all drive letter assignment information contained in the registry that correspond to the volumes of the converted disk. If a drive letter assignment cannot be restored, an error will be displayed at the console and in the log, so that you can manually perform the correct assignment of the drive letter.

Important: this code runs after the layout conversion has taken place, so the operation cannot be undone at this stage.

The conversion tool will obtain volume unique ID data before and after the layout conversion, organizing this information into a lookup table. It will then iterate through all the entries in **HKLM\SYSTEM\MountedDevices**, and for each entry do the following:

1. Check if the unique ID corresponds to any of the unique IDs for any of the volumes that are part of the converted disk.
2. If found, set the value to be the new unique ID, obtained after the layout conversion.
3. If the new unique ID cannot be set and the value name starts with \DosDevices, issue a console and log warning about the need for manual intervention in properly restoring the drive letter assignment.

Troubleshooting

The tool will display status information in its output. Both validation and conversion are clear if any errors are encountered. For example, if one or more partitions do not translate properly, this is displayed and the conversion not performed. To view more detail about any errors that are encountered, see the associated [log files](#).

Logs

Four log files are created by the MBR2GPT tool:

- diagerr.xml
- diagwrn.xml
- setupact.log
- setuperr.log

These files contain errors and warnings encountered during disk validation and conversion. Information in these files can be helpful in diagnosing problems with the tool. The setupact.log and setuperr.log files will have the most detailed information about disk layouts, processes, and other information pertaining to disk validation and conversion. Note: The setupact*.log files are different than the Windows Setup files that are found in the %Windir%\Panther directory.

The default location for all these log files in Windows PE is **%windir%**.

Interactive help

To view a list of options available when using the tool, type **mbr2gpt /?**

The following text is displayed:

```
C:\> mbr2gpt /?
```

```
Converts a disk from MBR to GPT partitioning without modifying or deleting data on the disk.
```

```
MBR2GPT.exe /validate|convert [/disk:<diskNumber>] [/logs:<logDirectory>] [/map:<source>=<destination>]  
[/allowFullLOS]
```

Where:

/validate

- Validates that the selected disk can be converted without performing the actual conversion.

/convert

- Validates that the selected disk can be converted and performs the actual conversion.

/disk:<diskNumber>

- Specifies the disk number of the disk to be processed. If not specified, the system disk is processed.

/logs:<logDirectory>

- Specifies the directory for logging. By default logs are created in the %windir% directory.

/map:<source>=<destination>

- Specifies the GPT partition type to be used for a given MBR partition type not recognized by Windows. Multiple /map switches are allowed.

/allowFullLOS

- Allows the tool to be used from the full Windows environment. By default, this tool can only be used from the Windows Preinstallation Environment.

Return codes

MBR2GPT has the following associated return codes:

RETURN CODE	DESCRIPTION
0	Conversion completed successfully.
1	Conversion was canceled by the user.
2	Conversion failed due to an internal error.
3	Conversion failed due to an initialization error.
4	Conversion failed due to invalid command-line parameters.
5	Conversion failed due to error reading the geometry and layout of the selected disk.

RETURN CODE	DESCRIPTION
6	Conversion failed because one or more volumes on the disk is encrypted.
7	Conversion failed because the geometry and layout of the selected disk do not meet requirements.
8	Conversion failed due to error while creating the EFI system partition.
9	Conversion failed due to error installing boot files.
10	Conversion failed due to error while applying GPT layout.
100	Conversion to GPT layout succeeded, but some boot configuration data entries could not be restored.

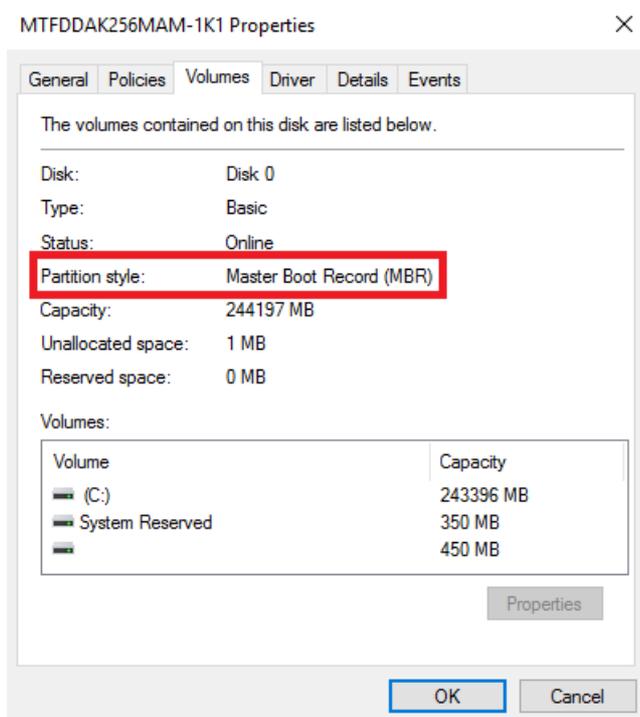
Determining the partition type

You can type the following command at a Windows PowerShell prompt to display the disk number and partition type. Example output is also shown:

```
PS C:\> Get-Disk | ft -Auto
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
0	MTFDDAK256MAM-1K1	13050928F47C	Healthy	Online	238.47 GB	MBR
1	ST1000DM003-1ER162	Z4Y3GD8F	Healthy	Online	931.51 GB	GPT

You can also view the partition type of a disk by opening the Disk Management tool, right-clicking the disk number, clicking **Properties**, and then clicking the **Volumes** tab. See the following example:



If Windows PowerShell and Disk Management are not available, such as when you are using Windows PE, you can determine the partition type at a command prompt with the DiskPart tool. To determine the partition style from a command line, type **diskpart** and then type **list disk**. See the following example:

```
X:\>DiskPart

Microsoft DiskPart version 10.0.15048.0

Copyright (C) Microsoft Corporation.
On computer: MININT-K71F13N

DISKPART> list disk

Disk ###  Status              Size               Free              Dyn  Gpt
-----  -
Disk 0    Online              238 GB             0 B
Disk 1    Online              931 GB             0 B                *
```

In this example, Disk 0 is formatted with the MBR partition style, and Disk 1 is formatted using GPT.

Related topics

[Windows 10 Enterprise system requirements](#)

[Windows 10 Specifications](#)

[Windows 10 IT pro forums](#)

Configure a PXE server to load Windows PE

6/18/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Summary

This walkthrough describes how to configure a PXE server to load Windows PE by booting a client computer from the network. Using the Windows PE tools and a Windows 10 image file, you can install Windows 10 from the network.

Prerequisites

- A deployment computer: A computer with the [Windows Assessment and Deployment Kit](#) (Windows ADK) installed.
- A DHCP server: A DHCP server or DHCP proxy configured to respond to PXE client requests is required.
- A PXE server: A server running the TFTP service that can host Windows PE boot files that the client will download.
- A file server: A server hosting a network file share.

All four of the roles specified above can be hosted on the same computer or each can be on a separate computer.

Step 1: Copy Windows PE source files

1. On the deployment computer, click **Start**, and type **deployment**.
2. Right-click **Deployment and Imaging Tools Environment** and then click **Run as administrator**. The Deployment and Imaging Tools Environment shortcut opens a Command Prompt window and automatically sets environment variables to point to all the necessary tools.
3. Run the following command to copy the base Windows PE files into a new folder. The script requires two arguments: hardware architecture and destination location. The value of **<architecture>** can be **x86**, **amd64**, or **arm** and **<destination>** is a path to a local directory. If the directory does not already exist, it will be created.

```
copype.cmd <architecture> <destination>
```

For example, the following command copies **amd64** architecture files to the **C:\winpe_amd64** directory:

```
copype.cmd amd64 C:\winpe_amd64
```

The script creates the destination directory structure and copies all the necessary files for that architecture. In the previous example, the following directories are created:

```
C:\winpe_amd64
C:\winpe_amd64\fwfiles
C:\winpe_amd64\media
C:\winpe_amd64\mount
```

4. Mount the base Windows PE image (winpe.wim) to the \mount directory using the DISM tool. Mounting an image file unpacks the file contents into a folder so that you can make changes directly or by using tools such as DISM. See the following example.

```
Dism /mount-image /imagefile:c:\winpe_amd64\media\sources\boot.wim /index:1
/mountdir:C:\winpe_amd64\mount
```

Verify that "The operation completed successfully" is displayed. Note: To view currently mounted images, type **dism /get-MountedWiminfo**.

5. Map a network share to the root TFTP directory on the PXE/TFTP server and create a \Boot folder. Consult your TFTP server documentation to determine the root TFTP server directory, then enable sharing for this directory, and verify it can be accessed on the network. In the following example, the PXE server name is PXE-1 and the TFTP root directory is shared using a network path of **\\PXE-1\TFTPRoot**:

```
net use y: \\PXE-1\TFTPRoot
y:
md boot
```

6. Copy the PXE boot files from the mounted directory to the \boot folder. For example:

```
copy c:\winpe_amd64\mount\windows\boot\pxe\*. * y:\boot
```

7. Copy the boot.sdi file to the PXE/TFTP server.

```
copy C:\winpe_amd64\media\boot\boot.sdi y:\boot
```

8. Copy the bootable Windows PE image (boot.wim) to the \boot folder.

```
copy C:\winpe_amd64\media\sources\boot.wim y:\boot
```

9. (Optional) Copy true type fonts to the \boot folder

```
copy C:\winpe_amd64\media\Boot\Fonts y:\boot\Fonts
```

Step 2: Configure boot settings and copy the BCD file

1. Create a BCD store using bcdedit.exe:

```
bcdedit /createstore c:\BCD
```

2. Configure RAMDISK settings:

```
bcdedit /store c:\BCD /create {ramdiskoptions} /d "Ramdisk options"
bcdedit /store c:\BCD /set {ramdiskoptions} ramdiskdevice boot
bcdedit /store c:\BCD /set {ramdiskoptions} ramdiskdipath \boot\boot.sdi
bcdedit /store c:\BCD /create /d "winpe boot image" /application osloader
```

The last command will return a GUID, for example:

```
The entry {a4f89c62-2142-11e6-80b6-00155da04110} was successfully created.
```

Copy this GUID for use in the next set of commands. In each command shown, replace "GUID1" with your GUID.

3. Create a new boot application entry for the Windows PE image:

```
bcdedit /store c:\BCD /set {GUID1} device ramdisk=[boot]\boot\boot.wim,{ramdiskoptions}
bcdedit /store c:\BCD /set {GUID1} path \windows\system32\winload.exe
bcdedit /store c:\BCD /set {GUID1} osdevice ramdisk=[boot]\boot\boot.wim,{ramdiskoptions}
bcdedit /store c:\BCD /set {GUID1} systemroot \windows
bcdedit /store c:\BCD /set {GUID1} detecthal Yes
bcdedit /store c:\BCD /set {GUID1} winpe Yes
```

4. Configure BOOTMGR settings (remember to replace GUID1 in the third command with your GUID):

```
bcdedit /store c:\BCD /create {bootmgr} /d "boot manager"
bcdedit /store c:\BCD /set {bootmgr} timeout 30
bcdedit /store c:\BCD -displayorder {GUID1} -addlast
```

5. Copy the BCD file to your TFTP server:

```
copy c:\BCD \\PXE-1\TFTPRoot\boot\BCD
```

Your PXE/TFTP server is now configured. You can view the BCD settings that have been configured using the command `bcdedit /store <BCD file location> /enum all`. See the following example. Note: Your GUID will be different than the one shown below.

```

C:\>bcdedit /store C:\BCD /enum all
Windows Boot Manager
-----
identifier           {bootmgr}
description          boot manager
displayorder        {a4f89c62-2142-11e6-80b6-00155da04110}
timeout              30

Windows Boot Loader
-----
identifier           {a4f89c62-2142-11e6-80b6-00155da04110}
device              ramdisk=[boot]\boot\boot.wim,{ramdiskoptions}
description          winpe boot image
osdevice            ramdisk=[boot]\boot\boot.wim,{ramdiskoptions}
systemroot          \Windows
detecthal           Yes
winpe               Yes

Setup Ramdisk Options
-----
identifier           {ramdiskoptions}
description          ramdisk options
ramdisksdidevice    boot
ramdisksdipath      \boot\boot.sdi

```

TIP

If you start the PXE boot process, but receive the error that "The boot configuration data for your PC is missing or contains errors" then verify that \boot directory is installed under the correct TFTP server root directory. In the example used here the name of this directory is TFTPRoot, but your TFTP server might be different.

PXE boot process summary

The following summarizes the PXE client boot process.

The following assumes that you have configured DHCP option 67 (Bootfile Name) to "boot\pxeboot.n12" which enables direct boot to PXE with no user interaction. For more information about DHCP options for network boot, see [Managing Network Boot Programs](#).

1. A client is directed by DHCP options 066 and 067 to download boot\pxeboot.n12 from the TFTP server.
2. pxeboot.n12 immediately begins a network boot.
3. The client downloads boot\bootmgr.exe and the boot\BCD file from the TFTP server. Note: The BCD store must reside in the \boot directory on the TFTP server and must be named BCD.
4. Bootmgr.exe reads the BCD operating system entries and downloads boot\boot.sdi and the Windows PE image (boot\boot.wim). Optional files that can also be downloaded include true type fonts (boot\Fonts\wgl4_boot.ttf) and the hibernation state file (\hiberfil.sys) if these files are present.
5. Bootmgr.exe starts Windows PE by calling winload.exe within the Windows PE image.
6. Windows PE loads, a command prompt opens and wpeinit.exe is run to initialize Windows PE.
7. The Windows PE client provides access to tools like imagex, diskpart, and bcdboot using the Windows PE command prompt. Using these tools together with a Windows 10 image file, the destination computer can be formatted properly to load a full Windows 10 operating system.

See Also

Concepts

[Windows PE Walkthroughs](#)

Windows ADK for Windows 10 scenarios for IT Pros

6/18/2019 • 2 minutes to read • [Edit Online](#)

The [Windows Assessment and Deployment Kit \(Windows ADK\)](#) contains tools that can be used by IT Pros to deploy Windows. For an overview of what's new in the Windows ADK for Windows 10, see [What's new in kits and tools](#).

In previous releases of Windows, the Windows ADK docs were published on both TechNet and the MSDN Hardware Dev Center. Starting with the Windows 10 release, Windows ADK documentation is available on the MSDN Hardware Dev Center. For the Windows 10 ADK reference content, see [Desktop manufacturing](#).

Here are some key scenarios that will help you find the content on the MSDN Hardware Dev Center.

Create a Windows image using command-line tools

[DISM](#) is used to mount and service Windows images.

Here are some things you can do with DISM:

- [Mount an offline image](#)
- [Add drivers to an offline image](#)
- [Enable or disable Windows features](#)
- [Add or remove packages](#)
- [Add language packs](#)
- [Add Universal Windows apps](#)
- [Upgrade the Windows edition](#)

[Sysprep](#) prepares a Windows installation for imaging and allows you to capture a customized installation.

Here are some things you can do with Sysprep:

- [Generalize a Windows installation](#)
- [Customize the default user profile](#)
- [Use answer files](#)

[Windows PE \(WinPE\)](#) is a small operating system used to boot a computer that does not have an operating system. You can boot to Windows PE and then install a new operating system, recover data, or repair an existing operating system.

Here are ways you can create a WinPE image:

- [Create a bootable USB drive](#)
- [Create a Boot CD, DVD, ISO, or VHD](#)

[Windows Recovery Environment \(Windows RE\)](#) is a recovery environment that can repair common operating system problems.

Here are some things you can do with Windows RE:

- [Customize Windows RE](#)
- [Push-button reset](#)

[Windows System Image Manager \(Windows SIM\)](#) helps you create answer files that change Windows settings and run scripts during installation.

Here are some things you can do with Windows SIM:

- [Create answer file](#)
- [Add a driver path to an answer file](#)
- [Add a package to an answer file](#)
- [Add a custom command to an answer file](#)

For a list of settings you can change, see [Unattended Windows Setup Reference](#) on the MSDN Hardware Dev Center.

Create a Windows image using Windows ICD

Introduced in Windows 10, [Windows Imaging and Configuration Designer \(ICD\)](#) streamlines the customizing and provisioning of a Windows 10 for desktop editions (Home, Pro, Enterprise, and Education), Windows 10 Mobile, or Windows 10 IoT Core (IoT Core) image.

Here are some things you can do with Windows ICD:

- [Build and apply a provisioning package](#)
- [Export a provisioning package](#)
- [Build and deploy an image for Windows 10 for desktop editions](#)

IT Pro Windows deployment tools

There are also a few tools included in the Windows ADK that are specific to IT Pros and this documentation is available on TechNet:

- [Volume Activation Management Tool \(VAMT\) Technical Reference](#)
- [User State Migration Tool \(USMT\) Technical Reference](#)

Deploy Windows To Go in your organization

6/18/2019 • 37 minutes to read • [Edit Online](#)

Applies to

- Windows 10

This topic helps you to deploy Windows To Go in your organization. Before you begin deployment, make sure that you have reviewed the topics [Windows To Go: feature overview](#) and [Prepare your organization for Windows To Go](#) to ensure that you have the correct hardware and are prepared to complete the deployment. You can then use the steps in this topic to start your Windows To Go deployment.

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

Deployment tips

The following is a list of items that you should be aware of before you start the deployment process:

- Only use recommended USB drives for Windows To Go. Use of other drives is not supported. Check the list at [Windows To Go: feature overview](#) for the latest USB drives certified for use as Windows To Go drives.
- After you provision a new workspace, always eject a Windows To Go drive using the **Safely Remove Hardware and Eject Media** control that can be found in the notification area or in Windows Explorer. Removing the drive from the USB port without ejecting it first can cause the drive to become corrupted.
- When running a Windows To Go workspace, always shutdown the workspace before unplugging the drive.
- System Center 2012 Configuration Manager SP1 and later includes support for user self-provisioning of Windows To Go drives. You can download Configuration Manager for evaluation from the [Microsoft TechNet Evaluation Center](#). For more information on this deployment option, see [How to Provision Windows To Go in Configuration Manager](#).
- If you are planning on using a USB drive duplicator to duplicate Windows To Go drives, do not configure offline domain join or BitLocker on the drive.

Basic deployment steps

Unless you are using a customized operating system image, your initial Windows To Go workspace will not be domain joined and will not contain applications. This is exactly like a new installation of Windows on a desktop or laptop computer. When planning your deployment, you should develop methods to join Windows to Go drives to the domain and install the standard applications that users in your organization require. These methods probably will be similar to the ones used for setting up desktop and laptop computers with domain privileges and applications. This section describes the instructions for creating the correct disk layout on the USB drive, applying the operating system image and the core Windows To Go specific configurations to the drive. The following steps are used in both small-scale and large-scale Windows To Go deployment scenarios.

Completing these steps will give you a generic Windows To Go drive that can be distributed to your users and then customized for their usage as needed. This drive is also appropriate for use with USB drive duplicators. Your

specific deployment scenarios will involve more than just these basic steps but these additional deployment considerations are similar to traditional PC deployment and can be incorporated into your Windows To Go deployment plan. For additional information, see [Windows Deployment Options](#).

WARNING

If you plan to use the generic Windows To Go drive as the master drive in a USB duplicator, the drive should not be booted. If the drive has been booted inadvertently it should be reprovisioned prior to duplication.

Create the Windows To Go workspace

In this step we are creating the operating system image that will be used on the Windows To Go drives. You can use the Windows To Go Creator Wizard or you can [do this manually](#) using a combination of Windows PowerShell and command-line tools.

WARNING

The preferred method to create a single Windows To Go drive is to use the Windows To Go Creator Wizard included in Windows 10 Enterprise and Windows 10 Education.

To create a Windows To Go workspace with the Windows To Go Creator Wizard

1. Sign into your Windows PC using an account with Administrator privileges.
2. Insert the USB drive that you want to use as your Windows To Go drive into your PC.
3. Verify that the .wim file location (which can be a network share, a DVD , or a USB drive) is accessible and that it contains a valid Windows 10 Enterprise or Windows 10 Education image that has been generalized using sysprep. Many environments can use the same image for both Windows To Go and desktop deployments.

NOTE

For more information about .wim files, see [Windows System Image Manager \(Windows SIM\) Technical Reference](#). For more information about using sysprep, see [Sysprep Overview](#).

4. Using Cortana, search for **Windows To Go** and then press **Enter**. If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then click **Yes**. The **Windows To Go Creator Wizard** opens.
5. On the **Choose the drive you want to use** page select the drive that represents the USB drive you inserted previously, then click **Next**.
6. On the **Choose a Windows image** page, click **Add Search Location** and then navigate to the .wim file location and click select folder. The wizard will display the installable images present in the folder; select the Windows 10 Enterprise or Windows 10 Education image you wish to use and then click **Next**.
7. (Optional) On the **Set a BitLocker password (optional)** page, you can select **Use BitLocker with my Windows To Go Workspace** to encrypt your Windows To Go drive. If you do not wish to encrypt the drive at this time, click **Skip**. If you decide you want to add BitLocker protection later, see [Enable BitLocker protection for your Windows To Go drive](#) for instructions. r

WARNING

If you plan to use a USB-Duplicator to create multiple Windows To Go drives, do not enable BitLocker. Drives protected with BitLocker should not be duplicated.

If you choose to encrypt the Windows To Go drive now:

- Type a password that is at least eight characters long and conforms to your organizations password complexity policy. This password will be provided before the operating system is started so any characters you use must be able to be interpreted by the firmware. Some firmware does not support non-ASCII characters.

>[!IMPORTANT]

>The BitLocker recovery password will be saved in the documents library of the computer used to create the workspace automatically. If your organization is using Active Directory Domain Services (AD DS) to store recovery passwords it will also be saved in AD DS under the computer account of the computer used to create the workspace. This password will be used only if you need to recover access to the drive because the BitLocker password specified in the previous step is not available, such as if a password is lost or forgotten. For more information about BitLocker and AD DS, see [Active Directory Domain Services considerations](https://go.microsoft.com/fwlink/p/?LinkId=619157).

8. Verify that the USB drive inserted is the one you want to provision for Windows To Go and then click **Create** to start the Windows To Go workspace creation process.

WARNING

The USB drive identified will be reformatted as part of the Windows To Go provisioning process and any data on the drive will be erased.

9. Wait for the creation process to complete, which can take 20 to 30 minutes. A completion page will be displayed that tells you when your Windows To Go workspace is ready to use. From the completion page you can configure the Windows To Go startup options to configure the current computer as a Windows To Go host computer.

Your Windows To Go workspace is now ready to be started. You can now [prepare a host computer](#) using the Windows To Go startup options and boot your Windows To Go drive.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. This procedure can only be used on PCs that are running Windows 10. Before starting, ensure that only the USB drive that you want to provision as a Windows To Go drive is connected to the PC.

1. Using Cortana, search for **powershell**, right-click **Windows PowerShell**, and then select **Run as administrator**.
2. In the Windows PowerShell session type the following commands to partition a master boot record (MBR) disk for use with a FAT32 system partition and an NTFS-formatted operating system partition. This disk layout can support computers that use either UEFI or BIOS firmware:

```

# The following command will set $Disk to all USB drives with >20 GB of storage

$Disk = Get-Disk | Where-Object {$_.Path -match "USBSTOR" -and $_.Size -gt 20Gb -and -not $_.IsBoot }

#Clear the disk. This will delete any data on the disk. (and will fail if the disk is not yet
initialized. If that happens, simply continue with 'New-Partition...') Validate that this is the correct
disk that you want to completely erase.
#
# To skip the confirmation prompt, append -confirm:$False
Clear-Disk -InputObject $Disk[0] -RemoveData

# This command initializes a new MBR disk
Initialize-Disk -InputObject $Disk[0] -PartitionStyle MBR

# This command creates a 350 MB system partition
$SystemPartition = New-Partition -InputObject $Disk[0] -Size (350MB) -IsActive

# This formats the volume with a FAT32 Filesystem
# To skip the confirmation dialog, append -Confirm:$False
Format-Volume -NewFileSystemLabel "UFD-System" -FileSystem FAT32 `
-Partition $SystemPartition

# This command creates the Windows volume using the maximum space available on the drive. The Windows
To Go drive should not be used for other file storage.
$OSPartition = New-Partition -InputObject $Disk[0] -UseMaximumSize
Format-Volume -NewFileSystemLabel "UFD-Windows" -FileSystem NTFS `
-Partition $OSPartition

# This command assigns drive letters to the new drive, the drive letters chosen should not already be
in use.
Set-Partition -InputObject $SystemPartition -NewDriveLetter "S"
Set-Partition -InputObject $OSPartition -NewDriveLetter "W"

# This command sets the NODEFAULTDRIVELETTER flag on the partition which prevents drive letters being
assigned to either partition when inserted into a different computer.
Set-Partition -InputObject $OSPartition -NoDefaultDriveLetter $TRUE

```

- Next you need to apply the operating system image that you want to use with Windows To Go to the operating system partition you just created on the disk (this may take 30 minutes or longer, depending on the size of the image and the speed of your USB connection). The following command shows how this can be accomplished using the [Deployment Image Servicing and Management](#) command-line tool (DISM):

TIP

The index number must be set correctly to a valid Enterprise image in the .WIM file.

```

#The WIM file must contain a sysprep generalized image.
dism /apply-image /imagefile:n:\imagefolder\deploymentimages\mywtgimage.wim /index:1 /applydir:W:\

```

- Now use the `bcdboot` command line tool to move the necessary boot components to the system partition on the disk. This helps ensure that the boot components, operating system versions, and architectures match. The `/f ALL` parameter indicates that boot components for UEFI and BIOS should be placed on the system partition of the disk. The following example illustrates this step:

```

``` syntax
W:\Windows\System32\bcdboot W:\Windows /f ALL /s S:
```

```

- Apply SAN policy—`OFFLINE_INTERNAL - "4"` to prevent the operating system from automatically

bringing online any internally connected disk. This is done by creating and saving a **san_policy.xml** file on the disk. The following example illustrates this step:

```
<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="offlineServicing">
    <component
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      language="neutral"
      name="Microsoft-Windows-PartitionManager"
      processorArchitecture="x86"
      publicKeyToken="31bf3856ad364e35"
      versionScope="nonSxS"
    >
      <SanPolicy>4</SanPolicy>
    </component>
  <component
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      language="neutral"
      name="Microsoft-Windows-PartitionManager"
      processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35"
      versionScope="nonSxS"
    >
      <SanPolicy>4</SanPolicy>
    </component>
  </settings>
</unattend>
```

6. Place the **san_policy.xml** file created in the previous step into the root directory of the Windows partition on the Windows To Go drive (W: from the previous examples) and run the following command:

```
Dism.exe /Image:W:\ /Apply-Unattend:W:\san_policy.xml
```

7. Create an answer file (unattend.xml) that disables the use of Windows Recovery Environment with Windows To Go. You can use the following code sample to create a new answer file or you can paste it into an existing answer file:

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="oobeSystem">
    <component name="Microsoft-Windows-WinRE-RecoveryAgent"
      processorArchitecture="x86"
      publicKeyToken="31bf3856ad364e35" language="neutral"
      versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <UninstallWindowsRE>true</UninstallWindowsRE>
    </component>
    <component name="Microsoft-Windows-WinRE-RecoveryAgent"
      processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral"
      versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <UninstallWindowsRE>true</UninstallWindowsRE>
    </component>
  </settings>
</unattend>
```

After the answer file has been saved, copy unattend.xml into the sysprep folder on the Windows To Go drive (for example, W:\Windows\System32\sysprep)

IMPORTANT

Setup unattend files are processed based on their location. Setup will place a temporary unattend file into the **%systemroot%\panther** folder which is the first location that setup will check for installation information. You should make sure that folder does not contain a previous version of an unattend.xml file to ensure that the one you just created is used.

If you do not wish to boot your Windows To Go device on this computer and want to remove it to boot it on another PC, be sure to use the **Safely Remove Hardware and Eject Media** option to safely disconnect the drive before physically removing it from the PC.

Your Windows To Go workspace is now ready to be started. You can now [prepare a host computer](#) using the Windows To Go startup options to test your workspace configuration, [configure the workspace for offline domain join](#), or [enable BitLocker protection for your Windows To Go drive](#).

To prepare a host computer

Computers running Windows 8 and later can be configured as host computers that use Windows To Go automatically whenever a Windows To Go workspace is available at startup. When the Windows To Go startup options are enabled on a host computer, Windows will divert startup to the Windows To Go drive whenever it is attached to the computer. This makes it easy to switch from using the host computer to using the Windows To Go workspace.

TIP

If you will be using a PC running Windows 7 as your host computer, see [Tips for configuring your BIOS settings to work with Windows To Go](#) for information to help you prepare the host computer.

If you want to use the Windows To Go workspace, simply shut down the computer, plug in the Windows To Go drive, and turn on the computer. To use the host computer, shut down the Windows To Go workspace, unplug the Windows To Go drive, and turn on the computer.

To set the Windows To Go Startup options for host computers running Windows 10:

1. Using Cortana, search for **Windows To Go startup options** and then press **Enter**.
2. In the **Windows To Go Startup Options** dialog box, select **Yes**, and then click **Save Changes** to configure the computer to boot from USB

For host computers running Windows 8 or Windows 8.1:

1. Press **Windows logo key+W**, search for **Windows To Go startup options**, and then press **Enter**.
2. In the **Windows To Go Startup Options** dialog box, select **Yes**, and then click **Save Changes** to configure the computer to boot from USB.

You can configure your organization's computers to automatically start from the USB drive by enabling the following Group Policy setting:

\\Computer Configuration\Administrative Templates\Windows Components\Portable Operating System\Windows To Go Default Startup Options

After this policy setting is enabled, automatic starting of a Windows To Go workspace will be attempted when a USB drive is connected to the computer when it is started. Users will not be able to use the Windows To Go

Startup Options to change this behavior. If you disable this policy setting, booting to Windows To Go when a USB drive is connected will not occur unless a user configures the option manually in the firmware. If you do not configure this policy setting, users who are members of the Administrators group can enable or disable booting from a USB drive using the Windows To Go Startup Options.

Your host computer is now ready to boot directly into Windows To Go workspace when it is inserted prior to starting the computer. Optionally you can perform [Configure Windows To Go workspace for offline domain join](#) and [Enable BitLocker protection for your Windows To Go drive](#).

Booting your Windows To Go workspace

After you have configured your host PC to boot from USB, you can use the following procedure to boot your Windows To Go workspace:

To boot your workspace

1. Make sure that the host PC is not in a sleep state. If the computer is in a sleep state, either shut it down or hibernate it.
2. Insert the Windows To Go USB drive directly into a USB 3.0 or USB 2.0 port on the PC. Do not use a USB hub or extender.
3. Turn on the PC. If your Windows To Go drive is protected with BitLocker you will be asked to type the password, otherwise the workspace will boot directly into the Windows To Go workspace.

Advanced deployment steps

The following steps are used for more advanced deployments where you want to have further control over the configuration of the Windows To Go drives, ensure that they are correctly configured for remote access to your organizational resources, and have been protected with BitLocker Drive Encryption.

Configure Windows To Go workspace for remote access

Making sure that Windows To Go workspaces are effective when used off premises is essential to a successful deployment. One of the key benefits of Windows To Go is the ability for your users to use the enterprise managed domain joined workspace on an unmanaged computer which is outside your corporate network. To enable this usage, typically you would provision the USB drive as described in the basic deployment instructions and then add the configuration to support domain joining of the workspace, installation of any line-of-business applications, and configuration of your chosen remote connectivity solution such as a virtual private network client or DirectAccess. Once these configurations have been performed the user can work from the workspace using a computer that is off-premises. The following procedure allows you to provision domain joined Windows To Go workspaces for workers that do not have physical access to your corporate network.

Prerequisites for remote access scenario

- A domain-joined computer running Windows 8 or later and is configured as a Windows To Go host computer
- A Windows To Go drive that hasn't been booted or joined to the domain using unattend settings.
- A domain user account with rights to add computer accounts to the domain and is a member of the Administrator group on the Windows To Go host computer
- [DirectAccess](#) configured on the domain

To configure your Windows To Go workspace for remote access

1. Start the host computer and sign in using a user account with privileges to add workstations to the domain and then run the following command from an elevated command prompt replacing the example placeholder parameters (denoted by <>) with the ones applicable for your environment:

```
djoin /provision /domain <exempldomain.com> /machine <examplewindowstogo_workspace_name> /certtemplate <WorkstationAuthentication_template> /policynames <DirectAccess Client Policy: {GUID}> /savefile <C:\example\path\domainmetadata\> /reuse
```

NOTE

The **/certtemplate** parameter supports the use of certificate templates for distributing certificates for DirectAccess, if your organization is not using certificate templates you can omit this parameter. Additionally, if are using djoin.exe with Windows Server 2008-based Domain Controllers, append the /downlevel switch during provisioning. For more information see the [Offline Domain Join Step-by-Step guide](#).

2. Insert the Windows To Go drive.
3. Launch an elevated Windows PowerShell prompt by right-clicking the Windows PowerShell shortcut in the taskbar, and then clicking **Run as Administrator**.
4. From the Windows PowerShell command prompt run:

```
# The following command will set $Disk to all USB drives with >20 GB of storage

$Disk = Get-Disk | Where-Object { $_.Path -match "USBSTOR" -and $_.Size -gt 20Gb -and -not $_.IsBoot }

#Clear the disk. This will delete any data on the disk. (and will fail if the disk is not yet
initialized. If that happens, simply continue with 'New-Partition...') Validate that this is the correct
disk that you want to completely erase.
#
# To skip the confirmation prompt, append -confirm:$False
Clear-Disk -InputObject $Disk[0] -RemoveData

# This command initializes a new MBR disk
Initialize-Disk -InputObject $Disk[0] -PartitionStyle MBR

# This command creates a 350 MB system partition
$SystemPartition = New-Partition -InputObject $Disk[0] -Size (350MB) -IsActive

# This formats the volume with a FAT32 Filesystem
# To skip the confirmation dialog, append -Confirm:$False
Format-Volume -NewFileSystemLabel "UFD-System" -FileSystem FAT32 `
-Partition $SystemPartition

# This command creates the Windows volume using the maximum space available on the drive. The Windows
To Go drive should not be used for other file storage.
$OSPartition = New-Partition -InputObject $Disk[0] -UseMaximumSize
Format-Volume -NewFileSystemLabel "UFD-Windows" -FileSystem NTFS `
-Partition $OSPartition

# This command assigns drive letters to the new drive, the drive letters chosen should not already be
in use.
Set-Partition -InputObject $SystemPartition -NewDriveLetter "S"
Set-Partition -InputObject $OSPartition -NewDriveLetter "W"

# This command toggles the NODEFAULTDRIVELETTER flag on the partition which prevents drive letters
being assigned to either partition when inserted into a different computer.
Set-Partition -InputObject $OSPartition -NoDefaultDriveLetter $TRUE
```

5. Next you need to apply the operating system image that you want to use with Windows To Go to the operating system partition you just created on the disk (this may take 30 minutes or longer, depending on the size of the image and the speed of your USB connection). The following command shows how this can be accomplished using the [Deployment Image Servicing and Management](#) command-line tool (DISM):

```
>[!TIP]
>The index number must be set correctly to a valid Enterprise image in the .WIM file.

``` syntax
#The WIM file must contain a sysprep generalized image.
dism /apply-image /imagefile:n:\imagefolder\deploymentimages\mywtgimage.wim /index:1 /applydir:W:\
```
```

6. After those commands have completed, run the following command:

```
djoin /requestodj /loadfile C:\example\path\domainmetadadatafile /windowspath W:\Windows
```

7. Next, we will need to edit the unattend.xml file to configure the first run (OOBE) settings. In this example we are hiding the Microsoft Software License Terms (EULA) page, configuring automatic updates to install important and recommended updates automatically, and identifying this workspace as part of a private office network. You can use other OOBE settings that you have configured for your organization if desired. For more information about the OOBE settings, see [OOBE](#):

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="oobeSystem">
    <component name="Microsoft-Windows-WinRE-RecoveryAgent"
      processorArchitecture="x86"
      publicKeyToken="31bf3856ad364e35" language="neutral"
      versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <UninstallWindowsRE>true</UninstallWindowsRE>
      <OOBE>
        <HideEULAPage>true</HideEULAPage>
        <ProtectYourPC>1</ProtectYourPC>
        <NetworkLocation>Work</NetworkLocation>
      </OOBE>
    </component>
    <component name="Microsoft-Windows-WinRE-RecoveryAgent"
      processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral"
      versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <UninstallWindowsRE>true</UninstallWindowsRE>
      <OOBE>
        <HideEULAPage>true</HideEULAPage>
        <ProtectYourPC>1</ProtectYourPC>
        <NetworkLocation>Work</NetworkLocation>
      </OOBE>
    </component>
  </settings>
</unattend>
```

8. Safely remove the Windows To Go drive.

9. From a host computer, either on or off premises, start the computer and boot the Windows To Go workspace.

- If on premises using a host computer with a direct network connection, sign on using your domain credentials.
- If off premises, join a wired or wireless network with internet access and then sign on again using your domain credentials.

NOTE

Depending on your DirectAccess configuration you might be asked to insert your smart card to log on to the domain.

You should now be able to access your organization's network resources and work from your Windows To Go workspace as you would normally work from your standard desktop computer on premises.

Enable BitLocker protection for your Windows To Go drive

Enabling BitLocker on your Windows To Go drive will help ensure that your data is protected from unauthorized use and that if your Windows To Go drive is lost or stolen it will not be easy for an unauthorized person to obtain confidential data or use the workspace to gain access to protected resources in your organization. When BitLocker is enabled, each time you boot your Windows To Go drive, you will be asked to provide the BitLocker password to unlock the drive. The following procedure provides the steps for enabling BitLocker on your Windows To Go drive:

Prerequisites for enabling BitLocker scenario

- A Windows To Go drive that can be successfully provisioned.
- A computer running Windows 8 configured as a Windows To Go host computer
- Review the following Group Policy settings for BitLocker Drive Encryption and modify the configuration as necessary:

\Windows Components\BitLocker Drive Encryption\Operating System Drives\Require additional authentication at startup. This policy allows the use of a password key protector with an operating system drive; this policy must be enabled to configure BitLocker from within the Windows To Go workspace. This policy setting allows you to configure whether BitLocker requires additional authentication each time the computer starts and whether you are using BitLocker with or without a Trusted Platform Module (TPM). You must enable this setting and select the **Allow BitLocker without a compatible TPM** check box and then enable the **Configure use of passwords for operating system drives** setting.

\Windows Components\BitLocker Drive Encryption\Operating System Drives\Configure use of passwords for operating system drives. This policy setting enables passwords to be used to unlock BitLocker-protected operating system drives and provides the means to configure complexity and length requirements on passwords for Windows To Go workspaces. For the complexity requirement setting to be effective the Group Policy setting **Password must meet complexity requirements** located in **Computer Configuration\Windows Settings\Security Settings\Account Policies>Password Policy** must be also enabled.

\Windows Components\BitLocker Drive Encryption\Operating System Drives\Enable use of BitLocker authentication requiring preboot keyboard input on slates. This policy setting allows users to enable authentication options that require user input from the preboot environment even if the platform indicates a lack of preboot input capability. If this setting is not enabled, passwords cannot be used to unlock BitLocker-protected operating system drives.

You can choose to enable BitLocker protection on Windows To Go drives before distributing them to users as part of your provisioning process or you can allow your end-users to apply BitLocker protection to them after they have taken possession of the drive. A step-by-step procedure is provided for both scenarios.

Enabling BitLocker during provisioning ensures that your operating system image is always protected by BitLocker. When enabling BitLocker during the provisioning process you can significantly reduce the time required for encrypting the drive by enabling BitLocker after configuring the disk and just prior to applying the image. If you use this method, you will need to give users their BitLocker password when you give them their Windows To Go workspace. Also, you should instruct your users to boot their workspace and change their

BitLocker password as soon as possible (this can be done with standard user privileges).

Enabling BitLocker after distribution requires that your users turn on BitLocker. This means that your Windows To Go workspaces are unprotected until the user enables BitLocker. Administrative rights on the Windows To Go workspace are required to enable BitLocker. For more information about BitLocker see the [BitLocker Overview](#).

BitLocker recovery keys

BitLocker recovery keys are the keys that can be used to unlock a BitLocker protected drive if the standard unlock method fails. It is recommended that your BitLocker recovery keys be backed up to Active Directory Domain Services (AD DS). If you do not want to use AD DS to store recovery keys you can save recovery keys to a file or print them. How BitLocker recovery keys are managed differs depending on when BitLocker is enabled.

- If BitLocker protection is enabled during provisioning, the BitLocker recovery keys will be stored under the computer account of the computer used for provisioning the drives. If backing up recovery keys to AD DS is not used, the recovery keys will need to be printed or saved to a file for each drive. The IT administrator must track which keys were assigned to which Windows To Go drive.
- **Warning**
If BitLocker is enabled after distribution, the recovery key will be backed up to AD DS under the computer account of the workspace. If backing up recovery keys to AD DS is not used, they can be printed or saved to a file by the user. If the IT administrator wants a central record of recovery keys, a process by which the user provides the key to the IT department must be put in place.

To enable BitLocker during provisioning

1. Start the host computer that is running Windows 8.
2. Insert your Windows To Go drive.
3. Launch an elevated Windows PowerShell prompt by right-clicking the Windows PowerShell shortcut in the taskbar, and then clicking **Run as Administrator**.
4. Provision the Windows To Go drive using the following cmdlets:

NOTE

If you used the [manual method for creating a workspace](#) you should have already provisioned the Windows To Go drive. If so, you can continue on to the next step.

```

# The following command will set $Disk to all USB drives with >20 GB of storage

$Disk = Get-Disk | Where-Object {$_.Path -match "USBSTOR" -and $_.Size -gt 20Gb -and -not $_.IsBoot }

#Clear the disk. This will delete any data on the disk. (and will fail if the disk is not yet
initialized. If that happens, simply continue with 'New-Partition...') Validate that this is the correct
disk that you want to completely erase.
#
# To skip the confirmation prompt, append -confirm:$False
Clear-Disk -InputObject $Disk[0] -RemoveData

# This command initializes a new MBR disk
Initialize-Disk -InputObject $Disk[0] -PartitionStyle MBR

# This command creates a 350 MB system partition
$SystemPartition = New-Partition -InputObject $Disk[0] -Size (350MB) -IsActive

# This formats the volume with a FAT32 Filesystem
# To skip the confirmation dialog, append -Confirm:$False
Format-Volume -NewFileSystemLabel "UFD-System" -FileSystem FAT32 `
-Partition $SystemPartition

# This command creates the Windows volume using the maximum space available on the drive. The Windows
To Go drive should not be used for other file storage.
$OSPartition = New-Partition -InputObject $Disk[0] -UseMaximumSize
Format-Volume -NewFileSystemLabel "UFD-Windows" -FileSystem NTFS `
-Partition $OSPartition

# This command assigns drive letters to the new drive, the drive letters chosen should not already be
in use.
Set-Partition -InputObject $SystemPartition -NewDriveLetter "S"
Set-Partition -InputObject $OSPartition -NewDriveLetter "W"

# This command toggles the NODEFAULTDRIVELETTER flag on the partition which prevents drive letters
being assigned to either partition when inserted into a different computer.
Set-Partition -InputObject $OSPartition -NoDefaultDriveLetter $TRUE

```

Next you need to apply the operating system image that you want to use with Windows To Go to the operating system partition you just created on the disk (this may take 30 minutes or longer, depending on the size of the image and the speed of your USB connection). The following command shows how this can be accomplished using the [Deployment Image Servicing and Management](#) command-line tool (DISM):

TIP

The index number must be set correctly to a valid Enterprise image in the .WIM file.

```

#The WIM file must contain a sysprep generalized image.
dism /apply-image /imagefile:n:\imagefolder\deploymentimages\mywtgimage.wim /index:1 /applydir:W:\

```

5. In the same PowerShell session use the following cmdlet to add a recovery key to the drive:

```

$BitlockerRecoveryProtector = Add-BitLockerKeyProtector W: -RecoveryPasswordProtector

```

6. Next, use the following cmdlets to save the recovery key to a file:

```
#The BitLocker Recovery key is essential if for some reason you forget the BitLocker password
#This recovery key can also be backed up into Active Directory using manage-bde.exe or the
#PowerShell cmdlet Backup-BitLockerKeyProtector.
$RecoveryPassword = $BitlockerRecoveryProtector.KeyProtector.RecoveryPassword
$RecoveryPassword > WTG-Demo_Bitlocker_Recovery_Password.txt
```

7. Then, use the following cmdlets to add the password as a secure string. If you omit the password the cmdlet will prompt you for the password before continuing the operation:

```
# Create a variable to store the password
$spwd = ConvertTo-SecureString -String <password> -AsPlainText -Force
Enable-BitLocker W: -PasswordProtector $spwd
```

WARNING

To have BitLocker only encrypt used space on the disk append the parameter `-UsedSpaceOnly` to the `Enable-BitLocker` cmdlet. As data is added to the drive BitLocker will encrypt additional space. Using this parameter will speed up the preparation process as a smaller percentage of the disk will require encryption. If you are in a time critical situation where you cannot wait for encryption to complete you can also safely remove the Windows To Go drive during the encryption process. The next time the drive is inserted in a computer it will request the BitLocker password. Once the password is supplied, the encryption process will continue. If you do this, make sure your users know that BitLocker encryption is still in process and that they will be able to use the workspace while the encryption completes in the background.

8. Copy the numerical recovery password and save it to a file in a safe location. The recovery password will be required if the password is lost or forgotten.

WARNING

If the **Choose how BitLocker-protected removable data drives can be recovered** Group Policy setting has been configured to back up recovery information to Active Directory Domain Services, the recovery information for the drive will be stored under the account of the host computer used to apply the recovery key.

If you want to have the recovery information stored under the account of the Windows To Go workspace you can turn BitLocker from within the Windows To Go workspace using the BitLocker Setup Wizard from the BitLocker Control Panel item as described in [To enable BitLocker after distribution](#).

9. Safely remove the Windows To Go drive.

The Windows To Go drives are now ready to be distributed to users and are protected by BitLocker. When you distribute the drives, make sure the users know the following:

- Initial BitLocker password that they will need to boot the drives.
- Current encryption status.
- Instructions to change the BitLocker password after the initial boot.
- Instructions for how to retrieve the recovery password if necessary. This may be a help desk process, an automated password retrieval site, or a person to contact.

To enable BitLocker after distribution

1. Insert your Windows To Go drive into your host computer (that is currently shut down) and then turn on the computer and boot into your Windows To Go workspace

2. Press **Windows logo key+W** to open **Search Settings**, type BitLocker and then select the item for BitLocker Drive Encryption.
3. The drives on the workspace are displayed, click **Turn BitLocker On** for the C: drive. The **BitLocker Setup Wizard** appears.
4. Complete the steps in the **BitLocker Setup Wizard** selecting the password protection option.

NOTE

If you have not configured the Group Policy setting **\Windows Components\BitLocker Drive Encryption\Operating System Drives\Require additional authentication at startup** to specify **Allow BitLocker without a compatible TPM** you will not be able to enable BitLocker from within the Windows To Go workspace.

Advanced deployment sample script

The following sample script supports the provisioning of multiple Windows To Go drives and the configuration of offline domain join.

The sample script creates an unattend file that streamlines the deployment process so that the initial use of the Windows To Go drive does not prompt the end user for any additional configuration information before starting up.

Prerequisites for running the advanced deployment sample script

- To run this sample script you must open a Windows PowerShell session as an administrator from a domain-joined computer using an account that has permission to create domain accounts.
- Using offline domain join is required by this script, since the script does not create a local administrator user account. However, domain membership will automatically put "Domain admins" into the local administrators group. Review your domain policies. If you are using DirectAccess you will need to modify the djoin.exe command to include the `polycynames` and potentially the `certtemplate` parameters.
- The script needs to use drive letters, so you can only provision half as many drives as you have free drive letters.

To run the advanced deployment sample script

1. Copy entire the code sample titled "Windows To Go multiple drive provisioning sample script" into a PowerShell script (.ps1) file.
2. Make the modifications necessary for it to be appropriate to your deployment and save the file.
3. Configure the PowerShell execution policy. By default PowerShell's execution policy is set to Restricted; that means that scripts won't run until you have explicitly given them permission to. To configure PowerShell's execution policy to allow the script to run, use the following command from an elevated PowerShell prompt:

```
Set-ExecutionPolicy RemoteSigned
```

The RemoteSigned execution policy will prevent unsigned scripts from the internet from running on the computer, but will allow locally created scripts to run. For more information on execution policies, see [Set-ExecutionPolicy](#).

TIP

To get online help for any Windows PowerShell cmdlet, whether or not it is installed locally type the following cmdlet, replacing <cmdlet-name> with the name of the cmdlet you want to see the help for:

```
Get-Help <cmdlet-name> -Online
```

This command causes Windows PowerShell to open the online version of the help topic in your default Internet browser.

Windows To Go multiple drive provisioning sample script

```
<#
.SYNOPSIS
Windows To Go multiple drive provisioning sample script.

.DESCRIPTION
This sample script will provision one or more Windows To Go drives, configure offline domain join (using
random machine names) and provides an option for BitLocker encryption. To provide a seamless first boot
experience, an unattend file is created that will set the first run (OOBE) settings to defaults. To improve
performance of the script, copy your install image to a local location on the computer used for provisioning
the drives.

.EXAMPLE
.\WTG_MultiProvision.ps1 -InstallWIMPath c:\companyImages\amd64_enterprise.wim
provision drives connected to your machine with the provided image.
#>
param (
    [parameter(Mandatory=$true)]
    [string]
#Path to install wim. If you have the full path to the wim or want to use a local file.
    $InstallWIMPath,

    [string]
#Domain to which to join the Windows To Go workspaces.
    $DomainName
)

<#
    In order to set BitLocker Group Policies for our offline WTG image we need to create a Registry.pol file
    in the System32\GroupPolicy folder. This file requires binary editing, which is not possible in PowerShell
    directly so we have some C# code that we can use to add a type in our PowerShell instance that will write
    the data for us.
#>
$Source = @"
using System;
using System.Collections.Generic;
using System.IO;
using System.Text;

namespace MS.PolicyFileEditor
{
    //The PolicyEntry represents the DWORD Registry Key/Value/Data entry that will
    //be written into the file.
    public class PolicyEntry
    {
        private List<byte> byteList;

        public string KeyName { get; set; }
        public string ValueName { get; set; }

        internal List<byte> DataBytes
        {
            get { return this.byteList; }
        }
    }
}
```

```

public PolicyEntry(
    string Key,
    string Value,
    uint data)
{
    KeyName = Key;
    ValueName = Value;
    this.byteList = new List<byte>();
    byte[] arrBytes = BitConverter.GetBytes(data);
    if (BitConverter.IsLittleEndian == false) { Array.Reverse(arrBytes); }
    this.byteList.AddRange(arrBytes);
}

~PolicyEntry()
{
    this.byteList = null;
}
}

public class PolicyFile
{
    private Dictionary<string, PolicyEntry> entries;

    public List<PolicyEntry> Entries
    {
        get
        {
            List<PolicyEntry> policyList = new List<PolicyEntry>(entries.Values);
            return policyList;
        }
    }

    public PolicyFile()
    {
        this.entries = new Dictionary<string, PolicyEntry>(StringComparer.OrdinalIgnoreCase);
    }

    public void SetDWORDValue(string key, string value, uint data)
    {
        PolicyEntry entry = new PolicyEntry(key, value, data);
        this.entries[entry.KeyName + "\\\" + entry.ValueName] = entry;
    }

    public void SaveFile(string file)
    {
        using (FileStream fs = new FileStream(file, FileMode.Create, FileAccess.Write))
        {
            fs.Write(new byte[] { 0x50, 0x52, 0x65, 0x67, 0x01, 0x00, 0x00, 0x00 }, 0, 8);
            byte[] openBracket = UnicodeEncoding.Unicode.GetBytes("[");
            byte[] closeBracket = UnicodeEncoding.Unicode.GetBytes("]");
            byte[] semicolon = UnicodeEncoding.Unicode.GetBytes(";");
            byte[] nullChar = new byte[] { 0, 0 };

            byte[] bytes;

            foreach (PolicyEntry entry in this.Entries)
            {
                fs.Write(openBracket, 0, 2);
                bytes = UnicodeEncoding.Unicode.GetBytes(entry.KeyName);
                fs.Write(bytes, 0, bytes.Length);
                fs.Write(nullChar, 0, 2);

                fs.Write(semicolon, 0, 2);
                bytes = UnicodeEncoding.Unicode.GetBytes(entry.ValueName);
                fs.Write(bytes, 0, bytes.Length);
                fs.Write(nullChar, 0, 2);

                fs.Write(semicolon, 0, 2);
            }
        }
    }
}

```



```

#return the file object
$unattendFile
}

Function CreateRegistryPolicyFile {

    $saveFileLocaiton = "" + (get-location) + "\registry.pol"

    $policyFile = New-Object MS.PolicyFileEditor.PolicyFile
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "UseAdvancedStartup", 1)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "EnableBDEWithNoTPM", 1)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "UseTPM", 2)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "UseTPMPIN", 2)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "UseTPMKey", 2)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "UseTPMKeyPIN", 2)
    $policyFile.SetDWORDValue("Software\Policies\Microsoft\FVE", "OSEnablePrebootInputProtectorsOnSlates", 1)
    $policyFile.SaveFile($saveFileLocaiton)

    $saveFileLocaiton
}

#####

if ( Test-Path $installWIMPath ){
    write-output "Image: $installWIMPath"
}
else{
    write-output "Unable to find image: $installWIMPath" "Exiting the script"
    exit
}

if ( (Get-WindowsImage -ImagePath $InstallWIMPath -Index 1).Architecture -eq 0 ){
    $Arch = "x86"
}
else{
    $Arch = "amd64"
}

$starttime = get-date

#Add type information for modifying the Registry Policy file
Add-Type -TypeDefinition $Source -Language CSharp

#Create helper files
$unattendFile = CreateUnattendFile -Arch $Arch
$registryPolFilePath = CreateRegistryPolicyFile

$Disks = Get-Disk | Where-Object { $_.Path -match "USBSTOR" -and $_.Size -gt 20Gb -and -not $_.IsBoot }
if ($Disks -eq $null)
{
    Write-Output "No USB Disks found, exiting the script. Please check that you have a device connected."
    exit
}

#We want to make sure that all non-boot connected USB drives are online, writeable and cleaned.
#This command will erase all data from all USB drives larger than 20Gb connected to your machine
#To automate this step you can add: -confirm:$False
Clear-Disk -InputObject $Disks -RemoveData -erroraction SilentlyContinue

# Currently the provisioning script needs drive letters (for dism and bcdboot.exe) and the script is more
# reliable when the main process determines all of the free drives and provides them to the sub-processes.
# Use a drive index starting at 1, since we need 2 free drives to proceed. (system & operating system)
$driveLetters = 68..90 | ForEach-Object { "${[char]$_}:" } |
    Where-Object {
        (new-object System.IO.DriveInfo $_).DriveType -eq 'noRootdirectory'
    }
$driveIndex = 1

foreach ($disk in $Disks)

```

```

for ($driveIndex = 0; $driveIndex -lt $driveLetters.count; $driveIndex++)
{
    if ( $driveIndex -lt $driveLetters.count )
    {
        Start-Job -ScriptBlock {
            $installWIMPath = $args[0]
            $unattendFile = $args[1]
            $Disk = $args[2]
            $SystemDriveLetter = $args[3]
            $OSDriveLetter = $args[4]
            $DomainName = $args[5]
            $policyFilePath = $args[6]

            #For compatibility between UEFI and legacy BIOS we use MBR for the disk.
            Initialize-Disk -InputObject $Disk -PartitionStyle MBR

            #A short sleep between creating a new partition and formatting helps ensure the partition
            #is ready before formatting.
            $SystemPartition = New-Partition -InputObject $Disk -Size (350MB) -IsActive
            Sleep 1
            Format-Volume -Partition $SystemPartition -FileSystem FAT32 -NewFileSystemLabel "UFD-System" -
confirm:$False | Out-Null

            $OSPartition = New-Partition -InputObject $Disk -UseMaximumSize
            Sleep 1
            Format-Volume -NewFileSystemLabel "UFD-Windows" -FileSystem NTFS -Partition $OSPartition -
confirm:$False | Out-Null

            #The No default drive letter prevents other computers from displaying contents of the drive when connected as
            a Data drive.
            Set-Partition -InputObject $OSPartition -NoDefaultDriveLetter $TRUE
            Set-Partition -InputObject $SystemPartition -NewDriveLetter $SystemDriveLetter
            Set-Partition -InputObject $OSPartition -NewDriveLetter $OSDriveLetter

            dism /apply-image /index:1 /applydir:${OSDriveLetter}:\ /imagefile:$InstallWIMPath
            if (!$?) {
                write-output "DISM image application failed, exiting."
                exit
            }

            copy $unattendFile ${OSDriveLetter}:\Windows\System32\sysprep\unattend.xml

            #Create the directory for the Machine Registry Policy file, suppressing the output and any error
            #and copy the pre-created Registry.pol file to that location.
            write-output "Set BitLocker default policies for WindowsToGo"
            md ${OSDriveLetter}:\windows\System32\GroupPolicy\Machine | out-null
            copy $policyFilePath ${OSDriveLetter}:\windows\System32\GroupPolicy\Machine

            #modify the registry of the image to set SanPolicy. This is also where you could set the default
            #keyboard type for USB keyboards.
            write-output "Modify SAN Policy"
            reg load HKLM\PW-System ${OSDriveLetter}:\Windows\System32\config\SYSTEM > info.log
            reg add HKLM\PW-System\ControlSet001\Services\Partmgr\Parameters /v SanPolicy /d 4 /t REG_DWORD /f
            > info.log
            reg unload HKLM\PW-System > info.log

            #We're running bcdboot from the newly applied image so we know that the correct boot files for the
            architecture and operating system are used.
            #This will fail if we try to run an amd64 bcdboot.exe on x86.
            cmd /c "$OSDriveLetter`:\Windows\system32\bcdboot $OSDriveLetter`:\Windows /f ALL /s
$SystemDriveLetter`:"
            if (!$?) {
                write-output "BCDBOOT.exe failed, exiting script."
                exit
            }

            <#
            If a domain name was provided to the script, we will create a random computer name

```

```

        if a domain name was provided to the script, we will create a random computer name
        and perform an offline domain join for the device. With this command we also suppress the
        Add User OOBE screen.

#>
    if ($DomainName)
    {
#using get-random, we will create a random computer name for the drive.
        $suffix = Get-Random
        $computername = "wtg-" + $suffix
        djoin /provision /domain $DomainName /savefile ${OSDriveLetter}:\tempBLOB.bin /reuse /machine
$computername
        djoin /requestodj /loadfile ${OSDriveLetter}:\tempBLOB.bin /windowspath
${OSDriveLetter}:\windows > info.log
        del ${OSDriveLetter}:\tempBLOB.bin

#add offline registry key to skip user account screen
        write-output "Add Offline Registry key for skipping UserAccount OOBE page."
        reg load HKLM\PW-Temp${OSDriveLetter}  ${OSDriveLetter}:\Windows\System32\config\SOFTWARE >
info.log
        reg add HKLM\PW-Temp${OSDriveLetter}\Microsoft\Windows\CurrentVersion\Setup\OOBE /v
UnattendCreatedUser /d 1 /t REG_DWORD > info.log
        reg unload HKLM\PW-Temp${OSDriveLetter} > info.log
    }

    try
    {
        Write-VolumeCache -DriveLetter ${OSDriveLetter}
        Write-Output "Disk is now ready to be removed."
    }
    catch [System.Management.Automation.CommandNotFoundException]
    {
        write-output "Flush Cache not supported, Be sure to safely remove the WTG device."
    }

    } -ArgumentList @($installWIMPath, $unattendFile, $disk, $driveLetters[$driveIndex-1][0],
$driveLetters[$driveIndex][0], $DomainName, $registryPolFilePath)
    }
    $driveIndex = $driveIndex + 2
}
#wait for all threads to finish
get-job | wait-job

#print output from all threads
get-job | receive-job

#delete the job objects
get-job | remove-job

#Cleanup helper files
del .\WtgUnattend.xml
del .\Registry.pol

$finishtime = get-date
$elapsedTime = new-timespan $starttime $finishtime
write-output "Provsioning completed in: $elapsedTime (hh:mm:ss.000)"
write-output "" "Provisioning script complete."

```

Considerations when using different USB keyboard layouts with Windows To Go

Before provisioning your Windows To Go drive you need to consider if your workspace will boot on a computer with a non-English USB keyboard attached. As described in [KB article 927824](#) there is a known issue where the plug and play ID causes the keyboard to be incorrectly identified as an English 101 key keyboard. To avoid this

problem, you can modify the provisioning script to set the override keyboard parameters.

In the PowerShell provisioning script, after the image has been applied, you can add the following commands that will correctly set the keyboard settings. The following example uses the Japanese keyboard layout:

```
reg load HKLM\WTG-Keyboard ${OSDriveLetter}:\Windows\System32\config\SYSTEM > info.log
reg add HKLM\WTG-Keyboard\ControlSet001\Services\i8042prt\Parameters /v LayerDriver /d
JPN:kbd106d11 /t REG_SZ /f
reg add HKLM\WTG-Keyboard\ControlSet001\Services\i8042prt\Parameters /v OverrideKeyboardIdentifier
/d PCAT_106KEY /t REG_SZ /f
reg add HKLM\WTG-Keyboard\ControlSet001\Services\i8042prt\Parameters /v OverrideKeyboardSubtype /d
2 /t REG_DWORD /f
reg add HKLM\WTG-Keyboard\ControlSet001\Services\i8042prt\Parameters /v OverrideKeyboardType /d 7
/t REG_DWORD /f
reg unload HKLM\WTG-Keyboard
```

Related topics

[Windows To Go: feature overview](#)

[Windows 10 forums](#)

[Prepare your organization for Windows To Go](#)

[Deployment considerations for Windows To Go](#)

[Security and data protection considerations for Windows To Go](#)

[BitLocker overview](#)

Windows To Go: feature overview

6/6/2019 • 7 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

Windows To Go is a feature in Windows 10 Enterprise and Windows 10 Education that enables the creation of a Windows To Go workspace that can be booted from a USB-connected external drive on PCs.

PCs that meet the Windows 7 or later [certification requirements](#) can run Windows 10 in a Windows To Go workspace, regardless of the operating system running on the PC. Windows To Go workspaces can use the same image enterprises use for their desktops and laptops and can be managed the same way. Windows To Go is not intended to replace desktops, laptops or supplant other mobility offerings. Rather, it provides support for efficient use of resources for alternative workplace scenarios. There are some additional considerations that you should keep in mind before you start to use Windows To Go:

- [Differences between Windows To Go and a typical installation of Windows](#)
- [Roaming with Windows To Go](#)
- [Prepare for Windows To Go](#)
- [Hardware considerations for Windows To Go](#)

Note Windows To Go is not supported on Windows RT.

Differences between Windows To Go and a typical installation of Windows

Windows To Go workspace operates just like any other installation of Windows with a few exceptions. These exceptions are:

- **Internal disks are offline.** To ensure data isn't accidentally disclosed, internal hard disks on the host computer are offline by default when booted into a Windows To Go workspace. Similarly if a Windows To Go drive is inserted into a running system, the Windows To Go drive will not be listed in Windows Explorer.
- **Trusted Platform Module (TPM) is not used.** When using BitLocker Drive Encryption a pre-operating system boot password will be used for security rather than the TPM since the TPM is tied to a specific computer and Windows To Go drives will move between computers.
- **Hibernate is disabled by default.** To ensure that the Windows To Go workspace is able to move between computers easily, hibernation is disabled by default. Hibernation can be re-enabled by using Group Policy settings.
- **Windows Recovery Environment is not available.** In the rare case that you need to recover your Windows To Go drive, you should re-image it with a fresh image of Windows.
- **Refreshing or resetting a Windows To Go workspace is not supported.** Resetting to the

manufacturer's standard for the computer doesn't apply when running a Windows To Go workspace, so the feature was disabled.

- **Upgrading a Windows To Go workspace is not supported.** Older Windows 8 or Windows 8.1 Windows To Go workspaces cannot be upgraded to Windows 10 workspaces, nor can Windows 10 Windows To Go workspaces be upgraded to future versions of Windows 10. For new versions, the workspace needs to be re-imaged with a fresh image of Windows.

Roaming with Windows To Go

Windows To Go drives can be booted on multiple computers. When a Windows To Go workspace is first booted on a host computer it will detect all hardware on the computer and install any needed drivers. When the Windows To Go workspace is subsequently booted on that host computer it will be able to identify the host computer and load the correct set of drivers automatically.

The applications that you want to use from the Windows To Go workspace should be tested to make sure they also support roaming. Some applications bind to the computer hardware which will cause difficulties if the workspace is being used with multiple host computers.

Prepare for Windows To Go

Enterprises install Windows on a large group of computers either by using configuration management software (such as System Center Configuration Manager), or by using standard Windows deployment tools such as DiskPart and the Deployment Image Servicing and Management (DISM) tool.

These same tools can be used to provision Windows To Go drive, just as you would if you were planning for provisioning a new class of mobile PCs. You can use the [Windows Assessment and Deployment Kit](#) to review deployment tools available.

Important Make sure you use the versions of the deployment tools provided for the version of Windows you are deploying. There have been many enhancements made to support Windows To Go. Using versions of the deployment tools released for earlier versions of Windows to provision a Windows To Go drive is not supported.

As you decide what to include in your Windows To Go image, be sure to consider the following questions:

Are there any drivers that you need to inject into the image?

How will data be stored and synchronized to appropriate locations from the USB device?

Are there any applications that are incompatible with Windows To Go roaming that should not be included in the image?

What should be the architecture of the image - 32bit/64bit?

What remote connectivity solution should be supported in the image if Windows To Go is used outside the corporate network?

For more information about designing and planning your Windows To Go deployment, see [Prepare your organization for Windows To Go](#).

Hardware considerations for Windows To Go

For USB drives

The devices listed in this section have been specially optimized and certified for Windows To Go and meet the necessary requirements for booting and running a full version of Windows 10 from a USB drive. The optimizations for Windows To Go include the following:

- Windows To Go certified USB drives are built for high random read/write speeds and support the thousands of random access I/O operations per second required for running normal Windows workloads smoothly.
- Windows To Go certified USB drives have been tuned to ensure they boot and run on hardware certified for use with Windows 7 and later.
- Windows To Go certified USB drives are built to last. Certified USB drives are backed with manufacturer warranties and should continue operating under normal usage. Refer to the manufacturer websites for warranty details.

As of the date of publication, the following are the USB drives currently certified for use as Windows To Go drives:

Warning Using a USB drive that has not been certified is not supported

- IronKey Workspace W700 (<http://www.ironkey.com/windows-to-go-drives/ironkey-workspace-w700.html>)
- IronKey Workspace W500 (<http://www.ironkey.com/windows-to-go-drives/ironkey-workspace-w500.html>)
- IronKey Workspace W300 (<http://www.ironkey.com/windows-to-go-drives/ironkey-workspace-w300.html>)
- Kingston DataTraveler Workspace for Windows To Go (<http://www.kingston.com/wtg/>)
- Spyrus Portable Workplace (<http://www.spyruswtg.com/>)

We recommend that you run the Spyrus Deployment Suite for Windows To Go to provision the Spyrus Portable Workplace.

- Spyrus Secure Portable Workplace (<http://www.spyruswtg.com/>)

Important You must use the Spyrus Deployment Suite for Windows To Go to provision the Spyrus Secure Portable Workplace. For more information about the Spyrus Deployment Suite for Windows To Go please refer to <http://www.spyruswtg.com/>.

- Spyrus Worksafe (<http://www.spyruswtg.com/>)

Tip This device contains an embedded smart card.

- Super Talent Express RC4 for Windows To Go

-and-

Super Talent Express RC8 for Windows To Go

(<http://www.supertalent.com/wtg/>)

- Western Digital My Passport Enterprise (<http://www.wd.com/wtg>)

We recommend that you run the WD Compass utility to prepare the Western Digital My Passport Enterprise drive for provisioning with Windows To Go. For more information about the WD Compass utility please refer to <http://www.wd.com/wtg>

For host computers

When assessing the use of a PC as a host for a Windows To Go workspace you should consider the following criteria:

- Hardware that has been certified for use with Windows 7 or later operating systems will work well with

Windows To Go.

- Running a Windows To Go workspace from a computer that is running Windows RT is not a supported scenario.
- Running a Windows To Go workspace on a Mac computer is not a supported scenario.

The following table details the characteristics that the host computer must have to be used with Windows To Go:

ITEM	REQUIREMENT
Boot process	Capable of USB boot
Firmware	USB boot enabled. (PCs certified for use with Windows 7 or later can be configured to boot directly from USB, check with the hardware manufacturer if you are unsure of the ability of your PC to boot from USB)
Processor architecture	Must support the image on the Windows To Go drive
External USB Hubs	Not supported; connect the Windows To Go drive directly to the host machine
Processor	1 Ghz or faster
RAM	2 GB or greater
Graphics	DirectX 9 graphics device with WDDM 1.2 or greater driver
USB port	USB 2.0 port or greater

Checking for architectural compatibility between the host PC and the Windows To Go drive

In addition to the USB boot support in the BIOS, the Windows 10 image on your Windows To Go drive must be compatible with the processor architecture and the firmware of the host PC as shown in the table below.

HOST PC FIRMWARE TYPE	HOST PC PROCESSOR ARCHITECTURE	COMPATIBLE WINDOWS TO GO IMAGE ARCHITECTURE
Legacy BIOS	32-bit	32-bit only
Legacy BIOS	64-bit	32-bit and 64-bit
UEFI BIOS	32-bit	32-bit only
UEFI BIOS	64-bit	64-bit only

Additional resources

- [Windows 10 forums](#)
- [Windows To Go Step by Step Wiki](#)
- [Tips for configuring your BIOS settings to work with Windows To Go](#)

Related topics

- [Deploy Windows To Go in your organization](#)
- [Windows To Go: frequently asked questions](#)
- [Prepare your organization for Windows To Go](#)
- [Deployment considerations for Windows To Go](#)
- [Security and data protection considerations for Windows To Go](#)
- [Best practice recommendations for Windows To Go](#)

Best practice recommendations for Windows To Go

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

The following are the best practice recommendations for using Windows To Go:

- Always shut down Windows and wait for shutdown to complete before removing the Windows To Go drive.
- Do not insert the Windows To Go drive into a running computer.
- Do not boot the Windows To Go drive from a USB hub. Always insert the Windows To Go drive directly into a port on the computer.
- If available, use a USB 3.0 port with Windows To Go.
- Do not install non-Microsoft core USB drivers on Windows To Go.
- Suspend BitLocker on Windows host computers before changing the BIOS settings to boot from USB and then resume BitLocker protection.

Additionally, we recommend that when you plan your deployment you should also plan a standard operating procedure for answering questions about which USB drives can be used for Windows To Go and how to enable booting from USB to assist your IT department or help desk in supporting users and work groups that want to use Windows To Go. It may be very helpful for your organization to work with your hardware vendors to create an IT standard for USB drives for use with Windows To Go, so that if groups within your organization want to purchase drives they can quickly determine which ones they should obtain.

More information

[Windows To Go: feature overview](#)

[Prepare your organization for Windows To Go](#)

[Deployment considerations for Windows To Go](#)

[Security and data protection considerations for Windows To Go](#)

[Windows To Go: frequently asked questions](#)

Deployment considerations for Windows To Go

6/6/2019 • 14 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

From the start, Windows To Go was designed to minimize differences between the user experience of working on a laptop and Windows To Go booted from a USB drive. Given that Windows To Go was designed as an enterprise solution, extra consideration was given to the deployment workflows that enterprises already have in place. Additionally, there has been a focus on minimizing the number of differences in deployment between Windows To Go workspaces and laptop PCs.

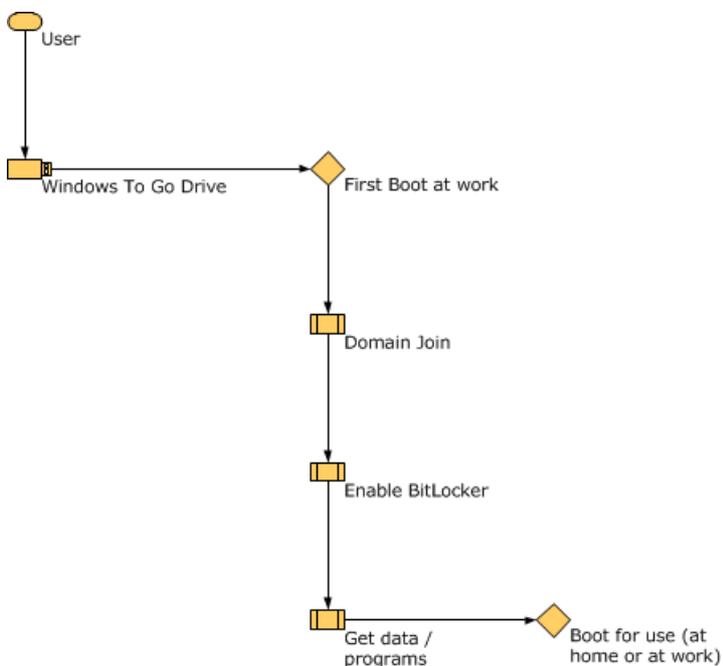
Note Windows To Go does not support operating system upgrades. Windows To Go is designed as a feature that is managed centrally. IT departments that plan to transition from one operating system version to a later version will need to incorporate re-imaging their existing Windows To Go drives as part of their upgrade deployment process.

The following sections discuss the boot experience, deployment methods, and tools that you can use with Windows To Go.

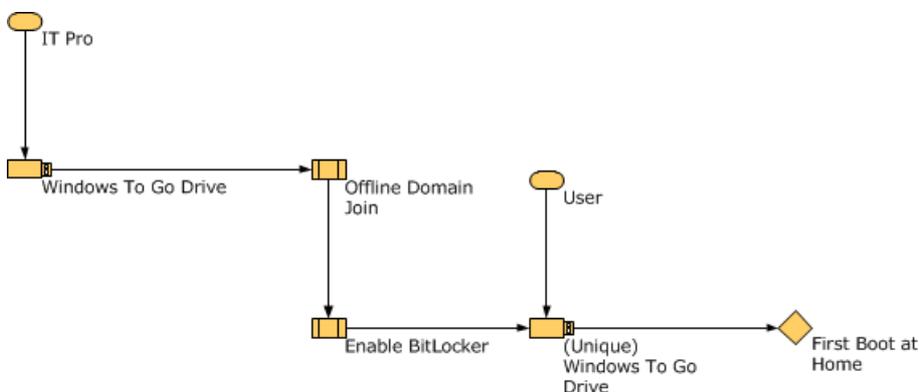
- [Initial boot experiences](#)
- [Image deployment and drive provisioning considerations](#)
- [Application installation and domain join](#)
- [Management of Windows To Go using Group Policy](#)
- [Supporting booting from USB](#)
- [Updating firmware](#)
- [Configure Windows To Go startup options](#)
- [Change firmware settings](#)

Initial boot experiences

The following diagrams illustrate the two different methods you could use to provide Windows To Go drives to your users. The experiences differ depending on whether the user will be booting the device initially on-premises or off-premises:



When a Windows To Go workspace is first used at the workplace, the Windows To Go workspace can be joined to the domain through the normal procedures that occur when a new computer is introduced. It obtains a lease, applicable policies are applied and set, and user account tokens are placed appropriately. BitLocker protection can be applied and the BitLocker recovery key automatically stored in Active Directory Domain Services. The user can access network resources to install software and get access to data sources. When the workspace is subsequently booted at a different location either on or off premises, the configuration required for it to connect back to the work network using either DirectAccess or a virtual private network connection can be configured. It is not necessary to configure the workspace for offline domain join. DirectAccess can make connecting to organizational resources easier, but is not required.



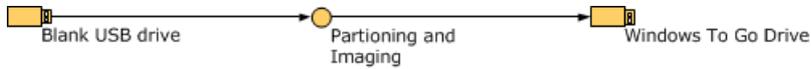
When the Windows To Go workspace is going to be used first on an off-premises computer, such as one at the employee's home, then the IT professional preparing the Windows To Go drives should configure the drive to be able to connect to organizational resources and to maintain the security of the workspace. In this situation, the Windows To Go workspace needs to be configured for offline domain join and BitLocker needs to be enabled before the workspace has been initialized.

Tip Applying BitLocker Drive Encryption to the drives before provisioning is a much faster process than encrypting the drives after data has already been stored on them due to a new feature called used-disk space only encryption. For more information, see [What's New in BitLocker](#).

DirectAccess can be used to ensure that the user can login with their domain credentials without needing a local account. For instructions on setting up a DirectAccess solution, for a small pilot deployment see [Deploy a Single Remote Access Server using the Getting Started Wizard](#) for a larger scale deployment, see [Deploy Remote Access in an Enterprise](#). If you do not want to use DirectAccess as an alternative users could log on using a local user account on the Windows To Go workspace and then use a virtual private network for remote access to your organizational network.

Image deployment and drive provisioning considerations

The Image Deployment process can be accomplished either by a centralized IT process for your organization or by individual users creating their own Windows To Go workspaces. You must have local Administrator access and access to a Windows 10 Enterprise or Windows 10 Education image to create a Windows To Go workspace, or you must be using System Center Configuration Manager 2012 Service Pack 1 or later to distribute Windows To Go workspaces to users. The image deployment process takes a blank USB drive and a Windows 10 Enterprise image (WIM) and turns it into a Windows To Go drive.



The simplest way to provision a Windows To Go drive is to use the Windows To Go Creator. After a single Windows To Go workspace has been created, it can be duplicated as many times as necessary using widely available USB duplicator products as long as the device has not been booted. After the Windows To Go drive is initialized, it should not be duplicated. Alternatively, Windows To Go Workspace Creator can be run multiple times to create multiple Windows To Go drives.

Tip When you create your Windows To Go image use `sysprep /generalize`, just as you do when you deploy Windows 10 to a standard PC. In fact, if appropriate, use the same image for both deployments.

Driver considerations

Windows includes most of the drivers that you will need to support a wide variety of host computers. However, you will occasionally need to download drivers from Windows Update to take advantage of the full functionality of a device. If you are using Windows To Go on a set of known host computers, you can add any additional drivers to the image used on Windows To Go to make Windows To Go drives more quickly usable by your employees. Especially ensure that network drivers are available so that the user can connect to Windows Update to get additional drivers if necessary.

Wi-Fi network adapter drivers are one of the most important drivers to make sure that you include in your standard image so that users can easily connect to the internet for any additional updates. IT administrators that are attempting to build Windows 10 images for use with Windows To Go should consider adding additional Wi-Fi drivers to their image to ensure that their users have the best chance of still having basic network connectivity when roaming between systems.

The following list of commonly used Wi-Fi network adapters that are not supported by the default drivers provided with Windows 10 is provided to help you ascertain whether or not you need to add drivers to your image.

Vendor name	Product description	HWID	Windows Update availability
Broadcom	802.11abgn Wireless SDIO adapter	sd\vid_02d0&pid_4330&fn_1	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_00d6106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_00f5106b&rev_02	Contact the system OEM or Broadcom for driver availability.

Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_00ef106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_00f4106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_010e106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_00e4106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_433114e4&rev_02	Contact the system OEM or Broadcom for driver availability.
Broadcom	802.11n Network Adapter	pci\ven_14e4&dev_4331&subsys_010f106b&rev_02	Contact the system OEM or Broadcom for driver availability.
Marvell	Yukon 88E8001/8003/8010 PCI Gigabit Ethernet	pci\ven_11ab&dev_4320&subsys_811a1043	32-bit driver 64-bit driver
Marvell	Libertas 802.11b/g Wireless	pci\ven_11ab&dev_1faa&subsys_6b001385&rev_03	32-bit driver 64-bit driver
Qualcomm	Atheros AR6004 Wireless LAN Adapter	sd\vid_0271&pid_0401	32-bit driver 64-bit driver not available
Qualcomm	Atheros AR5BWB222 Wireless Network Adapter	pci\ven_168c&dev_0034&subsys_20031a56	32-bit driver 64-bit driver not available
Qualcomm	Atheros AR5BWB222 Wireless Network Adapter	pci\ven_168c&dev_0034&subsys_020a1028&rev_01	Contact the system OEM or Qualcom for driver availability.
Qualcomm	Atheros AR5005G Wireless Network Adapter	pci\ven_168c&dev_001a&subsys_04181468&rev_01	32-bit driver 64-bit driver

Ralink	Wireless-G PCI Adapter	pci\ven_1814&dev_0301&subsys_00551737&rev_00	32-bit driver 64-bit driver
Ralink	Turbo Wireless LAN Card	pci\ven_1814&dev_0301&subsys_25611814&rev_00	32-bit driver 64-bit driver
Ralink	Wireless LAN Card V1	pci\ven_1814&dev_0302&subsys_3a711186&rev_00	32-bit driver 64-bit driver
Ralink	D-Link AirPlus G DWL-G510 Wireless PCI Adapter(rev.C)	pci\ven_1814&dev_0302&subsys_3c091186&rev_00	32-bit driver 64-bit driver

IT administrators that want to target Windows To Go images for specific systems should test their images to ensure that the necessary system drivers are in the image, especially for critical functionality like Wi-Fi that is not supported by class drivers. Some consumer devices require OEM specific driver packages, which may not be available on Windows Update. For more information on how to add a driver to a Windows Image, please refer to the [Basic Windows Deployment Step-by-Step Guide](#).

Application installation and domain join

Unless you are using a customized Windows image that includes unattended installation settings, the initial Windows To Go workspace will not be domain joined and will not contain applications. This is exactly like a new installation of Windows on a desktop or laptop computer. When planning your deployment, you should develop methods to join Windows to Go drives to the domain and install the standard applications that users in your organization require. These methods probably will be similar to the ones used for setting up desktop and laptop computers with domain privileges and applications

Management of Windows To Go using Group Policy

In general, management of Windows To Go workspaces is same as that for desktop and laptop computers. There are Windows To Go specific Group Policy settings that should be considered as part of Windows To Go deployment. Windows To Go Group Policy settings are located at

`\\Computer Configuration\Administrative Templates\Windows Components\Portable Operating System\` in the Local Group Policy Editor.

The use of the Store on Windows To Go workspaces that are running Windows 8 can also be controlled by Group Policy. This policy setting is located at

`\\Computer Configuration\Administrative Templates\Windows Components\Store\` in the Local Group Policy Editor.

The policy settings have specific implications for Windows To Go that you should be aware of when planning your deployment:

Settings for workspaces

- **Allow hibernate (S4) when started from a Windows To Go workspace**

This policy setting specifies whether the PC can use the hibernation sleep state (S4) when started from a Windows To Go workspace. By default, hibernation is disabled when using Windows To Go workspace, so enabling this setting explicitly turns this ability back on. When a computer enters hibernation, the contents of memory are written to disk. When the disk is resumed, it is important that the hardware attached to the system, as well as the disk itself, are unchanged. This is inherently incompatible with roaming between PC hosts. Hibernation should only be used when the Windows To Go workspace is not being used to roam

between host PCs.

Important For the host-PC to resume correctly when hibernation is enabled the Windows To Go workspace must continue to use the same USB port.

- **Disallow standby sleep states (S1-S3) when starting from a Windows To Go workspace**

This policy setting specifies whether the PC can use standby sleep states (S1–S3) when started from a Windows To Go workspace. The Sleep state also presents a unique challenge to Windows To Go users. When a computer goes to sleep, it appears as if it is shut down. It could be very easy for a user to think that a Windows To Go workspace in sleep mode was actually shut down and they could remove the Windows To Go drive and take it home. Removing the Windows To Go drive in this scenario is equivalent to an unclean shutdown which may result in the loss of unsaved user data or the corruption on the drive. Moreover, if the user now boots the drive on another PC and brings it back to the first PC which still happens to be in the sleep state, it will lead to an arbitrary crash and eventually corruption of the drive and result in the workspace becoming unusable. If you enable this policy setting, the Windows To Go workspace cannot use the standby states to cause the PC to enter sleep mode. If you disable or do not configure this policy setting, the Windows To Go workspace can place the PC in sleep mode.

Settings for host PCs

- **Windows To Go Default Startup Options**

This policy setting controls whether the host computer will boot to Windows To Go if a USB device containing a Windows To Go workspace is connected, and controls whether users can make changes using the **Windows To Go Startup Options** settings dialog. If you enable this policy setting, booting to Windows To Go when a USB device is connected will be enabled and users will not be able to make changes using the **Windows To Go Startup Options** settings dialog. If you disable this policy setting, booting to Windows To Go when a USB device is connected will not be enabled unless a user configures the option manually in the firmware. If you do not configure this policy setting, users who are members of the local Administrators group can enable or disable booting from USB using the **Windows To Go Startup Options** settings dialog.

Important Enabling this policy setting will cause PCs running Windows to attempt to boot from any USB device that is inserted into the PC before it is started.

Supporting booting from USB

The biggest hurdle for a user wanting to use Windows To Go is configuring their computer to boot from USB. This is traditionally done by entering the firmware and configuring the appropriate boot order options. To ease the process of making the firmware modifications required for Windows To Go, Windows includes a feature named **Windows To Go Startup Options** that allows a user to configure their computer to boot from USB from within Windows—without ever entering their firmware, as long as their firmware supports booting from USB.

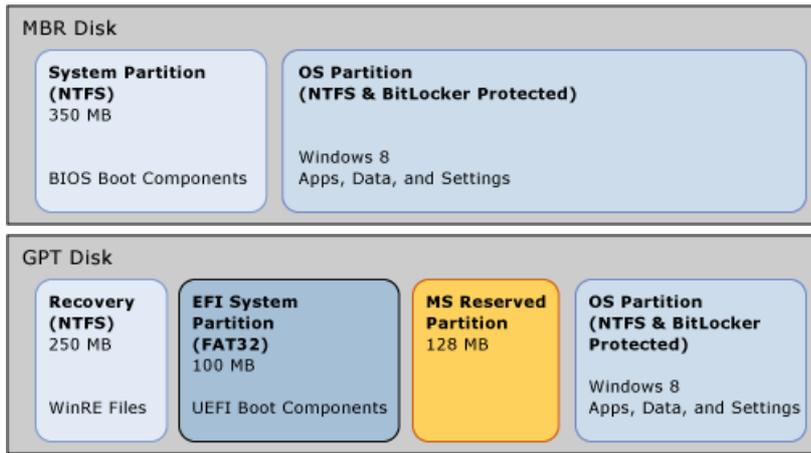
Note Enabling a system to always boot from USB first has implications that you should consider. For example, a USB device that includes malware could be booted inadvertently to compromise the system, or multiple USB drives could be plugged in to cause a boot conflict. For this reason, the Windows To Go startup options are disabled by default. In addition, administrator privileges are required to configure Windows To Go startup options.

If you are going to be using a Windows 7 computer as a host-PC, see the wiki article [Tips for configuring your BIOS settings to work with Windows To Go](#).

Roaming between different firmware types

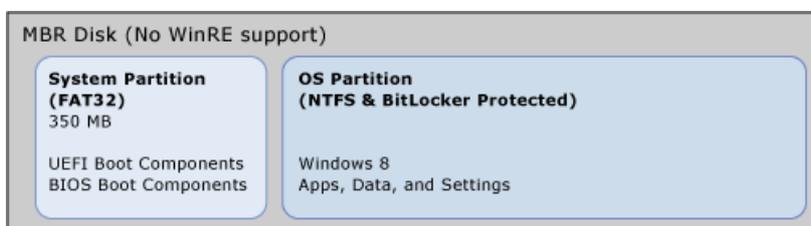
Windows supports two types of PC firmware: Unified Extensible Firmware Interface (UEFI), which is the new standard, and legacy BIOS firmware, which was used in most PCs shipping with Windows 7 or earlier version of

Windows. Each firmware type has completely different Windows boot components that are incompatible with each other. Beyond the different boot components, Windows supports different partition styles and layout requirements for each type of firmware as shown in the following diagrams.



This presented a unique challenge for Windows To Go because the firmware type is not easily determined by end-users—a UEFI computer looks just like a legacy BIOS computer and Windows To Go must boot on both types of firmware.

To enable booting Windows To Go on both types of firmware, a new disk layout is provided for Windows 8 or later that contains both sets of boot components on a FAT32 system partition and a new command-line option was added to bcdboot.exe to support this configuration. The **/f** option is used with the **bcdboot /s** command to specify the firmware type of the target system partition by appending either **UEFI**, **BIOS** or **ALL**. When creating Windows To Go drives manually you must use the **ALL** parameter to provide the Windows To Go drive the ability to boot on both types of firmware. For example, on volume H: (your Windows To Go USB drive letter), you would use the command **bcdboot C:\windows /s H: /f ALL**. The following diagram illustrates the disk layout that results from that command:



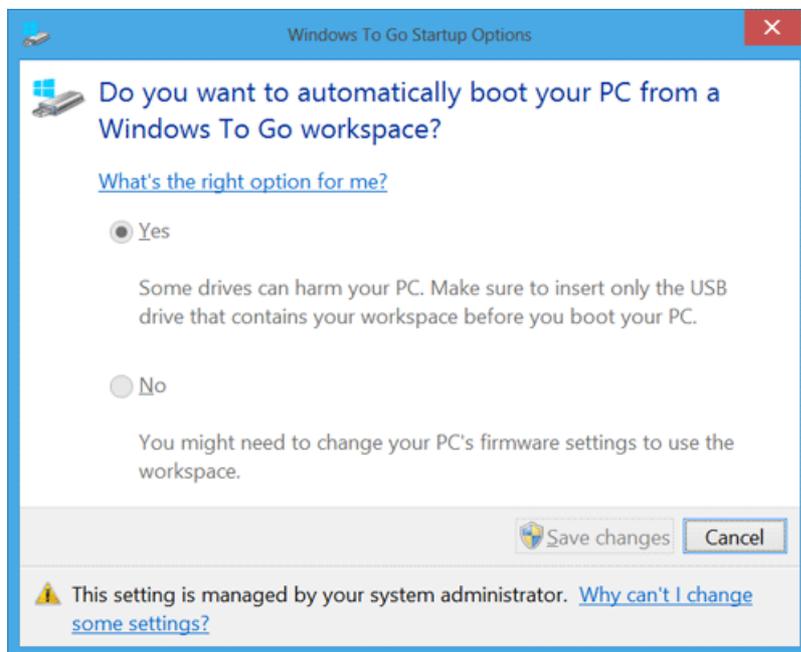
This is the only supported disk configuration for Windows To Go. With this disk configuration, a single Windows To Go drive can be booted on computers with UEFI and legacy BIOS firmware.

Configure Windows To Go startup options

Windows To Go Startup Options is a setting available on Windows 10-based PCs that enables the computer to be booted from a USB without manually changing the firmware settings of the PC. To configure Windows To Go Startup Options you must have administrative rights on the computer and the **Windows To Go Default Startup Options** Group Policy setting must not be configured.

To configure Windows To Go startup options

1. On the Start screen, type, type **Windows To Go Startup Options**, click **Settings** and then press Enter.



2. Select **Yes** to enable the startup options.

Tip If your computer is part of a domain, the Group Policy setting can be used to enable the startup options instead of the dialog.

3. Click **Save Changes**. If the User Account Control dialog box is displayed, confirm that the action it displays is what you want, and then click **Yes**.

Change firmware settings

If you choose to not use the Windows To Go startup options or are using a PC running Windows 7 as your host computer you will need to manually configure the firmware settings. The process used to accomplish this will depend on the firmware type and manufacturer. If your host computer is protected by BitLocker and running Windows 7 you should suspend BitLocker before making the change to the firmware settings. After the firmware settings have been successfully reconfigured, resume BitLocker protection. If you do not suspend BitLocker first, BitLocker will assume that the computer has been tampered with and will boot into BitLocker recovery mode.

Related topics

[Windows To Go: feature overview](#)

[Prepare your organization for Windows To Go](#)

[Security and data protection considerations for Windows To Go](#)

[Windows To Go: frequently asked questions](#)

Prepare your organization for Windows To Go

6/6/2019 • 9 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

The following information is provided to help you plan and design a new deployment of a Windows To Go in your production environment. It provides answers to the “what”, “why”, and “when” questions an IT professional might have when planning to deploy Windows To Go.

What is Windows To Go?

Windows To Go is a feature of Windows 10 Enterprise and Windows 10 Education that enables users to boot Windows from a USB-connected external drive. Windows To Go drives can use the same image that enterprises use for their desktops and laptops, and can be managed the same way. Offering a new mobility option, a Windows To Go workspace is not intended to replace desktops or laptops, or supplant other mobility offerings.

Enterprise customers utilizing Volume Activation Windows licensing will be able to deploy USB drives provisioned with Windows To Go workspace. These drives will be bootable on multiple compatible host computers. Compatible host computers are computers that are:

- USB boot capable
- Have USB boot enabled in the firmware
- Meet Windows 7 minimum system requirements
- Have compatible processor architectures (for example, x86 or AMD64) as the image used to create the Windows To Go workspace. ARM is not a supported processor for Windows To Go.
- Have firmware architecture that is compatible with the architecture of the image used for the Windows To Go workspace

Booting a Windows To Go workspace requires no specific software on the host computer. PCs certified for Windows 7 and later can host Windows To Go.

The following topics will familiarize you with how you can use a Windows To Go workspace and give you an overview of some of the things you should consider in your design.

Usage scenarios

The following scenarios are examples of situations in which Windows To Go workspaces provide a solution for an IT implementer:

- **Continuance of operations (COO).** In this scenario, selected employees receive a USB drive with a Windows To Go workspace, which includes all of the applications that the employees use at work. The employees can keep the device at home, in a briefcase, or wherever they want to store it until needed. When the users boot their home computer from the USB drive, it will create a corporate desktop experience so that they can quickly start working. On the very first boot, the employee sees that Windows

is installing devices; after that one time, the Windows To Go drive boots like a normal computer. If they have enterprise network access, employees can use a virtual private network (VPN) connection or DirectAccess to access corporate resources. If the enterprise network is available, the Windows To Go workspace will automatically be updated using your standard client management processes.

- **Contractors and temporary workers.** In this situation, an enterprise IT pro or manager would distribute the Windows To Go drive directly to the worker where they can be assisted with any necessary additional user education needs or address any possible compatibility issues. While the worker is on assignment, they can boot their computer exclusively from the Windows To Go drive and run all applications in that environment until the end of the assignment when the device is returned. No installation of software is required on the worker's personal computer.
- **Managed free seating.** The employee is issued a Windows To Go drive that is then used with the host computer assigned to that employee for a given session (this could be a vehicle, workspace, or standalone laptop). When the employee leaves the session, the next time they return they use the same USB flash drive but use a different host computer.
- **Work from home.** In this situation, the Windows To Go drive can be provisioned for employees using various methods including System Center Configuration Manager or other deployment tools and then distributed to employees. The employee is instructed to boot the Windows To Go drive initially at work, which caches the employee's credentials on the Windows To Go workspace and allows the initial data synchronization between the enterprise network and the Windows To Go workspace. The user can then bring the Windows To Go drive home where it can be used with their home computer, with or without enterprise network connectivity.
- **Travel lightly.** In this situation you have employees who are moving from site to site, but who always will have access to a compatible host computer on site. Using Windows To Go workspaces allows them to travel without the need to pack their PC.

Note If the employee wants to work offline for the majority of the time, but still maintain the ability to use the drive on the enterprise network, they should be informed of how often the Windows To Go workspace needs to be connected to the enterprise network. Doing so will ensure that the drive retains its access privileges and the workspace's computer object is not potentially deleted from Active Directory Domain Services (AD DS).

Infrastructure considerations

Because Windows To Go requires no additional software and minimal configuration, the same tools used to deploy images to other PCs can be used by an enterprise to install Windows To Go on a large group of USB devices. Moreover, because Windows To Go is compatible with connectivity and synchronization solutions already in use—such as Remote Desktop, DirectAccess and Folder Redirection—no additional infrastructure or management is necessary for this deployment. A Windows To Go image can be created on a USB drive that is identical to the hard drive inside a desktop. However, you may wish to consider making some modifications to your infrastructure to help make management of Windows To Go drives easier and to be able to identify them as a distinct device group.

Activation considerations

Windows To Go uses volume activation. You can use either Active Directory-based activation or KMS activation with Windows To Go. The Windows To Go workspace counts as another installation when assessing compliance with application licensing agreements.

Microsoft software, such as Microsoft Office, distributed to a Windows To Go workspace must also be activated. Office deployment is fully supported on Windows To Go. Please note, due to the retail subscription activation method associated with Office 365 ProPlus, Office 365 ProPlus subscribers are provided volume licensing activation rights for Office Professional Plus 2013 MSI for local installation on the Windows To Go drive. This is

available to organizations who purchase Office 365 ProPlus or Office 365 Enterprise SKUs containing Office 365 ProPlus via volume licensing channels. For more information about activating Microsoft Office, see [Volume activation methods in Office 2013](#).

You should investigate other software manufacturer's licensing requirements to ensure they are compatible with roaming usage before deploying them to a Windows To Go workspace.

Note Using Multiple Activation Key (MAK) activation is not a supported activation method for Windows To Go as each different PC-host would require separate activation. MAK activation should not be used for activating Windows, Office, or any other application on a Windows To Go drive.

See [Plan for Volume Activation](#) for more information about these activation methods and how they can be used in your organization.

Organizational unit structure and use of Group Policy Objects

You may find it beneficial to create additional Active Directory organizational unit (OU) structures to support your Windows To Go deployment; one for host computer accounts and one for Windows To Go workspace computer accounts. Creating an organizational unit for host computers allows you to enable the Windows To Go Startup Options using Group Policy for only the computers that will be used as Windows To Go hosts. Setting this policy helps to prevent computers from being accidentally configured to automatically boot from USB devices and allows closer monitoring and control of those computers which have the ability to boot from a USB device. The organizational unit for Windows To Go workspaces allows you to apply specific policy controls to them, such as the ability to use the Store application, power state controls, and line-of-business application installation.

If you are deploying Windows To Go workspaces for a scenario in which they are not going to be roaming, but are instead being used on the same host computer, such as with temporary or contract employees, you might wish to enable hibernation or the Windows Store.

For more information about Group Policy settings that can be used with Windows To Go, see [Deployment considerations for Windows To Go](#)

Computer account management

If you configure Windows To Go drives for scenarios where drives may remain unused for extended period of time such as use in continuance of operations scenarios, the AD DS computer account objects that correspond to Windows To Go drives have the potential to become stale and be pruned during maintenance operations. To address this issue, you should either have users log on regularly according to a schedule or modify any maintenance scripts to not clean up computer accounts in the Windows To Go device organizational unit.

User account and data management

People use computers to work with data and consume content - that is their core function. The data must be stored and retrievable for it to be useful. When users are working in a Windows To Go workspace, they need to have the ability to get to the data that they work with and to keep it accessible when the workspace is not being used. For this reason we recommend that you use folder redirection and offline files to redirect the path of local folders (such as the Documents folder) to a network location, while caching the contents locally for increased speed and availability. We also recommend that you use roaming user profiles to synchronize user specific settings so that users receive the same operating system and application settings when using their Windows To Go workspace and their desktop computer. When a user signs in using a domain account that is set up with a file share as the profile path, the user's profile is downloaded to the local computer and merged with the local profile (if present). When the user logs off the computer, the local copy of their profile, including any changes, is merged with the server copy of the profile. For more information, see [Folder Redirection, Offline Files, and Roaming User Profiles overview](#).

Windows To Go is fully integrated with your Microsoft account. Setting synchronization is accomplished by connecting a Microsoft account to a user account. Windows To Go devices fully support this feature and can be managed by Group Policy so that the customization and configurations you prefer will be applied to your Windows To Go workspace.

Remote connectivity

If you want Windows To Go to be able to connect back to organizational resources when it is being used off-premises a remote connectivity solution must be enabled. Windows Server 2012 DirectAccess can be used as can a virtual private network (VPN) solution. For more information about configuring a remote access solution, see the [Remote Access \(DirectAccess, Routing and Remote Access\) Overview](#).

Related topics

[Windows To Go: feature overview](#)

[Deployment considerations for Windows To Go](#)

[Security and data protection considerations for Windows To Go](#)

[Windows To Go: frequently asked questions](#)

Security and data protection considerations for Windows To Go

6/6/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

One of the most important requirements to consider when you plan your Windows To Go deployment is to ensure that the data, content, and resources you work with in the Windows To Go workspace is protected and secure.

Backup and restore

As long as you are not saving data on the Windows To Go drive, there is no need for a backup and restore solution for Windows To Go. If you are saving data on the drive and are not using folder redirection and offline files, you should back up all of your data to a network location, such as cloud storage or a network share after each work session. Review the new and improved features described in [Supporting Information Workers with Reliable File Services and Storage](#) for different solutions you could implement.

If the USB drive fails for any reason, the standard process to restore the drive to working condition is to reformat and re-provision the drive with Windows To Go, so all data and customization on the drive will be lost. This is another reason why using roaming user profiles, folder redirection and offline files with Windows To Go is strongly recommended. For more information, see [Folder Redirection, Offline Files, and Roaming User Profiles overview](#).

BitLocker

We recommend that you use BitLocker with your Windows To Go drives to protect the drive from being compromised if the drive is lost or stolen. When BitLocker is enabled, the user must provide a password to unlock the drive and boot the Windows To Go workspace, this helps prevent unauthorized users from booting the drive and using it to gain access to your network resources and confidential data. Because Windows To Go drives are meant to be roamed between computers, the Trusted Platform Module (TPM) cannot be used by BitLocker to protect the drive. Instead, you will be specifying a password that BitLocker will use for disk encryption and decryption. By default, this password must be eight characters in length and can enforce more strict requirements depending on the password complexity requirements defined by your organizations domain controller.

You can enable BitLocker while using the Windows To Go Creator wizard as part of the drive provisioning process before first use; or it can be enabled afterward by the user from within the Windows To Go workspace.

Tip If the Windows To Go Creator wizard is not able to enable BitLocker, see [Why can't I enable BitLocker from Windows To Go Creator?](#)

If you are using a host computer running Windows 7 that has BitLocker enabled, you should suspend BitLocker before changing the BIOS settings to boot from USB and then resume BitLocker protection. If BitLocker is not

suspended first, the next time the computer is started it will boot into recovery mode.

Disk discovery and data leakage

We recommend that you use the **NoDefaultDriveLetter** attribute when provisioning the USB drive to help prevent accidental data leakage. **NoDefaultDriveLetter** will prevent the host operating system from assigning a drive letter if a user inserts it into a running computer. This means the drive will not appear in Windows Explorer and an AutoPlay prompt will not be displayed to the user. This reduces the likelihood that an end-user will access the offline Windows To Go disk directly from another computer. If you use the Windows To Go Creator to provision a workspace, this attribute will automatically be set for you.

To prevent accidental data leakage between Windows To Go and the host system Windows 8 has a new SAN policy—OFFLINE_INTERNAL - "4" to prevent the operating system from automatically bringing online any internally connected disk. The default configuration for Windows To Go has this policy enabled. It is strongly recommended you do not change this policy to allow mounting of internal hard drives when booted into the Windows To Go workspace. If the internal drive contains a hibernated Windows 8 operating system, mounting the drive will lead to loss of hibernation state and therefore user state or any unsaved user data when the host operating system is booted. If the internal drive contains a hibernated Windows 7 or earlier operating system, mounting the drive will lead to corruption when the host operating system is booted.

For more information, see [How to Configure Storage Area Network \(SAN\) Policy in Windows PE](#).

Security certifications for Windows To Go

Windows To Go is a core capability of Windows when it is deployed on the drive and is configured following the guidance for the applicable security certification. Solutions built using Windows To Go can be submitted for additional certifications by the solution provider that cover the solution provider's specific hardware environment. For more details about Windows security certifications, see the following topics.

- [Windows Platform Common Criteria Certification](#)
- [FIPS 140 Evaluation](#)

Related topics

[Windows To Go: feature overview](#)

[Prepare your organization for Windows To Go](#)

[Deployment considerations for Windows To Go](#)

[Windows To Go: frequently asked questions](#)

Windows To Go: frequently asked questions

6/6/2019 • 23 minutes to read • [Edit Online](#)

Applies to

- Windows 10

IMPORTANT

Windows To Go is no longer being developed. The feature does not support feature updates and therefore does not enable you to stay current. It also requires a specific type of USB that is no longer supported by many OEMs.

The following list identifies some commonly asked questions about Windows To Go.

- [What is Windows To Go?](#)
- [Does Windows To Go rely on virtualization?](#)
- [Who should use Windows To Go?](#)
- [How can Windows To Go be deployed in an organization?](#)
- [Is Windows To Go supported on both USB 2.0 and USB 3.0 drives?](#)
- [Is Windows To Go supported on USB 2.0 and USB 3.0 ports?](#)
- [How do I identify a USB 3.0 port?](#)
- [Does Windows To Go run faster on a USB 3.0 port?](#)
- [Can the user self-provision Windows To Go?](#)
- [How can Windows To Go be managed in an organization?](#)
- [How do I make my computer boot from USB?](#)
- [Why isn't my computer booting from USB?](#)
- [What happens if I remove my Windows To Go drive while it is running?](#)
- [Can I use BitLocker to protect my Windows To Go drive?](#)
- [Why can't I enable BitLocker from Windows To Go Creator?](#)
- [What power states does Windows To Go support?](#)
- [Why is hibernation disabled in Windows To Go?](#)
- [Does Windows To Go support crash dump analysis?](#)
- [Do "Windows To Go Startup Options" work with dual boot computers?](#)
- [I plugged my Windows To Go drive into a running computer and I can't see the partitions on the drive. Why not?](#)
- [I'm booted into Windows To Go, but I can't browse to the internal hard drive of the host computer. Why not?](#)

- Why does my Windows To Go drive have an MBR disk format with a FAT32 system partition?
- Is Windows To Go secure if I use it on an untrusted machine?
- Does Windows To Go work with ARM processors?
- Can I synchronize data from Windows To Go with my other computer?
- What size USB Flash Drive do I need to make a Windows To Go drive?
- Do I need to activate Windows To Go every time I roam?
- Can I use all Windows features on Windows To Go?
- Can I use all my applications on Windows To Go?
- Does Windows To Go work slower than standard Windows?
- If I lose my Windows To Go drive, will my data be safe?
- Can I boot Windows To Go on a Mac?
- Are there any APIs that allow applications to identify a Windows To Go workspace?
- How is Windows To Go licensed?
- Does Windows Recovery Environment work with Windows To Go? What's the guidance for recovering a Windows To Go drive?
- Why won't Windows To Go work on a computer running Windows XP or Windows Vista?
- Why does the operating system on the host computer matter?
- My host computer running Windows 7 is protected by BitLocker Drive Encryption. Why did I need to use the recovery key to unlock and reboot my host computer after using Windows To Go?
- I decided to stop using a drive for Windows To Go and reformatted it – why doesn't it have a drive letter assigned and how can I fix it?
- Why do I keep on getting the message "Installing devices..." when I boot Windows To Go?
- How do I upgrade the operating system on my Windows To Go drive?

What is Windows To Go?

Windows To Go is a feature for users of Windows 10 Enterprise and Windows 10 Education that enables users to boot a full version of Windows from external USB drives on host PCs.

Does Windows To Go rely on virtualization?

No. Windows To Go is a native instance of Windows 10 that runs from a USB device. It is just like a laptop hard drive with Windows 8 that has been put into a USB enclosure.

Who should use Windows To Go?

Windows To Go was designed for enterprise usage and targets scenarios such as continuance of operations, contractors, managed free seating, traveling workers, and work from home.

How can Windows To Go be deployed in an organization?

Windows To Go can be deployed using standard Windows deployment tools like Diskpart and DISM. The

prerequisites for deploying Windows To Go are:

- A Windows To Go recommended USB drive to provision; See the list of currently available USB drives at [Hardware considerations for Windows To Go](#)
- A Windows 10 Enterprise or Windows 10 Education image
- A Windows 10 Enterprise, Windows 10 Education or Windows 10 Professional host PC that can be used to provision new USB keys

You can use a Windows PowerShell script to target several drives and scale your deployment for a large number of Windows To Go drives. You can also use a USB duplicator to duplicate a Windows To Go drive after it has been provisioned if you are creating a large number of drives. See the [Windows To Go Step by Step](#) article on the TechNet wiki for a walkthrough of the drive creation process.

Is Windows To Go supported on both USB 2.0 and USB 3.0 drives?

No. Windows To Go is supported on USB 3.0 drives that are certified for Windows To Go.

Is Windows To Go supported on USB 2.0 and USB 3.0 ports?

Yes. Windows To Go is fully supported on either USB 2.0 ports or USB 3.0 ports on PCs certified for Windows 7 or later.

How do I identify a USB 3.0 port?

USB 3.0 ports are usually marked blue or carry a SS marking on the side.

Does Windows To Go run faster on a USB 3.0 port?

Yes. Because USB 3.0 offers significantly faster speeds than USB 2.0, a Windows To Go drive running on a USB 3.0 port will operate considerably faster. This speed increase applies to both drive provisioning and when the drive is being used as a workspace.

Can the user self-provision Windows To Go?

Yes, if the user has administrator permissions they can self-provision a Windows To Go drive using the Windows To Go Creator wizard which is included in Windows 10 Enterprise, Windows 10 Education and Windows 10 Professional. Additionally, System Center 2012 Configuration Manager SP1 and later releases includes support for user self-provisioning of Windows To Go drives. Configuration Manager can be downloaded for evaluation from the [Microsoft TechNet Evaluation Center](#).

How can Windows To Go be managed in an organization?

Windows To Go can be deployed and managed like a traditional desktop PC using standard Windows enterprise software distribution tools like System Center Configuration Manager. Computer and user settings for Windows To Go workspaces can be managed using Group Policy setting also in the same manner that you manage Group Policy settings for other PCs in your organization. Windows To Go workspaces can be configured to connect to the organizational resources remotely using DirectAccess or a virtual private network connection so that they can connect securely to your network.

How do I make my computer boot from USB?

For host computers running Windows 10

- Using Cortana, search for **Windows To Go startup options**, and then press Enter.

- In the **Windows To Go Startup Options** dialog box, select **Yes**, and then click **Save Changes** to configure the computer to boot from USB.

For host computers running Windows 8 or Windows 8.1:

Press **Windows logo key+W** and then search for **Windows To Go startup options** and then press Enter.

In the **Windows To Go Startup Options** dialog box select **Yes** and then click **Save Changes** to configure the computer to boot from USB.

Note Your IT department can use Group Policy to configure Windows To Go Startup Options in your organization.

If the host computer is running an earlier version of the Windows operating system need to configure the computer to boot from USB manually.

To do this, early during boot time (usually when you see the manufacturer's logo), enter your firmware/BIOS setup. (This method to enter firmware/BIOS setup differs with different computer manufacturers, but is usually entered by pressing one of the function keys, such as F12, F2, F1, Esc, and so forth. You should check the manufacturer's site to be sure if you do not know which key to use to enter firmware setup.)

After you have entered firmware setup, make sure that boot from USB is enabled. Then change the boot order to boot from USB drives first.

Alternatively, if your computer supports it, you can try to use the one-time boot menu (often F12), to select USB boot on a per-boot basis.

For more detailed instructions, see the wiki article, [Tips for configuring your BIOS settings to work with Windows To Go](#).

Warning Configuring a computer to boot from USB will cause your computer to attempt to boot from any bootable USB device connected to your computer. This potentially includes malicious devices. Users should be informed of this risk and instructed to not have any bootable USB storage devices plugged in to their computers except for their Windows To Go drive.

Why isn't my computer booting from USB?

Computers certified for Windows 7 and later are required to have support for USB boot. Check to see if any of the following items apply to your situation:

1. Ensure that your computer has the latest BIOS installed and the BIOS is configured to boot from a USB device.
2. Ensure that the Windows To Go drive is connected directly to a USB port on the computer. Many computers don't support booting from a device connected to a USB 3 PCI add-on card or external USB hubs.
3. If the computer is not booting from a USB 3.0 port, try to boot from a USB 2.0 port.

If none of these items enable the computer to boot from USB, contact the hardware manufacturer for additional support.

What happens if I remove my Windows To Go drive while it is running?

If the Windows To Go drive is removed, the computer will freeze and the user will have 60 seconds to reinsert the Windows To Go drive. If the Windows To Go drive is reinserted into the same port it was removed from, Windows will resume at the point where the drive was removed. If the USB drive is not reinserted, or is reinserted into a different port, the host computer will turn off after 60 seconds.

Warning You should never remove your Windows To Go drive when your workspace is running. The computer freeze is a safety measure to help mitigate the risk of accidental removal. Removing the Windows To Go drive without shutting down the Windows To Go workspace could result in corruption of the Windows To Go drive.

Can I use BitLocker to protect my Windows To Go drive?

Yes. In Windows 8 and later, BitLocker has added support for using a password to protect operating system drives. This means that you can use a password to secure your Windows To Go workspace and you will be prompted to enter this password every time you use the Windows To Go workspace.

Why can't I enable BitLocker from Windows To Go Creator?

Several different Group Policies control the use of BitLocker on your organizations computers. These policies are located in the **Computer Configuration\Policies\Administrative Templates\Windows Components\BitLocker Drive Encryption** folder of the local Group Policy editor. The folder contains three sub-folders for fixed, operating system and removable data drive types.

When you are using Windows To Go Creator, the Windows To Go drive is considered a removable data drive by BitLocker. Review the following setting to see if these settings apply in your situation:

1. **Control use of BitLocker on removable drives**

If this setting is disabled BitLocker cannot be used with removable drives, so the Windows To Go Creator wizard will fail if it attempts to enable BitLocker on the Windows To Go drive.

2. **Configure use of smart cards on removable data drives**

If this setting is enabled and the option **Require use of smart cards on removable data drives** is also selected the creator wizard might fail if you have not already signed on using your smart card credentials before starting the Windows To Go Creator wizard.

3. **Configure use of passwords for removable data drives**

If this setting is enabled and the **Require password complexity option** is selected the computer must be able to connect to the domain controller to verify that the password specified meets the password complexity requirements. If the connection is not available, the Windows To Go Creator wizard will fail to enable BitLocker.

Additionally, the Windows To Go Creator will disable the BitLocker option if the drive does not have any volumes. In this situation, you should initialize the drive and create a volume using the Disk Management console before provisioning the drive with Windows To Go.

What power states does Windows To Go support?

Windows To Go supports all power states except the hibernate class of power states, which include hybrid boot, hybrid sleep, and hibernate. This default behavior can be modified by using Group Policy settings to enable hibernation of the Windows To Go workspace.

Why is hibernation disabled in Windows To Go?

When a Windows To Go workspace is hibernated, it will only successfully resume on the exact same hardware. Therefore, if a Windows To Go workspace is hibernated on one computer and roamed to another, the hibernation state (and therefore user state) will be lost. To prevent this from happening, the default settings for a Windows To Go workspace disable hibernation. If you are confident that you will only attempt to resume on the same computer, you can enable hibernation using the Windows To Go Group Policy setting, **Allow hibernate (S4) when started from a Windows To Go workspace** that is located at `\\Computer`

Configuration\Administrative Templates\Windows Components\Portable Operating System in the Local Group Policy Editor (gpedit.msc).

Does Windows To Go support crash dump analysis?

Yes. Windows 8 and later support crash dump stack analysis for both USB 2.0 and 3.0.

Do “Windows To Go Startup Options” work with dual boot computers?

Yes, if both operating systems are running the Windows 8 operating system. Enabling “Windows To Go Startup Options” should cause the computer to boot from the Windows To Go workspace when the drive is plugged in before the computer is turned on.

If you have configured a dual boot computer with a Windows operating system and another operating system it might work occasionally and fail occasionally. Using this configuration is unsupported.

I plugged my Windows To Go drive into a running computer and I can't see the partitions on the drive. Why not?

Windows To Go Creator and the recommended deployment steps for Windows To Go set the NO_DEFAULT_DRIVE_LETTER flag on the Windows To Go drive. This flag prevents Windows from automatically assigning drive letters to the partitions on the Windows To Go drive. That's why you can't see the partitions on the drive when you plug your Windows To Go drive into a running computer. This helps prevent accidental data leakage between the Windows To Go drive and the host computer. If you really need to access the files on the Windows To Go drive from a running computer, you can use diskmgmt.msc or diskpart to assign a drive letter.

Warning It is strongly recommended that you do not plug your Windows To Go drive into a running computer. If the computer is compromised, your Windows To Go workspace can also be compromised.

I'm booted into Windows To Go, but I can't browse to the internal hard drive of the host computer. Why not?

Windows To Go Creator and the recommended deployment steps for Windows To Go set SAN Policy 4 on Windows To Go drive. This policy prevents Windows from automatically mounting internal disk drives. That's why you can't see the internal hard drives of the host computer when you are booted into Windows To Go. This is done to prevent accidental data leakage between Windows To Go and the host system. This policy also prevents potential corruption on the host drives or data loss if the host operating system is in a hibernation state. If you really need to access the files on the internal hard drive, you can use diskmgmt.msc to mount the internal drive.

Warning It is strongly recommended that you do not mount internal hard drives when booted into the Windows To Go workspace. If the internal drive contains a hibernated Windows 8 or later operating system, mounting the drive will lead to loss of hibernation state and therefore user state or any unsaved user data when the host operating system is booted. If the internal drive contains a hibernated Windows 7 or earlier operating system, mounting the drive will lead to corruption when the host operating system is booted.

Why does my Windows To Go drive have an MBR disk format with a FAT32 system partition?

This is done to allow Windows To Go to boot from UEFI and legacy systems.

Is Windows To Go secure if I use it on an untrusted computer?

While you are more secure than if you use a completely untrusted operating system, you are still vulnerable to attacks from the firmware or anything that runs before Windows To Go starts. If you plug your Windows To Go

drive into a running untrusted computer, your Windows To Go drive can be compromised because any malicious software that might be active on the computer can access the drive.

Does Windows To Go work with ARM processors?

No. Windows RT is a specialized version of Windows designed for ARM processors. Windows To Go is currently only supported on PCs with x86 or x64-based processors.

Can I synchronize data from Windows To Go with my other computer?

To get your data across all your computers, we recommend using folder redirection and client side caching to store copies of your data on a server while giving you offline access to the files you need.

What size USB flash drive do I need to make a Windows To Go drive?

The size constraints are the same as full Windows. To ensure that you have enough space for Windows, your data, and your applications, we recommend USB drives that are a minimum of 20 GB in size.

Do I need to activate Windows To Go every time I roam?

No, Windows To Go requires volume activation; either using the [Key Management Service \(KMS\)](#) server in your organization or using [Active Directory](#) based volume activation. The Windows To Go workspace will not need to be reactivated every time you roam. KMS activates Windows on a local network, eliminating the need for individual computers to connect to Microsoft. To remain activated, KMS client computers must renew their activation by connecting to the KMS host on periodic basis. This typically occurs as soon as the user has access to the corporate network (either through a direct connection on-premises or a through remote connection using DirectAccess or a virtual private network connection), once activated the machine will not need to be activated again until the activation validity interval has passed. In a KMS configuration the activation validity interval is 180 days.

Can I use all Windows features on Windows To Go?

Yes, with some minor exceptions, you can use all Windows features with your Windows To Go workspace. The only currently unsupported features are using the Windows Recovery Environment and PC Reset & Refresh.

Can I use all my applications on Windows To Go?

Yes. Because your Windows To Go workspace is a full Windows 10 environment, all applications that work with Windows 10 should work in your Windows To Go workspace. However, any applications that use hardware binding (usually for licensing and/or digital rights management reasons) may not run when you roam your Windows To Go drive between different host computers, and you may have to use those applications on the same host computer every time.

Does Windows To Go work slower than standard Windows?

If you are using a USB 3.0 port and a Windows To Go certified device, there should be no perceivable difference between standard Windows and Windows To Go. However, if you are booting from a USB 2.0 port, you may notice some slowdown since USB 2.0 transfer speeds are slower than SATA speeds.

If I lose my Windows To Go drive, will my data be safe?

Yes! If you enable BitLocker on your Windows To Go drive, all your data will be encrypted and protected and a malicious user will not be able to access your data without your password. If you don't enable BitLocker, your

data will be vulnerable if you lose your Windows To Go drive.

Can I boot Windows To Go on a Mac?

We are committed to give customers a consistent and quality Windows 10 experience with Windows To Go. Windows To Go supports host devices certified for use with Windows 7 or later. Because Mac computers are not certified for use with Windows 7 or later, using Windows To Go is not supported on a Mac.

Are there any APIs that allow applications to identify a Windows To Go workspace?

Yes. You can use a combination of identifiers to determine if the currently running operating system is a Windows To Go workspace. First, check if the **PortableOperatingSystem** property is true. When that value is true it means that the operating system was booted from an external USB device.

Next, check if the **OperatingSystemSKU** property is equal to **4** (for Windows 10 Enterprise) or **121** (for Windows 10 Education). The combination of those two properties represents a Windows To Go workspace environment.

For more information, see the MSDN article on the [Win32_OperatingSystem class](#).

How is Windows To Go licensed?

Windows To Go allows organization to support the use of privately owned PCs at the home or office with more secure access to their organizational resources. With Windows To Go use rights under [Software Assurance](#), an employee will be able to use Windows To Go on any company PC licensed with Software Assurance as well as from their home PC.

Does Windows Recovery Environment work with Windows To Go? What's the guidance for recovering a Windows To Go drive?

No, use of Windows Recovery Environment is not supported on Windows To Go. It is recommended that you implement user state virtualization technologies like Folder Redirection to centralize and back up user data in the data center. If any corruption occurs on a Windows To Go drive, you should re-provision the workspace.

Why won't Windows To Go work on a computer running Windows XP or Windows Vista?

Actually it might. If you have purchased a computer certified for Windows 7 or later and then installed an older operating system, Windows To Go will boot and run as expected as long as you have configured the firmware to boot from USB. However, if the computer was certified for Windows XP or Windows Vista, it might not meet the hardware requirements for Windows To Go to run. Typically computers certified for Windows Vista and earlier operating systems have less memory, less processing power, reduced video rendering, and slower USB ports.

Why does the operating system on the host computer matter?

It doesn't other than to help visually identify if the PC has compatible hardware. For a PC to be certified for Windows 7 or later it had to support booting from USB. If a computer cannot boot from USB there is no way that it can be used with Windows To Go. The Windows To Go workspace is a full Windows 10 environment, so all of the hardware requirements of Windows 10 with respect to processing speed, memory usage, and graphics rendering need to be supported to be assured that it will work as expected.

My host computer running Windows 7 is protected by BitLocker Drive

Encryption. Why did I need to use the recovery key to unlock and reboot my host computer after using Windows To Go?

The default BitLocker protection profile in Windows 7 monitors the host computer for changes to the boot order as part of protecting the computer from tampering. When you change the boot order of the host computer to enable it to boot from the Windows To Go drive, the BitLocker system measurements will reflect that change and boot into recovery mode so that the computer can be inspected if necessary.

You can reset the BitLocker system measurements to incorporate the new boot order using the following steps:

1. Log on to the host computer using an account with administrator privileges.
2. Click **Start**, click **Control Panel**, click **System and Security**, and then click **BitLocker Drive Encryption**.
3. Click **Suspend Protection** for the operating system drive.

A message is displayed, informing you that your data will not be protected while BitLocker is suspended and asking if you want to suspend BitLocker Drive Encryption. Click **Yes** to continue and suspend BitLocker on the drive.

4. Restart the computer and enter the firmware settings to reset the boot order to boot from USB first. For more information on changing the boot order in the BIOS, see [Tips for configuring your BIOS settings to work with Windows To Go](#) on the TechNet wiki.
5. Restart the computer again and then log on to the host computer using an account with administrator privileges. (Neither your Windows To Go drive nor any other USB drive should be inserted.)
6. Click **Start**, click **Control Panel**, click **System and Security**, and then click **BitLocker Drive Encryption**.
7. Click **Resume Protection** to re-enable BitLocker protection.

The host computer will now be able to be booted from a USB drive without triggering recovery mode.

Note The default BitLocker protection profile in Windows 8 or later does not monitor the boot order.

I decided to stop using a drive for Windows To Go and reformatted it – why doesn't it have a drive letter assigned and how can I fix it?

Reformatting the drive erases the data on the drive, but doesn't reconfigure the volume attributes. When a drive is provisioned for use as a Windows To Go drive the NODEFAULTDRIVELETTER attribute is set on the volume. To remove this attribute, use the following steps:

1. Open a command prompt with full administrator permissions.

Note If your user account is a member of the Administrators group, but is not the Administrator account itself, then, by default, the programs that you run only have standard user permissions unless you explicitly choose to elevate them.

2. Start the `diskpart` command interpreter, by typing `diskpart` at the command prompt.
3. Use the `select disk` command to identify the drive. If you do not know the drive number, use the `list` command to display the list of disks available.
4. After selecting the disk, run the `clean` command to remove all data, formatting, and initialization information from the drive.

Why do I keep on getting the message "Installing devices..." when I boot Windows To Go?

One of the challenges involved in moving the Windows To Go drive between PCs while seamlessly booting Windows with access to all of their applications and data is that for Windows to be fully functional, specific drivers need to be installed for the hardware in each machine that runs Windows. Windows 8 or later has a process called respecialize which will identify new drivers that need to be loaded for the new PC and disable drivers which are not present on the new configuration. In general this feature is reliable and efficient when roaming between PCs of widely varying hardware configurations.

In certain cases, third party drivers for different hardware models or versions can reuse device ID's, driver file names, registry keys (or any other operating system constructs which do not support side-by-side storage) for similar hardware. For example, Touchpad drivers on different laptops often reuse the same device ID's, and video cards from the same manufacturer may often reuse service names. Windows handles these situations by marking the non-present device node with a flag that indicates the existing driver needs to be reinstalled before continuing to install the new driver.

This process will occur on any boot that a new driver is found and a driver conflict is detected. In some cases that will result in a respecialize progress message "Installing devices..." displaying every time that a Windows to Go drive is roamed between two PCs which require conflicting drivers.

How do I upgrade the operating system on my Windows To Go drive?

There is no support in Windows for upgrading a Windows To Go drive. Deployed Windows To Go drives with older versions of Windows will need to be re-imaged with a new version of Windows in order to transition to the new operating system version.

Additional resources

- [Windows 10 forums](#)
- [Windows To Go Step by Step Wiki](#)
- [Windows To Go: feature overview](#)
- [Prepare your organization for Windows To Go](#)
- [Deployment considerations for Windows To Go](#)
- [Security and data protection considerations for Windows To Go](#)

Volume Activation Management Tool (VAMT)

Technical Reference

6/6/2019 • 2 minutes to read • [Edit Online](#)

The Volume Activation Management Tool (VAMT) enables network administrators and other IT professionals to automate and centrally manage the Windows®, Microsoft® Office, and select other Microsoft products volume and retail-activation process. VAMT can manage volume activation using Multiple Activation Keys (MAKs) or the Windows Key Management Service (KMS). VAMT is a standard Microsoft Management Console (MMC) snap-in that requires the Microsoft Management Console (MMC) 3.0. VAMT can be installed on any computer that has one of the following Windows operating systems:

- Windows® 7 or above
- Windows Server 2008 R2 or above

Important VAMT is designed to manage volume activation for: Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 (or above), Microsoft Office 2010 (or above).

VAMT is only available in an EN-US (x86) package.

In this Section

TOPIC	DESCRIPTION
Introduction to VAMT	Provides a description of VAMT and common usages.
Active Directory-Based Activation Overview	Describes Active Directory-Based Activation scenarios.
Install and Configure VAMT	Describes how to install VAMT and use it to configure client computers on your network.
Add and Manage Products	Describes how to add client computers into VAMT.
Manage Product Keys	Describes how to add and remove a product key from VAMT.
Manage Activations	Describes how to activate a client computer by using a variety of activation methods.
Manage VAMT Data	Describes how to save, import, export, and merge a Computer Information List (CILX) file using VAMT.
VAMT Step-by-Step Scenarios	Provides step-by-step instructions for using VAMT in typical environments.
VAMT Known Issues	Lists known issues in VAMT.

Introduction to VAMT

6/6/2019 • 4 minutes to read • [Edit Online](#)

The Volume Activation Management Tool (VAMT) enables network administrators and other IT professionals to automate and centrally manage the Windows®, Microsoft® Office®, and select other Microsoft products volume and retail activation process. VAMT can manage volume activation using Multiple Activation Keys (MAKs) or the Windows Key Management Service (KMS). VAMT is a standard Microsoft Management Console (MMC) snap-in and can be installed on any computer that has one of the following Windows operating systems: Windows® 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 R2, or Windows Server 2012.

Note VAMT can be installed on, and can manage, physical or virtual instances. VAMT cannot detect whether or not the remote products are virtual. As long as the products can respond to Windows Management Instrumentation (WMI) calls, they will be discovered and activated.

In this Topic

- [Managing Multiple Activation Key \(MAK\) and Retail Activation](#)
- [Managing Key Management Service \(KMS\) Activation](#)
- [Enterprise Environment](#)
- [VAMT User Interface](#)

Managing Multiple Activation Key (MAK) and Retail Activation

You can use a MAK or a retail product key to activate Windows, Windows Server, or Office on an individual computer or a group of computers. VAMT enables two different activation scenarios:

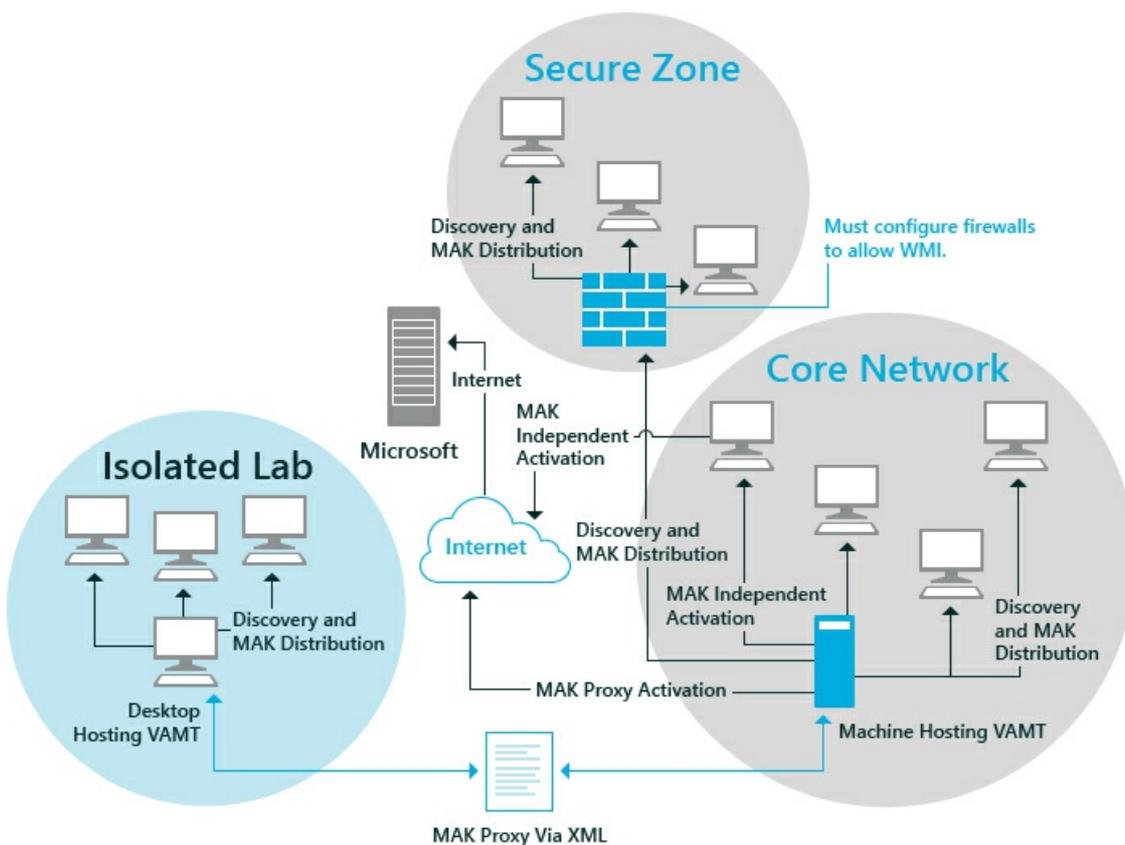
- **Online activation.** Many enterprises maintain a single Windows system image or Office installation package for deployment across the enterprise. Occasionally there is also a need to use retail product keys in special situations. Online activation enables you to activate over the Internet any products installed with MAK, KMS host, or retail product keys on one or more connected computers within a network. This process requires that each product communicate activation information directly to Microsoft.
- **Proxy activation.** This activation method enables you to perform volume activation for products installed on client computers that do not have Internet access. The VAMT host computer distributes a MAK, KMS Host key (CSVLK), or retail product key to one or more client products and collects the installation ID (IID) from each client product. The VAMT host sends the IIDs to Microsoft on behalf of the client products and obtains the corresponding Confirmation IDs (CIDs). The VAMT host then installs the CIDs on the client products to complete the activation. Using this method, only the VAMT host computer needs Internet access. You can also activate products installed on computers in a workgroup that is completely isolated from any larger network, by installing a second instance of VAMT on a computer within the workgroup. Then, use removable media to transfer activation data between this new instance of VAMT and the Internet-connected VAMT host.

Managing Key Management Service (KMS) Activation

In addition to MAK or retail activation, you can use VAMT to perform volume activation using the Key Management Service (KMS). VAMT can install and activate GVLK (KMS client) keys on client products. GVLKs are the default product keys used by Volume License editions of Windows Vista, Windows 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012 as well as Microsoft Office 2010. VAMT treats a KMS Host key (CSVLK) product key identically to a retail-type product key; therefore, the experience for product key entry and activation management are identical for both these product key types.

Enterprise Environment

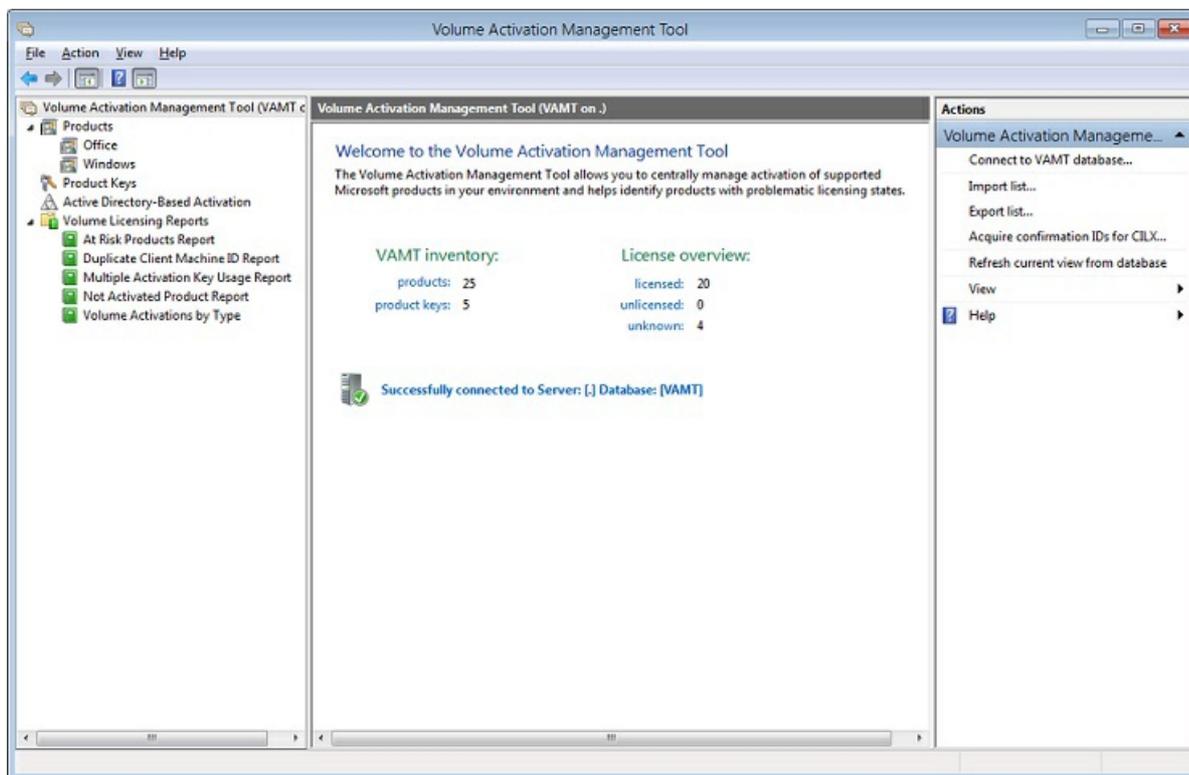
VAMT is commonly implemented in enterprise environments. The following illustrates three common environments—Core Network, Secure Zone, and Isolated Lab.



In the Core Network environment, all computers are within a common network managed by Active Directory® Domain Services (AD DS). The Secure Zone represents higher-security Core Network computers that have additional firewall protection. The Isolated Lab environment is a workgroup that is physically separate from the Core Network, and its computers do not have Internet access. The network security policy states that no information that could identify a specific computer or user may be transferred out of the Isolated Lab.

VAMT User Interface

The following screenshot shows the VAMT graphical user interface.



VAMT provides a single, graphical user interface for managing activations, and for performing other activation-related tasks such as:

- **Adding and removing computers.** You can use VAMT to discover computers in the local environment. VAMT can discover computers by querying AD DS, workgroups, by individual computer name or IP address, or via a general LDAP query.
- **Discovering products.** You can use VAMT to discover Windows, Windows Server, Office, and select other products installed on the client computers.
- **Monitoring activation status.** You can collect activation information about each product, including the last 5 characters of the product key being used, the current license state (such as Licensed, Grace, Unlicensed), and the product edition information.
- **Managing product keys.** You can store multiple product keys and use VAMT to install these keys to remote client products. You can also determine the number of activations remaining for MAKs.
- **Managing activation data.** VAMT stores activation data in a SQL database. VAMT can export this data to other VAMT hosts or to an archive in XML format.

Related topics

- [VAMT Step-by-Step Scenarios](#)

Active Directory-Based Activation overview

5/31/2019 • 2 minutes to read • [Edit Online](#)

Active Directory-Based Activation (ADBA) enables enterprises to activate computers through a connection to their domain. Many companies have computers at offsite locations that use products that are registered to the company. Previously these computers needed to either use a retail key or a Multiple Activation Key (MAK), or physically connect to the network in order to activate their products by using Key Management Services (KMS). ADBA provides a way to activate these products if the computers can join the company's domain. When the user joins their computer to the domain, the ADBA object automatically activates Windows installed on their computer, as long as the computer has a Generic Volume License Key (GVLK) installed. No single physical computer is required to act as the activation object, because it is distributed throughout the domain.

ADBA scenarios

You might use ADBA if you only want to activate domain joined devices.

If you have a server hosting the KMS service, it can be necessary to reactivate licenses if the server is replaced with a new host. This is not necessary when ADBA is used.

ADBA can also make load balancing easier when multiple KMS servers are present since the client can connect to any domain controller. This is simpler than using the DNS service to load balance by configuring priority and weight values.

Some VDI solutions also require that new clients activate during creation before they are added to the pool. In this scenario, ADBA can eliminate potential VDI issues that might arise due to a KMS outage.

ADBA methods

VAMT enables IT Professionals to manage and activate the ADBA object. Activation can be performed using the following methods:

- Online activation: To activate an ADBA forest online, the user selects the **Online activate forest** function, selects a KMS Host key (CSVLK) to use, and gives the ADBA Object a name.
- Proxy activation: For a proxy activation, the user first selects the **Proxy activate forest** function, selects a KMS Host key (CSVLK) to use, gives the ADBA Object a name, and provides a file name to save the CILx file that contains the Installation ID. Next, the user takes that file to a computer that is running VAMT with an Internet connection and then selects the **Acquire confirmation IDs for CILX** function on the VAMT landing page, and provides the original CILx file. When VAMT has loaded the Confirmation IDs into the original CILx file, the user takes this file back to the original VAMT instance, where the user completes the proxy activation process by selecting the **Apply confirmation ID to Active Directory domain** function.

Related topics

- [How to Activate an Active Directory Forest Online](#)
- [How to Proxy Activate an Active Directory Forest](#)

Install and Configure VAMT

5/31/2019 • 2 minutes to read • [Edit Online](#)

This section describes how to install and configure the Volume Activation Management Tool (VAMT).

In this Section

TOPIC	DESCRIPTION
VAMT Requirements	Provides system requirements for installing VAMT on a host computer.
Install VAMT	Describes how to get and install VAMT.
Configure Client Computers	Describes how to configure client computers on your network to work with VAMT.

Related topics

- [Introduction to VAMT](#)

VAMT Requirements

5/31/2019 • 2 minutes to read • [Edit Online](#)

This topic includes info about the product key and system requirements for VAMT.

Product Key Requirements

The Volume Activation Management Tool (VAMT) can be used to perform activations using any of the following types of product keys.

PRODUCT KEY TYPE	WHERE TO OBTAIN
<ul style="list-style-type: none">Multiple Activation Key (MAK)Key Management Service (KMS) host key (CSVLK)KMS client setup keys (GVLK)	Volume licensing keys can only be obtained with a signed contract from Microsoft. For more info, see the Microsoft Volume Licensing portal .
Retail product keys	Obtained at time of product purchase.

System Requirements

The following table lists the system requirements for the VAMT host computer.

ITEM	MINIMUM SYSTEM REQUIREMENT
Computer and Processor	1 GHz x86 or x64 processor
Memory	1 GB RAM for x86 or 2 GB RAM for x64
Hard Disk	16 GB available hard disk space for x86 or 20 GB for x64
External Drive	Removable media (Optional)
Display	1024x768 or higher resolution monitor
Network	Connectivity to remote computers via Windows® Management Instrumentation (TCP/IP) and Microsoft® Activation Web Service on the Internet via HTTPS
Operating System	Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 R2, or Windows Server 2012.
Additional Requirements	<ul style="list-style-type: none">Connection to a SQL Server database. For more info, see Install VAMT.PowerShell 3.0: For Windows 8, Windows 8.1, Windows 10, and Windows Server® 2012, PowerShell is included in the installation. For previous versions of Windows and

ITEM	MINIMUM SYSTEM REQUIREMENT
<p>Windows Server, you must download PowerShell 3.0. To download PowerShell, go to Download Windows PowerShell 3.0.</p> <ul style="list-style-type: none">• If installing on Windows Server 2008 R2, you must also install .NET Framework 3.51.	

Related topics

- [Install and Configure VAMT](#)

Install VAMT

6/10/2019 • 2 minutes to read • [Edit Online](#)

This topic describes how to install the Volume Activation Management Tool (VAMT).

Install VAMT

You install VAMT as part of the Windows Assessment and Deployment Kit (ADK) for Windows 10.

IMPORTANT

VAMT requires local administrator privileges on all managed computers in order to deposit confirmation IDs (CIDs), get the client products' license status, and install product keys. If VAMT is being used to manage products and product keys on the local host computer and you do not have administrator privileges, start VAMT with elevated privileges. For Active Directory-Based Activation use, for best results we recommend running VAMT while logged on as a domain administrator.

NOTE

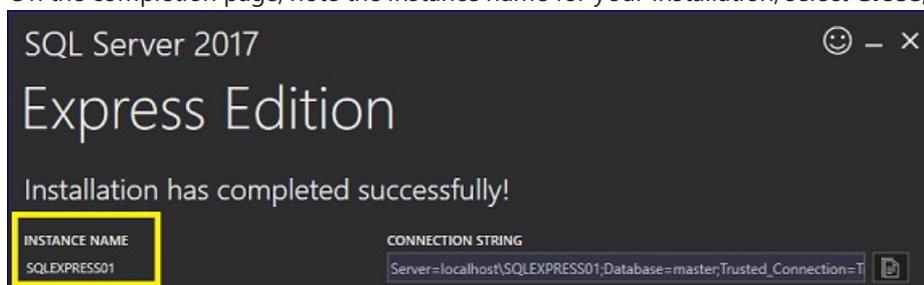
The VAMT Microsoft Management Console snap-in ships as an x86 package.

Requirements

- [Windows Server with Desktop Experience](#), with internet access and all updates applied
- [Windows 10, version 1809 ADK](#)
- [SQL Server 2017 Express](#)

Install SQL Server 2017 Express

1. Download and open the [SQL Server 2017 Express](#) package.
2. Select **Basic**.
3. Accept the license terms.
4. Enter an install location or use the default path, and then select **Install**.
5. On the completion page, note the instance name for your installation, select **Close**, and then select **Yes**.

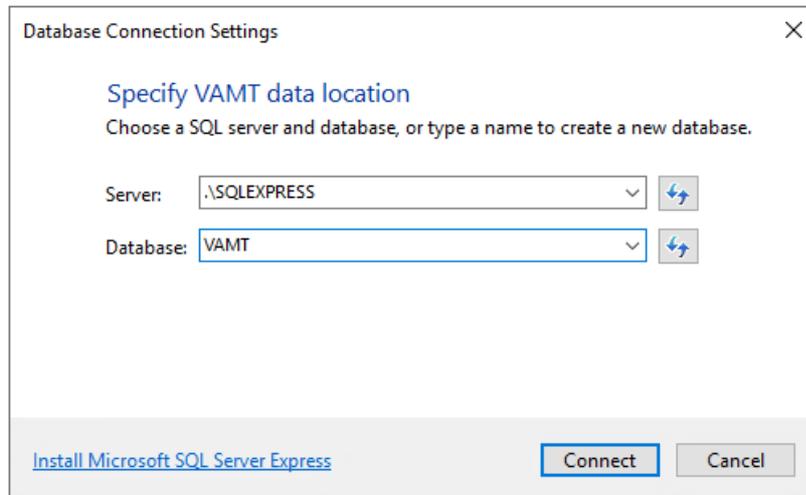


Install VAMT using the ADK

1. Download and open the [Windows 10, version 1809 ADK](#) package.
2. Enter an install location or use the default path, and then select **Next**.
3. Select a privacy setting, and then select **Next**.
4. Accept the license terms.
5. On the **Select the features you want to install** page, select **Volume Activation Management Tool (VAMT)**, and then select **Install**. (You can select additional features to install as well.)
6. On the completion page, select **Close**.

Configure VAMT to connect to SQL Server 2017 Express

1. Open **Volume Active Management Tool 3.1** from the Start menu.
2. Enter the server instance name and a name for the database, select **Connect**, and then select **Yes** to create the database. See the following image for an example.



Uninstall VAMT

To uninstall VAMT using the **Programs and Features** Control Panel:

1. Open **Control Panel** and select **Programs and Features**.
2. Select **Assessment and Deployment Kit** from the list of installed programs and click **Change**. Follow the instructions in the Windows ADK installer to remove VAMT.

2. Click **Windows Firewall with Advanced Security**.
3. Make your changes for each of the following three WMI items, for the applicable Network Profile (Domain, Public, Private):
 - Windows Management Instrumentation (ASync-In)
 - Windows Management Instrumentation (DCOM-In)
 - Windows Management Instrumentation (WMI-In)
4. In the **Windows Firewall with Advanced Security** dialog box, select **Inbound Rules** from the left-hand panel.
5. Right-click the desired rule and select **Properties** to open the **Properties** dialog box.
 - On the **General** tab, select the **Allow the connection** checkbox.
 - On the **Scope** tab, change the Remote IP Address setting from "Local Subnet" (default) to allow the specific access you need.
 - On the **Advanced** tab, verify selection of all profiles that are applicable to the network (Domain or Private/Public).

In certain scenarios, only a limited set of TCP/IP ports are allowed through a hardware firewall. Administrators must ensure that WMI (which relies on RPC over TCP/IP) is allowed through these types of firewalls. By default, the WMI port is a dynamically allocated random port above 1024. The following Microsoft knowledge article discusses how administrators can limit the range of dynamically-allocated ports. This is useful if, for example, the hardware firewall only allows traffic in a certain range of ports. For more info, see [How to configure RPC dynamic port allocation to work with firewalls](#).

Create a registry value for the VAMT to access workgroup-joined computer

Caution This section contains information about how to modify the registry. Make sure to back up the registry before you modify it; in addition, ensure that you know how to restore the registry, if a problem occurs. For more information about how to back up, restore, and modify the registry, see [Windows registry information for advanced users](#).

On the client computer, create the following registry key using regedit.exe.

1. Navigate to `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\system`
2. Enter the following details: **Value Name:** LocalAccountTokenFilterPolicy **Type:** DWORD **Value Data:** 1
Note To discover VAMT-manageable Windows computers in workgroups, you must enable network discovery on each client.

Deployment options

There are several options for organizations to configure the WMI firewall exception for computers:

- **Image.** Add the configurations to the master Windows image deployed to all clients.
- **Group Policy.** If the clients are part of a domain, then all clients can be configured using Group Policy. The Group Policy setting for the WMI firewall exception is found in GPMC.MSC at: **Computer Configuration\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Windows Firewall with Advanced Security\Inbound Rules**.
- **Script.** Execute a script using Microsoft System Center Configuration Manager or a third-party remote script execution facility.
- **Manual.** Configure the WMI firewall exception individually on each client. The above configurations will open an additional port through the Windows Firewall on target computers and should be performed on computers

that are protected by a network firewall. In order to allow VAMT to query the up-to-date licensing status, the WMI exception must be maintained. We recommend administrators consult their network security policies and make clear decisions when creating the WMI exception.

Related topics

- [Install and Configure VAMT](#)

Add and Manage Products

6/6/2019 • 2 minutes to read • [Edit Online](#)

This section describes how to add client computers into the Volume Activation Management Tool (VAMT). After the computers are added, you can manage the products that are installed on your network.

In this Section

TOPIC	DESCRIPTION
Add and Remove Computers	Describes how to add client computers to VAMT.
Update Product Status	Describes how to update the status of product license.
Remove Products	Describes how to remove a product from the product list.

Add and Remove Computers

6/6/2019 • 4 minutes to read • [Edit Online](#)

You can add computers that have any of the supported Windows or Office products installed to a Volume Activation Management Tool (VAMT) database by using the **Discover products** function. You can search for computers in an Active Directory domain, by individual computer name or IP address, in a workgroup, or by a general LDAP query. You can remove computers from a VAMT database by using the **Delete** function. After you add the computers, you can add the products that are installed on the computers by running the **Update license status** function.

Before adding computers, ensure that the Windows Management Instrumentation (WMI) firewall exception required by VAMT has been enabled on all target computers. For more information see [Configure Client Computers](#).

To add computers to a VAMT database

1. Open VAMT.
2. Click **Discover products** in the **Actions** menu in the right-side pane to open the **Discover Products** dialog box.
3. In the **Discover products** dialog box, click **Search for computers in the Active Directory** to display the search options, then click the search option you want to use. You can search for computers in an Active Directory domain, by individual computer name or IP address, in a workgroup, or by a general LDAP query.
 - To search for computers in an Active Directory domain, click **Search for computers in the Active Directory**, then under **Domain Filter Criteria**, in the list of domain names click the name of the domain you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for a specific computer within the domain. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only computer names that start with the letter "a".
 - To search by individual computer name or IP address, click **Manually enter name or IP address**, then enter the full name or IP address in the **One or more computer names or IP addresses separated by commas** text box. Separate multiple entries with a comma. Note that VAMT supports both IPv4 and IPV6 addressing.
 - To search for computers in a workgroup, click **Search for computers in the workgroup**, then under **Workgroup Filter Criteria**, in the list of workgroup names click the name of the workgroup you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for a specific computer within the workgroup. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only computer names that start with the letter "a".
 - To search for computers by using a general LDAP query, click **Search with LDAP query** and enter your query in the text box provided. VAMT will validate only the LDAP query syntax, but will otherwise run the query without further checks.
4. Click **Search**.
5. VAMT searches for the specified computers and adds them to the VAMT database. During the search, VAMT displays the **Finding computers** message shown below. To cancel the search, click **Cancel**. When the search is complete the names of the newly-discovered computers appear in the product list view in the center pane.



Important This step adds only the computers to the VAMT database, and not the products that are installed on the computers. To add the products, you need to run the **Update license status** function.

To add products to VAMT

1. In the **Products** list, select the computers that need to have their product information added to the VAMT database.
2. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
3. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
4. Click **Filter**. VAMT displays the filtered list in the center pane.
5. In the right-side **Actions** pane, click **Update license status** and then click a credential option. Choose **Alternate Credentials** only if you are updating products that require administrator credentials different from the ones you used to log into the computer. If you are supplying alternate credentials, in the **Windows Security** dialog box type the appropriate user name and password and click **OK**.
6. VAMT displays the **Collecting product information** dialog box while it collects the licensing status of all supported products on the selected computers. When the process is finished, the updated licensing status of each product will appear in the product list view in the center pane.

Note If a computer has more than one supported product installed, VAMT adds an entry for each product. The entry appears under the appropriate product heading.

To remove computers from a VAMT database

You can delete a computer by clicking on it in the product list view, and then clicking **Delete** in the **Selected Item** menu in the right-hand pane. In the **Confirm Delete Selected Products** dialog box that appears, click **Yes** to delete the computer. If a computer has multiple products listed, you must delete each product to completely remove the computer from the VAMT database.

Related topics

- [Add and Manage Products](#)

Update Product Status

6/6/2019 • 2 minutes to read • [Edit Online](#)

After you add computers to the VAMT database, you need to use the **Update license status** function to add the products that are installed on the computers. You can also use the **Update license status** at any time to retrieve the most current license status for any products in the VAMT database. To retrieve license status, VAMT must have administrative permissions on all selected computers and Windows Management Instrumentation (WMI) must be accessible through the Windows Firewall. In addition, for workgroup computers, a registry key must be created to enable remote administrative actions under User Account Control (UAC). For more information, see [Configure Client Computers](#).

Note The license-status query requires a valid computer name for each system queried. If the VAMT database contains computers that were added without Personally Identifiable Information, computer names will not be available for those computers, and the status for these computers will not be updated.

Update the license status of a product

1. Open VAMT.
2. In the **Products** list, select one or more products that need to have their status updated.
3. In the right-side **Actions** pane, click **Update license status** and then click a credential option. Choose **Alternate Credentials** only if you are updating products that require administrator credentials different from the ones you used to log into the computer.
4. If you are supplying alternate credentials, in the **Windows Security** dialog box type the appropriate user name and password and click **OK**.

VAMT displays the **Collecting product information** dialog box while it collects the status of all selected products. When the process is finished, the updated licensing status of each product will appear in the product list view in the center pane.

Note If a previously discovered Microsoft Office 2010 product has been uninstalled from the remote computer, updating its licensing status will cause the entry to be deleted from the **Office** product list view, and, consequently, the total number of discovered products will be smaller. However, the Windows installation of the same computer will not be deleted and will always be shown in the **Windows** products list view.

Related topics

- [Add and Manage Products](#)

Remove Products

5/31/2019 • 2 minutes to read • [Edit Online](#)

To remove one or more products from the Volume Activation Management Tool (VAMT), you can delete them from the product list view in the center pane.

To delete one or more products

1. Click a product node in the left-side pane.
2. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
3. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
4. Click **Filter**. VAMT displays the filtered list in the center pane.
5. Select the products you want to delete.
6. Click **Delete** in the **Selected Items** menu in the right-side pane.
7. On the **Confirm Delete Selected Products** dialog box, click **OK**.

Related topics

- [Add and Manage Products](#)

Manage Product Keys

6/6/2019 • 2 minutes to read • [Edit Online](#)

This section describes how to add and remove a product key from the Volume Activation Management Tool (VAMT). After you add a product key to VAMT, you can install that product key on a product or products you select in the VAMT database.

In this Section

TOPIC	DESCRIPTION
Add and Remove a Product Key	Describes how to add a product key to the VAMT database.
Install a Product Key	Describes how to install a product key for specific product.
Install a KMS Client Key	Describes how to install a GVLK (KMS client) key.

Add and Remove a Product Key

5/31/2019 • 2 minutes to read • [Edit Online](#)

Before you can use a Multiple Activation Key (MAK), retail, or KMS Host key (CSVLK) product key, you must first add it to the Volume Activation Management Tool (VAMT) database.

To Add a Product Key

1. Open VAMT.
2. In the left-side pane, right-click the **Product Keys** node to open the **Actions** menu.
3. Click **Add product keys** to open the **Add Product Keys** dialog box.
4. In the **Add Product Keys** dialog box, select from one of the following methods to add product keys:
 - To add product keys manually, click **Enter product key(s) separated by line breaks**, enter one or more product keys separated by line breaks, and click **Add Key(s)**.
 - To import a Comma Separated Values (CSV) file containing a list of product keys, click **Select a product key file to import**, browse to the file location, click **Open** to import the file, and then click **Add Key(s)**.

Note If you are activating a large number of products with a MAK, you should refresh the activation count of the MAK, to ensure that the MAK can support the required number of activations. In the product key list in the center pane, select the MAK and click **Refresh product key data online** in the right-side pane to contact Microsoft and retrieve the number of remaining activations for the MAK. This step requires Internet access. You can only retrieve the remaining activation count for MAKs.

Remove a Product Key

- To remove a product key from the list, simply select the key in the list and click **Delete** on the **Selected Items** menu in the right-side pane. Click **Yes** to confirm deletion of the product key. Removing a product key from the VAMT database will not affect the activation state of any products or computers on the network.

Related topics

- [Manage Product Keys](#)

Install a Product Key

6/6/2019 • 2 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) to install retail, Multiple Activation Key (MAK), and KMS Host key (CSVLK).

To install a Product key

1. Open VAMT.
2. In the left-side pane, click the product that you want to install keys onto.
3. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
5. Click **Filter**.
6. In the products list view in the center pane, sort the list if needed and then select the products that need to have keys installed. You can use the **CTRL** key or the **SHIFT** key to select more than one product.
7. Click **Install product key** in the **Selected Items** menu in the right-side pane to display the **Install Product Key** dialog box.
8. The **Select Product Key** dialog box displays the keys that are available to be installed. Under **Recommended MAKs**, VAMT might display one or more recommended MAK based on the selected products. You can select a recommended product key or a product key from the **All Product Keys** list. Use the scroll bar if you need to view the **Description** for each key. When you have selected the product key you want to install, click **Install Key**. Note that only one key can be installed at a time.
9. VAMT displays the **Installing product key** dialog box while it attempts to install the product key for the selected products. When the process is finished, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears.

The same status is shown under the **Status of Last Action** column in the product list view in the center pane.

Note Product key installation will fail if VAMT finds mismatched key types or editions. VAMT will display the failure status and will continue the installation for the next product in the list. For more information on choosing the correct MAK or KMS Host key (CSVLK), see [How to Choose the Right Volume License Key for Windows](#).

Related topics

- [Manage Product Keys](#)

Install a KMS Client Key

5/31/2019 • 2 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) to install Generic Volume License Key (GVLK), or KMS client, product keys. For example, if you are converting a MAK-activated product to KMS activation.

Note By default, volume license editions of Windows Vista, Windows® 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server® 2012, and Microsoft® Office 2010 use KMS for activation. GVLKs are already installed in volume license editions of these products.

To install a KMS Client key

1. Open VAMT.
2. In the left-side pane click **Products** to open the product list view in the center pane.
3. In the products list view in the center pane, select the products that need to have GVLKs installed. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
5. Click **Filter**. VAMT displays the filtered list in the center pane.
6. Click **Install product key** in the **Selected Items** menu in the right-side pane to display the **Install Product Key** dialog box.
7. The **Install Product Key** dialog box displays the keys that are available to be installed.
8. Select the **Automatically select an AD or KMS client key** option and then click **Install Key**.

VAMT displays the **Installing product key** dialog box while it attempts to install the product key for the selected products. When the process is finished, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears.

The same status is shown under the **Status of Last Action** column in the product list view in the center pane.

Related topics

- [Perform KMS Activation](#)

Manage Activations

6/6/2019 • 2 minutes to read • [Edit Online](#)

This section describes how to activate a client computer, by using a variety of activation methods.

In this Section

TOPIC	DESCRIPTION
Perform Online Activation	Describes how to activate a client computer over the Internet.
Perform Proxy Activation	Describes how to perform volume activation for client products that do not have Internet access.
Perform KMS Activation	Describes how perform volume activation using the Key Management Service (KMS).
Perform Local Reactivation	Describes how to reactivate an operating system or Office program that was reinstalled.
Activate an Active Directory Forest Online	Describes how to use Active Directory-Based Activation to online activate an Active Directory forest.
Activate by Proxy an Active Directory Forest	Describes how to use Active Directory-Based Activation to proxy activate an Active Directory forest that is not connected to the Internet.

Perform Online Activation

5/31/2019 • 2 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) to enable client products to be activated over the Internet. You can install the client products with any kind of product key that is eligible for online activation—Multiple Activation Key (MAK), retail, and Windows Key Management Services (KMS) host key.

Requirements

Before performing online activation, ensure that the network and the VAMT installation meet the following requirements:

- VAMT is installed on a central computer that has network access to all client computers.
- Both the VAMT host and client computers have Internet access.
- The products that you want to activate are added to VAMT.
- VAMT has administrative permissions on all computers that you intend to activate, and that Windows Management Instrumentation (WMI) can be accessed through the Windows firewall. For more information, see [Configure Client Computers](#).

The product keys that are installed on the client products must have a sufficient number of remaining activations. If you are activating a MAK key, you can retrieve the remaining number of activations for that key by selecting the MAK in the product key list in the center pane and then clicking **Refresh product key data online** in the right-side pane. This retrieves the number of remaining activations for the MAK from Microsoft. Note that this step requires Internet access and that the remaining activation count can only be retrieved for MAKs.

To Perform an Online Activation

To perform an online activation

1. Open VAMT.
2. In the products list view in the center pane, sort the list if necessary. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
3. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
4. Click **Filter**. VAMT displays the filtered list in the center pane.
5. Select the products that you want to activate. You can use the **CTRL** key or the **SHIFT** key to select more than one product.
6. Click **Activate** in the **Selected Items** menu in the right-side **Actions** pane and then point to **Activate**. If the **Actions** pane is not displayed, click the Show/Hide Action Pane button, which is located on the toolbar to the right of the Help button.
7. Point to **Online activate**, and then select the appropriate credential option. If you click the **Alternate Credentials** option, you will be prompted to enter an alternate user name and password.

8. VAMT displays the **Activating products** dialog box until it completes the requested action. When activation is complete, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears.

The same status is shown under the **Status of Last Action** column in the products list view in the center pane.

Note Online activation does not enable you to save the Confirmation IDs (CIDs). As a result, you cannot perform local reactivation.

Note

You can use online activation to select products that have different key types and activate the products at the same time.

Related topics

- [Manage Activations](#)

Perform Proxy Activation

6/6/2019 • 3 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) to perform activation for client computers that do not have Internet access. The client products can be installed with any type of product key that is eligible for proxy activation: Multiple activation Key (MAK), KMS Host key (CSVLK), or retail key.

In a typical proxy-activation scenario, the VAMT host computer distributes a MAK to one or more client computers and collects the installation ID (IID) from each computer. The VAMT host computer sends the IIDs to Microsoft on behalf of the client computers and obtains the corresponding Confirmation IDs (CIDs). The VAMT host computer then installs the CIDs on the client computer to complete the activation. Using this activation method, only the VAMT host computer needs Internet access.

Note For workgroups that are completely isolated from any larger network, you can still perform MAK, KMS Host key (CSVLK), or retail proxy activation. This requires installing a second instance of VAMT on a computer within the isolated group and using removable media to transfer activation data between that computer and another VAMT host computer that has Internet access. For more information about this scenario, see [Scenario 2: Proxy Activation](#). Similarly, you can proxy activate a KMS Host key (CSVLK) located in an isolated network. You can also proxy activate a KMS Host key (CSVLK) in the core network if you do not want the KMS host computer to connect to Microsoft over the Internet.

Requirements

Before performing proxy activation, ensure that your network and the VAMT installation meet the following requirements:

- There is an instance of VAMT that is installed on a computer that has Internet access. If you are performing proxy activation for an isolated workgroup, you also need to have VAMT installed on one of the computers in the workgroup.
- The products to be activated have been added to VAMT and are installed with a retail product key, a KMS Host key (CSVLK) or a MAK. If the products have not been installed with a proper product key, refer to the steps in the [Add and Remove a Product Key](#) section for instructions on how to install a product key.
- VAMT has administrative permissions on all products to be activated and Windows Management Instrumentation (WMI) is accessible through the Windows firewall.
- For workgroup computers, a registry key must be created to enable remote administrative actions under User Account Control (UAC). For more information, see [Configure Client Computers](#). The product keys that are installed on the client products must have a sufficient number of remaining activations. If you are activating a MAK key, you can retrieve the remaining number of activations for that key by selecting the MAK in the product key list in the center pane and then clicking **Refresh product key data online** in the right-side pane. This retrieves the number of remaining activations for the MAK from Microsoft. Note that this step requires Internet access and that the remaining activation count can only be retrieved for MAKs.

To Perform Proxy Activation

To perform proxy activation

1. Open VAMT.
2. If necessary, install product keys. For more information see:
 - [Install a Product Key](#) to install retail, MAK, or KMS Host key (CSVLK).

- [Install a KMS Client Key](#) to install GVLK (KMS client) keys.
3. In the **Products** list in the center pane, select the individual products to be activated. You can use the **Filter** function to narrow your search for products by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
 4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
 5. Click **Filter**. VAMT displays the filtered list in the center pane.
 6. In the right-side pane, click **Activate** and then click **Proxy activate** to open the **Proxy Activate** dialog box.
 7. In the **Proxy Activate** dialog box click **Apply Confirmation ID, apply to selected machine(s) and activate**.
 8. If you are activating products that require administrator credentials different from the ones you are currently using, select the **Use Alternate Credentials** checkbox.
 9. Click **OK**.
 10. VAMT displays the **Activating products** dialog box until it completes the requested action. If you selected the **Alternate Credentials** option, you will be prompted to enter the credentials.

Note You can use proxy activation to select products that have different key types and activate the products at the same time.

Perform KMS Activation

5/31/2019 • 3 minutes to read • [Edit Online](#)

The Volume Activation Management Tool (VAMT) can be used to perform volume activation using the Key Management Service (KMS). You can use VAMT to activate Generic Volume Licensing Keys, or KMS client keys, on products accessible to VAMT. GVLKs are the default product keys used by the volume-license editions of Windows Vista, Windows 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server® 2012, and Microsoft Office 2010. GVLKs are already installed in volume-license editions of these products.

Requirements

Before configuring KMS activation, ensure that your network and VAMT installation meet the following requirements:

- KMS host is set up and enabled.
- KMS clients can access the KMS host.
- VAMT is installed on a central computer with network access to all client computers.
- The products to be activated have been added to VAMT. For more information on adding product keys, see [Install a KMS Client Key](#).
- VAMT has administrative permissions on all computers to be activated, and Windows Management Instrumentation (WMI) is accessible through the Windows Firewall. For more information, see [Configure Client Computers](#).

To configure devices for KMS activation

To configure devices for KMS activation

1. Open VAMT.
2. If necessary, set up the KMS activation preferences. If you don't need to set up the preferences, skip to step 6 in this procedure. Otherwise, continue to step 2.
3. To set up the preferences, on the menu bar click **View**, then click **Preferences** to open the **Volume Activation Management Tool Preferences** dialog box.
4. Under **Key Management Services host selection**, select one of the following options:
 - **Find a KMS host automatically using DNS (default)**. If you choose this option, VAMT first clears any previously configured KMS host on the target computer and instructs the computer to query the Domain Name Service (DNS) to locate a KMS host and attempt activation.
 - **Find a KMS host using DNS in this domain for supported products**. Enter the domain name. If you choose this option, VAMT first clears any previously configured KMS host on the target computer and instructs the computer to query the DNS in the specified domain to locate a KMS host and attempt activation.
 - **Use specific KMS host**. Enter the KMS host name and KMS host port. For environments which do not use DNS for KMS host identification, VAMT sets the specified KMS host name and KMS host port on the target computer, and then instructs the computer to attempt activation with the specific KMS host.
5. Click **Apply**, and then click **OK** to close the **Volume Activation Management Tool Preferences** dialog box.
6. Select the products to be activated by selecting individual products in the product list view in the center pane. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box. In the **Filter Products** dialog box, you can filter the list by computer

name, product name, product key type, license status, or by any combination of these options.

- To filter the list by computer name, enter a name in the **Computer Name** box.
- To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.

7. Click **Filter**. VAMT displays the filtered list in the center pane.
8. In the right-side pane, click **Activate** in the **Selected Items** menu, and then click **Volume activate**.
9. Click a credential option. Choose **Alternate credentials** only if you are activating products that require administrator credentials different from the ones you are currently using.
10. If you are supplying alternate credentials, at the prompt, type the appropriate user name and password and click **OK**. VAMT displays the **Volume Activation** dialog box until it completes the requested action. When the process is finished, the updated activation status of each product appears in the product list view in the center pane.

Perform Local Reactivation

5/31/2019 • 2 minutes to read • [Edit Online](#)

If you reinstall Windows® or Microsoft® Office 2010 on a computer that was initially activated using proxy activation (MAK, retail, or CSLVK (KMS host)), and have not made significant changes to the hardware, use this local reactivation procedure to reactivate the program on that computer. Local reactivation relies upon data that was created during the initial proxy activation and stored in the Volume Activation Management Tool (VAMT) database. The database contains the installation ID (IID) and confirmation ID (Pending CID). Local reactivation uses this data to reapply the CID and reactivate those products. Reapplying the same CID conserves the remaining activations on the key.

Note During the initial proxy activation, the CID is bound to a digital “fingerprint”, which is calculated from values assigned to several different hardware components in the computer. If the computer has had significant hardware changes, this fingerprint will no longer match the CID. In this case, you must obtain a new CID for the computer from Microsoft.

To Perform a Local Reactivation

To perform a local reactivation

1. Open VAMT. Make sure that you are connected to the desired database.
2. In the left-side pane, click the product you want to reactivate to display the products list.
3. In the product list view in the center pane, select the desired products to be reactivated. You can sort the list by computer name by clicking on the **Computer Name** heading. You can also use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
5. Click **Filter**. VAMT displays the filtered list in the center pane.
6. In the right-side pane, click **Activate**, and then click **Apply Confirmation ID**.
7. Click a credential option. Choose **Alternate credentials** only if you are reactivating products that require administrator credentials different from the ones you are currently using.
8. If you are supplying alternate credentials, in the **Windows Security** dialog box type the appropriate user name and password and click **OK**.

VAMT displays the **Apply Confirmation ID** dialog box.
9. If you are using a different product key than the product key used for initial activation, you must complete a new activation to obtain a new CID.
10. If you are activating a product that requires administrator credentials different from the ones you are currently using, select the **Use Alternate Credentials** check box.
11. Click **OK**.

Related topics

- [Manage Activations](#)

Activate an Active Directory Forest Online

5/31/2019 • 2 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) Active Directory-Based Activation (ADBA) function to activate an Active Directory (AD) forest over the Internet. ADBA enables certain products to inherit activation from the domain.

Important ADBA is only applicable to Generic Volume License Keys (GVLKs) and KMS Host keys (CSVLKs). To use ADBA, one or more KMS Host keys (CSVLKs) must be installed on the AD forest, and client keys (GVLKs) must be installed on the client products.

Requirements

Before performing online activation, ensure that the network and the VAMT installation meet the following requirements:

- VAMT is installed on a host computer that has Internet access.
- VAMT has administrative permissions to the Active Directory domain.
- The KMS Host key (CSVLK) you intend to use is added to VAMT in the **Product Keys** node.

To perform an online Active Directory forest activation

1. Open VAMT.
2. In the left-side pane, click the **Active Directory-Based Activation** node.
3. In the right-side **Actions** pane, click **Online activate forest** to open the **Install Product Key** dialog box.
4. In the **Install Product Key** dialog box, select the KMS Host key (CSVLK) that you want to apply to the AD forest.
5. If required, enter a new Active Directory-Based Activation Object name

Important If you want to rename the ADBA object, you must do it now. After you click **Install Key**, the name cannot be changed.

6. Click **Install Key**.
7. VAMT displays the **Activating Active Directory** dialog box until it completes the requested action.

The activated object and the date that it was created appear in the **Active Directory-Based Activation** node in the center pane.

Related topics

- [Scenario 1: Online Activation](#)
- [Add and Remove Computers](#)

Activate by Proxy an Active Directory Forest

5/31/2019 • 3 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) Active Directory-Based Activation (ADBA) function to activate by proxy an Active Directory (AD) forest for an isolated workgroup that does not have Internet access. ADBA enables certain volume products to inherit activation from the domain.

Important ADBA is only applicable to Generic Volume License Keys (GVLKs) and KMS Host key (CSVLK). To use ADBA, one or more KMS Host keys (CSVLK) must be installed on the AD forest, and client keys (GVLKs) must be installed on the client products.

In a typical proxy-activation scenario, the VAMT host computer distributes a product key to one or more client computers and collects the installation ID (IID) from each computer. The VAMT host computer sends the IIDs to Microsoft on behalf of the client computers and obtains the corresponding Confirmation IDs (CIDs). The VAMT host computer then installs the CIDs on the client computer to complete the activation. If you use this activation method, only the VAMT host computer needs to have Internet access.

Note For workgroups that are isolated from any larger network, you can still perform an AD forest activation. This requires installing a second instance of VAMT on a computer in the isolated group and using removable media to transfer activation data between that computer and another VAMT host computer that has Internet access. You can also activate by proxy a KMS Host key (CSVLK) in the core network if you do not want the host computer to connect to Microsoft over the Internet.

Requirements

Before performing proxy activation, ensure that the network and the VAMT installation meet the following requirements:

- There is an instance of VAMT that is installed on a computer that has Internet access. If you are performing proxy activation for an isolated workgroup, you must also have VAMT installed on one of the computers in the workgroup.
- VAMT has administrative permissions to the Active Directory domain.

To perform an Active Directory forest proxy activation

1. Open VAMT.
2. In the left-side pane, click the **Active Directory-Based Activation** node.
3. In the right-side **Actions** pane, click **Proxy activate forest** to open the **Install Product Key** dialog box.
4. In the **Install Product Key** dialog box, select the KMS Host key (CSVLK) that you want to activate.
5. If you want to rename the ADBA object, enter a new Active Directory-Based Activation Object name. If you want to rename the ADBA object, you must do it now. After you click **Install Key**, the name cannot be changed.
6. Enter the name of the file where you want to save the offline installation ID, or browse to the file location and then click **Open**. If you are activating an AD forest in an isolated workgroup, save the .cilx file to a removable media device.
7. Click **Install Key**. VAMT displays the **Activating Active Directory** dialog box until it completes the requested action. The activated object and the date that it was created appear in the **Active Directory-Based Activation** node in the center pane.
8. Insert the removable media into the VAMT host that has Internet access. Make sure that you are on the root node, and that the **Volume Activation Management Tool** view is displayed in the center pane.
9. In the right-side **Actions** pane, click **Acquire confirmation IDs for CILX** to open the **Acquire confirmation**

IDs for file dialog box.

10. In the **Acquire confirmation IDs for file** dialog box, browse to where the .cilx file you exported from the isolated workgroup host computer is located. Select the file, and then click **Open**. VAMT displays an **Acquiring Confirmation IDs** message while it contacts Microsoft and acquires the CIDs.
11. When the CID collection process is complete, VAMT displays a **Volume Activation Management Tool** message that shows how many confirmation IDs were successfully acquired, and the name of the file to which the IDs were saved. Click **OK** to close the message.
12. Remove the storage device that contains the .cilx file from the Internet-connected VAMT host computer and insert it into the VAMT host computer in the isolated workgroup.
13. Open VAMT and then click the **Active Directory-Based Activation** node in the left-side pane.
14. In the right-side **Actions** pane, click **Apply confirmation ID to Active Directory domain**, browse to the .cilx file and then click **Open**.

VAMT displays the **Activating Active Directory** dialog box until it completes the requested action. The activated object and the date that it was created appear in the **Active Directory-Based Activation** node in the center pane.

Related topics

- [Add and Remove Computers](#)

Manage VAMT Data

5/31/2019 • 2 minutes to read • [Edit Online](#)

This section describes how to save, import, export, and merge a Computer Information List (CILX) file using the Volume Activation Management Tool (VAMT).

In this Section

TOPIC	DESCRIPTION
Import and Export VAMT Data	Describes how to import and export VAMT data.
Use VAMT in Windows PowerShell	Describes how to access Windows PowerShell and how to import the VAMT PowerShell module.

Import and Export VAMT Data

5/31/2019 • 2 minutes to read • [Edit Online](#)

You can use the Volume Activation Management Tool (VAMT) to import product-activation data from a Computer Information List (.cilx or .cil) file into SQL Server, and to export product-activation data into a .cilx file. A .cilx file is an XML file that stores computer and product-activation data. You can import data or export data during the following scenarios:

- Import and merge data from previous versions of VAMT.
- Export data to use to perform proxy activations.

Warning Editing a .cilx file using an application other than VAMT can corrupt the .cilx file and is not supported.

Import VAMT Data

To import data into VAMT

1. Open VAMT.
2. In the right-side **Actions** pane, click **Import list** to open the **Import List** dialog box.
3. In the **Import List** dialog box, navigate to the .cilx file location, select the file, and click **Open**.
4. In the **Volume Activation Management Tool** dialog box, click **OK** to begin the import. VAMT displays a progress message while the file is being imported. Click **OK** when a message appears and confirms that the import has completed successfully.

Export VAMT Data

Exporting VAMT data from a non-Internet-connected VAMT host computer is the first step of proxy activation using multiple VAMT hosts. To export product-activation data to a .cilx file:

1. In the left-side pane, you can click a product you want to export data for, or click **Products** if the list contains data for all products.
2. If you want to export only part of the data in a product list, in the product list view in the center pane select the products you want to export.
3. In the right-side **Actions** pane on, click **Export list** to open the **Export List** dialog box.
4. In the **Export List** dialog box, click **Browse** to navigate to the .cilx file.
5. Under **Export options**, select one of the following data-type options:
 - Export products and product keys
 - Export products only
 - Export proxy activation data only. Selecting this option ensures that the export contains only the licensing information required for the proxy web service to obtain CIDs from Microsoft. No Personally Identifiable Information (PII) is contained in the exported .cilx file when this selection is checked.
6. If you have selected products to export, select the **Export selected product rows only** check box.
7. Click **Save**. VAMT displays a progress message while the data is being exported. Click **OK** when a message appears and confirms that the export has completed successfully.

Related topics

- [Perform Proxy Activation](#)

Use VAMT in Windows PowerShell

6/6/2019 • 2 minutes to read • [Edit Online](#)

The Volume Activation Management Tool (VAMT) PowerShell cmdlets can be used to perform the same functions as the Vamt.exe command-line tool. **To install PowerShell 3.0**

- VAMT PowerShell cmdlets require Windows PowerShell, which is included in Windows 10, Windows 8 and Windows Server® 2012. You can download PowerShell for Windows 7 or other operating systems from the [Microsoft Download Center](#). **To install the Windows Assessment and Deployment Kit**
- In addition to PowerShell, you must import the VAMT PowerShell module. The module is included in the VAMT 3.0 folder after you install the Windows Assessment and Deployment Kit (Windows ADK). **To prepare the VAMT PowerShell environment**
- To open PowerShell with administrative credentials, click **Start** and type "PowerShell" to locate the program. Right-click **Windows PowerShell**, and then click **Run as administrator**. To open PowerShell in Windows 7, click **Start**, click **All Programs**, click **Accessories**, click **Windows PowerShell**, right-click **Windows PowerShell**, and then click **Run as administrator**.

Important

If you are using a computer that has an 64-bit processor, select **Windows PowerShell (x86)**. VAMT PowerShell cmdlets are supported for the x86 architecture only. You must use an x86 version of Windows PowerShell to import the VAMT module, which are available in these directories:

- The x86 version of PowerShell is available in
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
- The x86 version of the PowerShell ISE is available in
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell_ise.exe
- For all supported operating systems you can use the VAMT PowerShell module included with the Windows ADK. By default, the module is installed with the Windows ADK in the VAMT folder. Change directories to the directory where VAMT is located.

For example, if the Windows ADK is installed in the default location of

```
C:\Program Files(x86)\Windows Kits\10
```

, type:

```
cd "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\VAMT 3.0"
```

- Import the VAMT PowerShell module. To import the module, type the following at a command prompt:

```
Import-Module .\VAMT.psd1
```

Where **Import-Module** imports a module only into the current session. To import the module into all sessions, add an **Import-Module** command to a Windows PowerShell profile. For more information about profiles, type `get-help about_profiles`.

To Get Help for VAMT PowerShell cmdlets

You can view all of the help sections for a VAMT PowerShell cmdlet, or you can view only the section that you are interested in. To view all of the Help content for a VAMT cmdlet, type:

```
get-help <cmdlet name> -all
```

For example, type:

```
get-help get-VamtProduct -all
```

Warning

The update-help cmdlet is not supported for VAMT PowerShell cmdlets. To view online help for VAMT cmdlets, you can use the -online option with the get-help cmdlet. For more information, see [Volume Activation Management Tool \(VAMT\) Cmdlets in Windows PowerShell](#).

To view VAMT PowerShell Help sections

1. To get the syntax to use with a cmdlet, type the following at a command prompt:

```
get-help <cmdlet name>
```

For example, type:

```
get-help get-VamtProduct
```

2. To see examples using a cmdlet, type:

```
get-help <cmdlet name> -examples
```

For example, type:

```
get-help get-VamtProduct -examples
```

VAMT Step-by-Step Scenarios

5/31/2019 • 2 minutes to read • [Edit Online](#)

This section provides step-by-step instructions on implementing the Volume Activation Management Tool (VAMT) in typical environments. VAMT supports many common scenarios; the scenarios in this section describe some of the most common to get you started.

In this Section

TOPIC	DESCRIPTION
Scenario 1: Online Activation	Describes how to distribute Multiple Activation Keys (MAKs) to products installed on one or more connected computers within a network, and how to instruct these products to contact Microsoft over the Internet for activation.
Scenario 2: Proxy Activation	Describes how to use two VAMT host computers — the first one with Internet access and a second computer within an isolated workgroup — as proxies to perform MAK volume activation for workgroup computers that do not have Internet access.
Scenario 3: KMS Client Activation	Describes how to use VAMT to configure client products for Key Management Service (KMS) activation. By default, volume license editions of Windows 10, Windows Vista, Windows® 7, Windows 8, Windows Server 2008, Windows Server 2008 R2, or Windows Server® 2012, and Microsoft® Office 2010 use KMS for activation.

Related topics

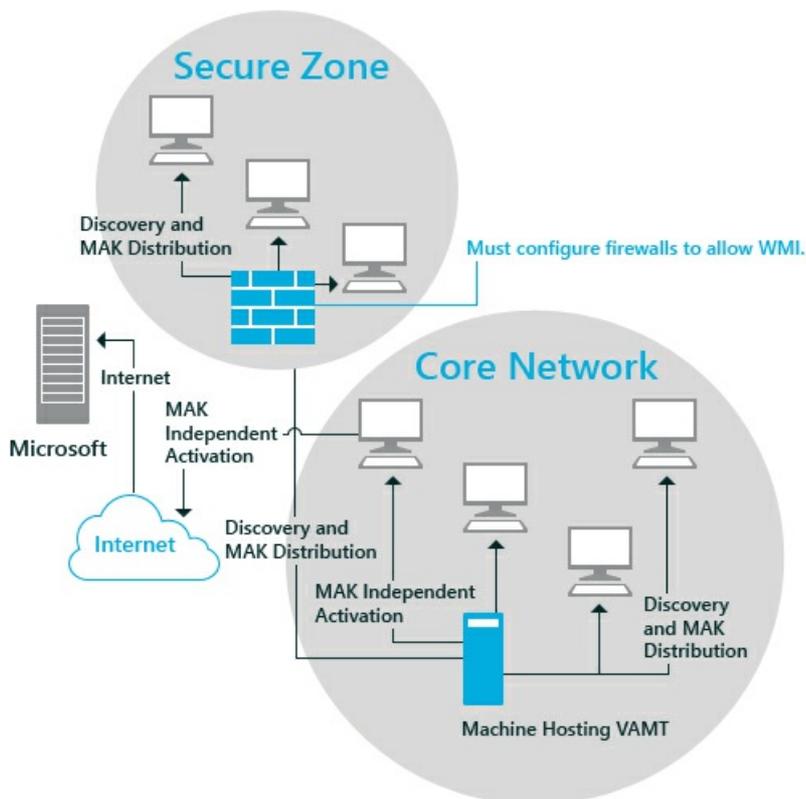
- [Introduction to VAMT](#)

Scenario 1: Online Activation

6/6/2019 • 10 minutes to read • [Edit Online](#)

In this scenario, the Volume Activation Management Tool (VAMT) is deployed in the Core Network environment. VAMT is installed on a central computer that has network access to all of the client computers. Both the VAMT host and the client computers have Internet access. The following illustration shows a diagram of an online activation scenario for Multiple Activation Keys (MAKs). You can use this scenario for online activation of the following key types:

- Multiple Activation Key (MAK)
- Windows Key Management Service (KMS) keys:
 - KMS Host key (CSVLK)
 - Generic Volume License Key (GVLK), or KMS client key
- Retail The Secure Zone represents higher-security Core Network computers that have additional firewall protection.



In This Topic

- [Install and start VAMT on a networked host computer](#)
- [Configure the Windows Management Instrumentation firewall exception on target computers](#)
- [Connect to VAMT database](#)
- [Discover products](#)
- [Sort and filter the list of computers](#)
- [Collect status information from the computers in the list](#)
- [Add product keys and determine the remaining activation count](#)
- [Install the product keys](#)
- [Activate the client products](#)

Step 1: Install and start VAMT on a networked host computer

1. Install VAMT on the host computer.
2. Click the VAMT icon in the **Start** menu to open VAMT.

Step 2: Configure the Windows Management Instrumentation firewall exception on target computers

- Ensure that the Windows Management Instrumentation (WMI) firewall exception has been enabled for all target computers. For more information, see [Configure Client Computers](#).

Note To retrieve product license status, VAMT must have administrative permissions on the remote computers and WMI must be available through the Windows Firewall. In addition, for workgroup computers, a registry key must be created to enable remote administrative actions under User Account Control (UAC). For more information, see [Configure Client Computers](#).

Step 3: Connect to a VAMT database

1. If you are not already connected to a database, the **Database Connection Settings** dialog box appears when you open VAMT. Select the server and database where the keys that must be activated are located.
2. Click **Connect**.
3. If you are already connected to a database, VAMT displays an inventory of the products and product keys in the center pane, and a license overview of the computers in the database. If you need to connect to a different database, click **Successfully connected to Server** to open **the Database Connection Settings** dialog box. For more information about how to create VAMT databases and adding VAMT data, see [Manage VAMT Data](#)

Step 4: Discover products

1. In the left-side pane, in the **Products** node Products, click the product that you want to activate.
2. To open the **Discover Products** dialog box, click **Discover products** in the **Actions** menu in the right-side pane.
3. In the **Discover Products** dialog box, click **Search for computers in the Active Directory** to display the search options, and then click the search options that you want to use. You can search for computers in an Active Directory domain, by individual computer name or IP address, in a workgroup, or by a general Lightweight Directory Access Protocol (LDAP) query:
 - To search for computers in an Active Directory domain, click **Search for computers in the Active Directory**. Then under **Domain Filter Criteria**, in the list of domain names click the name of the domain that you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for specific computers in the domain. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only those computer names that start with the letter "a".
 - To search by individual computer name or IP address, click **Manually enter name or IP address**. Then enter the full name or IP address in the **One or more computer names or IP addresses separated by commas** text box. Separate multiple entries with a comma. Note that VAMT supports both IPv4 and IPV6 addressing.
 - To search for computers in a workgroup, click **Search for computers in the workgroup**. Then under **Workgroup Filter Criteria**, in the list of workgroup names, click the name of the workgroup that you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for a specific computer in the workgroup. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only computer names that start with the letter "a".
 - To search for computers by using a general LDAP query, click **Search with LDAP query** and enter your

query in the text box that appears. VAMT will validate the LDAP query syntax, but will otherwise run the query without additional checks.

4. Click **Search**.

When the search is complete, the products that VAMT discovers appear in the product list view in the center pane.

Step 5: Sort and filter the list of computers

You can sort the list of products so that it is easier to find the computers that require product keys to be activated:

1. On the menu bar at the top of the center pane, click **Group by**, and then click **Product**, **Product Key Type**, or **License Status**.
2. To sort the list further, you can click one of the column headings to sort by that column.
3. You can also use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by product name, product key type, or license status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
5. Click **Filter**. VAMT displays the filtered list in the product list view in the center pane.

Step 6: Collect status information from the computers in the list

To collect the status from select computers in the database, you can select computers in the product list view by using one of the following methods:

- To select a block of consecutively listed computers, click the first computer that you want to select, and then click the last computer while pressing the **Shift** key.
- To select computers which are not listed consecutively, hold down the **Ctrl** key and select each computer for which you want to collect the status information. **To collect status information from the selected computers**
- In the right-side **Actions** pane, click **Update license status** in the **Selected Items** menu and then click a credential option. Choose **Alternate Credentials** only if you are updating products that require administrator credentials that are different from the ones that you used to log on to the computer. Otherwise, click **Current Credentials** and continue to step 2. If you are supplying alternate credentials, in the **Windows Security** dialog box, type the appropriate user name and password and then click **OK**.
- VAMT displays the **Collecting product information** dialog box while it collects the license status of all supported products on the selected computers. When the process is finished, the updated license status of each product will appear in the product list view in the center pane.

Note

If a computer has more than one supported product installed, VAMT adds an entry for each product. The entry appears under the appropriate product heading.

Step 7: Add product keys and determine the remaining activation count

1. Click the **Product Keys** node in the left-side pane, and then click **Add Product Keys** in the right-side pane to open the **Add Product Keys** dialog box.

2. In the **Add Product Key** dialog box, you can select from one of the following methods to add product keys:

- To add product keys manually, click **Enter product key(s) separated by line breaks**, enter one or more product keys, and then click **Add Key(s)**.
- To import a Comma Separated Values File (CSV) that contains a list of product keys, click **Select a product key file to import**, browse to the file location, click **Open** to import the file, and then click **Add Key(s)**.

The keys that you have added appear in the **Product Keys** list view in the center pane.

Important If you are activating many products with a MAK, refresh the activation count of the MAK to ensure that the MAK can support the required number of activations. In the product key list in the center pane, select the MAK and then click **Refresh product key data online** in the right-side pane to contact Microsoft and retrieve the number of remaining activations for the MAK. This step requires Internet access. You can only retrieve the remaining activation count for MAKs.

Step 8: Install the product keys

1. In the left-side pane, click the product that you want to install keys on to.
2. If necessary, sort and filter the list of products so that it is easier to find the computers that must have a product key installed. See [Step 5: Sort and filter the list of computers](#).
3. In the **Products** list view pane, select the individual products which must have keys installed. You can use the **CTRL** key or the **SHIFT** key to select more than one product.
4. Click **Install product key** in the **Selected Items** menu in the right-side pane to display the **Install Product Key** dialog box.
5. The **Select Product Key** dialog box displays the keys that are available to be installed. Under **Recommended MAKs**, VAMT might display one or more recommended MAKs based on the selected products. If you are installing a MAK you can select a recommended product key or any other MAK from the **All Product Keys List**. If you are not installing a MAK, select a product key from the **All Product Keys** list. Use the scroll bar if you want to view the **Description** for each key. When you have selected the product key that you want to install, click **Install Key**. Note that only one key can be installed at a time.
6. VAMT displays the **Installing product key** dialog box while it attempts to install the product key for the selected products. When the process is finished, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears.

The same status appears under the **Status of Last Action** column in the product list view in the center pane. **Note**

Product key installation will fail if VAMT finds mismatched key types or editions. VAMT will display the failure status and will continue the installation for the next product in the list. For more information on choosing the correct product key, see [How to Choose the Right Volume License Key for Windows](#).

Step 9: Activate the client products

1. Select the individual products that you want to activate in the list-view pane.
2. On the menu bar, click **Action**, point to **Activate** and point to **Online activate**. You can also right-click the selected computers(s) to display the **Action** menu, point to **Activate** and point to **Online activate**. You can also click **Activate** in the **Selected Items** menu in the right-hand pane to access the **Activate** option.
3. If you are activating product keys using your current credential, click **Current credential** and continue to step 5. If you are activating products that require an administrator credential that is different from the one

you are currently using, click the **Alternate credential** option.

4. Enter your alternate user name and password and click **OK**.
5. The **Activate** option contacts the Microsoft product-activation server over the Internet and requests activation for the selected products. VAMT displays the **Activating products** dialog box until the requested actions are completed.

Note Installing a MAK and overwriting the GVLK on client products must be done with care. If the RTM version of Windows Vista has been installed on the computer for more than 30 days, then its initial grace period has expired. As a result, it will enter Reduced Functionality Mode (RFM) if online activation is not completed successfully before the next logon attempt. However, you can use online activation to recover properly configured computers from RFM, as long as the computers are available on the network.

RFM only applies to the RTM version of Windows Vista or the retail editions of Microsoft Office 2010. Windows Vista with SP1 or later, Windows 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and volume editions of Office 2010 will not enter RFM.

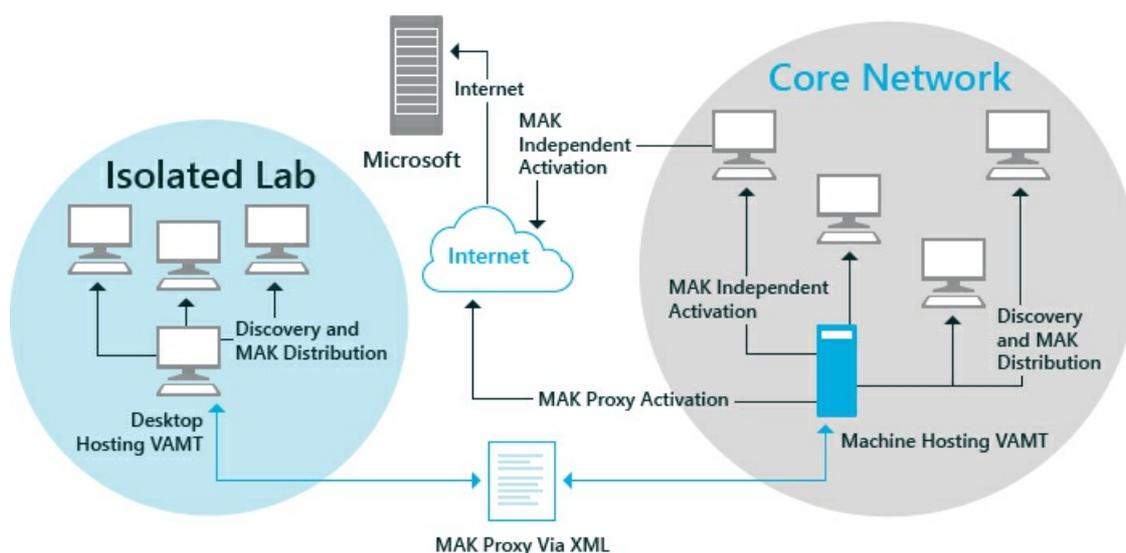
Related topics

- [VAMT Step-by-Step Scenarios](#)

Scenario 2: Proxy Activation

6/6/2019 • 15 minutes to read • [Edit Online](#)

In this scenario, the Volume Activation Management Tool (VAMT) is used to activate products that are installed on workgroup computers in an isolated lab environment. For workgroups which are isolated from the larger network, you can perform proxy activation of Multiple Activation Keys (MAKs), KMS Host keys (CSVLKs), Generic Volume License Keys (GVLKs) (or KMS client keys), or retail keys. Proxy activation is performed by installing a second instance of VAMT on a computer in the isolated workgroup. You can then use removable media to transfer VAMT Computer Information Lists (CILXs) between the instance of VAMT in the isolated workgroup and another VAMT host that has Internet access. The following diagram shows a Multiple Activation Key (MAK) proxy activation scenario:



Step 1: Install VAMT on a Workgroup Computer in the Isolated Lab

1. Install VAMT on a host computer in the isolated lab workgroup. This computer can be running Windows 7, Windows 8, Windows 10, Windows Server 2008 R2, or Windows Server® 2012.
2. Click the VAMT icon in the **Start** menu to open VAMT.

Step 2: Configure the Windows Management Instrumentation Firewall Exception on Target Computers

- Ensure that the Windows Management Instrumentation (WMI) firewall exception has been enabled for all target computers. For more information, see [Configure Client Computers](#).

Note To retrieve the license status on the selected computers, VAMT must have administrative permissions on the remote computers and WMI must be accessible through the Windows Firewall. In addition, for workgroup computers, a registry key must be created to enable remote administrative actions under User Account Control (UAC). For more information, see [Configure Client Computers](#).

Step 3: Connect to a VAMT Database

1. If the host computer in the isolated lab workgroup is not already connected to the database, the **Database Connection Settings** dialog box appears when you open VAMT. Select the server and database that contains the computers in the workgroup.

2. Click **Connect**.
3. If you are already connected to a database, in the center pane VAMT displays an inventory of the products and product keys, and a license overview of the computers in the database. If you need to connect to a different database, click **Successfully connected to the Server** to open the **Database Connection Settings** dialog box. For more information about how to create VAMT databases and adding VAMT data, see [Manage VAMT Data](#).

Step 4: Discover Products

1. In the left-side pane, in the **Products** node, click the product that you want to activate.
2. To open the **Discover Products** dialog box, click **Discover products** in the right-side pane.
3. In the **Discover Products** dialog box, click **Search for computers in the Active Directory** to display the search options, and then click the search options that you want to use. You can search for computers in an Active Directory domain, by individual computer name or IP address, in a workgroup, or by a general LDAP query:
 - To search for computers in an Active Directory domain, click **Search for computers in the Active Directory**. Then under **Domain Filter Criteria**, in the list of domain names, click the name of the domain that you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for specific computers in the domain. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only computer names that start with the letter "a".
 - To search by individual computer name or IP address, click **Manually enter name or IP address**. Then enter the full name or IP address in the **One or more computer names or IP addresses separated by commas** text box. Separate multiple entries with a comma. Note that both IPv4 and IPv6 addressing are supported.
 - To search for computers in a workgroup, click **Search for computers in the workgroup**. Then under **Workgroup Filter Criteria**, in the list of workgroup names, click the name of the workgroup that you want to search. You can narrow the search further by typing a name in the **Filter by computer name** field to search for a specific computer in the workgroup. This filter supports the asterisk (*) wildcard. For example, typing "a*" will display only those computer names that start with the letter "a".
 - To search for computers by using a general LDAP query, click **Search with LDAP query** and enter your query in the text box that appears. VAMT will validate the LDAP query syntax, but will otherwise run the query without additional checks.
4. Click **Search**.

The **Finding Computers** window appears and displays the search progress as the computers are located.

When the search is complete, the products that VAMT discovers appear in the list view in the center pane.

Step 5: Sort and Filter the List of Computers

You can sort the list of products so that it is easier to find the computers that require product keys to be activated:

1. On the menu bar at the top of the center pane, click **Group by**, and then click **Product**, **Product Key Type**, or **License Status**.
2. To sort the list further, you can click one of the column headings to sort by that column.
3. You can also use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
4. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by product name, product key type, or license status, click the list you want to use for the

filter and select an option. If necessary, click **clear all filters** to create a new filter.

5. Click **Filter**. VAMT displays the filtered list in the product list view in the center pane.

Step 6: Collect Status Information from the Computers in the Isolated Lab

To collect the status from select computers in the database, you can select computers in the product list view by using one of the following methods:

- To select a block of consecutively listed computers, click the first computer that you want to select, and then click the last computer while pressing the **Shift** key.
- To select computers which are not listed consecutively, hold down the **Ctrl** key and select each computer for which you want to collect the status information. **To collect status information from the selected computers**
- In the right-side **Actions** pane, click **Update license status** in the **Selected Items** menu and then click a credential option. Choose **Alternate Credentials** only if you are updating products that require administrator credentials that are different from the ones that you used to log on to the computer. Otherwise, click **Current Credentials** and continue to step 2. If you are supplying alternate credentials, in the **Windows Security** dialog box type the appropriate user name and password and then click **OK**.
- VAMT displays the **Collecting product information** dialog box while it collects the license status of all supported products on the selected computers. When the process is finished, the updated license status of each product will appear in the product list view in the center pane.

Note

If a computer has more than one supported product installed, VAMT adds an entry for each product. The entry appears under the appropriate product heading.

Step 7: Add Product Keys

1. Click the **Product Keys** node in the left-side pane, and then click **Add Product Keys** in the right-side pane to open the **Add Product Keys** dialog box.
2. In the **Add Product Keys** dialog box, you can select from one of the following methods to add product keys:
 - To add a single product key, click **Enter product key(s) separated by line breaks**, enter one or more product keys, and then click **Add key(s)**.
 - To import a Comma Separated Values File (CSV) that contains a list of product keys, click **Select a product key to import**, browse to the file location, click **Open** to import the file, and then click **Add Key(s)**.

The keys that you have added appear in the **Product Keys** list view in the center pane.

Step 8: Install the Product Keys on the Isolated Lab Computers

1. In the left-side pane, in the **Products** node click the product that you want to install keys onto.
2. If necessary, sort and filter the list of products so that it is easier to find the computers that must have a product key installed. See [Step 5: Sort and Filter the List of Computers](#).
3. In the **Products** list view pane, select the individual products which must have keys installed. You can use the **CTRL** key or the **SHIFT** key to select more than one product.
4. Click **Install product key** in the **Selected Items** menu in the right-side pane to display the **Install**

Product Key dialog box.

5. The **Select Product Key** dialog box displays the keys that are available to be installed. Under **Recommended MAKs**, VAMT might display one or more recommended MAKs based on the selected products. If you are installing a MAK you can select a recommended product key or any other MAK from the **All Product Keys List**. If you are not installing a MAK, select a product key from the **All Product Keys** list. Use the scroll bar if you need to view the **Description** for each key. When you have selected the product key that you want to install, click **Install Key**. Note that only one key can be installed at a time.
6. VAMT displays the **Installing product key** dialog box while it attempts to install the product key for the selected products. When the process is finished, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears.

The same status appears under the **Status of Last Action** column in the product list view in the center pane.

Note Product key installation will fail if VAMT finds mismatched key types or editions. VAMT displays the failure status and continues the installation for the next product in the list. For more information on choosing the correct product key, see [How to Choose the Right Volume License Key for Windows](#).

Note Installing a MAK and overwriting the GVLK on client products must be done with care. If the RTM version of Windows Vista has been installed on the computer for more than 30 days, then its initial grace period has expired. As a result, it will enter Reduced Functionality Mode (RFM) if online activation is not completed successfully before the next logon attempt. However, you can use online activation to recover properly configured computers from RFM, as long as the computers are available on the network. RFM only applies to the RTM version of Windows Vista or the retail editions of Microsoft Office 2010. Windows Vista with SP1 or later, Windows 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012, and volume editions of Office 2010 will not enter RFM.

Step 9: Export VAMT Data to a .cilx File

In this step, you export VAMT from the workgroup's host computer and save it in a .cilx file. Then you copy the .cilx file to removable media so that you can take it to a VAMT host computer that is connected to the Internet. In MAK proxy activation, it is critical to retain this file, because VAMT uses it to apply the Confirmation IDs (CIDs) to the proper products.

1. Select the individual products that successfully received a product key in Step 8. If needed, sort and filter the list to find the products.
2. In the right-side **Actions** pane, click **Export list** to open the **Export List** dialog box.
3. In the **Export List** dialog box, click **Browse** to navigate to the .cilx file, or enter the name of the .cilx file to which you want to export the data.
4. Under **Export options**, select one of the following data-type options:
 - Export products and product keys.
 - Export products only.
 - Export proxy activation data only. Selecting this option ensures that the export contains only the license information required for the proxy web service to obtain CIDs from Microsoft. No Personally Identifiable Information (PII) is contained in the exported .cilx file when this selection is selected. This option should be used when an enterprise's security policy states that no information that could identify a specific computer or user may be transferred out of the isolated lab and, therefore, this type of data must be excluded from the .cilx file that is transferred to the Core Network VAMT host.
5. If you have selected products to export, and not the entire set of data from the database, select the **Export**

selected product rows only check box.

6. Click **Save**. VAMT displays a progress message while the data is being exported. Click **OK** when a message appears and confirms that the export has completed successfully.
7. If you exported the list to a file on the host computer's hard drive, copy the file to removable media, such as a disk drive, CD/DVD, or USB storage device.

Important Choosing the **Export proxy activation data only** option excludes Personally Identifiable Information (PII) from being saved in the .cilx file. Therefore, the .cilx file must be re-imported into the SQL Server database on the isolated lab workgroup's VAMT host computer, so that the CIDs that are requested from Microsoft (discussed in Step 10) can be correctly assigned to the computers in the isolated lab group.

Step 10: Acquire Confirmation IDs from Microsoft on the Internet-Connected Host Computer

1. Insert the removable media into the VAMT host that has Internet access.
2. Open VAMT. Make sure you are on the root node, and that the **Volume Activation Management Tool** view is displayed in the center pane.
3. In the right-side **Actions** pane, click **Acquire confirmation IDs for CILX** to open the **Acquire confirmation IDs for file** dialog box.
4. In the **Acquire confirmation IDs for file** dialog box, browse to the location of the .cilx file that you exported from the isolated lab host computer, select the file, and then click **Open**. VAMT displays an **Acquiring Confirmation IDs** message while it contacts Microsoft and collects the CIDs.
5. When the CID collection process is complete, VAMT displays a **Volume Activation Management Tool** message that shows the number of confirmation IDs that were successfully acquired, and the name of the file where the IDs were saved. Click **OK** to close the message.

Step 11: Import the .cilx File onto the VAMT Host within the Isolated Lab Workgroup

1. Remove the storage device that contains the .cilx file from the Internet-connected VAMT host computer and insert it into the VAMT host computer in the isolated lab.
2. Open VAMT and verify that you are connected to the database that contains the computer with the product keys that you are activating.
3. In the right-side **Actions** pane, click **Import list** to open the **Import List** dialog box.
4. In the **Import list** dialog box, browse to the location of the .cilx file that contains the CIDs, select the file, and then click **Open**.
5. Click **OK** to import the file and to overwrite any conflicting data in the database with data from the file.
6. VAMT displays a progress message while the data is being imported. Click **OK** when a message appears and confirms that the data has been successfully imported.

Step 12: Apply the CIDs and Activate the Isolated Lab Computers

1. Select the products to which you want to apply CIDs. If needed, sort and filter the list to find the products.
2. In the right-side **Selected Items** menu, click **Activate**, click **Apply Confirmation ID**, and then select the appropriate credential option. If you click the **Alternate Credentials** option, you will be prompted to enter an alternate user name and password.

VAMT displays the **Applying Confirmation Id** dialog box while it installs the CIDs on the selected products. When VAMT finishes installing the CIDs, the status appears in the **Action Sataus** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done**

check box when the dialog box appears. The same status appears under the **Status of Last Action** column in the product list view in the center pane.

Step 13: (Optional) Reactivating Reimaged Computers in the Isolated Lab

If you have captured new images of the computers in the isolated lab, but the underlying hardware of those computers has not changed, VAMT can reactivate those computers using the CIDs that are stored in the database.

1. Redeploy products to each computer, using the same computer names as before.
2. Open VAMT.
3. In the right-side **Selected Items** menu, click **Activate**, click **Apply Confirmation ID**, and then select the appropriate credential option. If you click the **Alternate Credentials** option, you will be prompted to enter an alternate user name and password.

VAMT displays the **Applying Confirmation Id** dialog box while it installs the CIDs on the selected products. When VAMT finishes installing the CIDs, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box. You can also click the **Automatically close when done** check box when the dialog box appears. The same status appears under the **Status of Last Action** column in the product list view in the center pane.

Note Installing a MAK and overwriting the GVLK on the client products must be done with care. If the Windows activation initial grace period has expired, Windows will enter Reduced Functionality Mode (RFM) if online activation is not completed successfully before the next logon attempt. However, you can use online activation to recover properly configured computers from RFM, as long as the computers are accessible on the network.

RFM only applies to the RTM version of Windows Vista or the retail editions of Microsoft Office 2010. Windows Vista with SP1 or later, Windows 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012, and volume editions of Office 2010 will not enter RFM.

Note Reapplying the same CID conserves the remaining activations on the MAK.

Related topics

- [VAMT Step-by-Step Scenarios](#)

Scenario 3: KMS Client Activation

5/31/2019 • 3 minutes to read • [Edit Online](#)

In this scenario, you use the Volume Activation Management Tool (VAMT) to activate Key Management Service (KMS) client keys or Generic Volume License Keys (GVLKs). This can be performed on either Core Network or Isolated Lab computers. By default, volume license editions of Windows Vista, Windows® 7, Windows 8, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server® 2012, and Microsoft® Office 2010 use KMS for activation. GVLKs are already installed in volume license editions of these products. You do not have to enter a key to activate a product as a GVLK, unless you are converting a MAK-activated product to a KMS activation. For more information, see [Install a KMS Client Key](#).

The procedure that is described below assumes the following:

- The KMS Service is enabled and available to all KMS clients.
- VAMT has been installed and computers have been added to the VAMT database. See Parts 1 through 4 in either [Scenario 1: Online Activation](#) or [Scenario 2: Proxy Activation](#) for more information.

Activate KMS Clients

1. Open VAMT.
2. To set the KMS activation options, on the menu bar click **View**. Then click **Preferences** to open the **Volume Activation Management Tool Preferences** dialog box.
3. In the **Volume Activation Management Tool Preferences** dialog box, under **KMS Management Services host selection** select from the following options:
 - **Find a KMS host automatically using DNS**. This is the default setting. VAMT will instruct the computer to query the Domain Name Service (DNS) to locate a KMS host and perform activation. If the client contains a registry key with a valid KMS host, that value will be used instead.
 - **Find a KMS host using DNS in this domain for supported products**. Select this option if you use a specific domain, and enter the name of the domain.
 - **Use specific KMS host**. Select this option for environments which do not use DNS for KMS host identification, and manually enter the KMS host name and select the KMS host port. VAMT will set the specified KMS host name and KMS host port on the target computer, and then instruct the computer to perform activation with the specific KMS host.
4. In the left-side pane, in the **Products** node, click the product that you want to activate.
5. In the products list view in the center pane, sort the list if necessary. You can use the **Filter** function to narrow your search for computers by clicking **Filter** in the right-side pane to open the **Filter Products** dialog box.
6. In the **Filter Products** dialog box, you can filter the list by computer name, product name, product key type, license status, or by any combination of these options.
 - To filter the list by computer name, enter a name in the **Computer Name** box.
 - To filter the list by Product Name, Product Key Type, or License Status, click the list you want to use for the filter and select an option. If necessary, click **clear all filters** to create a new filter.
7. Click **Filter**. VAMT displays the filtered list in the center pane.
8. Select the products that you want to activate.
9. Click **Activate** in the **Selected Items** menu in the right-side **Actions** pane, click **Activate**, point to **Volume activate**, and then click the appropriate credential option. If you click the **Alternate Credentials** option, you will be prompted to enter an alternate user name and password.
10. VAMT displays the **Activating products** dialog box until it completes the requested action. When activation is complete, the status appears in the **Action Status** column of the dialog box. Click **Close** to close the dialog box.

You can also click the **Automatically close when done** check box when the dialog box appears.

The same status is shown under the **Status of Last Action** column in the products list view in the center pane.

Related topics

- [VAMT Step-by-Step Scenarios](#)

VAMT Known Issues

5/31/2019 • 2 minutes to read • [Edit Online](#)

The following list contains the current known issues with the Volume Activation Management Tool (VAMT) 3.0.

- The VAMT Windows Management Infrastructure (WMI) remote operations may take longer to execute if the target computer is in a sleep or standby state.
- Recovery of Non-Genuine computers is a two-step process. VAMT can be used to install a new product key and activate the computer. However, the computer itself must visit the [Windows Genuine Advantage](#) Web site to revalidate the computer's Genuine status. Upon successfully completing this step, the computer will be restored to full functionality. For more information on recovering Non-Genuine Windows computers, go to [Windows Volume Activation](#).
- When opening a Computer Information List (.cil file) saved in a previous version of VAMT, the edition information is not shown for each product in the center pane. Users must update the product status again to obtain the edition information.
- The remaining activation count can only be retrieved for MAKs.

User State Migration Tool (USMT) Technical Reference

5/31/2019 • 2 minutes to read • [Edit Online](#)

The User State Migration Tool (USMT) is included with the Windows Assessment and Deployment Kit (Windows ADK) for Windows 10. USMT provides a highly customizable user-profile migration experience for IT professionals.

Download the Windows ADK [from this website](#).

USMT support for Microsoft Office

USMT in the Windows ADK for Windows 10, version 1511 (10.1.10586.0) supports migration of user settings for installations of Microsoft Office 2003, 2007, 2010, and 2013.

USMT in the Windows ADK for Windows 10, version 1607 (10.1.14393.0) adds support for migration of user settings for installations of Microsoft Office 2016.

USMT includes three command-line tools:

- ScanState.exe
- LoadState.exe
- UsmtUtils.exe

USMT also includes a set of three modifiable .xml files:

- MigApp.xml
- MigDocs.xml
- MigUser.xml

Additionally, you can create custom .xml files to support your migration needs. You can also create a Config.xml file to specify files or settings to exclude from the migration.

USMT tools can be used on several versions of Windows operating systems, for more information, see [USMT Requirements](#). For more information about previous releases of the USMT tools, see [User State Migration Tool \(USMT\) 4.0 User's Guide](#).

In This Section

TOPIC	DESCRIPTION
User State Migration Tool (USMT) Overview Topics	Describes what's new in USMT, how to get started with USMT, and the benefits and limitations of using USMT.
User State Migration Tool (USMT) How-to topics	Includes step-by-step instructions for using USMT, as well as how-to topics for conducting tasks in USMT.
User State Migration Tool (USMT) Troubleshooting	Provides answers to frequently asked questions and common issues in USMT, as well as a reference for return codes used in USMT.

TOPIC	DESCRIPTION
User State Migration Toolkit (USMT) Reference	Includes reference information for migration planning, migration best practices, command-line syntax, using XML, and requirements for using USMT.

Related topics

- [Windows Assessment and Deployment Kit](#)

User State Migration Tool (USMT) Overview Topics

5/31/2019 • 2 minutes to read • [Edit Online](#)

The User State Migration Tool (USMT) 10.0 provides a highly customizable user-profile migration experience for IT professionals. USMT includes three command-line tools: ScanState.exe, LoadState.exe, and UsmtUtils.exe. USMT also includes a set of three modifiable .xml files: MigApp.xml, MigDocs.xml, and MigUser.xml. Additionally, you can create custom .xml files to support your migration needs. You can also create a Config.xml file to specify files or settings to exclude from the migration.

In This Section

TOPIC	DESCRIPTION
User State Migration Tool (USMT) Overview	Describes the benefits and limitations of using USMT.
Getting Started with the User State Migration Tool (USMT)	Describes the general process to follow to migrate files and settings, and provides links to more information.
Windows Upgrade and Migration Considerations	Discusses the Microsoft® tools you can use to move files and settings between installations, as well as special considerations for performing an upgrade or migration.

Related topics

- [User State Migration Tool \(USMT\) How-to topics](#)
- [User State Migration Tool \(USMT\) Troubleshooting](#)
- [User State Migration Toolkit \(USMT\) Reference](#)

User State Migration Tool (USMT) Overview

5/31/2019 • 2 minutes to read • [Edit Online](#)

You can use User State Migration Tool (USMT) 10.0 to streamline and simplify user state migration during large deployments of Windows operating systems. USMT captures user accounts, user files, operating system settings, and application settings, and then migrates them to a new Windows installation. You can use USMT for both PC replacement and PC refresh migrations. For more information, see [Common Migration Scenarios](#).

USMT enables you to do the following:

- Configure your migration according to your business needs by using the migration rule (.xml) files to control exactly which files and settings are migrated and how they are migrated. For more information about how to modify these files, see [USMT XML Reference](#).
- Fit your customized migration into your automated deployment process by using the ScanState and LoadState tools, which control collecting and restoring the user files and settings. For more information, see [User State Migration Tool \(USMT\) Command-line Syntax](#).
- Perform offline migrations. You can run migrations offline by using the ScanState command in Windows Preinstallation Environment (WinPE) or you can perform migrations from previous installations of Windows contained in Windows.old directories. For more information about migration types, see [Choose a Migration Store Type](#) and [Offline Migration Reference](#).

Benefits

USMT provides the following benefits to businesses that are deploying Windows operating systems:

- Safely migrates user accounts, operating system and application settings.
- Lowers the cost of deploying Windows by preserving user state.
- Reduces end-user downtime required to customize desktops and find missing files.
- Reduces help-desk calls.
- Reduces the time needed for the user to become familiar with the new operating system.
- Increases employee satisfaction with the migration experience.

Limitations

USMT is intended for administrators who are performing large-scale automated deployments. If you are only migrating the user states of a few computers, you can use [PCmover Express](#). PCmover Express is a tool created by Microsoft's partner, Laplink.

There are some scenarios in which the use of USMT is not recommended. These include:

- Migrations that require end-user interaction.
- Migrations that require customization on a machine-by-machine basis.

Related topics

- [User State Migration Tool \(USMT\) Technical Reference](#)

Getting Started with the User State Migration Tool (USMT)

6/14/2019 • 5 minutes to read • [Edit Online](#)

This topic outlines the general process that you should follow to migrate files and settings.

In this Topic

- [Step 1: Plan Your Migration](#)
- [Step 2: Collect files and settings from the source computer](#)
- [Step 3: Prepare the destination computer and restore files and settings](#)

Step 1: Plan your migration

1. [Plan Your Migration](#). Depending on whether your migration scenario is refreshing or replacing computers, you can choose an online migration or an offline migration using Windows Preinstallation Environment (WinPE) or the files in the Windows.old directory. For more information, see [Common Migration Scenarios](#).
2. [Determine What to Migrate](#). Data you might consider migrating includes end-user information, applications settings, operating-system settings, files, folders, and registry keys.
3. Determine where to store data. Depending on the size of your migration store, you can store the data remotely, locally in a hard-link migration store or on a local external storage device, or directly on the destination computer. For more information, see [Choose a Migration Store Type](#).
4. Use the **/GenMigXML** command-line option to determine which files will be included in your migration, and to determine whether any modifications are necessary. For more information see [ScanState Syntax](#)
5. Modify copies of the Migration.xml and MigDocs.xml files and create custom .xml files, if it is required. To modify the migration behavior, such as migrating the **Documents** folder but not the **Music** folder, you can create a custom .xml file or modify the rules in the existing migration .xml files. The document finder, or **MigXmlHelper.GenerateDocPatterns** helper function, can be used to automatically find user documents on a computer without creating extensive custom migration .xml files.

Important We recommend that you always make and modify copies of the .xml files included in User State Migration Tool (USMT) 10.0. Never modify the original .xml files.

You can use the MigXML.xsd file to help you write and validate the .xml files. For more information about how to modify these files, see [USMT XML Reference](#).

6. Create a [Config.xml File](#) if you want to exclude any components from the migration. To create this file, use the [ScanState Syntax](#) option together with the other .xml files when you use the **ScanState** command. For example, the following command creates a Config.xml file by using the MigDocs and MigApp.xml files:

```
scanstate /genconfig:config.xml /i:migdocs.xml /i:migapp.xml /v:13 /l:scanstate.log
```

7. Review the migration state of the components listed in the Config.xml file, and specify `migrate=no` for any components that you do not want to migrate.

Step 2: Collect files and settings from the source computer

1. Back up the source computer.
2. Close all applications. If some applications are running when you run the **ScanState** command, USMT might not migrate all of the specified data. For example, if Microsoft® Office Outlook® is open, USMT might not migrate PST files.

Note USMT will fail if it cannot migrate a file or setting unless you specify the **/C** option. When you specify the **/C** option, USMT will ignore the errors, and log an error every time that it encounters a file that is being used that USMT did not migrate. You can use the **<ErrorControl>** section in the Config.xml file to specify which errors should be ignored, and which should cause the migration to fail.

3. Run the **ScanState** command on the source computer to collect files and settings. You should specify all of the .xml files that you want the **ScanState** command to use. For example,

```
scanstate \\server\migration\mystore /config:config.xml /i:migdocs.xml /i:migapp.xml /v:13 /l:scan.log
```

Note If the source computer is running Windows 7, or Windows 8, you must run the **ScanState** command in **Administrator** mode. To run in **Administrator** mode, right-click **Command Prompt**, and then click **Run As Administrator**. If the source computer is running Windows XP, you must run the **ScanState** command from an account that has administrative credentials. For more information about the how the **ScanState** command processes and stores the data, see [How USMT Works](#).

4. Run the **USMTUtils** command with the **/Verify** option to ensure that the store you created is not corrupted.

Step 3: Prepare the destination computer and restore files and settings

1. Install the operating system on the destination computer.
2. Install all applications that were on the source computer. Although it is not always required, we recommend installing all applications on the destination computer before you restore the user state. This makes sure that migrated settings are preserved.

Note The application version that is installed on the destination computer should be the same version as the one on the source computer. USMT does not support migrating the settings for an older version of an application to a newer version. The exception to this is Microsoft® Office, which USMT can migrate from an older version to a newer version.

3. Close all applications. If some applications are running when you run the **LoadState** command, USMT might not migrate all of the specified data. For example, if Microsoft Office Outlook is open, USMT might not migrate PST files.

Note Use **/C** to continue your migration if errors are encountered, and use the **<ErrorControl>** section in the Config.xml file to specify which errors should be ignored, and which errors should cause the migration to fail.

4. Run the **LoadState** command on the destination computer. Specify the same set of .xml files that you specified when you used the **ScanState** command. However, you do not have to specify the Config.xml file, unless you want to exclude some of the files and settings that you migrated to the store. For example, you might want to migrate the My Documents folder to the store, but not to the destination computer. To do this, modify the Config.xml file and specify the updated file by using the **LoadState** command. Then, the **LoadState** command will migrate only the files and settings that you want to migrate. For more information about the how the **LoadState** command processes and migrates data, see [How USMT Works](#).

For example, the following command migrates the files and settings:

```
loadstate \\server\migration\mystore /config:config.xml /i:migdocs.xml /i:migapp.xml /v:13 /l:load.log
```

Note Run the **LoadState** command in administrator mode. To do this, right-click **Command Prompt**, and then click **Run As Administrator**.

5. Log off after you run the **LoadState** command. Some settings (for example, fonts, wallpaper, and screen saver settings) will not take effect until the next time that the user logs on.

Windows upgrade and migration considerations

6/14/2019 • 4 minutes to read • [Edit Online](#)

Files and application settings can be migrated to new hardware running the Windows® operating system, or they can be maintained during an operating system upgrade on the same computer. This topic summarizes the Microsoft® tools you can use to move files and settings between installations in addition to special considerations for performing an upgrade or migration.

Upgrade from a previous version of Windows

You can upgrade from an earlier version of Windows, which means you can install the new version of Windows and retain your applications, files, and settings as they were in your previous version of Windows. If you decide to perform a custom installation of Windows instead of an upgrade, your applications and settings will not be maintained. Your personal files, and all Windows files and directories, will be moved to a Windows.old folder. You can access your data in the Windows.old folder after Windows Setup is complete.

Migrate files and settings

Migration tools are available to transfer settings from one computer that is running Windows to another. These tools transfer only the program settings, not the programs themselves.

For more information about application compatibility, see the [Application Compatibility Toolkit \(ACT\)](#).

The User State Migration Tool (USMT) 10.0 is an application intended for administrators who are performing large-scale automated deployments. For deployment to a small number of computers or for individually customized deployments, you can use Windows Easy Transfer.

Migrate with Windows Easy Transfer

Windows Easy Transfer is a software wizard for transferring files and settings from one computer that is running Windows to another. It helps you select what to move to your new computer, enables you to set which migration method to use, and then performs the transfer. When the transfer has completed, Windows Easy Transfer Reports shows you what was transferred and provides a list of programs you might want to install on your new computer, in addition to links to other programs you might want to download.

With Windows Easy Transfer, files and settings can be transferred using a network share, a USB flash drive (UFD), or the Easy Transfer cable. However, you cannot use a regular universal serial bus (USB) cable to transfer files and settings with Windows Easy Transfer. An Easy Transfer cable can be purchased on the Web, from your computer manufacturer, or at an electronics store.

NOTE

Windows Easy Transfer is not available in Windows 10.

Migrate with the User State Migration Tool

You can use USMT to automate migration during large deployments of the Windows operating system. USMT uses configurable migration rule (.xml) files to control exactly which user accounts, user files, operating system settings, and application settings are migrated and how they are migrated. You can use USMT for both *side-by-side* migrations, where one piece of hardware is being replaced, or *wipe-and-load* (or *refresh*) migrations, when only the operating system is being upgraded.

Upgrade and migration considerations

Whether you are upgrading or migrating to a new version of Windows, you must be aware of the following issues and considerations:

Application compatibility

For more information about application compatibility in Windows, see [Use Upgrade Readiness to manage Windows upgrades](#).

Multilingual Windows image upgrades

When performing multilingual Windows upgrades, cross-language upgrades are not supported by USMT. If you are upgrading or migrating an operating system with multiple language packs installed, you can upgrade or migrate only to the system default user interface (UI) language. For example, if English is the default but you have a Spanish language pack installed, you can upgrade or migrate only to English.

If you are using a single-language Windows image that matches the system default UI language of your multilingual operating system, the migration will work. However, all of the language packs will be removed, and you will have to reinstall them after the upgrade is completed.

Errorhandler.cmd

When upgrading from an earlier version of Windows, if you intend to use Errorhandler.cmd, you must copy this file into the %WINDIR%\Setup\Scripts directory on the old installation. This makes sure that if there are errors during the down-level phase of Windows Setup, the commands in Errorhandler.cmd will run.

Data drive ACL migration

During the configuration pass of Windows Setup, the root access control list (ACL) on drives formatted for NTFS that do not appear to have an operating system will be changed to the default Windows XP ACL format. The ACLs on these drives are changed to enable authenticated users to modify access on folders and files.

Changing the ACLs may affect the performance of Windows Setup if the default Windows XP ACLs are applied to a partition with a large amount of data. Because of these performance concerns, you can change the following registry value to disable this feature:

```
Key: HKLM\System\Setup
Type: REG_DWORD
Value: "DDACLSys_Disabled" = 1
```

This feature is disabled if this registry key value exists and is configured to `1`.

Related topics

[User State Migration Tool \(USMT\) Overview Topics](#)

[Windows 10 upgrade paths](#)

[Windows 10 edition upgrade](#)

User State Migration Tool (USMT) How-to topics

5/31/2019 • 2 minutes to read • [Edit Online](#)

The following table lists topics that describe how to use User State Migration Tool (USMT) 10.0 to perform specific tasks.

In This Section

TOPIC	DESCRIPTION
Exclude Files and Settings	Create a custom .xml file to exclude files, file types, folders, or registry settings from your migration.
Extract Files from a Compressed USMT Migration Store	Recover files from a compressed migration store after installing the operating system.
Include Files and Settings	Create a custom .xml file to include files, file types, folders, or registry settings in your migration.
Migrate Application Settings	Migrate the settings of an application that the MigApp.xml file does not include by default.
Migrate EFS Files and Certificates	Migrate Encrypting File System (EFS) certificates by using USMT.
Migrate User Accounts	Specify the users to include and exclude in your migration.
Reroute Files and Settings	Create a custom .xml file to reroute files and settings during a migration.
Verify the Condition of a Compressed Migration Store	Determine whether a compressed migration store is intact, or whether it contains corrupt files or a corrupt catalog.

Related topics

- [User State Migration Tool \(USMT\) Overview Topics](#)
- [User State Migration Tool \(USMT\) Troubleshooting](#)
- [User State Migration Toolkit \(USMT\) Reference](#)

Exclude Files and Settings

6/14/2019 • 7 minutes to read • [Edit Online](#)

When you specify the migration .xml files, MigApp.xml, MigDocs, and MigUser.xml, the User State Migration Tool (USMT) 10.0 migrates the settings and components listed, as discussed in [What Does USMT Migrate?](#) You can create a custom .xml file to further specify what to include or exclude in the migration. In addition you can create a Config.xml file to exclude an entire component from a migration. You cannot, however, exclude users by using the migration .xml files or the Config.xml file. The only way to specify which users to include and exclude is by using the User options on the command line in the ScanState tool. For more information, see [ScanState Syntax](#).

In this topic:

- [Create a custom .xml file](#). You can use the following elements to specify what to exclude:
 - [include and exclude](#): You can use the <include> and <exclude> elements to exclude objects with conditions. For example, you can migrate all files located in the C:\ drive, except any .mp3 files. It is important to remember that [Conflicts and Precedence](#) apply to these elements.
 - [unconditionalExclude](#): You can use the <unconditionalExclude> element to globally exclude data. This element takes precedence over all other include and exclude rules in the .xml files. Therefore, this element excludes objects regardless of any other <include> rules that are in the .xml files. For example, you can exclude all .mp3 files on the computer, or you can exclude all files from C:\UserData.
- [Create a Config.xml File](#): You can create and modify a Config.xml file to exclude an entire component from the migration. For example, you can use this file to exclude the settings for one of the default applications. In addition, creating and modifying a Config.xml file is the only way to exclude the operating-system settings that are migrated to computers running Windows. Excluding components using this file is easier than modifying the migration .xml files because you do not need to be familiar with the migration rules and syntax.

Create a custom .xml file

We recommend that you create a custom .xml file instead of modifying the default migration .xml files. When you use a custom .xml file, you can keep your changes separate from the default .xml files, which makes it easier to track your modifications.

<include> and <exclude>

The migration .xml files, MigApp.xml, MigDocs, and MigUser.xml, contain the <component> element, which typically represents a self-contained component or an application such as Microsoft® Office Outlook® and Word. To exclude the files and registry settings that are associated with these components, use the <include> and <exclude> elements. For example, you can use these elements to migrate all files and settings with pattern X except files and settings with pattern Y, where Y is more specific than X. For the syntax of these elements, see [USMT XML Reference](#).

Note If you specify an <exclude> rule, always specify a corresponding <include> rule. Otherwise, if you do not specify an <include> rule, the specific files or settings will not be included. They will already be excluded from the migration. Thus, an unaccompanied <exclude> rule is unnecessary.

- [Example 1: How to migrate all files from C:\ except .mp3 files](#)
- [Example 2: How to migrate all files located in C:\Data except files in C:\Data\tmp](#)

- [Example 3: How to exclude the files in a folder but include all subfolders](#)
- [Example 4: How to exclude a file from a specific folder](#)
- [Example 5: How to exclude a file from any location](#)

Example 1: How to migrate all files from C:\ except .mp3 files

The following .xml file migrates all files located on the C: drive, except any .mp3 files.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/mp3files">
  <!-- This component migrates all files except those with .mp3 extension-->
  <component type="Documents" context="UserAndSystem">
    <displayName _locID="miguser.sharedvideo">MP3 Files</displayName>
    <role role="Data">
      <rules>
        <include filter='MigXmlHelper.IgnoreIrrelevantLinks()''>
          <objectSet>
            <pattern type="File">C:\* [*]</pattern>
          </objectSet>
        </include>
        <exclude>
          <objectSet>
            <pattern type="File">C:\* [*].mp3</pattern>
          </objectSet>
        </exclude>
      </rules>
    </role>
  </component>
</migration>
```

Example 2: How to migrate all files located in C:\Data except files in C:\Data\tmp

The following .xml file migrates all files and subfolders in C:\Data, except the files and subfolders in C:\Data\tmp.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
  <component type="Documents" context="System">
    <displayName _locID="miguser.sharedvideo">Test component</displayName>
    <role role="Data">
      <rules>
        <include>
          <objectSet>
            <pattern type="File">C:\Data\* [*]</pattern>
          </objectSet>
        </include>
        <exclude>
          <objectSet>
            <pattern type="File"> C:\Data\tmp\* [*]</pattern>
          </objectSet>
        </exclude>
      </rules>
    </role>
  </component>
</migration>
```

Example 3: How to exclude the files in a folder but include all subfolders

The following .xml file migrates any subfolders in C:\EngineeringDrafts, but excludes all files that are in C:\EngineeringDrafts.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents without subfolders</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
      <exclude>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\ [*]</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>
</migration>

```

Example 4: How to exclude a file from a specific folder

The following .xml file migrates all files and subfolders in C:\EngineeringDrafts, except for the Sample.doc file in C:\EngineeringDrafts.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents except Sample.doc</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
      <exclude>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\ [Sample.doc]</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>
</migration>

```

Example 5: How to exclude a file from any location

To exclude a Sample.doc file from any location on the C: drive, use the <pattern> element. If multiple files exist with the same name on the C: drive, all of these files will be excluded.

```

<pattern type="File"> C:\* [Sample.doc] </pattern>

```

To exclude a Sample.doc file from any drive on the computer, use the <script> element. If multiple files exist with the same name, all of these files will be excluded.

```

<script>MigXmlHelper.GenerateDrivePatterns("* [sample.doc]", "Fixed")</script>

```

Examples of how to use XML to exclude files, folders, and registry keys

Here are some examples of how to use XML to exclude files, folders, and registry keys. For more info, see [USMT XML Reference](#)

Example 1: How to exclude all .mp3 files

The following .xml file excludes all .mp3 files from the migration:

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/excludefiles">
  <component context="System" type="Documents">
    <displayName>Test</displayName>
    <role role="Data">
      <rules>
        <unconditionalExclude>
          <objectSet>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*].mp3", "Fixed")</script>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

Example 2: How to exclude all of the files on a specific drive

The following .xml file excludes only the files located on the C: drive.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/allfiles">
  <component type="Documents" context="System">
    <displayName>Test</displayName>
    <role role="Data">
      <rules>
        <unconditionalExclude>
          <objectSet>
            <pattern type="File">c:\*[*]</pattern>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

Example 3: How to exclude registry keys

The following .xml file unconditionally excludes the HKEY_CURRENT_USER registry key and all of its subkeys.

```
<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/miguser">
  <component type="Documents" context="User">
    <displayName>Test</displayName>
    <role role="Data">
      <rules>
        <include>
          <objectSet>
            <pattern type="Registry">HKCU\testReg[*]</pattern>
          </objectSet>
        </include>
        <unconditionalExclude>
          <objectSet>
            <pattern type="Registry">HKCU\*[*]</pattern>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

Example 4: How to Exclude C:\Windows and C:\Program Files

The following .xml file unconditionally excludes the system folders of `C:\Windows` and `C:\Program Files`. Note that all *.docx, *.xls and *.ppt files will not be migrated because the `<unconditionalExclude>` element takes precedence over the `<include>` element.

```
<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/miguser">
  <component type="Documents" context="System">
    <displayName>Test</displayName>
    <role role="Data">
      <rules>
        <include>
          <objectSet>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*.doc]", "Fixed")</script>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*.xls]", "Fixed")</script>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*.ppt]", "Fixed")</script>
          </objectSet>
        </include>
        <unconditionalExclude>
          <objectSet>
            <pattern type="File">C:\Program Files\* [*]</pattern>
          </objectSet>
        </unconditionalExclude>
        <pattern type="File">C:\Windows\* [*]</pattern>
        <objectSet>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

Create a Config XML File

You can create and modify a Config.xml file if you want to exclude components from the migration. Excluding components using this file is easier than modifying the migration .xml files because you do not need to be familiar with the migration rules and syntax. Config.xml is an optional file that you can create using the **/genconfig** command-line option with the ScanState tool. For example, you can use the Config.xml file to exclude the settings for one of the default applications. In addition, creating and modifying this file is the only way to exclude the operating-system settings that are migrated to computers running Windows.

- **To exclude the settings for a default application:** Specify `migrate="no"` for the application under the `<Applications>` section of the Config.xml file.
- **To exclude an operating system setting:** Specify `migrate="no"` for the setting under the `<WindowsComponents>` section.
- **To exclude My Documents:** Specify `migrate="no"` for My Documents under the `<Documents>` section. Note that any `<include>` rules in the .xml files will still apply. For example, if you have a rule that includes all the .docx files in My Documents, then only the .docx files will be migrated, but the rest of the files will not.

See [Config.xml File](#) for more information.

Note To exclude a component from the Config.xml file, set the **migrate** value to **"no"**. Deleting the XML tag for the component from the Config.xml file will not exclude the component from your migration.

Related topics

- [Customize USMT XML Files](#)
- [USMT XML Reference](#)

Extract Files from a Compressed USMT Migration Store

5/31/2019 • 3 minutes to read • [Edit Online](#)

When you migrate files and settings during a typical PC-refresh migration, you usually create a compressed migration store file on the intermediate store. This migration store is a single image file that contains all files being migrated as well as a catalog file. To protect the compressed file, you can encrypt it by using different encryption algorithms. When you migrate the file back to the source computer after the operating system is installed, you can run the **Usmtutils** command with the **/extract** option to recover the files from the compressed migration store. You can also use the **Usmtutils** command with the **/extract** option any time you need to recover data from a migration store.

Options used with the **/extract** option can specify:

- The cryptographic algorithm that was used to create the migration store.
- The encryption key or the text file that contains the encryption key.
- Include and exclude patterns for selective data extraction.

In addition, you can specify the file patterns that you want to extract by using the **/i** option to include file patterns or the **/e** option to exclude file patterns. When both the **/i** option and the **/e** option are used in the same command, include patterns take precedence over exclude patterns. Note that this is different from the include and exclude rules used in the ScanState and LoadState tools.

In this topic

- [To run the USMTutils tool with the /extract option](#)
- [To extract all files from a compressed migration store](#)
- [To extract specific file types from an encrypted compressed migration store](#)
- [To extract all but one, or more, file types from an encrypted compressed migration store](#)
- [To extract file types using the include pattern and the exclude pattern](#)

To run the USMTutils tool with the /extract option

To extract files from the compressed migration store onto the destination computer, use the following USMTutils syntax:

```
Cd /d <USMTpath> usmtutils /extract <filePath> <destinationPath> [/i:<includePattern>] [/e:<excludePattern>] [/l:<logfile>] [/decrypt[:<AlgID>] {/key:<keystring> | /keyfile:<filename>}] [/o]
```

Where the placeholders have the following values:

- *<USMTpath>* is the location where you have saved the USMT files and tools.
- *<filePath>* is the location of the migration store.
- *<destination path>* is the location of the file where you want the **/extract** option to put the extracted migration store contents.
- *<includePattern>* specifies the pattern for the files to include in the extraction.

- `<excludePattern>` specifies the pattern for the files to omit from the extraction.
- `<AlgID>` is the cryptographic algorithm that was used to create the migration store on the **ScanState** command line.
- `<logfile>` is the location and name of the log file.
- `<keystring>` is the encryption key that was used to encrypt the migration store.
- `<filename>` is the location and name of the text file that contains the encryption key.

To extract all files from a compressed migration store

To extract everything from a compressed migration store to a file on the C:\ drive, type:

```
usmtutils /extract D:\MyMigrationStore\USMT\store.mig C:\ExtractedStore
```

To extract specific file types from an encrypted compressed migration store

To extract specific files, such as .txt and .pdf files, from an encrypted compressed migration store, type:

```
usmtutils /extract D:\MyMigrationStore\USMT\store.mig /i:"*.txt,*.pdf" C:\ExtractedStore /decrypt
/keyfile:D:\encryptionKey.txt
```

In this example, the file is encrypted and the encryption key is located in a text file called encryptionKey.

To extract all but one, or more, file types from an encrypted compressed migration store

To extract all files except for one file type, such as .exe files, from an encrypted compressed migration store, type:

```
usmtutils /extract D:\MyMigrationStore\USMT\store.mig /e:*.exe C:\ExtractedStore /decrypt:AES_128
/key:password /l:C:\usmtutilslog.txt
```

To extract file types using the include pattern and the exclude pattern

To extract files from a compressed migration store, and to exclude files of one type (such as .exe files) while including only specific files, use both the include pattern and the exclude pattern, as in this example:

```
usmtutils /extract D:\MyMigrationStore\USMT\store.mig /i:myProject.* /e:*.exe C:\ExtractedStore /o
```

In this example, if there is a myProject.exe file, it will also be extracted because the include pattern option takes precedence over the exclude pattern option.

Related topics

[UsmtUtils Syntax](#)

[Return Codes](#)

[Verify the Condition of a Compressed Migration Store](#)

Include Files and Settings

5/31/2019 • 3 minutes to read • [Edit Online](#)

When you specify the migration .xml files, User State Migration Tool (USMT) 10.0 migrates the settings and components specified in [What Does USMT Migrate?](#) To include additional files and settings, we recommend that you create a custom .xml file and then include this file when using both the ScanState and LoadState commands. By creating a custom .xml file, you can keep your changes separate from the default .xml files, which makes it easier to track your modifications.

In this topic:

[Migrate a Single Registry Key](#)

[Migrate a Specific Folder](#)

[Migrate a Folder from a Specific Drive](#)

[Migrate a Folder from Any Location](#)

[Migrate a File Type Into a Specific Folder](#)

[Migrate a Specific File](#)

Migrate a Single Registry Key

The following .xml file migrates a single registry key.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
  <component type="Application" context="System">
    <displayName>Component to migrate only registry value string</displayName>
    <role role="Settings">
      <rules>
        <include>
          <objectSet>
            <pattern type="Registry">HKLM\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\Cache [Persistent]</pattern>
          </objectSet>
        </include>
      </rules>
    </role>
  </component>
</migration>
```

Migrate a Specific Folder

The following examples show how to migrate a folder from a specific drive, and from any location on the computer.

Migrate a Folder from a Specific Drive

- **Including subfolders.** The following .xml file migrates all files and subfolders from C:\EngineeringDrafts to the destination computer.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmltest/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents including subfolders</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File">C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

- **Excluding subfolders.** The following .xml file migrates all files from C:\EngineeringDrafts, but it does not migrate any subfolders within C:\EngineeringDrafts.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmltest/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents without subfolders</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

Migrate a Folder from Any Location

The following .xml file migrates all files and subfolders of the EngineeringDrafts folder from any drive on the computer. If multiple folders exist with the same name, then all files with this name are migrated.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmltest/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents folder on any drive on the computer
</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("\"EngineeringDrafts\"* [*] ", "Fixed")</script>
          <script>MigXmlHelper.GenerateDrivePatterns ("*\EngineeringDrafts\"* [*] ", "Fixed")</script>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

The following .xml file migrates all files and subfolders of the EngineeringDrafts folder from any location on the C:\ drive. If multiple folders exist with the same name, they are all migrated.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents EngineeringDrafts folder from where ever
it exists on the C: drive </displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
<pattern type="File"> C:\*\EngineeringDrafts\* [*]</pattern>
<pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

Migrate a File Type Into a Specific Folder

The following .xml file migrates .mp3 files located in the specified drives on the source computer into the C:\Music folder on the destination computer.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>All .mp3 files to My Documents</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("* [*].mp3", "Fixed")</script>
        </objectSet>
      </include>
      <!-- Migrates all the .mp3 files in the store to the C:\Music folder during LoadState -->
      <locationModify script="MigXmlHelper.Move('C:\Music')">
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("* [*].mp3", "Fixed")</script>
        </objectSet>
      </locationModify>
    </rules>
  </role>
</component>
</migration>

```

Migrate a Specific File

The following examples show how to migrate a file from a specific folder, and how to migrate a file from any location.

- **To migrate a file from a folder.** The following .xml file migrates only the Sample.doc file from C:\EngineeringDrafts on the source computer to the destination computer.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts Documents</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\ [Sample.doc]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>
```

- **To migrate a file from any location.** To migrate the Sample.doc file from any location on the C:\ drive, use the <pattern> element, as the following example shows. If multiple files exist with the same name on the C:\ drive, all of files with this name are migrated.

```
<pattern type="File"> C:\* [Sample.doc] </pattern>
```

To migrate the Sample.doc file from any drive on the computer, use <script> as the following example shows. If multiple files exist with the same name, all files with this name are migrated.

```
<script>MigXmlHelper.GenerateDrivePatterns("* [sample.doc]", "Fixed")</script>
```

Related topics

[Customize USMT XML Files](#)

[Custom XML Examples](#)

[Conflicts and Precedence](#)

[USMT XML Reference](#)

Migrate Application Settings

6/14/2019 • 12 minutes to read • [Edit Online](#)

You can create a custom .xml file to migrate specific line-of-business application settings or to change the default migration behavior of the User State Migration Tool (USMT) 10.0. For ScanState and LoadState to use this file, you must specify the custom .xml file on both command lines.

This topic defines how to author a custom migration .xml file that migrates the settings of an application that is not migrated by default using MigApp.xml. You should migrate the settings after you install the application, but before the user runs the application for the first time.

This topic does not contain information about how to migrate applications that store settings in an application-specific store, only the applications that store the information in files or in the registry. It also does not contain information about how to migrate the data that users create using the application. For example, if the application creates .doc files using a specific template, this topic does not discuss how to migrate the .doc files and templates themselves.

In this Topic

- [Before You Begin](#)
- [Step 1: Verify that the application is installed on the source computer, and that it is the same version as the version to be installed on the destination computer.](#)
- [Step 2: Identify settings to collect and determine where each setting is stored on the computer.](#)
- [Step 3: Identify how to apply the gathered settings.](#)
- [Step 4: Create the migration XML component for the application.](#)
- [Step 5: Test the application settings migration.](#)

Before You Begin

You should identify a test computer that contains the operating system of your source computers, and the application whose settings you want to migrate. For example, if you are planning on migrating from Windows 7 to Windows 10, install Windows 7 on your test computer and then install the application.

Step 1: Verify that the application is installed on the source computer, and that it is the same version as the version to be installed on the destination computer.

Before USMT migrates the settings, you need it to check whether the application is installed on the source computer, and that it is the correct version. If the application is not installed on the source computer, you probably do not want USMT to spend time searching for the application's settings. More importantly, if USMT collects settings for an application that is not installed, it may migrate settings that will cause the destination computer to function incorrectly. You should also investigate whether there is more than one version of the application. This is because the new version may not store the settings in the same place, which may lead to unexpected results on the destination computer.

There are many ways to detect if an application is installed. The best practice is to check for an application uninstall key in the registry, and then search the computer for the executable file that installed the application. It is

important that you check for both of these items, because sometimes different versions of the same application share the same uninstall key. So even if the key is there, it may not correspond to the version of the application that you want.

Check the registry for an application uninstall key.

When many applications are installed (especially those installed using the Microsoft® Windows® Installer technology), an application uninstall key is created under

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall. For example, when Adobe Acrobat Reader 7 is installed, it creates a key named

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall \{AC76BA86-7AD7-1033-7B44-A70000000000}. Therefore, if a computer contains this key, then Adobe Acrobat Reader 7 is installed on the computer. You can check for the existence of a registry key using the **DoesObjectExist** helper function.

Usually, you can find this key by searching under

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall for the name of the application, the name of the application executable file, or for the name of the company that makes the application. You can use the Registry Editor (**Regedit.exe** located in the **%SystemRoot%**) to search the registry.

Check the file system for the application executable file.

You should also check the application binaries for the executable that installed the application. To do this, you will first need to determine where the application is installed and what the name of the executable is. Most applications store the installation location of the application binaries in the registry. You should search the registry for the name of the application, the name of the application executable, or for the name of the company that makes the application, until you find the registry value that contains the installation path. Once you have determined the path to the application executable, you can use the **DoesFileVersionMatch** helper function to check for the correct version of the application executable. For an example of how to do this, see the Windows Live™ Messenger section of the MigApp.xml file.

Step 2: Identify settings to collect and determine where each setting is stored on the computer.

Next, you should go through the user interface and make a list of all of the available settings. You can reduce the list if there are settings that you do not want to migrate. To determine where each setting is stored, you will need to change each setting and monitor the activity on the registry and the file system. You do not need to migrate the binary files and registry settings that are made when the application is installed. This is because you will need to reinstall the application onto the destination computer. You only need to migrate those settings that are customizable.

How To Determine Where Each Setting is Stored

1. Download a file and registry monitoring tool, such as the Regmon and Filemon tools, from the [Windows Sysinternals Web site](#).
2. Shut down as many applications as possible to limit the registry and file system activity on the computer.
3. Filter the output of the tools so it only displays changes being made by the application.

Note Most applications store their settings under the user profile. That is, the settings stored in the file system are under the **%UserProfile%** directory, and the settings stored in the registry are under the **HKEY_CURRENT_USER** hive. For these applications you can filter the output of the file and registry monitoring tools to show activity only under these locations. This will considerably reduce the amount of output that you will need to examine.

4. Start the monitoring tool(s), change a setting, and look for registry and file system writes that occurred

when you changed the setting. Make sure the changes you make actually take effect. For example, if you are changing a setting in Microsoft Word by selecting a check box in the **Options** dialog box, the change typically will not take effect until you close the dialog box by clicking **OK**.

5. When the setting is changed, note the changes to the file system and registry. There may be more than one file or registry values for each setting. You should identify the minimal set of file and registry changes that are required to change this setting. This set of files and registry keys is what you will need to migrate in order to migrate the setting.

Note Changing an application setting invariably leads to writing to registry keys. If possible, filter the output of the file and registry monitor tool to display only writes to files and registry keys/values.

Step 3: Identify how to apply the gathered settings.

If the version of the application on the source computer is the same as the one on the destination computer, then you do not have to modify the collected files and registry keys. By default, USMT migrates the files and registry keys from the source location to the corresponding location on the destination computer. For example, if a file was collected from the C:\Documents and Settings\User1\My Documents folder and the profile directory on the destination computer is located at D:\Users\User1, then USMT will automatically migrate the file to D:\Users\User1\My Documents. However, you may need to modify the location of some settings in the following three cases:

Case 1: The version of the application on the destination computer is newer than the one on the source computer.

In this case, the newer version of the application may be able to read the settings from the source computer without modification. That is, the data collected from an older version of the application is sometimes compatible with the newer version of the application. However, you may need to modify the setting location if either of the following is true:

- **The newer version of the application has the ability to import settings from an older version.** This mapping usually happens the first time a user runs the newer version after the settings have been migrated. Some applications do this automatically after settings are migrated; however, other applications will only do this if the application was upgraded from the older version. When the application is upgraded, a set of files and/or registry keys is installed that indicates the older version of the application was previously installed. If you perform a clean installation of the newer version (which is the case in most migrations), the computer does not contain this set of files and registry keys so the mapping does not occur. In order to trick the newer version of the application into initiating this import process, your migration script may need to create these files and/or registry keys on the destination computer.

To identify which files and/or registry keys/values need to be created to cause the import, you should upgrade the older version of the application to the newer one and monitor the changes made to the file system and registry by using the same process described in [How To determine where each setting is stored](#). Once you know the set of files that the computer needs, you can use the `<addObjects>` element to add them to the destination computer.

- **The newer version of the application cannot read settings from the source computer and it is also unable to import the settings into the new format.** In this case, you will need to create a mapping for each setting from the old locations to the new locations. To do this, determine where the newer version stores each setting using the process described in [How to determine where each setting is stored](#). After you have created the mapping, apply the settings to the new location on the destination computer using the `<LocationModify>` element, and the **RelativeMove** and **ExactMove** helper functions.

Case 2: The destination computer already contains settings for the application.

We recommend that you migrate the settings after you install the application, but before the user runs the application for the first time. We recommend this because this ensures that there are no settings on the destination

computer when you migrate the settings. If you must install the application before the migration, you should delete any existing settings using the `<destinationCleanup>` element. If for any reason you want to preserve the settings that are on the destination computer, you can use the `<merge>` element and **DestinationPriority** helper function.

Case 3: The application overwrites settings when it is installed.

We recommend that you migrate the settings after you install the application, but before the user runs the application for the first time. We recommend this because this ensures that there are no settings on the destination computer when you migrate the settings. Also, when some applications are installed, they overwrite any existing settings that are on the computer. In this scenario, if you migrated the data before you installed the application, your customized settings would be overwritten. This is common for applications that store settings in locations that are outside of the user profile (typically these are settings that apply to all users). These universal settings are sometimes overwritten when an application is installed, and they are replaced by default values. To avoid this, you must install these applications before migrating the files and settings to the destination computer. By default with USMT, data from the source computer overwrites data that already exists in the same location on the destination computer.

Step 4: Create the migration XML component for the application

After you have completed steps 1 through 3, you will need to create a custom migration .xml file that migrates the application based on the information that you now have. You can use the MigApp.xml file as a model because it contains examples of many of the concepts discussed in this topic. You can also see [Custom XML Examples](#) for another sample .xml file.

Note We recommend that you create a separate .xml file instead of adding your script to the **MigApp.xml** file. This is because the **MigApp.xml** file is a very large file and it will be difficult to read and edit. In addition, if you reinstall USMT for some reason, the **MigApp.xml** file will be overwritten by the default version of the file and you will lose your customized version.

Important Some applications store information in the user profile that should not be migrated (for example, application installation paths, the computer name, and so on). You should make sure to exclude these files and registry keys from the migration.

Your script should do the following:

1. Check whether the application and correct version is installed by:
 - Searching for the installation uninstall key under **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall** using the **DoesObjectExist** helper function.
 - Checking for the correct version of the application executable file using the **DoesFileVersionMatch** helper function.
2. If the correct version of the application is installed, then ensure that each setting is migrated to the appropriate location on the destination computer.
 - If the versions of the applications are the same on both the source and destination computers, migrate each setting using the `<include>` and `<exclude>` elements.
 - If the version of the application on the destination computer is newer than the one on the source computer, and the application cannot import the settings, your script should either 1) add the set of files that trigger the import using the `<addObjects>` element or 2) create a mapping that applies the old settings to the correct location on the destination computer using the `<locationModify>` element, and the **RelativeMove** and **ExactMove** helper functions.
 - If you must install the application before migrating the settings, delete any settings that are already

on the destination computer using the `<destinationCleanup>` element.

For information about the .xml elements and helper functions, see [XML Elements Library](#).

Step 5: Test the application settings migration

On a test computer, install the operating system that will be installed on the destination computers. For example, if you are planning on migrating from Windows 7 to Windows 10, install Windows 10 and the application. Next, run LoadState on the test computer and verify that all settings migrate. Make corrections if necessary and repeat the process until all the necessary settings are migrated correctly.

To speed up the time it takes to collect and migrate the data, you can migrate only one user at a time, and you can exclude all other components from the migration except the application that you are testing. To specify only User1 in the migration, type: `/ue:* \ /ui:user1`. For more information, see [Exclude Files and Settings](#) and User options in the [ScanState Syntax](#) topic. To troubleshoot a problem, check the progress log, and the ScanState and LoadState logs, which contain warnings and errors that may point to problems with the migration.

Related topics

[USMT XML Reference](#)

[Conflicts and Precedence](#)

[XML Elements Library](#)

[Log Files](#)

Migrate EFS Files and Certificates

6/6/2019 • 2 minutes to read • [Edit Online](#)

This topic describes how to migrate Encrypting File System (EFS) certificates. For more information about the **/efs** options, see [ScanState Syntax](#).

To Migrate EFS Files and Certificates

Encrypting File System (EFS) certificates will be migrated automatically. However, by default, the User State Migration Tool (USMT) 10.0 fails if an encrypted file is found (unless you specify an **/efs** option). Therefore, you must specify **/efs:abort | skip | decryptcopy | copyraw | hardlink** with the ScanState command to migrate the encrypted files. Then, when you run the LoadState command on the destination computer, the encrypted file and the EFS certificate will be automatically migrated.

Note The **/efs** options are not used with the LoadState command.

Before using the ScanState tool for a migration that includes encrypted files and EFS certificates, you must ensure that all files in an encrypted folder are encrypted as well or remove the encryption attribute from folders that contain unencrypted files. If the encryption attribute has been removed from a file but not from the parent folder, the file will be encrypted during the migration using the credentials of the account used to run the LoadState tool.

You can run the Cipher tool at a Windows command prompt to review and change encryption settings on files and folders. For example, to remove encryption from a folder, at a command prompt type:

```
Cipher /D /S:<PATH>
```

Where *<Path>* is the full path of the topmost parent directory where the encryption attribute is set.

Related topics

[What Does USMT Migrate?](#)

[Identify File Types, Files, and Folders](#)

Migrate User Accounts

6/6/2019 • 2 minutes to read • [Edit Online](#)

By default, all users are migrated. The only way to specify which users to include and exclude is on the command line by using the User options. You cannot specify users in the migration XML files or by using the Config.xml file.

In this Topic

- [To migrate all user accounts and user settings](#)
- [To migrate two domain accounts \(User1 and User2\)](#)
- [To migrate two domain accounts \(User1 and User2\) and move User1 from the Contoso domain to the Fabrikam domain](#)

To migrate all user accounts and user settings

Links to detailed explanations of commands are available in the Related Topics section.

1. Log on to the source computer as an administrator, and specify the following in a **Command-Prompt** window:

```
scanstate \\server\share\migration\mystore /i:migdocs.xml /i:migapp.xml /o
```

2. Log on to the destination computer as an administrator.
3. Do one of the following:

- If you are migrating domain accounts, specify:

```
loadstate \\server\share\migration\mystore /i:migdocs.xml /i:migapp.xml
```

- If you are migrating local accounts along with domain accounts, specify:

```
loadstate \\server\share\migration\mystore /i:migdocs.xml /i:migapp.xml /lac /lae
```

Note You do not have to specify the **/lae** option, which enables the account that was created with the **/lac** option. Instead, you can create a disabled local account by specifying only the **/lac** option, and then a local administrator needs to enable the account on the destination computer.

To migrate two domain accounts (User1 and User2)

Links to detailed explanations of commands are available in the Related Topics section.

1. Log on to the source computer as an administrator, and specify:

```
scanstate \\server\share\migration\mystore /ue:*\* /ui:contoso\user1 /ui:fabrikam\user2 /i:migdocs.xml /i:migapp.xml /o
```

2. Log on to the destination computer as an administrator.
3. Specify the following:

```
loadstate \\server\share\migration\mystore /i:migdocs.xml /i:migapp.xml
```

To migrate two domain accounts (User1 and User2) and move User1

from the Contoso domain to the Fabrikam domain

Links to detailed explanations of commands are available in the Related Topics section.

1. Log on to the source computer as an administrator, and type the following at the command-line prompt:

```
scanstate \\server\share\migration\mystore /ue:* \* /ui:contoso\user1 /ui:contoso\user2 /i:migdocs.xml /i:migapp.xml /o
```

2. Log on to the destination computer as an administrator.
3. Specify the following:

```
loadstate \\server\share\migration\mystore /mu:contoso\user1:fabrikam\user2 /i:migdocs.xml /i:migapp.xml
```

Related topics

[Identify Users](#)

[ScanState Syntax](#)

[LoadState Syntax](#)

Reroute Files and Settings

5/31/2019 • 2 minutes to read • [Edit Online](#)

To reroute files and settings, create a custom .xml file and specify this file name on both the ScanState and LoadState commandlines. This enables you to keep your changes separate from the default .xml files, so that it is easier to track your modifications.

In this topic:

- [Reroute a Folder](#)
- [Reroute a Specific File Type](#)
- [Reroute a Specific File](#)

Reroute a Folder

The following custom .xml file migrates the directories and files from C:\EngineeringDrafts into the My Documents folder of every user. %CSIDL_PERSONAL% is the virtual folder representing the My Documents desktop item, which is equivalent to CSIDL_MYDOCUMENTS.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlexml/test">
<component type="Documents" context="User">
  <displayName>Engineering Drafts Documents to Personal Folder</displayName>
  <role role="Data">
    <rules>
      <!-- Migrate all directories and files present in c:\EngineeringDrafts folder -->
      <include>
        <objectSet>
          <pattern type="File">C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
      <!-- This migrates all files and directories from C:\EngineeringDrafts to every user's personal folder. -
      ->
      <locationModify script="MigXmlHelper.RelativeMove('C:\EngineeringDrafts','%CSIDL_PERSONAL%')">
        <objectSet>
          <pattern type="File">C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </locationModify>
    </rules>
  </role>
</component>
</migration>
```

Reroute a Specific File Type

The following custom .xml file reroutes .mp3 files located in the fixed drives on the source computer into the C:\Music folder on the destination computer.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="System">
  <displayName>All .mp3 files to My Documents</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("* [.mp3]", "Fixed")</script>
        </objectSet>
      </include>
      <!-- Migrates all the .mp3 files in the store to the C:\Music folder during LoadState -->
      <locationModify script="MigXmlHelper.Move('C:\Music')">
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("* [.mp3]", "Fixed")</script>
        </objectSet>
      </locationModify>
    </rules>
  </role>
</component>
</migration>

```

Reroute a Specific File

The following custom .xml file migrates the Sample.doc file from C:\EngineeringDrafts into the My Documents folder of every user. %CSIDL_PERSONAL% is the virtual folder representing the My Documents desktop item, which is equivalent to CSIDL_MYDOCUMENTS.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/test">
<component type="Documents" context="User">
<displayName>Sample.doc into My Documents</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\ [Sample.doc]</pattern>
        </objectSet>
      </include>
      <locationModify script="MigXmlHelper.RelativeMove('C:\EngineeringDrafts','%CSIDL_PERSONAL%')">
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\ [Sample.doc]</pattern>
        </objectSet>
      </locationModify>
    </rules>
  </role>
</component>
</migration>

```

Related topics

[Customize USMT XML Files](#)

[Conflicts and Precedence](#)

[USMT XML Reference](#)

Verify the Condition of a Compressed Migration Store

5/31/2019 • 3 minutes to read • [Edit Online](#)

When you migrate files and settings during a typical PC-refresh migration, the user state is usually stored in a compressed folder on the intermediate store. This compressed folder, also called the compressed migration store, is a single image file that contains:

- All of the files being migrated.
- The user's settings.
- A catalog file that contains metadata for all files in the migration store.

When you run the **LoadState** command to load the data from these files to the destination computer, LoadState requires a valid catalog file in order to open the migration store. You can run the **UsmtUtils** command with the **/verify** option to determine whether the compressed migration store is intact, or whether it contains corrupted files or a corrupted catalog. You should run the **/verify** option on the migration store before you overwrite the original user-state files and settings.

When you use the **/verify** option, you can specify what type of information to report in the UsmtUtils log file. These report types are:

- **Catalog**: Displays the status of only the catalog file.
- **All**: Displays the status of all files, including the catalog file.
- **Failure only**: Displays only the files that are corrupted.

In This Topic

The following sections demonstrate how to run the **UsmtUtils** command with the **/verify** option, and how to specify the information to display in the UsmtUtils log file.

- [The UsmtUtils syntax for the /verify option](#)
- [To verify that the migration store is intact](#)
- [To verify the status of only the catalog file](#)
- [To verify the status of all files](#)
- [To verify the status of the files and return only the corrupted files](#)

The UsmtUtils Syntax for the /verify Option

To verify the condition of a compressed migration store, use the following UsmtUtils syntax:

```
cd /d<USMTpath>usmtutils /verify[:<reportType>] <filePath> [/l:<logfile>] [/decrypt[:<AlgID>] {/key:  
<keystring> | /keyfile:<filename>}]
```

Where the placeholders have the following values:

- *<USMTpath>* is the location where you have saved the USMT files and tools.
- *<reportType>* specifies whether to report on all files, corrupted files only, or the status of the catalog.

- *<filePath>* is the location of the compressed migration store.
- *<logfile>* is the location and name of the log file.
- *<AlgID>* is the cryptographic algorithm that was used to create the migration store on the **ScanState** command line.
- *<keystring>* is the encryption key that was used to encrypt the migration store.
- *<filename>* is the location and name of the text file that contains the encryption key.

To Verify that the Migration Store is Intact

To verify whether the migration store is intact or whether it contains corrupted files or a corrupted catalog, type:

```
usmtutils /verify D:\MyMigrationStore\store.mig
```

Because no report type is specified, UsmtUtils displays the default summary report.

To Verify the Status of Only the Catalog File

To verify whether the catalog file is corrupted or intact, type:

```
usmtutils /verify:catalog D:\MyMigrationStore\store.mig
```

To Verify the Status of all Files

To verify whether there are any corrupted files in the compressed migration store, and to specify the name and location of the log file, type:

```
usmtutils /verify:all D:\MyMigrationStore\store.mig /decrypt /l:D:\UsmtUtilsLog.txt
```

In addition to verifying the status of all files, this example decrypts the files. Because no encryption algorithm is specified, UsmtUtils uses the default 3DES cryptographic algorithm.

To Verify the Status of the Files and Return Only the Corrupted Files

In this example, the log file will only list the files that became corrupted during the ScanState process. This list will include the catalog file if it is also corrupted.

```
usmtutils /verify:failureonly D:\MyMigrationStore\USMT\store.mig /decrypt:AES_192  
/keyfile:D:\encryptionKey.txt
```

This example also decrypts the files by specifying the cryptographic algorithm and the location of the file that contains the encryption key.

Next Steps

If the **/verify** option indicates that there are corrupted files in the migration store, you can use the **/extract** option in the UsmtUtils tool to recover data from some corrupted stores. For more information, see [Extract Files from a Compressed USMT Migration Store](#).

Related topics

[UsmtUtils Syntax](#)

[Return Codes](#)

User State Migration Tool (USMT) Troubleshooting

6/6/2019 • 2 minutes to read • [Edit Online](#)

The following table describes topics that address common User State Migration Tool (USMT) 10.0 issues and questions. These topics describe tools that you can use to troubleshoot issues that arise during your migration.

In This Section

Common Issues	Find troubleshooting solutions for common problems in USMT.
Frequently Asked Questions	Find answers to questions about how to use USMT.
Log Files	Learn how to enable logging to help you troubleshoot issues in USMT.
Return Codes	Learn how to use return codes to identify problems in USMT.
USMT Resources	Find more information and support for using USMT.

Related topics

[USMT Best Practices](#)

[User State Migration Tool \(USMT\) Overview Topics](#)

[User State Migration Tool \(USMT\) How-to topics](#)

[User State Migration Toolkit \(USMT\) Reference](#)

Common Issues

6/14/2019 • 14 minutes to read • [Edit Online](#)

The following sections discuss common issues that you might see when you run the User State Migration Tool (USMT) 10.0 tools. USMT produces log files that describe in further detail any errors that occurred during the migration process. These logs can be used to troubleshoot migration failures.

In This Topic

[User Account Problems](#)

[Command-line Problems](#)

[XML File Problems](#)

[Migration Problems](#)

[Offline Migration Problems](#)

[Hard Link Migration Problems](#)

[USMT does not migrate the Start layout](#)

General Guidelines for Identifying Migration Problems

When you encounter a problem or error message during migration, you can use the following general guidelines to help determine the source of the problem:

- Examine the ScanState, LoadState, and UsmtUtils logs to obtain the exact USMT error messages and Windows® application programming interface (API) error messages. For more information about USMT return codes and error messages, see [Return Codes](#). For more information about Windows API error messages, type **nethelpmsg** on the command line.

In most cases, the ScanState and LoadState logs indicate why a USMT migration is failing. We recommend that you use the **/v:5** option when testing your migration. This verbosity level can be adjusted in a production migration; however, reducing the verbosity level might make it more difficult to diagnose failures that are encountered during production migrations. You can use a verbosity level higher than 5 if you want the log files output to go to a debugger.

Note

Running the ScanState and LoadState tools with the **/v:5** option creates a detailed log file. Although this option makes the log file large, the extra detail can help you determine where migration errors occurred.

- Use the **/Verify** option in the UsmtUtils tool to determine whether any files in a compressed migration store are corrupted. For more information, see [Verify the Condition of a Compressed Migration Store](#).
- Use the **/Extract** option in the UsmtUtils tool to extract files from a compressed migration store. For more information, see [Extract Files from a Compressed USMT Migration Store](#).
- Create a progress log using the **/Progress** option to monitor your migration.
- For the source and destination computers, obtain operating system information, and versions of applications such as Internet Explorer and any other relevant programs. Then verify the exact steps that are needed to reproduce the problem. This information might help you to understand what is wrong and to reproduce the issue in your testing environment.

- Log off after you run the LoadState tool. Some settings—for example, fonts, desktop backgrounds, and screen-saver settings—will not take effect until the next time the end user logs on.
- Close all applications before running ScanState or LoadState tools. If some applications are running during the ScanState or LoadState process, USMT might not migrate some data. For example, if Microsoft Outlook® is open, USMT might not migrate PST files.

Note

USMT will fail if it cannot migrate a file or setting unless you specify the **/c** option. When you specify the **/c** option, USMT ignores errors. However, it logs an error when it encounters a file that is in use that did not migrate.

User Account Problems

The following sections describe common user account problems. Expand the section to see recommended solutions.

I'm having problems creating local accounts on the destination computer.

Resolution: For more information about creating accounts and migrating local accounts, see [Migrate User Accounts](#).

Not all of the user accounts were migrated to the destination computer.

Causes/Resolutions There are two possible causes for this problem:

When running the ScanState tool on Windows Vista, or the ScanState and LoadState tools on Windows 7, Windows 8, or Windows 10, you must run them in Administrator mode from an account with administrative credentials to ensure that all specified users are migrated. To run in Administrator mode:

1. Click **Start**.
2. Click **All Programs**.
3. Click **Accessories**.
4. Right-click **Command Prompt**.
5. Click **Run as administrator**.

Then specify your LoadState or ScanState command. If you do not run USMT in Administrator mode, only the user profile that is logged on will be included in the migration.

Any user accounts on the computer that have not been used will not be migrated. For example, if you add User1 to the computer, but User1 never logs on, then USMT will not migrate the User1 account.

User accounts that I excluded were migrated to the destination computer.

Cause: The command that you specified might have had conflicting **/ui** and **/ue** options. If a user is specified with the **/ui** option and is also specified to be excluded with either the **/ue** or **/uel** options, the user will be included in the migration. For example, if you specify `/ui:domain1* /ue:domain1\user1`, then User1 will be migrated because the **/ui** option takes precedence.

Resolution: For more information about how to use the **/ui** and **/ue** options together, see the examples in the [ScanState Syntax](#) topic.

I am using the /uel option, but many accounts are still being included in the migration.

Cause The **/uel** option depends on the last modified date of the users' NTUser.dat file. There are scenarios in which this last modified date might not match the users' last logon date.

Resolution This is a limitation of the **/uel** option. You might need to exclude these users manually with the **/ue**

option.

The LoadState tool reports an error as return code 71 and fails to restore a user profile during a migration test.

Cause: During a migration test, if you run the ScanState tool on your test computer and then delete user profiles in order to test the LoadState tool on the same computer, you may have a conflicting key present in the registry. Using the **net use** command to remove a user profile will delete folders and files associated with that profile, but will not remove the registry key.

Resolution: To delete a user profile, use the **User Accounts** item in Control Panel. To correct an incomplete deletion of a user profile:

1. Open the registry editor by typing `regedit` at an elevated command prompt.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList`.

Each user profile is stored in a System Identifier key under `ProfileList`.

3. Delete the key for the user profile you are trying to remove.

Files that were not encrypted before the migration are now encrypted with the account used to run the LoadState tool.

Cause: The ScanState tool was run using the **/EFS: copyraw** option to migrate encrypted files and Encrypting File System (EFS) certificates. The encryption attribute was set on a folder that was migrated, but the attribute was removed from file contents of that folder prior to migration.

Resolution: Before using the ScanState tool for a migration that includes encrypted files and EFS certificates, you can run the Cipher tool at the command prompt to review and change encryption settings on files and folders. You must remove the encryption attribute from folders that contain unencrypted files or encrypt the contents of all files within an encrypted folder.

To remove encryption from files that have already been migrated incorrectly, you must log on to the computer with the account that you used to run the LoadState tool and then remove the encryption from the affected files.

The LoadState tool reports an error as return code 71 and a Windows Error 2202 in the log file.

Cause: The computer name was changed during an offline migration of a local user profile.

Resolution: You can use the **/mu** option when you run the LoadState tool to specify a new name for the user. For example,

```
loadstate /i:migapp.xml /i:migdocs.xml \\server\share\migration\mystore  
/progress:prog.log /l:load.log /mu:fareast\user1:farwest\user1
```

Command-line Problems

The following sections describe common command-line problems. Expand the section to see recommended solutions.

I received the following error message: "Usage Error: You cannot specify a file path with any of the command-line options that exceeds 256 characters."

Cause: You might receive this error message in some cases even if you do not specify a long store or file path, because the path length is calculated based on the absolute path. For example, if you run the **scanstate.exe /o store** command from `C:\Program Files\USMT40`, then each character in "`C:\Program Files\USMT40`" will be added to the length of "store" to get the length of the path.

Resolution: Ensure that the total path length—the store path plus the current directory—does not exceed 256 characters.

I received the following error message: "USMT was unable to create the log file(s). Ensure that you have write access to the log directory."

Cause: If you are running the ScanState or LoadState tools from a shared network resource, you will receive this error message if you do not specify `/l`.

Resolution: To fix this issue in this scenario, specify the `/l:scan.log` or `/l:load.log` option.

XML File Problems

The following sections describe common XML file problems. Expand the section to see recommended solutions.

I used the `/genconfig` option to create a `Config.xml` file, but I see only a few applications and components that are in `MigApp.xml`. Why does `Config.xml` not contain all of the same applications?

Cause: `Config.xml` will contain only operating system components, applications, and the user document sections that are in both of the `.xml` files and are installed on the computer when you run the `/genconfig` option. Otherwise, these applications and components will not appear in the `Config.xml` file.

Resolution: Install all of the desired applications on the computer before running the `/genconfig` option. Then run ScanState with all of the `.xml` files. For example, run the following:

```
scanstate /genconfig:config.xml /i:migdocs.xml /i:migapp.xml /v:5 /l:scanstate.log
```

I am having problems with a custom `.xml` file that I authored, and I cannot verify that the syntax is correct.

Resolution: You can load the XML schema (`MigXML.xsd`), included with USMT, into your XML authoring tool. For examples, see the [Visual Studio Development Center](#). Then, load your `.xml` file in the authoring tool to see if there is a syntax error. In addition, see [USMT XML Reference](#) for more information about using the XML elements.

I am using a MigXML helper function, but the migration isn't working the way I expected it to. How do I troubleshoot this issue?

Cause: Typically, this issue is caused by incorrect syntax used in a helper function. You receive a Success return code, but the files you wanted to migrate did not get collected or applied, or weren't collected or applied in the way you expected.

Resolution: You should search the ScanState or LoadState log for either the component name which contains the MigXML helper function, or the MigXML helper function title, so that you can locate the related warning in the log file.

Migration Problems

The following sections describe common migration problems. Expand the section to see recommended solutions.

Files that I specified to exclude are still being migrated.

Cause: There might be another rule that is including the files. If there is a more specific rule or a conflicting rule, the files will be included in the migration.

Resolution: For more information, see [Conflicts and Precedence](#) and the Diagnostic Log section in [Log Files](#).

I specified rules to move a folder to a specific location on the destination computer, but it has not migrated correctly.

Cause: There might be an error in the XML syntax.

Resolution: You can use the USMT XML schema (`MigXML.xsd`) to write and validate migration `.xml` files. Also see the XML examples in the following topics:

[Conflicts and Precedence](#)

[Exclude Files and Settings](#)

[Reroute Files and Settings](#)

[Include Files and Settings](#)

[Custom XML Examples](#)

After LoadState completes, the new desktop background does not appear on the destination computer.

There are three typical causes for this issue.

Cause #1: Some settings such as fonts, desktop backgrounds, and screen-saver settings are not applied by LoadState until after the destination computer has been restarted.

Resolution: To fix this issue, log off, and then log back on to see the migrated desktop background.

Cause #2: If the source computer was running Windows® XP and the desktop background was stored in the *Drive:\WINDOWS\Web\Wallpaper* folder—the default folder where desktop backgrounds are stored in Windows XP—the desktop background will not be migrated. Instead, the destination computer will have the default Windows® desktop background. This will occur even if the desktop background was a custom picture that was added to the *\WINDOWS\Web\Wallpaper* folder. However, if the end user sets a picture as the desktop background that was saved in another location, for example, My Pictures, then the desktop background will migrate.

Resolution: Ensure that the desktop background images that you want to migrate are not in the *\WINDOWS\Web\Wallpaper* folder on the source computer.

Cause #3: If ScanState was not run on Windows XP from an account with administrative credentials, some operating system settings will not migrate. For example, desktop background settings, screen-saver selections, modem options, media-player settings, and Remote Access Service (RAS) connection phone book (.pbk) files and settings will not migrate.

Resolution: Run the ScanState and LoadState tools from within an account with administrative credentials.

I included MigApp.xml in the migration, but some PST files aren't migrating.

Cause: The MigApp.xml file migrates only the PST files that are linked to Outlook profiles.

Resolution: To migrate PST files that are not linked to Outlook profiles, you must create a separate migration rule to capture these files.

USMT does not migrate the Start layout

Description: You are using USMT to migrate profiles from one installation of Windows 10 to another installation of Windows 10 on different hardware. After migration, the user signs in on the new device and does not have the Start menu layout they had previously configured.

Cause: A code change in the Start Menu with Windows 10 version 1607 and later is incompatible with this USMT function.

Resolution: The following workaround is available:

1. With the user signed in, back up the Start layout using the following Windows PowerShell command. You can specify a different path if desired:

```
Export-StartLayout -Path "C:\Layout\user1.xml"
```

2. Migrate the user's profile with USMT.
3. Before the user signs in on the new device, import the Start layout using the following Windows PowerShell command:

```
Import-StartLayout -LayoutPath "C:\Layout\user1.xml" -MountPath %systemdrive%
```

This workaround changes the Default user's Start layout. The workaround does not scale to a mass migrations or multiuser devices, but it can potentially unblock some scenarios. If other users will sign on to the device you should delete layoutmodification.xml from the Default user profile. Otherwise, all users who sign on to that device will use the imported Start layout.

Offline Migration Problems

The following sections describe common offline migration problems. Expand the section to see recommended solutions.

Some of my system settings do not migrate in an offline migration.

Cause: Some system settings, such as desktop backgrounds and network printers, are not supported in an offline migration. For more information, see [What Does USMT Migrate?](#)

Resolution: In an offline migration, these system settings must be restored manually.

The ScanState tool fails with return code 26.

Cause: A common cause of return code 26 is that a temp profile is active on the source computer. This profile maps to c:\users\temp. The ScanState log shows a MigStartupOfflineCaught exception that includes the message "User profile duplicate SID error".

Resolution: You can reboot the computer to get rid of the temp profile or you can set MIG_FAIL_ON_PROFILE_ERROR=0 to skip the error and exclude the temp profile.

Include and Exclude rules for migrating user profiles do not work the same offline as they do online.

Cause: When offline, the DNS server cannot be queried to resolve the user name and SID mapping.

Resolution: Use a Security Identifier (SID) to include a user when running the ScanState tool. For example:

```
Scanstate /ui:S1-5-21-124525095-708259637-1543119021*
```

The wild card (*) at the end of the SID will migrate the *SID_Classes* key as well.

You can also use patterns for SIDs that identify generic users or groups. For example, you can use the */ue:*-500* option to exclude the local administrator accounts. For more information about Windows SIDs, see [this Microsoft Web site](#).

My script to wipe the disk fails after running the ScanState tool on a 64-bit system.

Cause: The HKLM registry hive is not unloaded after the ScanState tool has finished running.

Resolution: Reboot the computer or unload the registry hive at the command prompt after the ScanState tool has finished running. For example, at a command prompt, type:

```
reg.exe unload hklm\%dest$software
```

Hard-Link Migration Problems

The following sections describe common hard-link migration problems. Expand the section to see recommended solutions.

EFS files are not restored to the new partition.

Cause: EFS files cannot be moved to a new partition with a hard link. The **/efs:hardlink** command-line option is only applicable to files migrated on the same partition.

Resolution: Use the **/efs:copyraw** command-line option to copy EFS files during the migration instead of creating hard links, or manually copy the EFS files from the hard-link store.

The ScanState tool cannot delete a previous hard-link migration store.

Cause: The migration store contains hard links to locked files.

Resolution: Use the UsmtUtils tool to delete the store or change the store name. For example, at a command prompt, type:

```
USMTutils /rd <storedir>
```

You should also reboot the machine.

Related topics

[User State Migration Tool \(USMT\) Troubleshooting](#)

[Frequently Asked Questions](#)

[Return Codes](#)

[UsmtUtils Syntax](#)

Frequently Asked Questions

5/31/2019 • 5 minutes to read • [Edit Online](#)

The following sections provide frequently asked questions and recommended solutions for migrations using User State Migration Tool (USMT) 10.0.

General

How much space is needed on the destination computer?

The destination computer needs enough available space for the following:

- Operating system
- Applications
- Uncompressed store

Can I store the files and settings directly on the destination computer or do I need a server?

You do not need to save the files to a server. If you are moving the user state to a new computer, you can create the store on a shared folder, on media that you can remove, such as a USB flash drive (UFD), or you can store it directly on the destination computer, as in the following steps:

1. Create and share the directory C:\store on the destination computer.
2. Run the ScanState tool on the source computer and save the files and settings to \\DestinationComputerName\store
3. Run the LoadState tool on the destination computer and specify C:\store as the store location.

Can I migrate data between operating systems with different languages?

No. USMT does not support migrating data between operating systems with different languages; the source computer's operating-system language must match the destination computer's operating-system language.

Can I change the location of the temporary directory on the destination computer?

Yes. The environment variable USMT_WORKING_DIR can be changed to an alternative temporary directory. There are some offline migration scenarios where this is necessary, for example, when the USMT binaries are located on read-only Windows Preinstallation Environment (WinPE) boot media.

How do I install USMT?

Because USMT is included in Windows Assessment and Deployment Kit (Windows ADK), you need to install the Windows ADK package on at least one computer in your environment. However, the USMT binaries are designed to be deployed using xcopy. This means that they are installed on a computer simply by recursively copying the USMT directory from the computer containing the Windows ADK to each client computer.

How do I uninstall USMT?

If you have installed the Windows ADK on the computer, uninstalling Windows ADK will uninstall USMT. For client computers that do not have the Windows ADK installed, you can simply delete the USMT directory to uninstall USMT.

Files and Settings

How can I exclude a folder or a certain type of file from the migration?

You can use the **<unconditionalExclude>** element to globally exclude data from the migration. For example, you can use this element to exclude all MP3 files on the computer or to exclude all files from C:\UserData. This element excludes objects regardless of any other **<include>** rules that are in the .xml files. For an example, see **<unconditionalExclude>** in the [Exclude Files and Settings](#) topic. For the syntax of this element, see [XML Elements Library](#).

What happens to files that were located on a drive that does not exist on the destination computer?

USMT migrates the files to the %SystemDrive% while maintaining the correct folder hierarchy. For example, if E:\data\File.pst is on the source computer, but the destination computer does not have an E:\ drive, the file will be migrated to C:\data\File.pst, if C:\ is the system drive. This holds true even when **<locationModify>** rules attempt to move data to a drive that does not exist on the destination computer.

USMT .xml Files

Where can I get examples of USMT .xml files?

The following topics include examples of USMT .xml files:

- [Exclude Files and Settings](#)
- [Reroute Files and Settings](#)
- [Include Files and Settings](#)
- [Custom XML Examples](#)

Can I use custom .xml files that were written for USMT 5.0?

Yes. You can use custom .xml files that were written for USMT 5.0 with USMT for Windows 10. However, in order to use new USMT functionality, you must revisit your custom USMT files and refresh them to include the new command-line options and XML elements.

How can I validate the .xml files?

You can use the USMT XML Schema (MigXML.xsd) to write and validate migration .xml files.

Why must I list the .xml files with both the ScanState and LoadState commands?

The .xml files are not copied to the store as in previous versions of USMT. Because the ScanState and LoadState tools need the .xml files to control the migration, you must specify the same set of .xml files for the **ScanState** and **LoadState** commands. If you used a particular set of mig*.xml files in the ScanState tool, either called through the **/auto** option, or individually through the **/i** option, then you should use same option to call the exact same mig*.xml files in the LoadState tool. However, you do not have to specify the Config.xml file, unless you want to exclude some of the files and settings that you migrated to the store. For example, you might want to migrate the My Documents folder to the store, but not to the destination computer. To do this, modify the Config.xml file and specify the updated file with the **LoadState** command. **LoadState** will migrate only the files and settings that you want to migrate.

If you exclude an .xml file from the **LoadState** command, then all of the data that is in the store that was migrated with the missing .xml files will be migrated. However, the migration rules that were specified for the **ScanState** command will not apply. For example, if you exclude a MigApp.xml file that has a rerouting rule such as `MigsysHelperFunction.RelativeMove("c:\data", "%CSIDL_PERSONAL%")`, USMT will not reroute the files. Instead, it will migrate them to C:\data.

Which files can I modify and specify on the command line?

You can specify the MigUser.xml and MigApp.xml files on the command line. You can modify each of these files. The migration of operating system settings is controlled by the manifests, which you cannot modify. If you want to exclude certain operating-system settings or any other components, create and modify the Config.xml file.

What happens if I do not specify the .xml files on the command line?

- **ScanState**

If you do not specify any files with the **ScanState** command, all user accounts and default operating system components are migrated.

- **LoadState**

If you do not specify any files with the **LoadState** command, all data that is in the store is migrated. However, any target-specific migration rules that were specified in .xml files with the **ScanState** command will not apply. For example, if you exclude a MigApp.xml file that has a rerouting rule such as

```
MigsysHelperFunction.RelativeMove("c:\data", "%CSIDL_PERSONAL%")
```

, USMT will not reroute the files. Instead, it will migrate them to C:\data.

Conflicts and Precedence

What happens when there are conflicting XML rules or conflicting objects on the destination computer?

For more information, see [Conflicts and Precedence](#).

Related topics

[User State Migration Tool \(USMT\) Troubleshooting](#)

[Extract Files from a Compressed USMT Migration Store](#)

[Verify the Condition of a Compressed Migration Store](#)

Log Files

6/26/2019 • 9 minutes to read • [Edit Online](#)

You can use User State Migration Tool (USMT) 10.0 logs to monitor your migration and to troubleshoot errors and failed migrations. This topic describes the available command-line options to enable USMT logs, and new XML elements that configure which types of errors are fatal and should halt the migration, which types are non-fatal and should be skipped so that the migration can continue.

[Log Command-Line Options](#)

[ScanState and LoadState Logs](#)

[Progress Log](#)

[List Files Log](#)

[Diagnostic Log](#)

Log Command-Line Options

The following table describes each command-line option related to logs, and it provides the log name and a description of what type of information each log contains.

COMMAND LINE OPTION	FILE NAME	DESCRIPTION
<i>/l[Path]FileName</i>	Scanstate.log or LoadState.log	Specifies the path and file name of the ScanState.log or LoadState log.
<i>/progress[Path]FileName</i>	Specifies the path and file name of the Progress log.	Provides information about the status of the migration, by percentage complete.
<i>/v[VerbosityLevel]</i>	Not applicable	See the "Monitoring Options" section in ScanState Syntax .
<i>/listfiles[Path]FileName</i>	Specifies the path and file name of the Listfiles log.	Provides a list of the files that were migrated.
Set the environment variable MIG_ENABLE_DIAG to a path to an XML file.	USMTDiag.xml	The diagnostic log contains detailed system environment information, user environment information, and information about the migration units (migunits) being gathered and their contents.

Note You cannot store any of the log files in *StorePath*. If you do, the log will be overwritten when USMT is run.

ScanState and LoadState Logs

ScanState and LoadState logs are text files that are create when you run the ScanState and LoadState tools. You

can use these logs to help monitor your migration. The content of the log depends on the command-line options that you use and the verbosity level that you specify. For more information about verbosity levels, see [Monitoring Options in ScanState Syntax](#).

Progress Log

You can create a progress log using the **/progress** option. External tools, such as Microsoft System Center Operations Manager 2007, can parse the progress log to update your monitoring systems. The first three fields in each line are fixed as follows:

- **Date:** Date, in the format of *day shortNameOfTheMonth year*. For example: 08 Jun 2006.
- **Local time:** Time, in the format of *hrs:minutes:seconds* (using a 24-hour clock). For example: 13:49:13.
- **Migration time:** Duration of time that USMT was run, in the format of *hrs:minutes:seconds*. For example: 00:00:10.

The remaining fields are key/value pairs as indicated in the following table.

KEY	VALUE
program	ScanState.exe or LoadState.exe.
productVersion	The full product version number of USMT.
computerName	The name of the source or destination computer on which USMT was run.
commandLine	The full command used to run USMT.
PHASE	Reports that a new phase in the migration is starting. This can be one of the following: <ul style="list-style-type: none"> • Initializing • Scanning • Collecting • Saving • Estimating • Applying
detectedUser	<ul style="list-style-type: none"> • For the ScanState tool, these are the users USMT detected on the source computer that can be migrated. • For the LoadState tool, these are the users USMT detected in the store that can be migrated.
includedInMigration	Defines whether the user profile/component is included for migration. Valid values are Yes or No.

KEY	VALUE
forUser	<p>Specifies either of the following:</p> <ul style="list-style-type: none"> • The user state being migrated. • <i>This Computer</i>, meaning files and settings that are not associated with a user.
detectedComponent	<p>Specifies a component detected by USMT.</p> <ul style="list-style-type: none"> • For ScanState, this is a component or application that is installed on the source computer. • For LoadState, this is a component or application that was detected in the store.
totalSizeInMBToTransfer	<p>Total size of the files and settings to migrate in megabytes (MB).</p>
totalPercentageCompleted	<p>Total percentage of the migration that has been completed by either ScanState or LoadState.</p>
collectingUser	<p>Specifies which user ScanState is collecting files and settings for.</p>
totalMinutesRemaining	<p>Time estimate, in minutes, for the migration to complete.</p>
error	<p>Type of non-fatal error that occurred. This can be one of the following:</p> <ul style="list-style-type: none"> • UnableToCopy: Unable to copy to store because the disk on which the store is located is full. • UnableToOpen: Unable to open the file for migration because the file is opened in non-shared mode by another application or service. • UnableToCopyCatalog: Unable to copy because the store is corrupted. • UnableToAccessDevice: Unable to access the device. • UnableToApply: Unable to apply the setting to the destination computer.
objectName	<p>The name of the file or setting that caused the non-fatal error.</p>

KEY	VALUE
action	Action taken by USMT for the non-fatal error. The values are: <ul style="list-style-type: none"> • Ignore: Non-fatal error ignored and the migration continued because the /c option was specified on the command line. • Abort: Stopped the migration because the /c option was not specified.
errorCode	The errorCode or return value.
numberOfIgnoredErrors	The total number of non-fatal errors that USMT ignored.
message	The message corresponding to the errorCode.

List Files Log

The List files log (Listfiles.txt) provides a list of the files that were migrated. This list can be used to troubleshoot XML issues or can be retained as a record of the files that were gathered into the migration store. The List Files log is only available for ScanState.exe.

Diagnostic Log

You can obtain the diagnostic log by setting the environment variable MIG_ENABLE_DIAG to a path to an XML file.

The diagnostic log contains:

- Detailed system environment information
- Detailed user environment information
- Information about the migration units (migunits) being gathered and their contents

Using the Diagnostic Log

The diagnostic log is essentially a report of all the migration units (migunits) included in the migration. A migunit is a collection of data that is identified by the component it is associated with in the XML files. The migration store is made up of all the migunits in the migration. The diagnostic log can be used to verify which migunits were included in the migration and can be used for troubleshooting while authoring migration XML files.

The following examples describe common scenarios in which you can use the diagnostic log.

Why is this file not migrating when I authored an "include" rule for it?

Let's imagine that we have the following directory structure and that we want the "data" directory to be included in the migration along with the "New Text Document.txt" file in the "New Folder." The directory of **C:\data** contains:

```

01/21/2009 10:08 PM <DIR>      .
01/21/2009 10:08 PM <DIR>      ..
01/21/2009 10:08 PM <DIR>      New Folder
01/21/2009 09:19 PM          13 test (1).txt
01/21/2009 09:19 PM          13 test.txt
                2 File(s)          26 bytes

```

The directory of **C:\data\New Folder** contains:

```

01/21/2009 10:08 PM <DIR>      .
01/21/2009 10:08 PM <DIR>      ..
01/21/2009 10:08 PM          0 New Text Document.txt
                1 File(s)          0 bytes

```

To migrate these files you author the following migration XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/TestSuite_BUGFIX">

<component context="System" type="Application">
  <displayName>DATA1</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File">c:\data\ [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

However, upon testing the migration you notice that the "New Text Document.txt" file isn't included in the migration. To troubleshoot this failure, the migration can be repeated with the environment variable MIG_ENABLE_DIAG set such that the diagnostic log is generated. Upon searching the diagnostic log for the component "DATA1", the following XML section is discovered:

```

<MigUnitList>
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)" Context="System" ConfidenceLevel="100" Group="Applications"
Role="UserData" Agent="CMXEAgent" Selected="true" Supported="true">
<Patterns Type="Include">
<Pattern Type="File" Path="C:\data [*]" />
</Patterns>
</MigUnit>
</MigUnitList>
<Perform Name="Gather" User="System">
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)">
<Operation Name="Store" Type="File" Path="C:\data" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test (1).txt]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test.txt]" SimObj="false" Success="true"/>
</MigUnit>
</Perform>

```

Analysis of this XML section reveals the migunit that was created when the migration rule was processed. The <Perform> section details the actual files that were scheduled for gathering and the result of the gathering operation. The "New Text Document.txt" file doesn't appear in this section, which confirms that the migration rule was not correctly authored.

An analysis of the XML elements reference topic reveals that the <pattern> tag needs to be modified as follows:

```
<pattern type="File">c:\data\* [*]</pattern>
```

When the migration is performed again with the modified tag, the diagnostic log reveals the following:

```
<MigUnitList>
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)" Context="System" ConfidenceLevel="100" Group="Applications"
Role="UserData" Agent="CMXEAgent" Selected="true" Supported="true">
<Patterns Type="Include">
<Pattern Type="File" Path="C:\data\* [*]" />
</Patterns>
</MigUnit>
</MigUnitList>
<Perform Name="Gather" User="System">
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)">
<Operation Name="Store" Type="File" Path="C:\data" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test (1).txt]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test.txt]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder [New Text Document.txt]" SimObj="false"
Success="true"/>
</MigUnit>
</Perform>
```

This diagnostic log confirms that the modified <pattern> value enables the migration of the file.

Why is this file migrating when I authored an exclude rule excluding it?

In this scenario, you have the following directory structure and you want all files in the "data" directory to migrate, except for text files. The **C:\Data** folder contains:

```
Directory of C:\Data

01/21/2009  10:08 PM    <DIR>          .
01/21/2009  10:08 PM    <DIR>          ..
01/21/2009  10:08 PM    <DIR>          New Folder
01/21/2009  09:19 PM                13 test (1).txt
01/21/2009  09:19 PM                13 test.txt
                2 File(s)                26 bytes
```

The **C:\Data\New Folder** contains:

```
01/21/2009  10:08 PM    <DIR>          .
01/21/2009  10:08 PM    <DIR>          ..
01/21/2009  10:08 PM                0 New Text Document.txt
                1 File(s)                0 bytes
```

You author the following migration XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/TestSuite_BUGFIX">

<component context="System" type="Application">
  <displayName>DATA1</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File">c:\data\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
    <rules>
      <exclude>
        <objectSet>
          <pattern type="File">c:\* [* .txt]</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>

```

However, upon testing the migration you notice that all the text files are still included in the migration. In order to troubleshoot this issue, the migration can be performed with the environment variable MIG_ENABLE_DIAG set so that the diagnostic log is generated. Upon searching the diagnostic log for the component "DATA1", the following XML section is discovered:

```

<MigUnitList>
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)" Context="System" ConfidenceLevel="100" Group="Applications"
Role="UserData" Agent="CMXEAgent" Selected="true" Supported="true">
<Patterns Type="Include">
<Pattern Type="File" Path="C:\data\* [*]"/>
</Patterns>
<Patterns Type="Exclude">
<Pattern Type="File" Path="C:\* [* .txt]"/>
</Patterns>
</MigUnit>
</MigUnitList>
<Perform Name="Gather" User="System">
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)">
<Operation Name="Store" Type="File" Path="C:\data" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test (1).txt]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test.docx]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test.txt]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder [New Text Document.txt]" SimObj="false"
Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder [test.docx]" SimObj="false" Success="true"/>
</MigUnit>
</Perform>

```

Upon reviewing the diagnostic log, you confirm that the files are still migrating, and that it is a problem with the authored migration XML rule. You author an update to the migration XML script as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/TestSuite_BUGFIX">

<component context="System" type="Application">
  <displayName>DATA1</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File">c:\data\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
    <rules>
      <exclude>
        <objectSet>
          <pattern type="File">c:\data\* [*].txt</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>

</migration>

```

Your revised migration XML script excludes the files from migrating, as confirmed in the diagnostic log:

```

<MigUnitList>
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)" Context="System" ConfidenceLevel="100" Group="Applications"
Role="UserData" Agent="CMXEAgent" Selected="true" Supported="true">
<Patterns Type="Include">
<Pattern Type="File" Path="C:\data\* [*]"/>
</Patterns>
<Patterns Type="Exclude">
<Pattern Type="File" Path="C:\data\* [*].txt"/>
</Patterns>
</MigUnit>
</MigUnitList>
<Perform Name="Gather" User="System">
<MigUnit Name="&lt;System&gt;\DATA1 (CMXEAgent)">
<Operation Name="Store" Type="File" Path="C:\data" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data [test.docx]" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder" SimObj="false" Success="true"/>
<Operation Name="Store" Type="File" Path="C:\data\New Folder [test.docx]" SimObj="false" Success="true"/>
</MigUnit>
</Perform>

```

Related topics

[XML Elements Library](#)

[ScanState Syntax](#)

[LoadState Syntax](#)

Return Codes

6/6/2019 • 14 minutes to read • [Edit Online](#)

This topic describes User State Migration Tool (USMT) 10.0 return codes and error messages. Also included is a table listing the USMT return codes with their associated mitigation steps. In addition, this topic provides tips to help you use the logfiles to determine why you received an error.

Understanding the requirements for running USMT can help minimize errors in your USMT migrations. For more information, see [USMT Requirements](#).

In This Topic

[USMT Return Codes](#)

[USMT Error Messages](#)

[Troubleshooting Return Codes and Error Messages](#)

USMT Return Codes

If you encounter an error in your USMT migration, you can use return codes and the more specific information provided in the associated USMT error messages to troubleshoot the issue and to identify mitigation steps.

Return codes are grouped into the following broad categories that describe their area of error reporting:

Success or User Cancel

Invalid Command Lines

Setup and Initialization

Non-fatal Errors

Fatal Errors

As a best practice, we recommend that you set verbosity level to 5, `/v:5`, on the **ScanState**, **LoadState**, and **USMTUtils** command lines so that the most detailed reporting is available in the respective USMT logs. You can use a higher verbosity level if you want the log files output to go to a debugger.

USMT Error Messages

Error messages provide more detailed information about the migration problem than the associated return code. For example, the **ScanState**, **LoadState**, or **USMTUtils** tool might return a code of "11" (for "USMT_INVALID_PARAMETERS") and a related error message that reads `"/key and /keyfile both specified"`. The error message is displayed at the command prompt and is identified in the **ScanState**, **LoadState**, or **USMTUtils** log files to help you determine why the return code was received.

You can obtain more information about any listed Windows application programming interface (API) system error codes by typing `net helpmsg` on the command line and, then typing the error code number. For more information about System Error Codes, see [this Microsoft Web site](#).

Troubleshooting Return Codes and Error Messages

The following table lists each return code by numeric value, along with the associated error messages and

suggested troubleshooting actions.

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
0	USMT_SUCCESS	Successful run	Not applicable	Success or Cancel
1	USMT_DISPLAY_HELP	Command line help requested	Not applicable	Success or Cancel
2	USMT_STATUS_CANCELLED	Gather was aborted because of an EFS file	Not applicable	
		User chose to cancel (such as pressing CTRL+C)	Not applicable	Success or Cancel
3	USMT_WOULD_HAVE_FAILED	At least one error was skipped as a result of /c	Review ScanState, LoadState, or UsmtUtils log for details about command-line errors.	
11	USMT_INVALID_PARAMETERS	/all conflicts with /ui, /ue or /uel	Review ScanState log or LoadState log for details about command-line errors.	
		/auto expects an optional parameter for the script folder	Review ScanState log or LoadState log for details about command-line errors.	
		/encrypt can't be used with /nocompress	Review ScanState log or LoadState log for details about command-line errors.	
		/encrypt requires /key or /keyfile	Review ScanState log or LoadState log for details about command-line errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		/genconfig can't be used with most other options	Review ScanState log or LoadState log for details about command-line errors.	
		/genmigxml can't be used with most other options	Review ScanState log or LoadState log for details about command-line errors.	
		/hardlink requires /nocompress	Review ScanState log or LoadState log for details about command-line errors.	
		/key and /keyfile both specified	Review ScanState log or LoadState log for details about command-line errors.	
		/key or /keyfile used without enabling encryption	Review ScanState log or LoadState log for details about command-line errors.	
		/lae is only used with /lac	Review ScanState log or LoadState log for details about command-line errors.	
		/listfiles cannot be used with /p	Review ScanState log or LoadState log for details about command-line errors.	
		/offline requires a valid path to an XML file describing offline paths	Review ScanState log or LoadState log for details about command-line errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		/offlinewindir requires a valid path to offline windows folder	Review ScanState log or LoadState log for details about command-line errors.	
		/offlinewinold requires a valid path to offline windows folder	Review ScanState log or LoadState log for details about command-line errors.	
		A command was already specified	Verify that the command-line syntax is correct and that there are no duplicate commands.	
		An option argument is missing	Review ScanState log or LoadState log for details about command-line errors.	
		An option is specified more than once and is ambiguous	Review ScanState log or LoadState log for details about command-line errors.	
		By default /auto selects all users and uses the highest log verbosity level. Switches like /all, /ui, /ue, /v are not allowed.	Review ScanState log or LoadState log for details about command-line errors.	
		Command line arguments are required. Specify /? for options.	Review ScanState log or LoadState log for details about command-line errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		Command line option is not valid	Review ScanState log or LoadState log for details about command-line errors.	
		EFS parameter specified is not valid for /efs	Review ScanState log or LoadState log for details about command-line errors.	
		File argument is invalid for /genconfig	Review ScanState log or LoadState log for details about command-line errors.	
		File argument is invalid for /genmigxml	Review ScanState log or LoadState log for details about command-line errors.	
		Invalid space estimate path. Check the parameters and/or file system permissions	Review ScanState log or LoadState log for details about command-line errors.	
		List file path argument is invalid for /listfiles	Review ScanState log or LoadState log for details about command-line errors.	
		Retry argument must be an integer	Review ScanState log or LoadState log for details about command-line errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		Settings store argument specified is invalid	Review ScanState log or LoadState log for details about command-line errors. Make sure that the store path is accessible and that the proper permission levels are set.	
		Specified encryption algorithm is not supported	Review ScanState log or LoadState log for details about command-line errors.	
		The /efs:hardlink requires /hardlink	Review ScanState log or LoadState log for details about command-line errors.	
		The /targetWindows7 option is only available for Windows XP, Windows Vista, and Windows 7	Review ScanState log or LoadState log for details about command-line errors.	
		The store parameter is required but not specified	Review ScanState log or LoadState log for details about command-line errors.	
		The source-to-target domain mapping is invalid for /md	Review ScanState log or LoadState log for details about command-line errors.	
		The source-to-target user account mapping is invalid for /mu	Review ScanState log or LoadState log for details about command-line errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		Undefined or incomplete command line option	Review ScanState log or LoadState log for details about command-line errors.	Invalid Command Lines
		Use /nocompress, or provide an XML file path with /p"pathtofile" to get a compressed store size estimate	Review ScanState log or LoadState log for details about command-line errors.	
		User exclusion argument is invalid	Review ScanState log or LoadState log for details about command-line errors.	
		Verbosity level must be specified as a sum of the desired log options: Verbose (0x01), Record Objects (0x04), Echo to debug port (0x08)	Review ScanState log or LoadState log for details about command-line errors.	
		Volume shadow copy feature is not supported with a hardlink store	Review ScanState log or LoadState log for details about command-line errors.	
		Wait delay argument must be an integer	Review ScanState log or LoadState log for details about command-line errors.	
12	USMT_ERROR_OPTION_PARAM_TOO_LARGE	Command line arguments cannot exceed 256 characters	Review ScanState log or LoadState log for details about command-line errors.	Invalid Command Lines

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		Specified settings store path exceeds the maximum allowed length of 256 characters	Review ScanState log or LoadState log for details about command-line errors.	
13	USMT_INIT_LOGFILE_FAILED	Log path argument is invalid for /l	When /l is specified in the ScanState command line, USMT validates the path. Verify that the drive and other information, for example file system characters, are correct.	Invalid Command Lines
14	USMT_ERROR_US E_LAC	Unable to create a local account because /lac was not specified	When creating local accounts, the command-line options /lac and /lae should be used.	Invalid Command Lines
26	USMT_INIT_ERROR	Multiple Windows installations found	Listfiles.txt could not be created. Verify that the location you specified for the creation of this file is valid.	Setup and Initialization
		Software malfunction or unknown exception	Check all loaded .xml files for errors, common error when using /l to load the Config.xml file.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		Unable to find a valid Windows directory to proceed with requested offline operation; Check if offline input file is present and has valid entries	Verify that the offline input file is present and that it has valid entries. USMT could not find valid offline operating system. Verify your offline directory mapping.	
27	USMT_INVALID_STORE_LOCATION	A store path can't be used because an existing store exists; specify /o to overwrite	Specify /o to overwrite an existing intermediate or migration store.	Setup and Initialization
		A store path is missing or has incomplete data	Make sure that the store path is accessible and that the proper permission levels are set.	
		An error occurred during store creation	Make sure that the store path is accessible and that the proper permission levels are set. Specify /o to overwrite an existing intermediate or migration store.	
		An inappropriate device such as a floppy disk was specified for the store	Make sure that the store path is accessible and that the proper permission levels are set.	
		Invalid store path; check the store parameter and/or file system permissions	Invalid store path; check the store parameter and/or file system permissions	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		The file layout and/or file content is not recognized as a valid store	Make sure that the store path is accessible and that the proper permission levels are set. Specify /o to overwrite an existing intermediate or migration store.	
		The store path holds a store incompatible with the current USMT version	Make sure that the store path is accessible and that the proper permission levels are set.	
		The store save location is read-only or does not support a requested storage option	Make sure that the store path is accessible and that the proper permission levels are set.	
28	USMT_UNABLE_GET_SCRIPTFILES	Script file is invalid for /i	Check all specified migration .xml files for errors. This is a common error when using /i to load the Config.xml file.	Setup and Initialization
		Unable to find a script file specified by /i	Verify the location of your script files, and ensure that the command-line options are correct.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
29	USMT_FAILED_MIGSTARTUP	A minimum of 250 MB of free space is required for temporary files	Verify that the system meets the minimum temporary disk space requirement of 250 MB. As a workaround, you can set the environment variable USMT_WORKING_DIR= <path> to redirect the temporary files working directory.	Setup and Initialization
		Another process is preventing migration; only one migration tool can run at a time	Check the ScanState log file for migration .xml file errors.	
		Failed to start main processing, look in log for system errors or check the installation	Check the ScanState log file for migration .xml file errors.	
		Migration failed because of an XML error; look in the log for specific details	Check the ScanState log file for migration .xml file errors.	
		Unable to automatically map the drive letters to match the online drive letter layout; Use /offline to provide a mapping table	Check the ScanState log file for migration .xml file errors.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
31	USMT_UNABLE_F INDMIGUNITS	An error occurred during the discover phase; the log should have more specific information	Check the ScanState log file for migration .xml file errors.	Setup and Initialization
32	USMT_FAILED_SE TMIGRATIONTYPE	An error occurred processing the migration system	Check the ScanState log file for migration .xml file errors, or use online Help by typing /? on the command line.	Setup and Initialization
33	USMT_UNABLE_R EADKEY	Error accessing the file specified by the /keyfile parameter	Check the ScanState log file for migration .xml file errors, or use online Help by typing /? on the command line.	Setup and Initialization
		The encryption key must have at least one character	Check the ScanState log file for migration .xml file errors, or use online Help by typing /? on the command line.	
34	USMT_ERROR_IN SUFFICIENT_RIG HTS	Directory removal requires elevated privileges	Log on as Administrator, and run with elevated privileges.	Setup and Initialization
		No rights to create user profiles; log in as Administrator; run with elevated privileges	Log on as Administrator, and run with elevated privileges.	
		No rights to read or delete user profiles; log in as Administrator, run with elevated privileges	Log on as Administrator, and run with elevated privileges.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
35	USMT_UNABLE_DELETE_STORE	A reboot is required to remove the store	Reboot to delete any files that could not be deleted when the command was executed.	Setup and Initialization
		A store path can't be used because it contains data that could not be overwritten	A migration store could not be deleted. If you are using a hardlink migration store you might have a locked file in it. You should manually delete the store, or use USMTUtils /rd command to delete the store.	
		There was an error removing the store	Review ScanState log or LoadState log for details about command-line errors.	
36	USMT_ERROR_UN_SUPPORTED_PLATFORM	Compliance check failure; please check the logs for details	Investigate whether there is an active temporary profile on the system.	Setup and Initialization
		Use of /offline is not supported during apply	The /offline command was not used while running in the Windows Preinstallation Environment (WinPE).	
		Use /offline to run gather on this platform	The /offline command was not used while running in WinPE.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
37	USMT_ERROR_NO_INVALID_KEY	The store holds encrypted data but the correct encryption key was not provided	Verify that you have included the correct encryption /key or /keyfile.	Setup and Initialization
38	USMT_ERROR_CORRUPTED_NOT_ENCRYPTED_STORE	An error occurred during store access	Review ScanState log or LoadState log for details about command-line errors. Make sure that the store path is accessible and that the proper permission levels are set.	Setup and Initialization
39	USMT_UNABLE_TO_READ_CONFIG_FILE	Error reading Config.xml	Review ScanState log or LoadState log for details about command-line errors in the Config.xml file.	Setup and Initialization
		File argument is invalid for /config	Check the command line you used to load the Config.xml file. You can use online Help by typing /? on the command line.	
40	USMT_ERROR_UNABLE_CREATE_PROGRESS_LOG	Error writing to the progress log	The Progress log could not be created. Verify that the location is valid and that you have write access.	Setup and Initialization
		Progress log argument is invalid for /progress	The Progress log could not be created. Verify that the location is valid and that you have write access.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
41	USMT_PREFLIGHT_FILE_CREATION_FAILED	Can't overwrite existing file	The Progress log could not be created. Verify that the location is valid and that you have write access.	Setup and Initialization
		Invalid space estimate path. Check the parameters and/or file system permissions	Review ScanState log or LoadState log for details about command-line errors.	
42	USMT_ERROR_CORRUPTED_STORE	The store contains one or more corrupted files	Review UsmtUtils log for details about the corrupted files. For information on how to extract the files that are not corrupted, see Extract Files from a Compressed USMT Migration Store .	
61	USMT_MIGRATION_STOPPED_NON_FATAL	Processing stopped due to an I/O error	USMT exited but can continue with the /c command-line option, with the optional configurable <ErrorControl> section or by using the /vsc command-line option.	Non-fatal Errors
71	USMT_INIT_OPERATING_ENVIRONMENT_FAILED	A Windows Win32 API error occurred	Data transfer has begun, and there was an error during the creation of migration store or during the apply phase. Review the ScanState log or LoadState log for details.	Fatal Errors

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		An error occurred when attempting to initialize the diagnostic mechanisms such as the log	Data transfer has begun, and there was an error during the creation of migration store or during the apply phase. Review the ScanState log or LoadState log for details.	
		Failed to record diagnostic information	Data transfer has begun, and there was an error during the creation of migration store or during the apply phase. Review the ScanState log or LoadState log for details.	
		Unable to start. Make sure you are running USMT with elevated privileges	Exit USMT and log in again with elevated privileges.	
72	USMT_UNABLE_DOMIGRATION	An error occurred closing the store	Data transfer has begun, and there was an error during migration-store creation or during the apply phase. Review the ScanState log or LoadState log for details.	Fatal Errors
		An error occurred in the apply process	Data transfer has begun, and there was an error during migration-store creation or during the apply phase. Review the ScanState log or LoadState log for details.	

RETURN CODE VALUE	RETURN CODE	ERROR MESSAGE	TROUBLESHOOTING, MITIGATION, WORKAROUNDS	CATEGORY
		An error occurred in the gather process	Data transfer has begun, and there was an error during migration-store creation or during the apply phase. Review the ScanState log or LoadState log for details.	
		Out of disk space while writing the store	Data transfer has begun, and there was an error during migration-store creation or during the apply phase. Review the ScanState log or LoadState log for details.	
		Out of temporary disk space on the local system	Data transfer has begun, and there was an error during migration-store creation or during the apply phase. Review the ScanState log or LoadState log for details.	

Related topics

[User State Migration Tool \(USMT\) Troubleshooting](#)

[Log Files](#)

USMT Resources

5/31/2019 • 2 minutes to read • [Edit Online](#)

USMT Online Resources

- [ADK Release Notes](#)
- Microsoft Visual Studio
 - You can use the User State Migration Tool (USMT) XML schema (the MigXML.xsd file) to validate the migration .xml files using an XML authoring tool such as Microsoft® Visual Studio®.

For more information about how to use the schema with your XML authoring environment, see the environment's documentation.
- [Ask the Directory Services Team blog](#)
- Forums:
 - [Microsoft Deployment Toolkit](#)
 - [Configuration Manager Operating System Deployment](#)

Related topics

[User State Migration Tool \(USMT\) Overview Topics](#)

User State Migration Toolkit (USMT) Reference

6/6/2019 • 2 minutes to read • [Edit Online](#)

In This Section

USMT Requirements	Describes operating system, hardware, and software requirements, and user prerequisites.
USMT Best Practices	Discusses general and security-related best practices when using USMT.
How USMT Works	Learn about the processes behind the ScanState and LoadState tools.
Plan Your Migration	Choose what to migrate and the best migration scenario for your enterprise.
User State Migration Tool (USMT) Command-line Syntax	Explore command-line options for the ScanState, LoadState, and UsmtUtils tools.
USMT XML Reference	Learn about customizing a migration with XML files.
Offline Migration Reference	Find requirements, best practices, and other considerations for performing a migration offline.

Related topics

[User State Migration Tool \(USMT\) Overview Topics](#)

[User State Migration Tool \(USMT\) How-to topics](#)

[User State Migration Tool \(USMT\) Troubleshooting](#)

USMT Requirements

6/6/2019 • 3 minutes to read • [Edit Online](#)

In This Topic

- [Supported Operating Systems](#)
- [Windows PE](#)
- [Credentials](#)
- [Config.xml](#)
- [LoadState](#)
- [Hard Disk Requirements](#)
- [User Prerequisites](#)

Supported Operating Systems

The User State Migration Tool (USMT) 10.0 does not have any explicit RAM or CPU speed requirements for either the source or destination computers. If your computer complies with the system requirements of the operating system, it also complies with the requirements for USMT. You need an intermediate store location large enough to hold all of the migrated data and settings, and the same amount of hard disk space on the destination computer for the migrated files and settings.

The following table lists the operating systems supported in USMT.

OPERATING SYSTEMS	SCANSTATE (SOURCE COMPUTER)	LOADSTATE (DESTINATION COMPUTER)
32-bit versions of Windows 7	X	X
64-bit versions of Windows 7	X	X
32-bit versions of Windows 8	X	X
64-bit versions of Windows 8	X	X
32-bit versions of Windows 10	X	X
64-bit versions of Windows 10	X	X

Note You can migrate a 32-bit operating system to a 64-bit operating system. However, you cannot migrate a 64-bit operating system to a 32-bit operating system.

USMT does not support any of the Windows Server® operating systems, Windows 2000, Windows XP, or any of the starter editions for Windows Vista or Windows 7.

USMT for Windows 10 should not be used for migrating from Windows 7 to Windows 8.1. It is meant to migrate to Windows 10. For more information about previous releases of the USMT tools, see [User State Migration Tool \(USMT\) 4.0 User's Guide](#).

Windows PE

- **Must use latest version of Windows PE.** For example, to migrate to Windows 10, you'll need Windows PE 5.1. For more info, see [What's New in Windows PE](#).

Credentials

- **Run as administrator** When manually running the **ScanState** and **LoadState** tools on Windows 7, Windows 8 or Windows 10 you must run them from an elevated command prompt to ensure that all specified users are migrated. If you do not run USMT from an elevated prompt, only the user profile that is logged on will be included in the migration.

To open an elevated command prompt:

1. Click **Start**.
2. Enter **cmd** in the search function.
3. Depending on the OS you are using, **cmd** or **Command Prompt** is displayed.
4. Right-click **cmd** or **Command Prompt**, and then click **Run as administrator**.
5. If the current user is not already an administrator, you will be prompted to enter administrator credentials.

Important

You must run USMT using an account with full administrative permissions, including the following privileges:

- SeBackupPrivilege (Back up files and directories)
- SeDebugPrivilege (Debug programs)
- SeRestorePrivilege (Restore files and directories)
- SeSecurityPrivilege (Manage auditing and security log)
- SeTakeOwnership Privilege (Take ownership of files or other objects)

Config.xml

- **Specify the /c option and <ErrorControl> settings in the Config.xml file.**

USMT will fail if it cannot migrate a file or setting, unless you specify the **/c** option. When you specify the **/c** option, USMT logs an error each time it encounters a file that is in use that did not migrate, but the migration will not be interrupted. In USMT, you can specify in the Config.xml file which types of errors should allow the migration to continue, and which should cause the migration to fail. For more information about error reporting, and the **<ErrorControl>** element, see [Config.xml File](#), [Log Files](#), and [XML Elements Library](#).

LoadState

- **Install applications before running the LoadState command.**

Install all applications on the destination computer before restoring the user state. This ensures that migrated settings are preserved.

Hard-Disk Requirements

Ensure that there is enough available space in the migration-store location and on the source and destination computers. For more information, see [Estimate Migration Store Size](#).

User Prerequisites

This documentation assumes that IT professionals using USMT understand command-line tools. The documentation also assumes that IT professionals using USMT to author MigXML rules understand the following:

- The navigation and hierarchy of the Windows registry.
- The files and file types that applications use.
- The methods to extract application and setting information manually from applications created by internal software-development groups and non-Microsoft software vendors.
- XML-authoring basics.

Related topics

[Plan Your Migration](#)

[Estimate Migration Store Size](#)

[User State Migration Tool \(USMT\) Overview Topics](#)

USMT Best Practices

6/14/2019 • 6 minutes to read • [Edit Online](#)

This topic discusses general and security-related best practices when using User State Migration Tool (USMT) 10.0.

General Best Practices

- **Install applications before running the LoadState tool**

Though it is not always essential, it is best practice to install all applications on the destination computer before restoring the user state. This helps ensure that migrated settings are preserved.

- **Do not use MigUser.xml and MigDocs.xml together**

If you use both .xml files, some migrated files may be duplicated if conflicting instructions are given about target locations. You can use the **/genmigxml** command-line option to determine which files will be included in your migration, and to determine if any modifications are necessary. For more information, see [Identify File Types, Files, and Folders](#).

- **Use MigDocs.xml for a better migration experience**

If your data set is unknown or if many files are stored outside of the standard user-profile folders, the MigDocs.xml file is a better choice than the MigUser.xml file, because the MigDocs.xml file will gather a broader scope of data. The MigDocs.xml file migrates folders of data based on location, and on registered file type by querying the registry for registered application extensions. The MigUser.xml file migrates only the files with the specified file extensions.

- **Close all applications before running either the ScanState or LoadState tools**

Although using the **/vsc** switch can allow the migration of many files that are open with another application it is a best practice to close all applications in order to ensure all files and settings migrate. Without the **/vsc** or **/c** switch USMT will fail when it cannot migrate a file or setting. When you use the **/c** option USMT will ignore any files or settings that it cannot migrate and log an error each time.

- **Log off after you run the LoadState**

Some settings, such as fonts, wallpaper, and screensaver settings, will not take effect until the next time the user logs on. For this reason, you should log off after you run the LoadState tool.

- **Managed environment**

To create a managed environment, you can move all of the end user's documents into My Documents (%CSIDL_PERSONAL%). We recommend that you migrate files into the smallest-possible number of folders on the destination computer. This will help you to clean up files on the destination computer, if the LoadState command fails before completion.

- **Chkdsk.exe**

We recommend that you run Chkdsk.exe before running the ScanState and LoadState tools. Chkdsk.exe creates a status report for a hard disk drive and lists and corrects common errors. For more information about the Chkdsk.exe tool, see [Chkdsk](#).

- **Migrate in groups**

If you decide to perform the migration while users are using the network, it is best to migrate user accounts in groups. To minimize the impact on network performance, determine the size of the groups based on the size of each user account. Migrating in phases also allows you to make sure each phase is successful before starting the next phase. Using this method, you can make any necessary modifications to your plan between groups.

Security Best Practices

As the authorized administrator, it is your responsibility to protect the privacy of the users and maintain security during and after the migration. In particular, you must consider the following issues:

- **Encrypting File System (EFS)**

Take extreme caution when migrating encrypted files, because the end user does not need to be logged on to capture the user state. By default, USMT fails if an encrypted file is found. For more information about EFS best practices, see this article in the [Microsoft Knowledge Base](#). For specific instructions about EFS best practices, see [Migrate EFS Files and Certificates](#).

Important If you migrate an encrypted file without also migrating the certificate, end users will not be able to access the file after the migration.

- **Encrypt the store**

Consider using the **/encrypt** option with the ScanState command and the **/decrypt** option with the LoadState command. However, use extreme caution with this set of options, because anyone who has access to the ScanState command-line script also has access to the encryption key.

- **Virus Scan**

We recommend that you scan both the source and destination computers for viruses before running USMT. In addition, you should scan the destination computer image. To help protect data from viruses, we strongly recommend running an antivirus utility before migration.

- **Maintain security of the file server and the deployment server**

We recommend that you manage the security of the file and deployment servers. It is important to make sure that the file server where you save the store is secure. You must also secure the deployment server, to ensure that the user data that is in the log files is not exposed. We also recommend that you only transmit data over a secure Internet connection, such as a virtual private network. For more information about network security, see [Microsoft Security Compliance Manager](#).

- **Password Migration**

To ensure the privacy of the end users, USMT does not migrate passwords, including those for applications such as Windows Live™ Mail, Microsoft Internet Explorer®, as well as Remote Access Service (RAS) connections and mapped network drives. It is important to make sure that end users know their passwords.

- **Local Account Creation**

Before you migrate local accounts, see the Migrating Local Accounts section in the [Identify Users](#) topic.

XML File Best Practices

- **Specify the same set of mig*.xml files in both the ScanState and the LoadState tools**

If you used a particular set of mig*.xml files in the ScanState tool, either called through the "/auto" option, or individually through the "/i" option, then you should use same option to call the exact same mig*.xml files in the LoadState tool.

- **The <CustomFileName> in the migration urlid should match the name of the file**

Although it is not a requirement, it is good practice for <CustomFileName> to match the name of the file. For example, the following is from the MigApp.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/migapp">
```

- **Use the XML Schema (MigXML.xsd) when authoring .xml files to validate syntax**

The MigXML.xsd schema file should not be included on the command line or in any of the .xml files.

- **Use the default migration XML files as models**

To create a custom .xml file, you can use the migration .xml files as models to create your own. If you need to migrate user data files, model your custom .xml file on MigUser.xml. To migrate application settings, model your custom .xml file on the MigApp.xml file.

- **Consider the impact on performance when using the <context> parameter**

Your migration performance can be affected when you use the <context> element with the <component> element; for example, as in when you want to encapsulate logical units of file- or path-based <include> and <exclude> rules.

In the **User** context, a rule is processed one time for each user on the system.

In the **System** context, a rule is processed one time for the system.

In the **UserAndSystem** context, a rule is processed one time for each user on the system and one time for the system.

Note The number of times a rule is processed does not affect the number of times a file is migrated. The USMT migration engine ensures that each file migrates only once.

- **We recommend that you create a separate .xml file instead of adding your .xml code to one of the existing migration .xml files**

For example, if you have code that migrates the settings for an application, you should not just add the code to the MigApp.xml file.

- **You should not create custom .xml files to alter the operating system settings that are migrated**

These settings are migrated by manifests and you cannot modify those files. If you want to exclude certain operating system settings from the migration, you should create and modify a Config.xml file.

- **You can use the asterisk (*) wildcard character in any migration XML file that you create**

Note The question mark is not valid as a wildcard character in USMT .xml files.

Related topics

[Migration Store Encryption](#)

[Plan Your Migration](#)

How USMT Works

6/6/2019 • 8 minutes to read • [Edit Online](#)

USMT includes two tools that migrate settings and data: ScanState and LoadState. ScanState collects information from the source computer, and LoadState applies that information to the destination computer.

- [ScanState Process](#)
- [LoadState Process](#)

Note For more information about how USMT processes the rules and the XML files, see [Conflicts and Precedence](#).

The ScanState Process

When you run the ScanState tool on the source computer, it goes through the following process:

1. It parses and validates the command-line parameters, creates the ScanState.log file, and then begins logging.
2. It collects information about all of the migration components that need to be migrated. A *migration component* is a logical group of files, registry keys, and values. For example, the set of files, registry keys, and values that store the settings of Adobe Acrobat is grouped into a single migration component.

There are three types of components:

- Components that migrate the operating system settings
- Components that migrate application settings
- Components that migrate users' files

The ScanState tool collects information about the application settings and user data components from the .xml files that are specified on the command line.

In Windows 7, and Windows 8, the manifest files control how the operating-system settings are migrated. You cannot modify these files. If you want to exclude certain operating-system settings, you must create and modify a Config.xml file.

3. ScanState determines which user profiles should be migrated. By default, all user profiles on the source computer are migrated. However, you can include and exclude users using the User Options. The public profile in a source computer running Windows 7, Windows 8, and Windows 10 is always migrated, and you cannot exclude these profiles from the migration.
4. In the "Scanning" phase, ScanState does the following for each user profile selected for migration:
 - a. For each component, ScanState checks the type of the component. If the current user profile is the system profile and the component type is "System" or "UserAndSystem", the component is selected for this user. Otherwise, the component is ignored. Alternatively, if the current user profile is not the system profile and the component type is "User" or "UserAndSystem", the component is selected for this user. Otherwise, this component is ignored.

Note From this point on, ScanState does not distinguish between components that migrate operating-system settings, those that migrate application settings, and those that migrate users' files. ScanState processes all components in the same way.

- b. Each component that is selected in the previous step is processed further. Any profile-specific variables (such as CSIDL_PERSONAL) are evaluated in the context of the current profile. For example, if the profile that is being processed belongs to "User1", then CSIDL_PERSONAL would expand to C:\Users\User1\Documents, assuming that the user profiles are stored in the C:\Users directory.
- c. For each selected component, ScanState evaluates the <detects> section. If the condition in the <detects> section evaluates to false, the component is not processed any further. Otherwise, the processing of this component continues.
- d. For each selected component, ScanState evaluates the <rules> sections. For each <rules> section, if the current user profile is the system profile and the context of the <rules> section is "System" or "UserAndSystem", the rule is processed further. Otherwise, this rule is ignored. Alternatively, if the current user profile is not the system profile and the context of the <rules> section is "User" or "UserAndSystem", the rule is processed further. Otherwise, this rule is ignored.
- e. ScanState creates a list of migration units that need to be migrated by processing the various subsections under this <rules> section. Each unit is collected if it is mentioned in an <include> subsection, as long as there is not a more specific rule for it in an <exclude> subsection in the same <rules> section. For more information about precedence in the .xml files, see [Conflicts and Precedence](#).

In addition, any migration unit (such as a file, registry key, or set of registry values) that is in an <UnconditionalExclude> section is not migrated.

Note ScanState ignores some subsections such as <destinationCleanup> and <locationModify>. These sections are evaluated only on the destination computer.

5. In the "Collecting" phase, ScanState creates a master list of the migration units by combining the lists that were created for each selected user profile.
6. In the "Saving" phase, ScanState writes the migration units that were collected to the store location.

Note ScanState does not modify the source computer in any way.

The LoadState Process

The LoadState process is very similar to the ScanState process. The ScanState tool collects migration units such as file, registry key, or registry values from the source computer and saves them to the store. Similarly, the LoadState tool collects migration units from the store and applies them to the destination computer.

1. ScanState parses and validates the command-line parameters, creates the ScanState.log file, and then begins logging.
2. LoadState collects information about the migration components that need to be migrated.

LoadState obtains information for the application-settings components and user-data components from the migration .xml files that are specified by the LoadState command.

In Windows 7, and Windows 8, the manifest files control how the operating-system settings are migrated. You cannot modify these files. If you want to exclude certain operating-system settings, you must create and modify a Config.xml file.

3. LoadState determines which user profiles should be migrated. By default, all user profiles present on the source computer are migrated. However, you can include and exclude users using the User Options. The system profile, the "All users" profile in a source computer running Windows XP, or the Public profile in a source computer running Windows Vista, Windows 7, and Windows 8, is always migrated and you cannot exclude these profiles from the migration.

- If you are migrating local user accounts and if the accounts do not already exist on the destination computer, you must use the **/lac** command-line option. If you do not specify the **/lac** option, any local user accounts that are not already present on the destination computer, are not migrated.
- The **/md** and **/mu** options are processed to rename the user profile on the destination computer, if they have been included when the LoadState command was specified.
- For each user profile selected from the store, LoadState creates a corresponding user profile on the destination computer. The destination computer does not need to be connected to the domain for domain user profiles to be created. If USMT cannot determine a domain, it attempts to apply the settings to a local account. For more information, see [Identify Users](#).

4. In the "Scanning" phase, LoadState does the following for each user profile:

- a. For each component, LoadState checks the type of the component. If the current user profile is the system profile and the component type is "System" or "UserAndSystem", the component is selected for this user. Otherwise, the component is ignored. Alternatively, if the current user profile is not the system profile and the component type is "User" or "UserAndSystem", the component is selected for this user. Otherwise, this component is ignored.

Note

From this point on, LoadState does not distinguish between components that migrate operating-system settings, those that migrate application settings, and those that migrate users' files. LoadState evaluates all components in the same way.

- b. Each component that is selected is processed further. Any profile-specific variables (such as CSIDL_PERSONAL) are evaluated in the context of the current profile. For example, if the profile being processed belongs to "User1", then CSIDL_PERSONAL would expand to C:\Users\User1\Documents (assuming that the user profiles are stored in the C:\Users directory).

Note

LoadState ignores the <detects> section specified in a component. At this point, all specified components are considered to be detected and are selected for migration.

- c. For each selected component, LoadState evaluates the <rules> sections. For each <rules> section, if the current user profile is the system profile and the context of the <rules> section is "System" or "UserAndSystem", the rule is processed further. Otherwise, this rule is ignored. Alternatively, if the current user profile is not the system profile and the context of the <rules> section is "User" or "UserAndSystem", the rule is processed further. Otherwise, this rule is ignored.
- d. LoadState creates a master list of migration units by processing the various subsections under the <rules> section. Each migration unit that is in an <include> subsection is migrated as long, as there is not a more specific rule for it in an <exclude> subsection in the same <rules> section. For more information about precedence, see [Conflicts and Precedence](#).
- e. LoadState evaluates the destination computer-specific subsections; for example, the <destinationCleanup> and <locationModify> subsections.
- f. If the destination computer is running Windows 7 or Windows 8 then the migunits that were collected by ScanState using downlevel manifest files are processed by LoadState using the corresponding Component Manifest for Windows 7. The downlevel manifest files are not used during LoadState.

Important

It is important to specify the .xml files with the LoadState command if you want LoadState to use them. Otherwise, any destination-specific rules, such as <locationModify>, in these .xml files are ignored, even if the same .xml files were provided when the ScanState command ran.

5. In the "Apply" phase, LoadState writes the migration units that were collected to the various locations on the destination computer. If there are conflicts and there is not a <merge> rule for the object, the default behavior for the registry is for the source to overwrite the destination. The default behavior for files is for the source to be renamed incrementally, for example, OriginalFileName(1).OriginalExtension. Some settings, such as fonts, wallpaper, and screen-saver settings, do not take effect until the next time the user logs on. For this reason, you should log off when the LoadState command actions have completed.

Related topics

[User State Migration Tool \(USMT\) Command-line Syntax](#)

Plan Your Migration

6/6/2019 • 2 minutes to read • [Edit Online](#)

Before you use the User State Migration Tool (USMT) 10.0 to perform your migration, we recommend that you plan your migration carefully. Planning can help your migration proceed smoothly and can reduce the risk of migration failure.

In migration planning, both organizations and individuals must first identify what to migrate, including user settings, applications and application settings, and personal data files and folders. Identifying the applications to migrate is especially important so that you can avoid capturing data about applications that may be phased out.

One of the most important requirements for migrating settings and data is restoring only the information that the destination computer requires. Although the data that you capture on the source computer may be more comprehensive than the restoration data for backup purposes, restoring data or settings for applications that you will not install on the destination system is redundant. This can also introduce instability in a newly deployed computer.

In This Section

Common Migration Scenarios	Determine whether you will perform a refresh migration or a replace migration.
What Does USMT Migrate?	Learn which applications, user data, and operating system components USMT migrates.
Choose a Migration Store Type	Choose an uncompressed, compressed, or hard-link migration store.
Determine What to Migrate	Identify user accounts, application settings, operating system settings, and files that you want to migrate inside your organization.
Test Your Migration	Test your migration before you deploy Windows to all users.

Related topics

[USMT XML Reference](#)

Common Migration Scenarios

6/14/2019 • 6 minutes to read • [Edit Online](#)

You use the User State Migration Tool (USMT) 10.0 when hardware and/or operating system upgrades are planned for a large number of computers. USMT manages the migration of an end-user's digital identity by capturing the user's operating-system settings, application settings, and personal files from a source computer and reinstalling them on a destination computer after the upgrade has occurred.

One common scenario when only the operating system, and not the hardware, is being upgraded is referred to as *PC refresh*. A second common scenario is known as *PC replacement*, where one piece of hardware is being replaced, typically by newer hardware and a newer operating system.

In This Topic

PC Refresh

[Scenario One: PC-refresh offline using Windows PE and a hard-link migration store](#)

[Scenario Two: PC-refresh using a compressed migration store](#)

[Scenario Three: PC-refresh using a hard-link migration store](#)

[Scenario Four: PC-refresh using Windows.old folder and a hard-link migration store](#)

PC Replacement

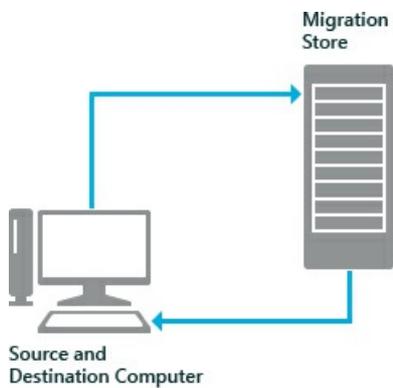
[Scenario One: Offline migration using Windows PE and an external migration store](#)

[Scenario Two: Manual network migration](#)

[Scenario Three: Managed network migration](#)

PC-Refresh

The following diagram shows a PC-refresh migration, also known as a computer refresh migration. First, the administrator migrates the user state from a source computer to an intermediate store. After installing the operating system, the administrator migrates the user state back to the source computer.



1. ScanState collects user state and saves data to Migration Store
2. New operating system is installed
3. LoadState restores user state

Scenario One: PC-refresh offline using Windows PE and a hard-link migration store

A company has just received funds to update the operating system on all of its computers in the accounting department to Windows 10. Each employee will keep the same computer, but the operating system on each computer will be updated. In this scenario, the update is being handled completely offline, without a network connection. An administrator uses Windows Preinstallation Environment (WinPE) and a hard-link migration store to save each user state to their respective computer.

1. On each computer, the administrator boots the machine into WinPE and runs the ScanState command-line tool, specifying the **/hardlink /nocompress** command-line options. ScanState saves the user state to a hard-link migration store on each computer, improving performance by minimizing network traffic as well as minimizing migration failures on computers with very limited space available on the hard drive.
2. On each computer, the administrator installs the company's standard operating environment (SOE) which includes Windows 10 and other company applications.
3. The administrator runs the LoadState command-line tool on each computer. LoadState restores each user state back to each computer.

Scenario Two: PC-refresh using a compressed migration store

A company has just received funds to update the operating system on all of its computers to Windows 10. Each employee will keep the same computer, but the operating system on each computer will be updated. In this scenario, an administrator uses a compressed migration store to save the user states to a server.

1. The administrator runs the ScanState command-line tool on each computer. ScanState saves each user state to a server.
2. On each computer, the administrator installs the company's standard SOE which includes Windows 10 and other company applications.
3. The administrator runs the LoadState command-line tool on each source computer, and LoadState restores each user state back to the computer.

Scenario Three: PC-refresh using a hard-link migration store

A company has just received funds to update the operating system on all of its computers to Windows 10. Each employee will keep the same computer, but the operating system on each computer will be updated. In this scenario, an administrator uses a hard-link migration store to save each user state to their respective computer.

1. The administrator runs the ScanState command-line tool on each computer, specifying the **/hardlink /nocompress** command-line options. ScanState saves the user state to a hard-link migration store on each

computer, improving performance by minimizing network traffic as well as minimizing migration failures on computers with very limited space available on the hard drive.

2. On each computer, the administrator installs the company's SOE which includes Windows 10 and other company applications.
3. The administrator runs the LoadState command-line tool on each computer. LoadState restores each user state back on each computer.

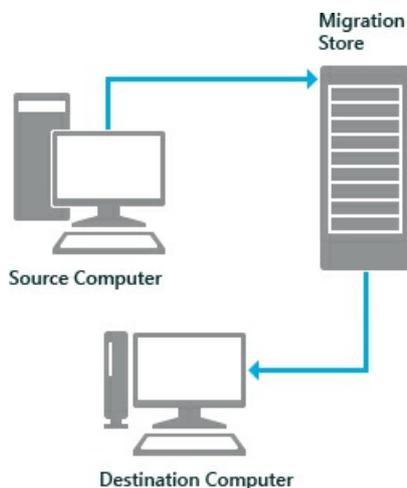
Scenario Four: PC-refresh using Windows.old folder and a hard-link migration store

A company has decided to update the operating system on all of its computers to Windows 10. Each employee will keep the same computer, but the operating system on each computer will be updated. In this scenario, an administrator uses Windows.old and a hard-link migration store to save each user state to their respective computer.

1. The administrator clean installs Windows 10 on each computer, making sure that the Windows.old directory is created by installing Windows 10 without formatting or repartitioning and by selecting a partition that contains the previous version of Windows.
2. On each computer, the administrator installs the company's SOE which includes company applications.
3. The administrator runs the ScanState and LoadState command-line tools successively on each computer while specifying the **/hardlink /nocompress** command-line options.

PC-Replacement

The following diagram shows a PC-replacement migration. First, the administrator migrates the user state from the source computer to an intermediate store. After installing the operating system on the destination computer, the administrator migrates the user state from the store to the destination computer.



1. ScanState collects user state from Source computer and saves data to Migration Store
2. New operating system is installed on Destination Computer
3. LoadState restores user state on Destination Computer

Scenario One: Offline migration using WinPE and an external migration store

A company is allocating 20 new computers to users in the accounting department. The users each have a source computer with their files and settings. In this scenario, migration is being handled completely offline, without a

network connection.

1. On each source computer, an administrator boots the machine into WinPE and runs ScanState to collect the user state to either a server or an external hard disk.
2. On each new computer, the administrator installs the company's SOE which includes Windows 10 and other company applications.
3. On each of the new computers, the administrator runs the LoadState tool, restoring each user state from the migration store to one of the new computers.

Scenario Two: Manual network migration

A company receives 50 new laptops for their managers and needs to reallocate 50 older laptops to new employees. In this scenario, an administrator runs the ScanState tool from the cmd prompt on each computer to collect the user states and save them to a server in a compressed migration store.

1. The administrator runs the ScanState tool on each of the manager's old laptops, and saves each user state to a server.
2. On the new laptops, the administrator installs the company's SOE, which includes Windows 10 and other company applications.
3. The administrator runs the LoadState tool on the new laptops to migrate the managers' user states to the appropriate computer. The new laptops are now ready for the managers to use.
4. On the old computers, the administrator installs the company's SOE, which includes Windows 10, Microsoft Office, and other company applications. The old computers are now ready for the new employees to use.

Scenario Three: Managed network migration

A company is allocating 20 new computers to users in the accounting department. The users each have a source computer that contains their files and settings. An administrator uses a management technology such as a logon script or a batch file to run ScanState on each source computer to collect the user states and save them to a server in a compressed migration store.

1. On each source computer, the administrator runs the ScanState tool using Microsoft System Center Configuration Manager (SCCM), Microsoft Deployment Toolkit (MDT), a logon script, a batch file, or a non-Microsoft management technology. ScanState collects the user state from each source computer and then saves it to a server.
2. On each new computer, the administrator installs the company's SOE, which includes Windows 10 and other company applications.
3. On each of the new computers, the administrator runs the LoadState tool using System Center Configuration Manager, a logon script, a batch file, or a non-Microsoft management technology. LoadState migrates each user state from the migration store to one of the new computers.

Related topics

[Plan Your Migration](#)

[Choose a Migration Store Type](#)

[Offline Migration Reference](#)

What does USMT migrate?

6/6/2019 • 8 minutes to read • [Edit Online](#)

In this topic

- [Default migration scripts](#)
- [User Data](#)
- [Operating-system components](#)
- [Supported applications](#)
- [What USMT does not migrate](#)

Default migration scripts

The User State Migration Tool (USMT) 10.0 is designed so that an IT engineer can precisely define migrations using the USMT .xml scripting language. USMT provides the following sample scripts:

- **MigApp.XML.** Rules to migrate application settings.
- **MigDocs.XML.** Rules that use the **MigXmlHelper.GenerateDocPatterns** helper function, which can be used to automatically find user documents on a computer without the need to author extensive custom migration .xml files.
- **MigUser.XML.** Rules to migrate user profiles and user data.

MigUser.xml gathers everything in a user's profile and then does a file extension- based search of most of the system for other user data. If data doesn't match either of these criteria, the data won't be migrated. For the most part, this file describes a "core" migration.

The following data does not migrate with MigUser.xml:

- Files outside the user profile that don't match one of the file extensions in MigUser.xml.
- Access control lists (ACLs) for folders outside the user profile.

User data

This section describes the user data that USMT migrates by default, using the MigUser.xml file. It also defines how to migrate ACLs.

- **Folders from each user profile.** When you specify the MigUser.xml file, USMT migrates everything in a user's profiles including the following:

My Documents, My Video, My Music, My Pictures, desktop files, Start menu, Quick Launch settings, and Favorites.

IMPORTANT

Starting in Windows 10, version 1607 the USMT does not migrate the Start menu layout. To migrate a user's Start menu, you must export and then import settings using the Windows PowerShell cmdlets **Export-StartLayout** and **Import-StartLayout**. For more information, see [USMT common issues](#).

- **Folders from the All Users and Public profiles.** When you specify the MigUser.xml file, USMT also migrates the following from the **All Users** profile in Windows® XP, or the **Public** profile in Windows Vista, Windows 7, or Windows 8:

- Shared Documents
- Shared Video
- Shared Music
- Shared desktop files
- Shared Pictures
- Shared Start menu
- Shared Favorites

- **File types.** When you specify the MigUser.xml file, the ScanState tool searches the fixed drives, collects and then migrates files with any of the following file extensions:

.accdb, .ch3, .csv, .dif, .doc*, .dot*, .dqy, .iqy, .mcw, .mdb*, .mpp, .one*, .oqy, .or6, .pot*, .ppa, .pps*, .ppt*, .pre, .pst, .pub, .qdf, .qel, .qph, .qsd, .rqy, .rtf, .scd, .sh3, .slk, .txt, .vl*, .vsd, .wk*, .wpd, .wps, .wq1, .wri, .xl*, .xla, .xlb, .xls*.

Note The asterisk (*) stands for zero or more characters.

- **Access control lists.** USMT migrates ACLs for specified files and folders from computers running both Windows® XP and Windows Vista. For example, if you migrate a file named File1.txt that is read-only for User1 and read/write for User2, these settings will still apply on the destination computer after the migration.

Important To migrate ACLs, you must specify the directory to migrate in the MigUser.xml file. Using file patterns like *.doc will not migrate a directory. The source ACL information is migrated only when you explicitly specify the directory. For example, `<pattern type="File">c:\test docs</pattern>`.

Operating-system components

USMT migrates operating-system components to a destination computer from computers running Windows 7 and Windows 8

The following components are migrated by default using the manifest files:

- Accessibility settings
- Address book
- Command-prompt settings
- *Desktop wallpaper
- EFS files
- Favorites
- Folder options
- Fonts
- Group membership. USMT migrates users' group settings. The groups to which a user belongs can be found by right-clicking **My Computer** on the Start menu and then clicking **Manage**. When running an offline migration, the use of a **<ProfileControl>** section in the Config.xml file is required.

- *Windows Internet Explorer® settings
- Microsoft® Open Database Connectivity (ODBC) settings
- Mouse and keyboard settings
- Network drive mapping
- *Network printer mapping
- *Offline files
- *Phone and modem options
- RAS connection and phone book (.pbk) files
- *Regional settings
- Remote Access
- *Taskbar settings
- User personal certificates (all)
- Windows Mail.
- *Windows Media Player
- Windows Rights Management

* These settings are not available for an offline migration. For more information, see [Offline Migration Reference](#).

Important This list may not be complete. There may be additional components that are migrated.

Note Some settings, such as fonts, are not applied by the LoadState tool until after the destination computer has been restarted. For this reason, restart the destination computer after you run the LoadState tool.

Supported applications

Although it is not required for all applications, it is good practice to install all applications on the destination computer before restoring the user state. Installing applications before migrating settings helps to ensure that the migrated settings are not overwritten by the application installers.

Note The versions of installed applications must match on the source and destination computers. USMT does not support migrating the settings of an earlier version of an application to a later version, except for Microsoft Office.

Note USMT migrates only the settings that have been used or modified by the user. If there is an application setting on the source computer that was not touched by the user, the setting may not migrate.

When you specify the MigApp.xml file, USMT migrates the settings for the following applications:

PRODUCT	VERSION
Adobe Acrobat Reader	9
AOL Instant Messenger	6.8
Adobe Creative Suite	2

PRODUCT	VERSION
Adobe Photoshop CS	8, 9
Adobe ImageReady CS	
Apple iTunes	6, 7, 8
Apple QuickTime Player	5, 6, 7
Apple Safari	3.1.2
Google Chrome	beta
Google Picasa	3
Google Talk	beta
IBM Lotus 1-2-3	9
IBM Lotus Notes	6,7, 8
IBM Lotus Organizer	5
IBM Lotus WordPro	9.9
Intuit Quicken Deluxe	2009
Money Plus Business	2008
Money Plus Home	2008
Mozilla Firefox	3
Microsoft Office	2003, 2007, 2010
Microsoft Office Access®	2003, 2007, 2010
Microsoft Office Excel®	2003, 2007, 2010

PRODUCT	VERSION
Microsoft Office FrontPage®	2003, 2007, 2010
Microsoft Office OneNote®	2003, 2007, 2010
Microsoft Office Outlook®	2003, 2007, 2010
Microsoft Office PowerPoint®	2003, 2007, 2010
Microsoft Office Publisher	2003, 2007, 2010
Microsoft Office Word	2003, 2007, 2010
Opera Software Opera	9.5
Microsoft Outlook Express	(only mailbox file)
Microsoft Project	2003, 2007
Microsoft Office Visio®	2003, 2007
RealPlayer Basic	11
Sage Peachtree	2009
Skype	3.8
Windows Live Mail	12, 14
Windows Live Messenger	8.5, 14
Windows Live MovieMaker	14
Windows Live Photo Gallery	12, 14
Windows Live Writer	12, 14
Windows Mail	(Windows 7 and 8)

PRODUCT	VERSION
Microsoft Works	9
Yahoo Messenger	9
Microsoft Zune™ Software	3

What USMT does not migrate

The following is a list of the settings that USMT does not migrate. If you are having a problem that is not listed here, see [Common Issues](#).

Application settings

USMT does not migrate the following application settings:

- Settings from earlier versions of an application. The versions of each application must match on the source and destination computers. USMT does not support migrating the settings of an earlier version of an application to a later version, except for Microsoft Office. USMT can migrate from an earlier version of Microsoft Office to a later version.
- Application settings and some operating-system settings when a local account is created. For example, if you run /lac to create a local account on the destination computer, USMT will migrate the user data, but only some of the operating-system settings, such as wallpaper and screensaver settings, and no application settings will migrate.
- Microsoft Project settings, when migrating from Office 2003 to Office 2007 system.
- ICQ Pro settings, if ICQ Pro is installed in a different location on the destination computer. To successfully migrate the settings of ICQ Pro, you must install ICQ Pro in the same location on the destination computer as it was on the source computer. Otherwise, after you run the LoadState tool, the application will not start. You may encounter problems when:
 - You change the default installation location on 32-bit destination computers.
 - You attempt to migrate from a 32-bit computer to a 64-bit computer. This is because the ICQ Pro default installation directory is different on the two types of computers. When you install ICQ Pro on a 32-bit computer, the default location is "C:\Program Files\...". The ICQ Pro default installation directory on an x64-based computer, however, is "C:\Program Files (x86)\...".

Operating-System settings

USMT does not migrate the following operating-system settings.

- Local printers, hardware-related settings, drivers, passwords, application binary files, synchronization files, DLL files, or other executable files.
- Permissions for shared folders. After migration, you must manually re-share any folders that were shared on the source computer.
- Files and settings migrating between operating systems with different languages. The operating system of the source computer must match the language of the operating system on the destination computer.
- Customized icons for shortcuts may not migrate.
- Taskbar settings, when the source computer is running Windows XP.

You should also note the following:

- You should run USMT from an account with administrative credentials. Otherwise, some data will not migrate. When running the ScanState and LoadState tools you must run the tools in Administrator mode from an account with administrative credentials. If you do not run USMT in Administrator mode, only the user profile that is logged on will be included in the migration. In addition, you must run the ScanState tool on Windows XP from an account with administrative credentials. Otherwise, some operating-system settings will not migrate. To run in Administrator mode, click **Start**, click **All Programs**, click **Accessories**, right-click **Command Prompt**, and then click **Run as administrator**.
- You can use the **/localonly** option to exclude the data from removable drives and network drives mapped on the source computer. For more information about what is excluded when you specify **/localonly**, see [ScanState Syntax](#).

Start menu layout

Starting in Windows 10, version 1607 the USMT does not migrate the Start menu layout. To migrate a user's Start menu, you must export and then import settings using the Windows PowerShell cmdlets **Export-StartLayout** and **Import-StartLayout**. For more information, see [USMT common issues](#).

Related topics

[Plan your migration](#)

Choose a Migration Store Type

6/14/2019 • 2 minutes to read • [Edit Online](#)

One of the main considerations for planning your migration is to determine which migration store type best meets your needs. As part of these considerations, determine how much space is required to run the User State Migration Tool (USMT) 10.0 components on your source and destination computers, and how much space is needed to create and host the migration store, whether you are using a local share, network share, or storage device. The final consideration is ensuring that user data integrity is maintained by encrypting the migration store.

In This Section

Migration Store Types Overview	Choose the migration store type that works best for your needs and migration scenario.
Estimate Migration Store Size	Estimate the amount of disk space needed for computers in your organization based on information about your organization's infrastructure.
Hard-Link Migration Store	Learn about hard-link migration stores and the scenarios in which they are used.
Migration Store Encryption	Learn about the using migration store encryption to protect user data integrity during a migration.

Related topics

[Plan Your Migration](#)

[User State Migration Tool \(USMT\) How-to topics](#)

Migration Store Types Overview

6/14/2019 • 3 minutes to read • [Edit Online](#)

When planning your migration, you should determine which migration store type best meets your needs. As part of these considerations, determine how much space is required to run the User State Migration Tool (USMT) 10.0 components on your source and destination computers. You should also determine the space needed to create and host the migration store, whether you are using a local share, network share, or storage device.

In This Topic

[Migration Store Types](#)

[Local Store vs. Remote Store](#)

[The /localonly Command-Line Option](#)

Migration Store Types

This section describes the three migration store types available in USMT.

Uncompressed (UNC)

The uncompressed (UNC) migration store is an uncompressed directory with a mirror image of the folder hierarchy being migrated. Each directory and file retains the same access permissions that it has on the local file system. You can use Windows Explorer to view this migration store type. Settings are stored in a catalog file that also describes how to restore files on the destination computer.

Compressed

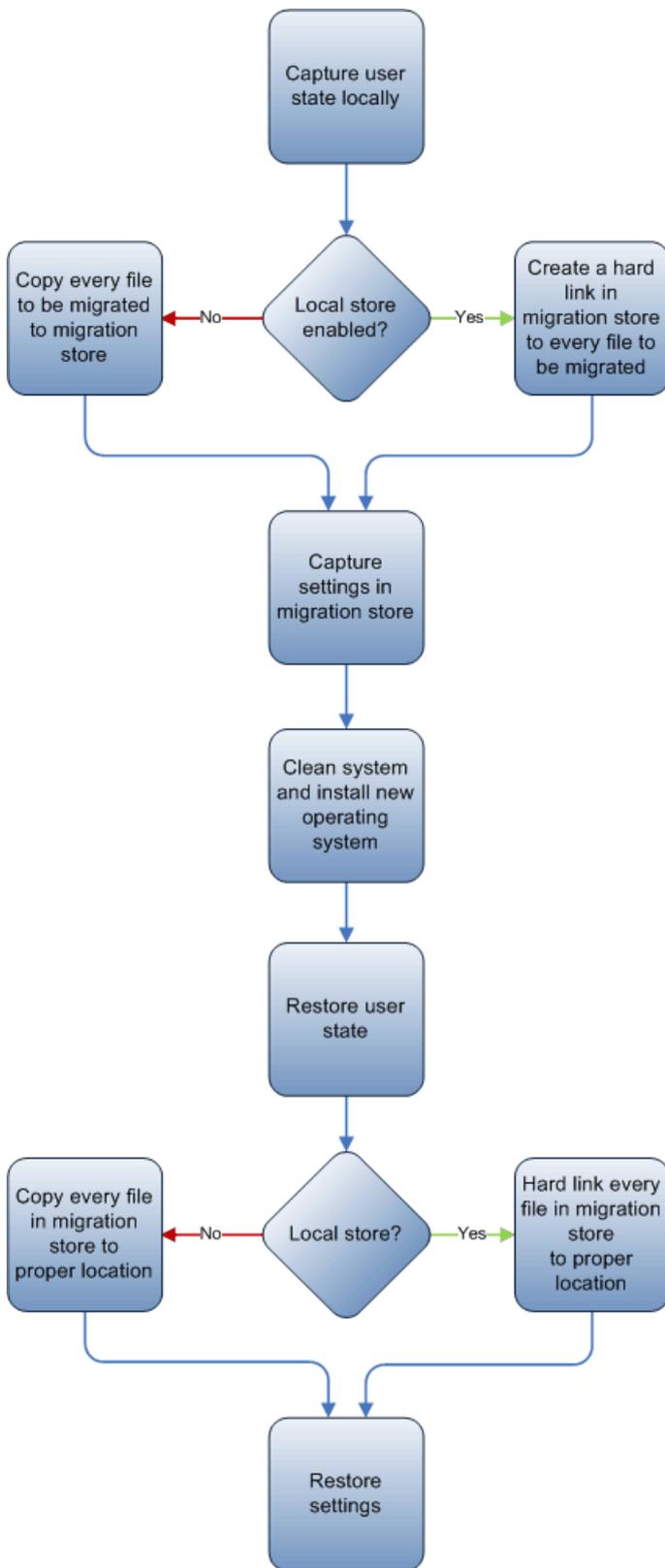
The compressed migration store is a single image file that contains all files being migrated and a catalog file. This image file is often encrypted and protected with a password, and cannot be navigated with Windows Explorer.

Hard-Link

A hard-link migration store functions as a map that defines how a collection of bits on the hard disk are “wired” into the file system. You use the new USMT hard-link migration store in the PC Refresh scenario only. This is because the hard-link migration store is maintained on the local computer while the old operating system is removed and the new operating system is installed. Using a hard-link migration store saves network bandwidth and minimizes the server use needed to accomplish the migration.

You use a command-line option, **/hardlink**, to create a hard-link migration store, which functions the same as an uncompressed migration store. Files are not duplicated on the local computer when user state is captured, nor are they duplicated when user state is restored. For more information, see [Hard-Link Migration Store](#).

The following flowchart illustrates the procedural differences between a local migration store and a remote migration store. In this example, a hard-link migration store is used for the local store.



Local Store vs. Remote Store

If you have enough space and you are migrating the user state back to the same computer, storing data on a local device is normally the best option to reduce server storage costs and network performance issues. You can store the data locally either on a different partition or on a removable device such as a USB flash drive (UFD). Also, depending on the imaging technology that you are using, you might be able to store the data on the partition that is being re-imaged, if the data will be protected from deletion during the process. To increase performance, store the data on high-speed drives that use a high-speed network connection. It is also good practice to ensure that the migration is the only task the server is performing.

If there is not enough local disk space, or if you are moving the user state to another computer, then you must store

the data remotely. For example, you can store it in on a shared folder, on removable media such as a UFD drive, or you can store it directly on the destination computer. For example, create and share C:\store on the destination computer. Then run the ScanState command on the source computer and save the files and settings to \\DestinationComputerName\store. Then, run the **LoadState** command on the destination computer and specify **C:\Store** as the store location. By doing this, you do not need to save the files to a server.

Important If possible, have users store their data within their %UserProfile%\My Documents and %UserProfile%\Application Data folders. This will reduce the chance of USMT missing critical user data that is located in a directory that USMT is not configured to check.

The /localonly Command-Line Option

You should use this option to exclude the data from removable drives and network drives mapped on the source computer. For more information about what is excluded when you specify **/LocalOnly**, see [ScanState Syntax](#).

Related topics

[Plan Your Migration](#)

Estimate Migration Store Size

6/26/2019 • 6 minutes to read • [Edit Online](#)

The disk space requirements for a migration are dependent on the size of the migration store and the type of migration. You can estimate the amount of disk space needed for computers in your organization based on information about your organization's infrastructure. You can also calculate the disk space requirements using the ScanState tool.

In This Topic

- [Hard Disk Space Requirements](#). Describes the disk space requirements for the migration store and other considerations on the source and destination computers.
- [Calculate Disk Space Requirements Using the ScanState Tool](#). Describes how to use the ScanState tool to determine how big the migration store will be on a particular computer.
- [Estimate Migration Store Size](#). Describes how to estimate the average size of migration stores for the computers in your organization, based on your infrastructure.

Hard Disk Space Requirements

- **Store.** For non-hard-link migrations, you should ensure that there is enough available disk space at the location where you will save your store to contain the data being migrated. You can save your store to another partition, an external storage device such as a USB flash drive or a server. For more information, see [Choose a Migration Store Type](#).
- **Source Computer.** The source computer needs enough available space for the following:
 - [E250 megabytes \(MB\) minimum of hard disk space](#). Space is needed to support the User State Migration Tool (USMT) 10.0 operations, for example, growth in the page file. Provided that every volume involved in the migration is formatted as NTFS, 250 MB should be enough space to ensure success for almost every hard-link migration, regardless of the size of the migration. The USMT tools will not create the migration store if 250 MB of disk space is not available.
 - [Temporary space for USMT to run](#). Additional disk space for the USMT tools to operate is required. This does not include the minimum 250 MB needed to create the migration store. The amount of temporary space required can be calculated using the ScanState tool.
 - [Hard-link migration store](#). It is not necessary to estimate the size of a hard-link migration store. The only case where the hard-link store can be quite large is when non-NTFS file systems exist on the system and contain data being migrated.
- **Destination computer.** The destination computer needs enough available space for the following:
 - [Operating system](#).
 - [Applications](#).
 - [Data being migrated](#). It is important to consider that in addition to the files being migrated, registry information will also require hard disk space for storage.
 - [Temporary space for USMT to run](#). Additional disk space for the USMT tools to operate is required. The amount of temporary space required can be calculated using the ScanState tool.

Calculate Disk Space Requirements using the ScanState Tool

You can use the ScanState tool to calculate the disk space requirements for a particular compressed or uncompressed migration. It is not necessary to estimate the migration store size for a hard-link migration since this method does not create a separate migration store. The ScanState tool provides disk space requirements for the state of the computer at the time the tool is run. The state of the computer may change during day to day use so it is recommended that you use the calculations as an estimate when planning your migration.

To run the ScanState tool on the source computer with USMT installed,

1. Open a command prompt with administrator privileges.
2. Navigate to the USMT tools. For example, type

```
cd /d "C:\Program Files (x86)\Windows Kits\8.0\Assessment and Deployment Kit\User State Migration Tool\  
<architecture>"
```

Where *<architecture>* is x86 or amd64.

3. Run the **ScanState** tool to generate an XML report of the space requirements. At the command prompt, type

```
ScanState.exe <StorePath> /p:<path to a file>
```

Where *<StorePath>* is a path to a directory where the migration store will be saved and *<path to a file>* is the path and filename where the XML report for space requirements will be saved. For example,

```
ScanState.exe c:\store /p:c:\spaceRequirements.xml
```

The migration store will not be created by running this command, but `StorePath` is a required parameter.

The ScanState tool also allows you to estimate disk space requirements based on a customized migration. For example, you might not want to migrate the My Documents folder to the destination computer. You can specify this in a configuration file when you run the ScanState tool. For more information, see [Customize USMT XML Files](#).

Note To preserve the functionality of existing applications or scripts that require the previous behavior of USMT, the **/p** option, without specifying *<path to a file>* is still available in USMT.

The space requirements report provides two elements, **<storeSize>** and **<temporarySpace>**. The **<temporarySpace>** value shows the disk space, in bytes, that USMT uses to operate during the migration—this does not include the minimum 250 MB needed to support USMT. The **<storeSize>** value shows the disk space, in bytes, required to host the migration store contents on both the source and destination computers. The following example shows a report generated using **/p:<path to a file>**.

```
<?xml version="1.0" encoding="UTF-8"?>  
<PreMigration>  
  <storeSize>  
    <size clusterSize="4096">11010592768</size>  
  </storeSize>  
  <temporarySpace>  
    <size>58189144</size>  
  </temporarySpace>  
</PreMigration>
```

Additionally, USMT performs a compliance check for a required minimum of 250 MB of available disk space and

will not create a store if the compliance check fails.

Estimate Migration Store Size

Determine how much space you will need to store the migrated data. You should base your calculations on the volume of e-mail, personal documents, and system settings for each user. The best way to estimate these is to survey several computers to arrive at an average for the size of the store that you will need.

The amount of space that is required in the store will vary, depending on the local storage strategies your organization uses. For example, one key element that determines the size of migration data sets is e-mail storage. If e-mail is stored centrally, data sets will be smaller. If e-mail is stored locally, such as offline-storage files, data sets will be larger. Mobile users will typically have larger data sets than workstation users. You should perform tests and inventory the network to determine the average data set size in your organization.

Note You can create a space-estimate file (Usmtsizetext), by using the legacy **/p** command-line option to estimate the size of the store.

When trying to determine how much disk space you will need, consider the following issues:

- **E-mail** : If users deal with a large volume of e-mail or keep e-mail on their local computers instead of on a mail server, the e-mail can take up as much disk space as all other user files combined. Prior to migrating user data, make sure that users who store e-mail locally synchronize their inboxes with their mail server.
- **User documents**: Frequently, all of a user's documents fit into less than 50 MB of space, depending on the types of files involved. This estimate assumes typical office work, such as word-processing documents and spreadsheets. This estimate can vary substantially based on the types of documents that your organization uses. For example, an architectural firm that predominantly uses computer-aided design (CAD) files needs much more space than a law firm that primarily uses word-processing documents. You do not need to migrate the documents that users store on file servers through mechanisms such as Folder Redirection, as long as users will have access to these locations after the migration.
- **User system settings** Five megabytes is usually adequate space to save the registry settings. This requirement can fluctuate, however, based on the number of applications that have been installed. It is rare, however, for the user-specific portion of the registry to exceed 5 MB.

Related topics

[Common Migration Scenarios](#)

Hard-Link Migration Store

6/6/2019 • 10 minutes to read • [Edit Online](#)

A *hard-link migration store* enables you to perform an in-place migration where all user state is maintained on the computer while the old operating system is removed and the new operating system is installed; this is why it is best suited for the computer-refresh scenario. Use of a hard-link migration store for a computer-refresh scenario drastically improves migration performance and significantly reduces hard-disk utilization, reduces deployment costs and enables entirely new migration scenarios.

In This Topic

[When to Use a Hard-Link Migration](#)

[Understanding a Hard-Link Migration](#)

[Scenario](#)

[Hard-Link Migration Store Details](#)

[Hard Disk Space](#)

[Hard-Link Store Size Estimation](#)

[Migration Store Path on Multiple Volumes](#)

[Location Modifications](#)

[Migrating Encrypting File System \(EFS\) Certificates and Files](#)

[Migrating Locked Files With the Hard-Link Migration Store](#)

[XML Elements in the Config.xml File](#)

When to Use a Hard-Link Migration

You can use a hard-link migration store when your planned migration meets both of the following criteria:

- You are upgrading the operating system on existing hardware rather than migrating to new computers.
- You are upgrading the operating system on the same volume of the computer.

You cannot use a hard-link migration store if your planned migration includes any of the following:

- You are migrating data from one computer to a second computer.
- You are migrating data from one volume on a computer to another volume, for example from C: to D:.
- You are formatting or repartitioning the disk outside of Windows Setup, or specifying a disk format or repartition during Windows Setup that will remove the migration store.

Understanding a Hard-Link Migration

The hard-link migration store is created using the command-line option, **/hardlink**, and is equivalent to other migration-store types. However, it differs in that hard links are utilized to keep files stored on the source computer during the migration. Keeping the files in place on the source computer eliminates the redundant work of duplicating files. It also enables the performance benefits and reduction in disk utilization that define this scenario.

When you create a hard link, you give an existing file an additional path. For instance, you could create a hard link to `c:\file1.txt` called `c:\hard link\myFile.txt`. These are two paths to the same file. If you open `c:\file1.txt`, make changes, and save the file, you will see those changes when you open `c:\hard link\myFile.txt`. If you delete `c:\file1.txt`, the file still exists on your computer as `c:\hardlink\myFile.txt`. You must delete both references to the file in order to delete the file.

Note A hard link can only be created for a file on the same volume. If you copy a hard-link migration store to another drive or external device, the files, and not the links, are copied, as in a non-compressed migration-store scenario.

For more information about hard links, please see [Hard Links and Junctions](#)

In most aspects, a hard-link migration store is identical to an uncompressed migration store. It is located where specified by the Scanstate command-line tool and you can view the contents of the store by using Windows® Explorer. Once created, it can be deleted or copied to another location without changing user state. Restoring a hard-link migration store is similar to restoring any other migration store; however, as with creating the store, the same hard-link functionality is used to keep files in-place.

As a best practice, we recommend that you delete the hard-link migration store after you confirm that the Loadstate tool has successfully migrated the files. Since Loadstate has created new paths to the files on your new installation of a Windows operating system, deleting the hard links in the migration store will only delete one path to the files and will not delete the actual files or the paths to them from your new operating system.

Important Using the `/c` option will force the Loadstate tool to continue applying files when non-fatal errors occur. If you use the `/c` option, you should verify that no errors are reported in the logs before deleting the hard-link migration store in order to avoid data loss.

Keeping the hard-link migration store can result in additional disk space being consumed or problems with some applications for the following reasons:

- Applications reporting file-system statistics, for example, space used and free space, might incorrectly report these statistics while the hard-link migration store is present. The file may be reported twice because of the two paths that reference that file.
- A hard link may lose its connection to the original file. Some applications save changes to a file by creating a temporary file and then renaming the original to a backup filename. The path that was not used to open the file in this application will continue to refer to the unmodified file. The unmodified file that is not in use is taking up additional disk space. You should create the hard-link migration store just before you perform the migration, and not use applications once the store is created, in order to make sure you are migrating the latest versions of all files.
- Editing the file by using different paths simultaneously may result in data corruption.

Important The read-only file attribute on migrated files is lost when the hard-link migration store is deleted. This is due to a limitation in NTFS file system hard links.

Hard-Link Migration Scenario

For example, a company has decided to deploy Windows 10 on all of their computers. Each employee will keep the same computer, but the operating system on each computer will be updated.

1. An administrator runs the ScanState command-line tool on each computer, specifying the `/hardlink` command-line option. The ScanState tool saves the user state to a hard-link migration store on each computer, improving performance by reducing file duplication, except in certain specific instances.

Note As a best practice, we recommend that you do not create your hard-link migration store until just before you perform the migration in order to migrate the latest versions of your files. You should not use

your software applications on the computer after creating the migration store until you have finished migrating your files with Loadstate.

2. On each computer, an administrator installs the company's standard operating environment (SOE), which includes Windows 7 and other applications the company currently uses.
3. An administrator runs the LoadState command-line tool on each computer. The LoadState tool restores user state back on each computer.

Hard-Link Migration Store Details

This section provides details about hard-link migration stores.

Hard Disk Space

The **/hardlink** command-line option proceeds with creating the migration store only if there is 250 megabytes (MB) of free space on the hard disk. Provided that every volume involved in the migration is formatted as NTFS, 250 MB should be enough space to ensure success for almost every hard-link migration, regardless on the size of the migration.

Hard-Link Store Size Estimation

It is not necessary to estimate the size of a hard-link migration store. Estimating the size of a migration store is only useful in scenarios where the migration store is very large, and on NTFS volumes the hard-link migration store will require much less incremental space than other store options. The only case where the local store can be quite large is when non-NTFS file systems exist on the system and contain data being migrated. Since NTFS has been the default file system format for Windows XP and newer operating systems, this situation is unusual.

Migration Store Path on Multiple Volumes

Separate hard-link migration stores are created on each NTFS volume that contain data being migrated. In this scenario, the primary migration-store location will be specified on the command line, and should be the operating-system volume. Migration stores with identical names and directory names will be created on every volume containing data being migrated. For example:

```
Scanstate /hardlink c:\USMTMIG [...]
```

Running this command on a system that contains the operating system on the C: drive and the user data on the D: drive will generate migration stores in the following locations, assuming that both drives are NTFS:

C:\USMTMIG\

D:\USMTMIG\

The drive you specify on the command line for the hard-link migration store is important, because it defines where the *master migration store* should be placed. The *master migration store* is the location where data migrating from non-NTFS volumes is stored. This volume must have enough space to contain all of the data that comes from non-NTFS volumes. As in other scenarios, if a migration store already exists at the specified path, the **/o** option must be used to overwrite the existing data in the store.

Location Modifications

Location modifications that redirect migrated content from one volume to a different volume have an adverse impact on the performance of a hard-link migration. This is because the migrating data that must cross system volumes cannot remain in the hard-link migration store, and must be copied across the system volumes.

Migrating Encrypting File System (EFS) Certificates and Files

To migrate Encrypting File System (EFS) files to a new installation of an operating system on the same volume of the computer, specify the **/efs:hardlink** option in the Scanstate command-line syntax.

If the EFS files are being restored to a different partition, you should use the **/efs:copyraw** option instead of the

/efs:hardlink option. Hard links can only be created for files on the same volume. Moving the files to another partition during the migration requires a copy of the files to be created on the new partition. The **/efs:copyraw** option will copy the files to the new partition in encrypted format.

For more information, see [Migrate EFS Files and Certificates](#) and the Encrypted File Options in [ScanState Syntax](#).

Migrating Locked Files with the Hard-Link Migration Store

Files that are locked by an application or the operating system are handled differently when using a hard-link migration store.

Files that are locked by the operating system cannot remain in place and must be copied into the hard-link migration store. As a result, selecting many operating-system files for migration significantly reduces performance during a hard-link migration. As a best practice, we recommend that you do not migrate any files out of the \Windows directory, which minimizes performance-related issues.

Files that are locked by an application are treated the same in hard-link migrations as in other scenarios when the volume shadow-copy service is not being utilized. The volume shadow-copy service cannot be used in conjunction with hard-link migrations. However, by modifying the new **<HardLinkStoreControl>** section in the Config.xml file, it is possible to enable the migration of files locked by an application.

Important There are some scenarios in which modifying the **<HardLinkStoreControl>** section in the Config.xml file makes it more difficult to delete a hard-link migration store. In these scenarios, you must use USMTutils.exe to schedule the migration store for deletion on the next restart.

XML Elements in the Config.xml File

A new section in the Config.xml file allows optional configuration of some of the hard-link migration behavior introduced with the **/HardLink** option.

<Policies>	This element contains elements that describe the policies that USMT follows while creating a migration store.
<HardLinkStoreControl>	This element contains elements that describe how to handle files during the creation of a hard link migration store.
<fileLocked>	This element contains elements that describe how to handle files that are locked for editing.
<createHardLink>	This element defines a standard MigXML pattern that describes file paths where hard links should be created, even if the file is locked for editing by another application. Syntax: <code><createHardLink> [pattern] </createHardLink></code>
<errorHardLink>	This element defines a standard MigXML pattern that describes file paths where hard links should not be created, if the file is locked for editing by another application. <code><errorHardLink> [pattern] </errorHardLink></code>

Important You must use the **/nocompress** option with the **/HardLink** option.

The following XML sample specifies that files locked by an application under the \Users directory can remain in

place during the migration. It also specifies that locked files that are not located in the \Users directory should result in the **File in Use** error. It is important to exercise caution when specifying the paths using the **File in Use** `<createhardlink>` tag in order to minimize scenarios that make the hard-link migration store more difficult to delete.

```
<Policies>
  <HardLinkStoreControl>
    <fileLocked>
      <createHardLink>c:\Users\* [*]</createHardLink>
      <errorHardLink>C:\* [*]</errorHardLink>
    </fileLocked>
  </HardLinkStoreControl>
</Policies>
```

Related topics

[Plan Your Migration](#)

Migration Store Encryption

6/6/2019 • 2 minutes to read • [Edit Online](#)

This topic discusses User State Migration Tool (USMT) 10.0 options for migration store encryption to protect the integrity of user data during a migration.

USMT Encryption Options

USMT enables support for stronger encryption algorithms, called Advanced Encryption Standard (AES), in several bit-level options. AES is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data.

The encryption algorithm you choose must be specified for both the **ScanState** and the **LoadState** commands, so that these commands can create or read the store during encryption and decryption. The new encryption algorithms can be specified on the **ScanState** and the **LoadState** command lines by using the **/encrypt:"encryptionstrength"** and the **/decrypt:"encryptionstrength"** command-line options. All of the encryption application programming interfaces (APIs) used by USMT are available in Windows 7, Windows 8, and Windows 10 operating systems. However, export restrictions might limit the set of algorithms that are available to computers in certain locales. You can use the Usmtutils.exe file to determine which encryption algorithms are available to the computers' locales before you begin the migration.

The following table describes the command-line encryption options in USMT.

COMPONENT	OPTION	DESCRIPTION
ScanState	/encrypt <AES, AES_128, AES_192, AES_256, 3DES, 3DES_112>	This option and argument specify that the migration store is encrypted and which algorithm to use. When the algorithm argument is not provided, the ScanState tool employs the 3DES algorithm.
LoadState	/decrypt <AES, AES_128, AES_192, AES_256, 3DES, 3DES_112>	This option and argument specify that the store must be decrypted and which algorithm to use. When the algorithm argument is not provided, the LoadState tool employs the 3DES algorithm.

Important Some encryption algorithms may not be available on your systems. You can verify which algorithms are available by running the UsmtUtils command with the **/ec** option. For more information see [UsmtUtils Syntax](#)

Related topics

[Plan Your Migration](#)

Determine What to Migrate

6/14/2019 • 2 minutes to read • [Edit Online](#)

By default, User State Migration Tool (USMT) 10.0 migrates the items listed in [What Does USMT Migrate?](#), depending on the migration .xml files you specify. These default settings are often enough for a basic migration.

However, when considering what settings to migrate, you should also consider what settings you would like the user to be able to configure, if any, and what settings you would like to standardize. Many organizations use their migration as an opportunity to create and begin enforcing a better-managed environment. Some of the settings that users can configure on unmanaged computers prior to the migration can be locked on the new, managed computers. For example, standard wallpaper, Internet Explorer security settings, and desktop configuration are some of the items you can choose to standardize.

To reduce complexity and increase standardization, your organization should consider creating a *standard operating environment (SOE)*. An SOE is a combination of hardware and software that you distribute to all users. This means selecting a baseline for all computers, including standard hardware drivers; core operating system features; core productivity applications, especially if they are under volume licensing; and core utilities. This environment should also include a standard set of security features, as outlined in the organization's corporate policy. Using a standard operating environment can vastly simplify the migration and reduce overall deployment challenges.

In This Section

Identify Users	Use command-line options to specify which users to migrate and how they should be migrated.
Identify Applications Settings	Determine which applications you want to migrate and prepare a list of application settings to be migrated.
Identify Operating System Settings	Use migration to create a new standard environment on each of the destination computers.
Identify File Types, Files, and Folders	Determine and locate the standard, company-specified, and non-standard locations of the file types, files, folders, and settings that you want to migrate.

Related topics

[What Does USMT Migrate?](#)

Identify Users

6/6/2019 • 2 minutes to read • [Edit Online](#)

It is important to carefully consider how you plan to migrate users. By default, all users are migrated by User State Migration Tool (USMT) 5.0. You must specify which users to include by using the command line. You cannot specify users in the .xml files. For instructions on how to migrate users, see [Migrate User Accounts](#).

In This Topic

- [Migrating Local Accounts](#)
- [Migrating Domain Accounts](#)
- [Command-Line Options](#)

Migrating Local Accounts

Before migrating local accounts, note the following:

- [You must explicitly specify that local accounts that are not on the destination computer should be migrated.](#) If you are migrating local accounts and the local account does not exist on the destination computer, you must use the **/lac** option when using the LoadState command. If the **/lac** option is not specified, no local user accounts will be migrated.
- [Consider whether to enable user accounts that are new to the destination computer.](#) The **/lae** option enables the account that was created with the **/lac** option. However, if you create a disabled local account by using only the **/lac** option, a local administrator must enable the account on the destination computer.
- [Be careful when specifying a password for local accounts.](#) If you create the local account with a blank password, anyone could log on to that account on the destination computer. If you create the local account with a password, the password is available to anyone with access to the USMT command-line tools.

Note

If there are multiple users on a computer, and you specify a password with the **/lac** option, all migrated users will have the same password.

Migrating Domain Accounts

The source and destination computers do not need to be connected to the domain for domain user profiles to be migrated.

Command-Line Options

USMT provides several options to migrate multiple users on a single computer. The following command-line options specify which users to migrate.

- [Specifying users.](#) You can specify which users to migrate with the **/all**, **/ui**, **/uel**, and **/ue** options with both the ScanState and LoadState command-line tools.

Important The **/uel** option excludes users based on the **LastModified** date of the Ntuser.dat file. The **/uel** option is not valid in offline migrations.

- [Moving users to another domain.](#) You can move user accounts to another domain using the **/md** option

with the LoadState command-line tool.

- [Creating local accounts](#). You can create and enable local accounts using the **/lac** and **/lae** options with the LoadState command-line tool.
- [Renaming user accounts](#). You can rename user accounts using the **/mu** option.

Note By default, if a user name is not specified in any of the command-line options, the user will be migrated.

Related topics

[Determine What to Migrate](#)

[ScanState Syntax](#)

[LoadState Syntax](#)

Identify Applications Settings

5/31/2019 • 2 minutes to read • [Edit Online](#)

When planning for your migration, you should identify which applications and settings you want to migrate. For more information about how to create a custom .xml file to migrate the settings of another application, see [Customize USMT XML Files](#).

Applications

First, create and prioritize a list of applications that to be migrated. It may be helpful to review the application lists and decide which applications will be redeployed and which applications will be retired. Often, the applications are prioritized based on a combination of how widely the application is used and how complex the application is.

Next, identify an application owner to be in charge of each application. This is necessary because the developers will not be experts on all of the applications in the organization. The application owner should have the most experience with an application. The application owner provides insight into how the organization installs, configures, and uses the application.

Application Settings

Next, determine and locate the application settings to be migrated. You can acquire much of the information that you need for this step when you are testing the new applications for compatibility with the new operating system.

After completing the list of applications to be migrated, review the list and work with each application owner on a list of settings to be migrated. For each setting, determine whether it needs to be migrated or if the default settings are adequate. Then, determine where the setting is located; for example, in the registry or in an .ini file. Next, consider the following questions to determine what needs to be done to migrate the setting successfully:

- Is the destination version of the application newer than the source version?
- Do these settings work with the new version?
- Do the settings need to be moved or altered?
- Can the first-run process force the application to appear as if it had run already? If so, does this work correctly, or does it break the application?

After answering these questions, create a custom .xml file to migrate settings. Work with the application owner to develop test cases and to determine the file types that need to be migrated for the application.

Locating Where Settings Are Stored

See [Migrate Application Settings](#) and follow the directions.

Related topics

[Determine What to Migrate](#)

Identify Operating System Settings

6/6/2019 • 2 minutes to read • [Edit Online](#)

When planning for your migration, you should identify which operating system settings you want to migrate and to what extent you want to create a new standard environment on each of the computers. User State Migration Tool (USMT) 10.0 enables you to migrate select settings and keep the default values for all others. The operating system settings include the following:

- **Appearance.**

This includes items such as wallpaper, colors, sounds, and the location of the taskbar.

- **Action.**

This includes items such as the key-repeat rate, whether double-clicking a folder opens it in a new window or the same window, and whether you need to single-click or double-click an item to open it.

- **Internet.**

These are the settings that let you connect to the Internet and control how your browser operates. This includes items such as your home page URL, favorites, bookmarks, cookies, security settings, dial-up connections, and proxy settings.

- **Mail.**

This includes the information that you need to connect to your mail server, your signature file, views, mail rules, local mail, and contacts.

To help you decide which settings to migrate, you should consider any previous migration experiences as well as the results of any surveys and tests that you have conducted. You should also consider the number of help-desk calls related to operating-system settings that you have had in the past, and are able to handle in the future. Also decide how much of the new operating-system functionality you want to take advantage of.

You should migrate any settings that users need to get their jobs done, those that make the work environment comfortable, and those that will reduce help-desk calls after the migration. Although it is easy to dismiss migrating user preferences, you should consider that users can spend a significant amount of time restoring items such as wallpaper, screen savers, and other customizable user-interface features. Most users do not remember how these settings were applied. Although these items are not critical to migration success, migrating these items increases user productivity and overall satisfaction of the migration process.

Note For more information about how to change the operating-system settings that are migrated, see [User State Migration Tool \(USMT\) How-to topics](#).

For information about the operating-system settings that USMT migrates, see [What Does USMT Migrate?](#)

Related topics

[Determine What to Migrate](#)

Identify File Types, Files, and Folders

5/31/2019 • 2 minutes to read • [Edit Online](#)

When planning for your migration, if not using MigDocs.xml, you should identify the file types, files, folders, and settings that you want to migrate. First, you should determine the standard file locations on each computer, such as **My Documents**, **C:\Data**, and company-specified locations, such as **\EngineeringDrafts**. Next, you should determine and locate the non-standard locations. For non-standard locations, consider the following:

- **File types.** Consider which file types need to be included and excluded from the migration. You can create this list based on common applications used in your organization. Applications normally use specific file name extensions. For example, Microsoft Office Word primarily uses .doc, .docx and .dotx file name extension. However, it also uses other file types, such as templates (.dot files), on a less frequent basis.
- **Excluded locations.** Consider the locations on the computer that should be excluded from the migration (for example, %WINDIR% and Program Files).
- **New locations.** Decide where files should be migrated to on the destination computer for example, \My Documents, a designated folder, or a folder matching the files' name and location on the source computer. For example, you might have shared data on source machine or you might wish to clean up documents outside the user profiles on the source system. Identify any data that needs to be redirected to a new location in the apply phase. This can be accomplished with location modify rules.

Once you have verified which files and file types that the end users work with regularly, you will need to locate them. Files may be saved to a single folder or scattered across a drive. A good starting point for finding files types to include is to look at the registered file types on the computer.

To find the registered file types on a computer running Windows 7 or Windows 8

1. Click **Start**. Open **Control Panel**, click **Control Panel Home**, and click **Programs**.
2. Click **Default Programs**, and click **Associate a file type or protocol with a program**.
3. On this screen, the registered file types are displayed.

For more information about how to change the file types, files, and folders that are migrated when you specify the MigUser.xml file, see [User State Migration Tool \(USMT\) How-to topics](#).

Related topics

[Determine What to Migrate](#)

Test Your Migration

6/6/2019 • 2 minutes to read • [Edit Online](#)

Always test your migration plan in a controlled laboratory setting before you deploy it to your entire organization. In your test environment, you need at least one computer for each type of operating system from which you are migrating data.

After you have thoroughly tested the entire migration process on a single computer running each of your source operating systems, conduct a pilot migration with a small group of users. After migrating a few typical user states to the intermediate store, note the space required and adjust your initial calculations accordingly. For details about estimating the space needed for your migration, see [Estimate Migration Store Size](#). You might also need to adjust the registry-setting and file-location information in your migration-rule files. If you make changes, test the migration again. Then verify that all data and settings have migrated as expected. A pilot migration also gives you an opportunity to test your space estimates for the intermediate store.

If your test migration encounters any errors, examine the ScanState and LoadState logs to obtain the exact User State Migration Tool (USMT) 10.0 return code and associated error messages or Windows application programming interface (API) error message. For more information about USMT return codes and error messages, see [Return Codes](#). You can also obtain more information about a Windows API error message by typing **net helpmsg** and the error message number on the command line.

In most cases, the ScanState and LoadState logs indicate why a USMT migration is failing. We recommend that you use the **/v:5** option when testing your migration. This verbosity level can be adjusted in a production migration. Reducing the verbosity level might make it more difficult to diagnose failures that are encountered during production migrations. You can use a higher verbosity level if you want the log files output to go to a debugger.

Note Running the ScanState and LoadState tools with the **/v:5** option creates a detailed log file. Although this option makes the log file large, it is helpful in determining where migration errors occurred.

After you have determined that the pilot migration successfully migrated the specified files and settings, you are ready to add USMT to the server that is running Microsoft® System Center Configuration Manager (SCCM), or a non-Microsoft management technology. For more information, see [Configuration Manager](#).

Note For testing purposes, you can create an uncompressed store using the **/hardlink /nocompress** option. When compression is disabled, the ScanState tool saves the files and settings to a hidden folder named "File" at *StorePath*\USMT. You can use the uncompressed store to view what USMT has stored or to troubleshoot a problem, or you can run an antivirus utility against the files. Additionally, you can also use the **/listfiles** command-line option and the diagnostic log to list the files that were gathered and to troubleshoot problems with your migration.

Related topics

[Plan Your Migration](#)

[Log Files](#)

User State Migration Tool (USMT) Command-line Syntax

6/14/2019 • 2 minutes to read • [Edit Online](#)

The User State Migration Tool (USMT) 10.0 migrates user files and settings during large deployments of Windows. To improve and simplify the migration process, USMT captures desktop, network, and application settings in addition to a user's files. USMT then migrates these items to a new Windows installation.

In This Section

ScanState Syntax	Lists the command-line options for using the ScanState tool.
LoadState Syntax	Lists the command-line options for using the LoadState tool.
UsmtUtils Syntax	Lists the command-line options for using the UsmtUtils tool.

ScanState Syntax

6/6/2019 • 21 minutes to read • [Edit Online](#)

The ScanState command is used with the User State Migration Tool (USMT) 10.0 to scan the source computer, collect the files and settings, and create a store.

In This Topic

[Before You Begin](#)

[Syntax](#)

[Storage Options](#)

[Migration Rule Options](#)

[Monitoring Options](#)

[User Options](#)

[Encrypted File Options](#)

[Incompatible Command-Line Options](#)

Before You Begin

Before you run the **ScanState** command, note the following:

- To ensure that all operating system settings migrate, in most cases you must run the **ScanState** commands in administrator mode from an account with administrative credentials.
- If you encrypt the migration store, you will be required to enter an encryption key or a path to a file containing the encryption key. Be sure to make note of the key or the key file location, because this information is not kept anywhere in the migration store. You will need this information when you run the LoadState command to decrypt the migration store, or if you need to run the recovery utility. An incorrect or missing key or key file results in an error message.
- For information about software requirements for running the **ScanState** command, see [USMT Requirements](#).
- Unless otherwise noted, you can use each option only once when running a tool on the command line.
- You can gather domain accounts without the source computer having domain controller access. This functionality is available without any additional configuration.
- The [Incompatible Command-Line Options](#) table lists which options you can use together and which command-line options are incompatible.
- The directory location where you save the migration store will be excluded from the scan. For example, if you save the migration store to the root of the D drive, the D drive and all of its subdirectories will be excluded from the scan.

Syntax

This section explains the syntax and usage of the **ScanState** command-line options. The options can be

specified in any order. If the option contains a parameter, you can use either a colon or a space separator.

The **ScanState** command's syntax is:

```
scanstate [StorePath] [/apps] [/ppkg:FileName] [/i:[Path\]FileName] [/o] [/v:VerbosityLevel] [/nocompress]
[/localonly] [/encrypt /key:KeyString]/keyfile:[Path\]FileName] [/l:[Path\]FileName] [/progress:
[Path\]FileName] [/r:TimesToRetry] [/w:SecondsBeforeRetry] [/c] [/p] [/all] [/ui:
[DomainName|ComputerName\]UserName] [/ue:[DomainName|ComputerName\]UserName]
[/uel:NumberOfDays|YYYY/MM/DD|0] [/efs:abort|skip|decryptcopy|copyraw] [/genconfig:
[Path\]FileName[/config:[Path\]FileName] [/?|help]
```

For example:

To create a Config.xml file in the current directory, use:

```
scanstate /i:migapp.xml /i:migdocs.xml /genconfig:config.xml /v:13
```

To create an encrypted store using the Config.xml file and the default migration .xml files, use:

```
scanstate \\server\share\migration\mystore /i:migapp.xml /i:migdocs.xml /o /config:config.xml /v:13 /encrypt
/key: "mykey"
```

Storage Options

COMMAND-LINE OPTION	DESCRIPTION
<i>StorePath</i>	Indicates a folder where files and settings will be saved. Note that <i>StorePath</i> cannot be C:\ . You must specify the <i>StorePath</i> option in the ScanState command, except when using the /genconfig option. You cannot specify more than one <i>StorePath</i> location.
/apps	Scans the image for apps and includes them and their associated registry settings.
/ppkg [<FileName>]	Exports to a specific file location.
/o	Required to overwrite any existing data in the migration store or Config.xml file. If not specified, the ScanState command will fail if the migration store already contains data. You cannot use this option more than once on a command line.
/vsc	This option enables the volume shadow-copy service to migrate files that are locked or in use. This command-line option eliminates most file-locking errors that are typically encountered by the < ErrorControl > section. This option can be used only with the ScanState executable file and cannot be combined with the /hardlink option.
/hardlink	Enables the creation of a hard-link migration store at the specified location. The /nocompress option must be specified with the /hardlink option.

COMMAND-LINE OPTION	DESCRIPTION
<p>/encrypt [{/key:<KeyString> /keyfile:<file>}]</p>	<p>Encrypts the store with the specified key. Encryption is disabled by default. With this option, you will need to specify the encryption key in one of the following ways:</p> <ul style="list-style-type: none"> • /key:KeyString specifies the encryption key. If there is a space in <i>KeyString</i>, you will need to surround <i>KeyString</i> with quotation marks. • /keyfile:FilePathAndName specifies a text (.txt) file that contains the encryption key. <p>We recommend that <i>KeyString</i> be at least eight characters long, but it cannot exceed 256 characters. The /key and /keyfile options cannot be used on the same command line. The /encrypt and /nocompress options cannot be used on the same command line.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Important</p> <p>You should use caution with this option, because anyone who has access to the ScanState command-line script will also have access to the encryption key.</p> </div> <p>The following example shows the ScanState command and the /key option:</p> <pre style="border: 1px solid gray; padding: 5px; margin: 10px 0;">scanstate /i:migdocs.xml /i:migapp.xml \server\share\migration\mystore /encrypt /key:mykey</pre>
<p>/encrypt:<EncryptionStrength></p>	<p>The /encrypt option accepts a command-line parameter to define the encryption strength to be used for encryption of the migration store. For more information about supported encryption algorithms, see Migration Store Encryption.</p>
<p>/nocompress</p>	<p>Disables compression of data and saves the files to a hidden folder named "File" at <i>StorePath</i>\USMT. Compression is enabled by default. Combining the /nocompress option with the /hardlink option generates a hard-link migration store. You can use the uncompressed store to view what USMT stored, troubleshoot a problem, or run an antivirus utility against the files. You should use this option only in testing environments, because we recommend that you use a compressed store during your actual migration, unless you are combining the /nocompress option with the /hardlink option.</p> <p>The /nocompress and /encrypt options cannot be used together in one statement on the command line. However, if you do choose to migrate an uncompressed store, the LoadState command will migrate each file directly from the store to the correct location on the destination computer without a temporary location.</p> <p>For example:</p> <pre style="border: 1px solid gray; padding: 5px; margin: 10px 0;">scanstate /i:migdocs.xml /i:migapp.xml \server\share\migration\mystore /nocompress</pre>

Run the ScanState Command on an Offline Windows System

You can run the **ScanState** command in Windows Preinstallation Environment (WinPE). In addition, USMT supports migrations from previous installations of Windows contained in Windows.old directories. The offline directory can be a Windows directory when you run the **ScanState** command in WinPE or a Windows.old directory when you run the **ScanState** command in Windows.

There are several benefits to running the **ScanState** command on an offline Windows image, including:

- **Improved Performance.**

Because WinPE is a thin operating system, there are fewer running services. In this environment, the **ScanState** command has more access to the local hardware resources, enabling **ScanState** to perform migration operations more quickly.

- **Simplified end to end deployment process.**

Migrating data from Windows.old simplifies the end-to-end deployment process by enabling the migration process to occur after the new operating system is installed.

- **Improved success of migration.**

The migration success rate is increased because files will not be locked for editing while offline, and because WinPE provides administrator access to files in the offline Windows file system, eliminating the need for administrator-level access to the online system.

- **Ability to recover an unbootable computer.**

It might be possible to recover and migrate data from an unbootable computer.

Offline Migration Options

COMMAND-LINE OPTION	DEFINITION
/offline: "path to an offline.xml file"	This option is used to define a path to an offline .xml file that might specify other offline migration options, for example, an offline Windows directory or any domain or folder redirection required in your migration.
/offlinewindir: "path to a Windows directory"	This option specifies the offline Windows directory that the ScanState command gathers user state from. The offline directory can be Windows.old when you run the ScanState command in Windows or a Windows directory when you run the ScanState command in WinPE.
/offlinewinold: "Windows.old directory"	This command-line option enables the offline migration mode and starts the migration from the location specified. It is only intended to be used in Windows.old migration scenarios, where the migration is occurring from a Windows.old directory.

Migration Rule Options

USMT provides the following options to specify what files you want to migrate.

COMMAND-LINE OPTION	DESCRIPTION
<p><i>/i:[Path]FileName</i></p>	<p>(include)</p> <p>Specifies an .xml file that contains rules that define what user, application or system state to migrate. You can specify this option multiple times to include all of your .xml files (MigApp.xml, MigDocs.xml, and any custom .xml files that you create). <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> must be located in the current directory. For more information about which files to specify, see the "XML Files" section of the Frequently Asked Questions topic.</p>
<p><i>/genconfig:[Path]FileName</i></p>	<p>(Generate Config.xml)</p> <p>Generates the optional Config.xml file, but does not create a migration store. To ensure that this file contains every component, application and setting that can be migrated, you should create this file on a source computer that contains all the components, applications and settings that will be present on the destination computers. In addition, you should specify the other migration .xml files, using the <i>/i</i> option, when you specify this option.</p> <p>After you create this file, you will need to make use of it with the ScanState command using the <i>/config</i> option.</p> <p>The only options that you can specify with this option are the <i>/i</i>, <i>/v</i>, and <i>/l</i> options. You cannot specify <i>StorePath</i>, because the <i>/genconfig</i> option does not create a store. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> will be created in the current directory.</p> <p>Examples:</p> <ul style="list-style-type: none"> The following example creates a Config.xml file in the current directory: <pre data-bbox="874 1406 1355 1462">scanstate /i:migapp.xml /i:migdocs.xml /genconfig:config.xml /v:13</pre>

COMMAND-LINE OPTION	DESCRIPTION
<p>/config:<i>[Path]FileName</i></p>	<p>Specifies the Config.xml file that the ScanState command should use to create the store. You cannot use this option more than once on the command line. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> must be located in the current directory.</p> <p>The following example creates a store using the Config.xml file, MigDocs.xml, and MigApp.xml files:</p> <pre data-bbox="842 472 1396 555">scanstate \server\share\migration\mystore /config:config.xml /i:migdocs.xml /i:migapp.xml /v:13 /l:scan.log</pre> <p>The following example migrates the files and settings to the destination computer using the Config.xml, MigDocs.xml, and MigApp.xml files:</p> <pre data-bbox="842 689 1396 772">loadstate \server\share\migration\mystore /config:config.xml /i:migdocs.xml /i:migapp.xml /v:13 /l:load.log</pre>
<p>/auto:<i>path to script files</i></p>	<p>This option enables you to specify the location of the default .xml files and then begin the migration. If no path is specified, USMT will reference the directory where the USMT binaries are located. The /auto option has the same effect as using the following options: /i:MigDocs.xml /i:MigApp.xml /v:5.</p>
<p>/genmigxml:<i>path to a file</i></p>	<p>This option specifies that the ScanState command should use the document finder to create and export an .xml file that defines how to migrate all of the files on the computer on which the ScanState command is running.</p>
<p>/targetwindows8</p>	<p>Optimizes Scanstate.exe when using USMT 10.0 to migrate a user state to Windows 8 or Windows 8.1 instead of Windows 10. You should use this command line option in the following scenarios:</p> <ul data-bbox="842 1487 1342 1883" style="list-style-type: none"> • To create a Config.xml file by using the /genconfig option. Using the /targetwindows8 option optimizes the Config.xml file so that it only contains components that relate to Windows 8 or Windows 8.1. • To create a migration store. Using the /targetwindows8 option ensures that the ScanState tool gathers the correct set of operating system settings. Without the /targetwindows8 command-line option, some settings can be lost during the migration.

COMMAND-LINE OPTION	DESCRIPTION								
<p>/targetwindows7</p>	<p>Optimizes Scanstate.exe when using USMT 10.0 to migrate a user state to Windows 7 instead of Windows 10. You should use this command line option in the following scenarios:</p> <ul style="list-style-type: none"> • To create a Config.xml file by using the /genconfig option. Using the /targetwindows7 option optimizes the Config.xml file so that it only contains components that relate to Windows 7. • To create a migration store. Using the /targetwindows7 option ensures that the ScanState tool gathers the correct set of operating system settings. Without the /targetwindows7 command-line option, some settings can be lost during the migration. 								
<p>/localonly</p>	<p>Migrates only files that are stored on the local computer, regardless of the rules in the .xml files that you specify on the command line. You should use this option when you want to exclude the data from removable drives on the source computer, such as USB flash drives (UFDs), some external hard drives, and so on, and when there are network drives mapped on the source computer. If the /localonly option is not specified, then the ScanState command will copy files from these removable or network drives into the store.</p> <p>Anything that is not considered a fixed drive by the OS will be excluded by /localonly. In some cases large external hard drives are considered fixed drives. These drives can be explicitly excluded from migration by using a custom.xml file. For more information about how to exclude all files on a specific drive, see Exclude Files and Settings.</p> <p>The /localonly command-line option includes or excludes data in the migration as identified in the following table:</p> <table border="1" data-bbox="839 1462 1437 1827"> <thead> <tr> <th data-bbox="839 1462 1139 1529">DRIVE TYPE</th> <th data-bbox="1139 1462 1437 1529">BEHAVIOR WITH /LOCALONLY</th> </tr> </thead> <tbody> <tr> <td data-bbox="839 1529 1139 1648">Removable drives such as a USB flash drive</td> <td data-bbox="1139 1529 1437 1648">Excluded</td> </tr> <tr> <td data-bbox="839 1648 1139 1738">Network drives</td> <td data-bbox="1139 1648 1437 1738">Excluded</td> </tr> <tr> <td data-bbox="839 1738 1139 1827">Fixed drives</td> <td data-bbox="1139 1738 1437 1827">Included</td> </tr> </tbody> </table>	DRIVE TYPE	BEHAVIOR WITH /LOCALONLY	Removable drives such as a USB flash drive	Excluded	Network drives	Excluded	Fixed drives	Included
DRIVE TYPE	BEHAVIOR WITH /LOCALONLY								
Removable drives such as a USB flash drive	Excluded								
Network drives	Excluded								
Fixed drives	Included								

Monitoring Options

USMT provides several options that you can use to analyze problems that occur during migration.

Note

The ScanState log is created by default, but you can specify the name and location of the log with the **/l** option.

COMMAND-LINE OPTION	DESCRIPTION
/listfiles: < FileName >	You can use the /listfiles command-line option with the ScanState command to generate a text file that lists all of the files included in the migration.
/!:[Path]FileName	<p>Specifies the location and name of the ScanState log.</p> <p>You cannot store any of the log files in <i>StorePath</i>. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then the log will be created in the current directory. You can use the /v option to adjust the amount of output.</p> <p>If you run the ScanState or LoadState commands from a shared network resource, you must specify this option or USMT will fail with the following error: "USMT was unable to create the log file(s)". To fix this issue, use the /!:scan.log command.</p>

COMMAND-LINE OPTION

/v:< *VerbosityLevel* >

DESCRIPTION**(Verbosity)**

Enables verbose output in the ScanState log file. The default value is 0.

You can set the *VerbosityLevel* to one of the following levels:

LEVEL	EXPLANATION
0	Only the default errors and warnings are enabled.
1	Enables verbose output.
4	Enables error and status output.
5	Enables verbose and status output.
8	Enables error output to a debugger.
9	Enables verbose output to a debugger.
12	Enables error and status output to a debugger.
13	Enables verbose, status, and debugger output.

For example:

```
scanstate \server\share\migration\mystore /v:13 /i:migdocs.xml /i:migapp.xml
```

COMMAND-LINE OPTION	DESCRIPTION
<p>/progress:<i>[Path]FileName</i></p>	<p>Creates the optional progress log. You cannot store any of the log files in <i>StorePath</i>. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> will be created in the current directory.</p> <p>For example:</p> <pre data-bbox="842 376 1398 456">scanstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /progress:prog.log /l:scanlog.log</pre>
<p>/c</p>	<p>When this option is specified, the ScanState command will continue to run, even if non-fatal errors occur. Any files or settings that cause an error are logged in the progress log. For example, if there is a large file that will not fit in the store, the ScanState command will log an error and continue with the migration. In addition, if a file is open or in use by an application, USMT may not be able to migrate the file and will log an error. Without the /c option, the ScanState command will exit on the first error.</p> <p>You can use the new <ErrorControl> section in the Config.xml file to specify which file or registry read/write errors can be safely ignored and which might cause the migration to fail. This enables the /c command-line option to safely skip all input/output (I/O) errors in your environment. In addition, the /genconfig option now generates a sample <ErrorControl> section that is enabled by specifying error messages and desired behaviors in the Config.xml file.</p>
<p>/r:<<i>TimesToRetry</i>></p>	<p>(Retry)</p> <p>Specifies the number of times to retry when an error occurs while saving the user state to a server. The default is three times. This option is useful in environments where network connectivity is not reliable.</p> <p>While storing the user state, the /r option will not be able to recover data that is lost due to a network-hardware failure, such as a faulty or disconnected network cable, or when a virtual private network (VPN) connection fails. The retry option is intended for large, busy networks where connectivity is satisfactory, but communication latency is a problem.</p>
<p>/w:<<i>SecondsBeforeRetry</i>></p>	<p>(Wait)</p> <p>Specifies the time to wait, in seconds, before retrying a network file operation. The default is 1 second.</p>

COMMAND-LINE OPTION	DESCRIPTION
/p: <pathToFile>	<p>When the ScanState command runs, it will create an .xml file in the path specified. This .xml file includes improved space estimations for the migration store. The following example shows how to create this .xml file:</p> <pre>Scanstate.exe C:\MigrationLocation [additional parameters]</pre> <pre>/p: "C:\MigrationStoreSize.xml"</pre> <p>For more information, see Estimate Migration Store Size.</p> <p>To preserve the functionality of existing applications or scripts that require the previous behavior of USMT, you can use the /p option, without specifying "<i>pathToFile</i>", in USMT. If you specify only the /p option, the storage space estimations are created in the same manner as with USMT3.x releases.</p>
/? or /help	Displays Help at the command line.

User Options

By default, all users are migrated. The only way to specify which users to include and exclude is by using the following options. You cannot exclude users in the migration .xml files or using the Config.xml file. For more information, see [Identify Users](#) and [Migrate User Accounts](#).

COMMAND-LINE OPTION	DESCRIPTION
/all	<p>Migrates all of the users on the computer.</p> <p>USMT migrates all user accounts on the computer, unless you specifically exclude an account with either the /ue or /uel options. For this reason, you do not need to specify this option on the command line. However, if you choose to specify the /all option, you cannot also use the /ui, /ue or /uel options.</p>

COMMAND-LINE OPTION	DESCRIPTION
<pre data-bbox="199 181 552 208">/ui:<DomainName>\<UserName></pre> <p data-bbox="199 235 225 262">or</p> <pre data-bbox="199 284 624 311">/ui:<ComputerName>\<LocalUserName></pre>	<p data-bbox="837 181 991 208">(User include)</p> <p data-bbox="837 235 1398 515">Migrates the specified users. By default, all users are included in the migration. Therefore, this option is helpful only when used with the /ue or /uel options. You can specify multiple /ui options, but you cannot use the /ui option with the /all option. <i>DomainName</i> and <i>UserName</i> can contain the asterisk (*) wildcard character. When you specify a user name that contains spaces, you will need to surround it with quotation marks.</p> <div data-bbox="837 533 1437 725" style="border: 1px solid black; padding: 5px;"><p data-bbox="860 562 911 589">Note</p><p data-bbox="860 598 1374 687"><i>If a user is specified for inclusion with the /ui option, and also is specified to be excluded with either the /ue or /uel options, the user will be included in the migration.</i></p></div> <p data-bbox="837 754 970 781">For example:</p> <p data-bbox="837 804 1337 866">To include only User2 from the Fabrikam domain, type:</p> <pre data-bbox="837 887 1153 913" style="border: 1px solid gray; padding: 2px;">/ue:* * /ui:fabrikam\user2</pre> <p data-bbox="837 943 1351 1066">To migrate all users from the Fabrikam domain, and only the user accounts from other domains that have been active or otherwise modified in the last 30 days, type:</p> <pre data-bbox="837 1086 1107 1113" style="border: 1px solid gray; padding: 2px;">/uel:30 /ui:fabrikam *</pre> <p data-bbox="837 1142 1342 1232">In this example, a user account from the Contoso domain that was last modified 2 months ago will not be migrated.</p> <p data-bbox="837 1261 1394 1323">For more examples, see the descriptions of the /ue and /ui options in this table.</p>

COMMAND-LINE OPTION	DESCRIPTION
<p><code>/uel:<NumberOfDays></code></p> <p>or</p> <p><code>/uel:<YYYY/MM/DD></code></p> <p>or</p> <p><code>/uel:0</code></p>	<p>(User exclude based on last logon)</p> <p>Migrates the users that logged onto the source computer within the specified time period, based on the Last Modified date of the Ntuser.dat file on the source computer. The <code>/uel</code> option acts as an include rule. For example, the <code>/uel:30</code> option migrates users who logged on, or whose account was modified, within the last 30 days from the date when the ScanState command is run.</p> <p>You can specify a number of days or you can specify a date. You cannot use this option with the <code>/all</code> option. USMT retrieves the last logon information from the local computer, so the computer does not need to be connected to the network when you run this option. In addition, if a domain user has logged onto another computer, that logon instance is not considered by USMT.</p> <div data-bbox="842 779 1433 902" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>The <code>/uel</code> option is not valid in offline migrations.</p> </div> <ul style="list-style-type: none"> • <code>/uel:0</code> migrates any users who are currently logged on. • <code>/uel:90</code> migrates users who have logged on, or whose accounts have been otherwise modified, within the last 90 days. • <code>/uel:1</code> migrates users whose account has been modified within the last 24 hours. • <code>/uel:2002/1/15</code> migrates users who have logged on or been modified January 15, 2002 or afterwards. <p>For example:</p> <div data-bbox="842 1379 1394 1435" style="border: 1px solid black; padding: 5px;"> <pre>scanstate /i:migapp.xml /i:migdocs.xml \\server\share\migration\mystore /uel:0</pre> </div>
<p><code>/ue:<DomainName>\<UserName></code></p> <p>-or-</p> <p><code>/ue:<ComputerName>\<LocalUserName></code></p>	<p>(User exclude)</p> <p>Excludes the specified users from the migration. You can specify multiple <code>/ue</code> options. You cannot use this option with the <code>/all</code> option. <code><DomainName></code> and <code><UserName></code> can contain the asterisk (*) wildcard character. When you specify a user name that contains spaces, you need to surround it with quotation marks.</p> <p>For example:</p> <div data-bbox="842 1823 1394 1901" style="border: 1px solid black; padding: 5px;"> <pre>scanstate /i:migdocs.xml /i:migapp.xml \\server\share\migration\mystore /ue:contoso\user1</pre> </div>

How to Use /ui and /ue

The following examples apply to both the `/ui` and `/ue` options. You can replace the `/ue` option with the `/ui` option to include, rather than exclude, the specified users.

BEHAVIOR	COMMAND
Exclude the user named User One in the Fabrikam domain.	<code>/ue:"fabrikam\user one"</code>
Exclude the user named User1 in the Fabrikam domain.	<code>/ue:fabrikam\user1</code>
Exclude the local user named User1.	<code>/ue:%computername%\user1</code>
Exclude all domain users.	<code>/ue:Domain*</code>
Exclude all local users.	<code>/ue:%computername%*</code>
Exclude users in all domains named User1, User2, and so on.	<code>/ue:*\user*</code>

Using the Options Together

You can use the `/uel`, `/ue` and `/ui` options together to migrate only the users that you want migrated.

The `/ui` option has precedence over the `/ue` and `/uel` options. If a user is specified to be included using the `/ui` option, and also specified to be excluded using either the `/ue` or `/uel` options, the user will be included in the migration. For example, if you specify `/ui:contoso* /ue:contoso\user1`, then User1 will be migrated, because the `/ui` option takes precedence over the `/ue` option.

The `/uel` option takes precedence over the `/ue` option. If a user has logged on within the specified time period set by the `/uel` option, that user's profile will be migrated even if they are excluded by using the `/ue` option. For example, if you specify `/ue:fixed\user1 /uel:14`, the User1 will be migrated if they have logged on to the computer within the last 14 days.

BEHAVIOR	COMMAND
Include only User2 from the Fabrikam domain and exclude all other users.	<code>/ue:** /ui:fabrikam\user2</code>
Include only the local user named User1 and exclude all other users.	<code>/ue:** /ui:user1</code>
Include only the domain users from Contoso, except Contoso\User1.	<p>This behavior cannot be completed using a single command. Instead, to migrate this set of users, you will need to specify the following:</p> <ul style="list-style-type: none"> On the ScanState command line, type: <code>/ue:** /ui:contoso*</code> On the LoadState command line, type: <code>/ue:contoso\user1</code>

BEHAVIOR	COMMAND
Include only local (non-domain) users.	<code>/ue:* * /ui:%computername% *</code>

Encrypted File Options

You can use the following options to migrate encrypted files. In all cases, by default, USMT fails if an encrypted file is found unless you specify an **/efs** option. To migrate encrypted files, you must change the default behavior.

For more information, see [Migrate EFS Files and Certificates](#).

Note

EFS certificates will be migrated automatically when migrating to Windows 7, Windows 8 or Windows 10. Therefore, you should specify the **/efs:copyraw** option with the **ScanState** command to migrate the encrypted files

Caution

Take caution when migrating encrypted files. If you migrate an encrypted file without also migrating the certificate, end users will not be able to access the file after the migration.

COMMAND-LINE OPTION	EXPLANATION
/efs:hardlink	Creates a hard link to the EFS file instead of copying it. Use only with the /hardlink and the /nocompress options.
/efs:abort	Causes the ScanState command to fail with an error code, if an Encrypting File System (EFS) file is found on the source computer. Enabled by default.
/efs:skip	Causes the ScanState command to ignore EFS files.
/efs:decryptcopy	Causes the ScanState command to decrypt the file, if possible, before saving it to the migration store, and to fail if the file cannot be decrypted. If the ScanState command succeeds, the file will be unencrypted in the migration store, and once you run the LoadState command, the file will be copied to the destination computer.

COMMAND-LINE OPTION	EXPLANATION
/efs:copyraw	<p>Causes the ScanState command to copy the files in the encrypted format. The files will be inaccessible on the destination computer until the EFS certificates are migrated. EFS certificates will be automatically migrated; however, by default USMT fails if an encrypted file is found, unless you specify an /efs option. Therefore you should specify the /efs:copyraw option with the ScanState command to migrate the encrypted file. Then, when you run the LoadState command, the encrypted file and the EFS certificate will be automatically migrated.</p> <p>For example:</p> <pre>ScanState /i:migdocs.xml /i:migapp.xml \server\share\migration\mystore /efs:copyraw</pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Important</p> <p>All files must be encrypted if the parent folder is encrypted. If the encryption attribute on a file inside an encrypted folder has been removed, the file will be encrypted during the migration using the credentials of the account used to run the LoadState tool. For more information, see Migrate EFS Files and Certificates.</p> </div>

Incompatible Command-Line Options

The following table indicates which command-line options are not compatible with the **ScanState** command. If the table entry for a particular combination is blank, the options are compatible and you can use them together. The X symbol means that the options are not compatible. For example, you cannot use the **/nocompress** option with the **/encrypt** option.

COMMAND-LINE OPTION	/KEYFILE	/NOCOMPRESS	/GENCONFIG	/ALL
/i				
/o				
/v				
/nocompress			X	N/A
/localonly			X	
/key	X		X	
/encrypt	Required*	X	X	

COMMAND-LINE OPTION	/KEYFILE	/NOCOMPRESS	/GENCONFIG	/ALL
/keyfile	N/A		X	
/l				
/progress			X	
/r			X	
/w			X	
/c			X	
/p			X	N/A
/all			X	
/ui			X	X
/ue			X	X
/uel			X	X
/efs:<option>			X	
/genconfig			N/A	
/config			X	
<StorePath>			X	

Note

You must specify either the **/key** or **/keyfile** option with the **/encrypt** option.

Related topics

[XML Elements Library](#)

LoadState Syntax

6/6/2019 • 14 minutes to read • [Edit Online](#)

This topic discusses the **LoadState** command syntax and options.

In This Topic

[Before You Begin](#)

[Syntax](#)

[Storage Options](#)

[Migration Rule Options](#)

[Monitoring Options](#)

[User Options](#)

[Incompatible Command-Line Options](#)

Before You Begin

Before you run the **LoadState** command, note the following:

- To ensure that all operating system settings migrate, we recommend that you run the **LoadState** commands in administrator mode from an account with administrative credentials.
- For information about software requirements for running the **LoadState** command, see [USMT Requirements](#).
- You should log off after you run the **LoadState** command. Some settings (for example, fonts, wallpaper, and screensaver settings) will not take effect until the next time the user logs in.
- Unless otherwise specified, you can use each option only once when running a tool on the command line.
- **LoadState** does not require domain controller access to apply domain profiles. This functionality is available without any additional configuration. It is not necessary for the source computer to have had domain controller access when the user profile was gathered using **ScanState**. However, domain profiles are inaccessible until the destination computer is joined to the domain.
- The [Incompatible Command-Line Options](#) table lists which options you can use together and which command-line options are incompatible.

Syntax

This section explains the syntax and usage of the command-line options available when you use the **LoadState** command. The options can be specified in any order. If the option contains a parameter, you can specify either a colon or space separator.

The **LoadState** command's syntax is:

```
loadstate StorePath [/i:[Path]\FileName] [/v:VerbosityLevel] [/nocompress] [/decrypt /key:KeyString]/keyfile:  
[Path]\FileName] [/l:[Path]\FileName] [/progress:[Path]\FileName] [/r:TimesToRetry] [/w:SecondsToWait] [/c]  
[/all] [/ui:[DomainName\ComputerName\UserName] [/ue:[[DomainName\ComputerName\UserName]
```

```
[/uel:NumberOfDays|YYYY/MM/DD|0] [/md:OldDomain:NewDomain] [/mu:OldDomain\OldUserName:
[NewDomain\]NewUserName] [/lac:[Password]] [/lae] [/config:[Path\]FileName] [/?|help]
```

For example, to decrypt the store and migrate the files and settings to a computer running Windows 7 type the following on the command line:

```
loadstate \\server\share\migration\mystore /i:migapp.xml /i:migdocs.xml /v:13 /decrypt /key:"mykey"
```

Storage Options

USMT provides the following options that you can use to specify how and where the migrated data is stored.

COMMAND-LINE OPTION	DESCRIPTION
<i>StorePath</i>	Indicates the folder where the files and settings data are stored. You must specify <i>StorePath</i> when using the LoadState command. You cannot specify more than one <i>StorePath</i> .
<p>/decrypt /key:<i>KeyString</i></p> <p>or</p> <p>/decrypt /key:"<i>Key String</i>"</p> <p>or</p> <p>/decrypt /keyfile:[<i>Path</i>]<i>FileName</i></p>	<p>Decrypts the store with the specified key. With this option, you will need to specify the encryption key in one of the following ways:</p> <ul style="list-style-type: none"> • /key:<i>KeyString</i> specifies the encryption key. If there is a space in <i>KeyString</i>, you must surround the argument with quotation marks. • /keyfile:<i>FilePathAndName</i> specifies a text (.txt) file that contains the encryption key <p><i>KeyString</i> cannot exceed 256 characters.</p> <p>The /key and /keyfile options cannot be used on the same command line.</p> <p>The /decrypt and /nocompress options cannot be used on the same command line.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Important</p> <p>Use caution with this option, because anyone who has access to the LoadState command-line script will also have access to the encryption key.</p> </div> <p>For example:</p> <pre>loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /decrypt /key:mykey</pre>
/decrypt: " <i>encryption strength</i> "	The /decrypt option accepts a command-line parameter to define the encryption strength specified for the migration store encryption. For more information about supported encryption algorithms, see Migration Store Encryption .
/hardlink	Enables user-state data to be restored from a hard-link migration store. The /nocompress parameter must be specified with /hardlink option.

COMMAND-LINE OPTION	DESCRIPTION
/nocompress	<p>Specifies that the store is not compressed. You should only use this option in testing environments. We recommend that you use a compressed store during your actual migration. This option cannot be used with the /decrypt option.</p> <p>For example:</p> <pre>loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /nocompress</pre>

Migration Rule Options

USMT provides the following options to specify what files you want to migrate.

COMMAND-LINE OPTION	DESCRIPTION
/i:[Path]FileName	<p>(include)</p> <p>Specifies an .xml file that contains rules that define what state to migrate. You can specify this option multiple times to include all of your .xml files (MigApp.xml, MigSys.xml, MigDocs.xml and any custom .xml files that you create). <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> must be located in the current directory.</p> <p>For more information about which files to specify, see the "XML files" section of the Frequently Asked Questions topic.</p>
/config:[Path]FileName	<p>Specifies the Config.xml file that the LoadState command should use. You cannot specify this option more than once on the command line. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then the <i>FileName</i> must be located in the current directory.</p> <p>This example migrates the files and settings based on the rules in the Config.xml, MigDocs.xml, and MigApp.xml files:</p> <pre>loadstate \server\share\migration\mystore /config:config.xml /i:migdocs.xml /i:migapp.xml /v:5 /l:loadstate.log</pre>
/auto:"path to script files"	<p>This option enables you to specify the location of the default .xml files and then launch your migration. If no path is specified, USMT will use the directory where the USMT binaries are located. The /auto option has the same effect as using the following options: /i:MigDocs.xml /i:MigApp.xml /v:5.</p>

Monitoring Options

USMT provides several command-line options that you can use to analyze problems that occur during migration.

COMMAND-LINE OPTION	DESCRIPTION																		
<p>/l:[Path]FileName</p>	<p>Specifies the location and name of the LoadState log. You cannot store any of the log files in <i>StorePath</i>. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then the log will be created in the current directory. You can specify the /v option to adjust the amount of output.</p> <p>If you run the LoadState command from a shared network resource, you must specify this option or USMT will fail with the error: "USMT was unable to create the log file(s)". To fix this issue, use the /l:load.log option.</p>																		
<p>/v:<VerbosityLevel></p>	<p>(Verbosity)</p> <p>Enables verbose output in the LoadState log file. The default value is 0.</p> <p>You can set the <i>VerbosityLevel</i> to one of the following levels:</p> <table border="1" data-bbox="823 801 1434 1854"> <thead> <tr> <th data-bbox="823 801 1129 869">LEVEL</th> <th data-bbox="1129 801 1434 869">EXPLANATION</th> </tr> </thead> <tbody> <tr> <td data-bbox="823 869 1129 1016">0</td> <td data-bbox="1129 869 1434 1016">Only the default errors and warnings are enabled.</td> </tr> <tr> <td data-bbox="823 1016 1129 1106">1</td> <td data-bbox="1129 1016 1434 1106">Enables verbose output.</td> </tr> <tr> <td data-bbox="823 1106 1129 1227">4</td> <td data-bbox="1129 1106 1434 1227">Enables error and status output.</td> </tr> <tr> <td data-bbox="823 1227 1129 1348">5</td> <td data-bbox="1129 1227 1434 1348">Enables verbose and status output.</td> </tr> <tr> <td data-bbox="823 1348 1129 1469">8</td> <td data-bbox="1129 1348 1434 1469">Enables error output to a debugger.</td> </tr> <tr> <td data-bbox="823 1469 1129 1590">9</td> <td data-bbox="1129 1469 1434 1590">Enables verbose output to a debugger.</td> </tr> <tr> <td data-bbox="823 1590 1129 1738">12</td> <td data-bbox="1129 1590 1434 1738">Enables error and status output to a debugger.</td> </tr> <tr> <td data-bbox="823 1738 1129 1854">13</td> <td data-bbox="1129 1738 1434 1854">Enables verbose, status, and debugger output.</td> </tr> </tbody> </table> <p>For example:</p> <pre data-bbox="823 1921 1394 1980">loadstate \server\share\migration\mystore /v:5 /i:migdocs.xml /i:migapp.xml</pre>	LEVEL	EXPLANATION	0	Only the default errors and warnings are enabled.	1	Enables verbose output.	4	Enables error and status output.	5	Enables verbose and status output.	8	Enables error output to a debugger.	9	Enables verbose output to a debugger.	12	Enables error and status output to a debugger.	13	Enables verbose, status, and debugger output.
LEVEL	EXPLANATION																		
0	Only the default errors and warnings are enabled.																		
1	Enables verbose output.																		
4	Enables error and status output.																		
5	Enables verbose and status output.																		
8	Enables error output to a debugger.																		
9	Enables verbose output to a debugger.																		
12	Enables error and status output to a debugger.																		
13	Enables verbose, status, and debugger output.																		

COMMAND-LINE OPTION	DESCRIPTION
/progress: <i>[Path]FileName</i>	<p>Creates the optional progress log. You cannot store any of the log files in <i>StorePath</i>. <i>Path</i> can be either a relative or full path. If you do not specify the <i>Path</i> variable, then <i>FileName</i> will be created in the current directory.</p> <p>For example:</p> <pre>loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /progress:prog.log /l:scanlog.log</pre>
/c	<p>When this option is specified, the LoadState command will continue to run, even if non-fatal errors occur. Any files or settings that cause an error are logged in the progress log. For example, if there is a large file that will not fit on the computer, the LoadState command will log an error and continue with the migration. Without the /c option, the LoadState command will exit on the first error. You can use the new <ErrorControl> section in the Config.xml file to specify which file or registry read/write errors can be safely ignored and which might cause the migration to fail. This enables the /c command-line option to safely skip all input/output (I/O) errors in your environment. In addition, the /genconfig option now generates a sample <ErrorControl> section that is enabled by specifying error messages and desired behaviors in the Config.xml file.</p>
/r: <i><TimesToRetry></i>	<p>(Retry)</p> <p>Specifies the number of times to retry when an error occurs while migrating the user state from a server. The default is three times. This option is useful in environments where network connectivity is not reliable.</p> <p>While restoring the user state, the /r option will not recover data that is lost due to a network-hardware failure, such as a faulty or disconnected network cable, or when a virtual private network (VPN) connection fails. The retry option is intended for large, busy networks where connectivity is satisfactory, but communication latency is a problem.</p>
/w: <i><SecondsBeforeRetry></i>	<p>(Wait)</p> <p>Specifies the time to wait, in seconds, before retrying a network file operation. The default is 1 second.</p>
/? or /help	Displays Help on the command line.

User Options

By default, all users are migrated. The only way to specify which users to include and exclude is by using the following options. You cannot exclude users in the migration .xml files or by using the Config.xml file. For more information, see [Identify Users](#).

COMMAND-LINE OPTION	DESCRIPTION
<p>/all</p>	<p>Migrates all of the users on the computer.</p> <p>USMT migrates all user accounts on the computer, unless you specifically exclude an account with the /ue or /uel options. For this reason, you do not need to specify this option on the command line. However, if you choose to use the /all option, you cannot also use the /ui, /ue or /uel options.</p>
<p>/ui:DomainNameUserName</p> <p>or</p> <p>/ui:"DomainNameUser Name"</p> <p>or</p> <p>/ui:ComputerNameLocalUserName</p>	<p>(User include)</p> <p>Migrates the specified user. By default, all users are included in the migration. Therefore, this option is helpful only when used with the /ue option. You can specify multiple /ui options, but you cannot use the /ui option with the /all option. <i>DomainName</i> and <i>UserName</i> can contain the asterisk (*) wildcard character. When you specify a user name that contains spaces, you will need to surround it with quotations marks.</p> <p><i>For example:</i></p> <ul style="list-style-type: none"> To include only User2 from the Corporate domain, type: <pre data-bbox="858 949 919 981" style="border: 1px solid gray; padding: 2px;">/ue:</pre> <pre data-bbox="823 1010 1027 1041">* /ui:corporate\user2</pre> <div data-bbox="823 1059 1434 1281" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note</p> <p>If a user is specified for inclusion with the /ui option, and also is specified to be excluded with either the /ue or /uel options, the user will be included in the migration.</p> </div> <p>For more examples, see the descriptions of the /uel, /ue, and /ui options in this table.</p>

COMMAND-LINE OPTION	DESCRIPTION
<p><code>/uel:<NumberOfDays></code></p> <p>or</p> <p><code>/uel:<YYYY/MM/DD></code></p> <p>or</p> <p><code>/uel:0</code></p>	<p>(User exclude based on last logon)</p> <p>Migrates only the users that logged onto the source computer within the specified time period, based on the Last Modified date of the Ntuser.dat file on the source computer. The <code>/uel</code> option acts as an include rule. For example, the <code>/uel:30</code> option migrates users who logged on, or whose user account was modified, within the last 30 days from the date when the ScanState command is run. You can specify a number of days or you can specify a date. You cannot use this option with the <code>/all</code> option. USMT retrieves the last logon information from the local computer, so the computer does not need to be connected to the network when you run this option. In addition, if a domain user has logged onto another computer, that logon instance is not considered by USMT.</p> <div data-bbox="826 696 1434 819" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>The <code>/uel</code> option is not valid in offline migrations.</p> </div> <p>Examples:</p> <ul style="list-style-type: none"> • <code>/uel:0</code> migrates accounts that were logged on to the source computer when the ScanState command was run. • <code>/uel:90</code> migrates users who have logged on, or whose accounts have been otherwise modified, within the last 90 days. • <code>/uel:1</code> migrates users whose accounts have been modified within the last 24 hours. • <code>/uel:2002/1/15</code> migrates users who have logged on or whose accounts have been modified since January 15, 2002. <p>For example:</p> <div data-bbox="826 1391 1394 1451" style="border: 1px solid black; padding: 5px;"> <pre>loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /uel:0</pre> </div>
<p><code>/ue:DomainNameUserName</code></p> <p>or</p> <p><code>/ue:"DomainNameUser Name"</code></p> <p>or</p> <p><code>/ue:ComputerNameLocalUserName</code></p>	<p>(User exclude)</p> <p>Excludes the specified users from the migration. You can specify multiple <code>/ue</code> options but you cannot use the <code>/ue</code> option with the <code>/all</code> option. <i>DomainName</i> and <i>UserName</i> can contain the asterisk () <i>wildcard character</i>. <i>When you specify a user name that contains spaces, you will need to surround it with quotation marks.</i></p> <p>For example:</p> <div data-bbox="826 1839 1394 1899" style="border: 1px solid black; padding: 5px;"> <pre>Loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /ue:contoso\user1</pre> </div> <p>For more examples, see the descriptions of the <code>/uel</code>, <code>/ue</code>, and <code>/ui</code> options in this table.</p>

COMMAND-LINE OPTION	DESCRIPTION
<p>/md:<i>OldDomain:NewDomain</i></p> <p>or</p> <p>/md:<i>LocalComputerName:NewDomain</i></p>	<p>(move domain)</p> <p>Specifies a new domain for the user. Use this option to change the domain for users on a computer or to migrate a local user to a domain account. <i>OldDomain</i> may contain the asterisk (*) wildcard character.</p> <p>You can specify this option more than once. You may want to specify multiple /md options if you are consolidating users across multiple domains to a single domain. For example, you could specify the following to consolidate the users from the Corporate and FarNorth domains into the Fabrikam domain:</p> <pre data-bbox="826 568 1136 633">/md:corporate:fabrikam and /md:farnorth:fabrikam .</pre> <p>If there are conflicts between two /md commands, the first rule that you specify is applied. For example, if you specify the <code>/md:corporate:fabrikam</code> and <code>/md:corporate:farnorth</code> commands, then Corporate users would be mapped to the Fabrikam domain.</p> <div data-bbox="826 842 1434 1189" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>If you specify an <i>OldDomain</i> that did not exist on the source computer, the LoadState command will appear to complete successfully, without an error or warning. However, in this case, users will not be moved to <i>NewDomain</i> but will remain in their original domain. For example, if you misspell "contoso" and you specify <code>/md:contso:fabrikam</code>, the users will remain in contoso on the destination computer.</p> </div> <p>For example:</p> <pre data-bbox="826 1267 1394 1402">loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /progress:prog.log /l:load.log /md:contoso:fabrikam</pre>
<p>/mu:<i>OldDomain</i>OldUserName: [<i>NewDomain</i>]<i>NewUserName</i></p> <p>or</p> <p>/mu:<i>OldLocalUserName:NewDomain</i> NewUserNa me</p>	<p>Specifies a new user name for the specified user. If the store contains more than one user, you can specify multiple /mu options. You cannot use wildcard characters with this option.</p> <p>For example:</p> <pre data-bbox="826 1675 1394 1809">loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore /progress:prog.log /l:load.log /mu:contoso\user1:fabrikam\user1</pre>

COMMAND-LINE OPTION	DESCRIPTION
<p>/lac:<i>[Password]</i></p>	<p>(local account create)</p> <p>Specifies that if a user account is a local (non-domain) account, and it does not exist on the destination computer, USMT will create the account on the destination computer but it will be disabled. To enable the account, you must also use the /lae option.</p> <p>If the /lac option is not specified, any local user accounts that do not already exist on the destination computer will not be migrated.</p> <p><i>Password</i> is the password for the newly created account. An empty password is used by default.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Caution</p> <p>Use the <i>Password</i> variable with caution because it is provided in plain text and can be obtained by anyone with access to the computer that is running the LoadState command.</p> <p>Also, if the computer has multiple users, all migrated users will have the same password.</p> </div> <p>For example:</p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore</pre> <p>For instructions, see Migrate User Accounts.</p>
<p>/lae</p>	<p>(local account enable)</p> <p>Enables the account that was created with the /lac option. You must specify the /lac option with this option.</p> <p>For example:</p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">loadstate /i:migapp.xml /i:migdocs.xml \server\share\migration\mystore</pre> <pre style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">/progress:prog.log /l:load.log /lac:password /lae</pre> <p>For instructions, see Migrate User Accounts.</p>

Examples for the /ui and /ue options

The following examples apply to both the **/ui** and **/ue** options. You can replace the **/ue** option with the **/ui** option to include, rather than exclude, the specified users.

BEHAVIOR	COMMAND
<p>Exclude the user named User One in the Corporate domain.</p>	<pre style="border: 1px solid #ccc; padding: 5px;">/ue:"corporate\user one"</pre>
<p>Exclude the user named User1 in the Corporate domain.</p>	<pre style="border: 1px solid #ccc; padding: 5px;">/ue:corporate\user1</pre>
<p>Exclude the local user named User1.</p>	<pre style="border: 1px solid #ccc; padding: 5px;">/ue:%computername%\user1</pre>

BEHAVIOR	COMMAND
Exclude all domain users.	<code>/ue:Domain</code>
Exclude all local users.	<code>/ue:%computername%</code>
Exclude users in all domains named User1, User2, and so on.	<code>/ue:\user</code>

Using the Options Together

You can use the **/uel**, **/ue** and **/ui** options together to migrate only the users that you want migrated.

The /ui option has precedence over the /ue and /uel options. If a user is specified to be included using the **/ui** option, and also specified to be excluded using either the **/ue** or **/uel** options, the user will be included in the migration. For example, if you specify `/ui:contoso*` `/ue:contoso\user1`, then User1 will be migrated, because the **/ui** option takes precedence over the **/ue** option.

The /uel option takes precedence over the /ue option. If a user has logged on within the specified time period set by the **/uel** option, that user's profile will be migrated even if they are excluded by using the **/ue** option. For example, if you specify `/ue:contoso\user1` `/uel:14`, the User1 will be migrated if they have logged on to the computer within the last 14 days.

BEHAVIOR	COMMAND
Include only User2 from the Fabrikam domain and exclude all other users.	<code>/ue:* /ui:fabrikam\user2</code>
Include only the local user named User1 and exclude all other users.	<code>/ue:* /ui:user1</code>
Include only the domain users from Contoso, except Contoso\User1.	<p>This behavior cannot be completed using a single command. Instead, to migrate this set of users, you will need to specify the following:</p> <ul style="list-style-type: none"> Using the ScanState command-line tool, type: <code>/ue:* /ui:contoso</code> Using the LoadState command-line tool, type: <code>/ue:contoso\user1</code>
Include only local (non-domain) users.	<code>/ue: /ui:%computername%*</code>

Incompatible Command-Line Options

The following table indicates which command-line options are not compatible with the **LoadState** command. If the table entry for a particular combination is blank, the options are compatible and you can use them together. The X symbol means that the options are not compatible. For example, you cannot use the **/nocompress** option with the **/encrypt** option.

COMMAND-LINE OPTION	/KEYFILE	/NOCOMPRESS	/GENCONFIG	/ALL
/i				
/v				
/nocompress		N/A	X	
/key	X		X	
/decrypt	Required*	X	X	
/keyfile	N/A		X	
/l				
/progress			X	
/r			X	
/w			X	
/c			X	
/p			X	N/A
/all			X	
/ui			X	X
/ue			X	X
/uel			X	X
/genconfig			N/A	
/config			X	
<i>StorePath</i>				

COMMAND-LINE OPTION	/KEYFILE	/NOCOMPRESS	/GENCONFIG	/ALL
/md				
/mu				
/lae				
/lac				

Note

You must specify either the **/key** or **/keyfile** option with the **/encrypt** option.

Related topics

[XML Elements Library](#)

UsmtUtils Syntax

6/6/2019 • 6 minutes to read • [Edit Online](#)

This topic describes the syntax for the utilities available in User State Migration Tool (USMT) 10.0 through the command-line interface. These utilities:

- Improve your ability to determine cryptographic options for your migration.
- Assist in removing hard-link stores that cannot otherwise be deleted due to a sharing lock.
- Verify whether the catalog file or any of the other files in the compressed migration store have become corrupted.
- Extract files from the compressed migration store when you migrate files and settings to the destination computer.

In This Topic

[Usmtutils.exe](#)

[Verify Options](#)

[Extract Options](#)

Usmtutils.exe

The following table lists command-line options for USMTutils.exe. The sections that follow provide further command-line options for the **/verify** and the **/extract** options.

The syntax for UsmtUtils.exe is:

```
usmtutils [/ec | /rd <storeDir> | /verify <filepath> [options] | /extract <filepath> <destinationPath> [options]]
```

COMMAND-LINE OPTION	DESCRIPTION
/ec	Returns a list of supported cryptographic algorithms (AlgIDs) on the current system. You can use this on a destination computer to determine which algorithm to use with the /encrypt command before you run the ScanState tool on the source computer.
/rd<storeDir>	Removes the directory path specified by the <storeDir> argument on the computer. You can use this command to delete hard-link migration stores that cannot otherwise be deleted at a command prompt due to a sharing lock. If the migration store spans multiple volumes on a given drive, it will be deleted from all of these volumes. For example: <pre>usmtutils /rd D:\MyHardLinkStore</pre>

COMMAND-LINE OPTION	DESCRIPTION
/y	Overrides the accept deletions prompt when used with the /rd option. When you use the /y option with the /rd option, you will not be prompted to accept the deletions before USMT deletes the directories.
/verify	Returns information on whether the compressed migration store is intact or whether it contains corrupted files or a corrupted catalog. See Verify Options for syntax and options to use with /verify .
/extract	Recovers files from a compressed USMT migration store. See Extract Options for syntax and options to use with /extract .

Verify Options

Use the **/verify** option when you want to determine whether a compressed migration store is intact or whether it contains corrupted files or a corrupted catalog. For more information on how to use the **/verify** option, see [Verify the Condition of a Compressed Migration Store](#).

The syntax for **/verify** is:

```
usmtutils /verify[:<reportType>] <filePath> [/l:<logfile>] [/v:VerbosityLevel] [/decrypt[:<AlgID>]
{/key:<keystring> | /keyfile:<filename>}]
```

COMMAND-LINE OPTION	DESCRIPTION
---------------------	-------------

COMMAND-LINE OPTION	DESCRIPTION
<p><i><reportType></i></p>	<p>Specifies whether to report on all files, corrupted files only, or the status of the catalog.</p> <ul style="list-style-type: none"> • Summary. Returns both the number of files that are intact and the number of files that are corrupted in the migration store. If no algorithm is specified, the summary report is displayed as a default. • all. Returns a tab-delimited list of all of the files in the compressed migration store and the status for each file. Each line contains the file name followed by a tab spacing, and either "CORRUPTED" or "OK" depending on the status of the file. The last entry reports the corruption status of the "CATALOG" of the store. A catalog file contains metadata for all files in a migration store. The LoadState tool requires a valid catalog file in order to open the migration store. Returns "OK" if the catalog file is intact and LoadState can open the migration store and "CORRUPTED" if the migration store is corrupted. • failureonly. Returns a tab-delimited list of only the files that are corrupted in the compressed migration store. • Catalog. Returns only the status of the catalog file.
<p>/l: <i><logfilePath></i></p>	<p>Specifies the location and name of the log file.</p>

COMMAND-LINE OPTION

/v: < *VerbosityLevel* >

DESCRIPTION**(Verbosity)**

Enables verbose output in the UsmtUtils log file. The default value is 0.

You can set the *VerbosityLevel* to one of the following levels:

LEVEL	EXPLANATION
0	Only the default errors and warnings are enabled.
1	Enables verbose output.
4	Enables error and status output.
5	Enables verbose and status output.
8	Enables error output to a debugger.
9	Enables verbose output to a debugger.
12	Enables error and status output to a debugger.
13	Enables verbose, status, and debugger output.

COMMAND-LINE OPTION	DESCRIPTION
<pre>/decrypt<AlgID>/:<KeyString></pre> <p>or</p> <pre>/decrypt<AlgID>/:<"Key String"></pre> <p>or</p> <pre>/decrypt:<AlgID>/keyfile:<FileName></pre>	<p>Specifies that the /encrypt option was used to create the migration store with the ScanState tool. To decrypt the migration store, specify a /key or /keyfile option as follows:</p> <ul style="list-style-type: none"> • <AlgID> specifies the cryptographic algorithm that was used to create the migration store on the ScanState command line. If no algorithm is specified, ScanState and UsmtUtils use the 3DES algorithm as a default. <AlgID> valid values include: AES_128, AES_192, AES_256, 3DES, or 3DES_112. • /key:<KeyString> specifies the encryption key. If there is a space in <KeyString>, you must surround the argument with quotation marks. • /keyfile: <FileName> specifies the location and name of a text (.txt) file that contains the encryption key. <p>For more information about supported encryption algorithms, see Migration Store Encryption</p>

Some examples of **/verify** commands:

- `usmtutils /verify D:\MyMigrationStore\store.mig`
- `usmtutils /verify:catalog D:\MyMigrationStore\store.mig`
- `usmtutils /verify:all D:\MyMigrationStore\store.mig /decrypt /l:D:\UsmtUtilsLog.txt`
- `usmtutils /verify:failureonly D:\MyMigrationStore\store.mig /decrypt:AES_192 /keyfile:D:\encryptionKey.txt`

Extract Options

Use the **/extract** option to recover files from a compressed USMT migration store if it will not restore normally with loadstate. For more information on how to use the **/extract** option, see [Extract Files from a Compressed USMT Migration Store](#).

The syntax for **/extract** is:

```
/extract <filePath> <destinationPath> [/i:<includePattern>] [/e: <excludePattern>] [/l: <logfile>] [/v: VerboseLevel>] [/decrypt:<AlgID>] {key: <keystring> | /keyfile: <filename>}} [/o]
```

COMMAND-LINE OPTION	DESCRIPTION
<filePath>	<p>Path to the USMT migration store.</p> <p>For example:</p> <pre>D:\MyMigrationStore\USMT\store.mig</pre>
<destinationPath>	<p>Path to the folder where the tool puts the individual files.</p>

COMMAND-LINE OPTION	DESCRIPTION
<i>/i:</i> <includePattern>	Specifies a pattern for files to include in the extraction. You can specify more than one pattern. Separate patterns with a comma or a semicolon. You can use <i>/i:</i> <includePattern> and <i>/e:</i> <excludePattern> options in the same command. When both include and exclude patterns are used on the command line, include patterns take precedence over exclude patterns.
<i>/e:</i> <excludePattern>	Specifies a pattern for files to omit from the extraction. You can specify more than one pattern. Separate patterns with a comma or a semicolon. You can use <i>/i:</i> <includePattern> and <i>/e:</i> <excludePattern> options in the same command. When both include and exclude patterns are used on the command line, include patterns take precedence over exclude patterns.
<i>/l:</i> <logfilePath>	Specifies the location and name of the log file.

COMMAND-LINE OPTION

/v: < *VerbosityLevel* >

DESCRIPTION**(Verbosity)**

Enables verbose output in the UsmtUtils log file. The default value is 0.

You can set the *VerbosityLevel* to one of the following levels:

LEVEL	EXPLANATION
0	Only the default errors and warnings are enabled.
1	Enables verbose output.
4	Enables error and status output.
5	Enables verbose and status output.
8	Enables error output to a debugger.
9	Enables verbose output to a debugger.
12	Enables error and status output to a debugger.
13	Enables verbose, status, and debugger output.

COMMAND-LINE OPTION	DESCRIPTION
<p>/decrypt<AlgID>/key:<KeyString></p> <p>or</p> <p>/decrypt<AlgID>/:<"Key String"></p> <p>or</p> <p>/decrypt:<AlgID>/keyfile:<FileName></p>	<p>Specifies that the /encrypt option was used to create the migration store with the ScanState tool. To decrypt the migration store, you must also specify a /key or /keyfile option as follows:</p> <ul style="list-style-type: none"> • <AlgID> specifies the cryptographic algorithm that was used to create the migration store on the ScanState command line. If no algorithm is specified, ScanState and UsmtUtils use the 3DES algorithm as a default. <p><AlgID> valid values include: AES_128, AES_192, AES_256, 3DES, or 3DES_112.</p> <ul style="list-style-type: none"> • /key: <KeyString> specifies the encryption key. If there is a space in <KeyString>, you must surround the argument with quotation marks. • /keyfile:<FileName> specifies a text (.txt) file that contains the encryption key <p>For more information about supported encryption algorithms, see Migration Store Encryption.</p>
/o	Overwrites existing output files.

Some examples of **/extract** commands:

- `usmtutils /extract D:\MyMigrationStore\USMT\store.mig C:\ExtractedStore`
- `usmtutils /extract D:\MyMigrationStore\USMT\store.mig /i:"*.txt, *.pdf" C:\ExtractedStore /decrypt /keyfile:D:\encryptionKey.txt`
- `usmtutils /extract D:\MyMigrationStore\USMT\store.mig /e:*.exe C:\ExtractedStore /decrypt:AES_128 /key:password /l:C:\usmtlog.txt`
- `usmtutils /extract D:\MyMigrationStore\USMT\store.mig /i:myProject.* /e:*.exe C:\ExtractedStore /o`

Related topics

[User State Migration Tool \(USMT\) Command-line Syntax](#)

[Return Codes](#)

USMT XML Reference

6/6/2019 • 2 minutes to read • [Edit Online](#)

This section contains topics that you can use to work with and to customize the migration XML files.

In This Section

Understanding Migration XML Files	Provides an overview of the default and custom migration XML files and includes guidelines for creating and editing a customized version of the MigDocs.xml file.
Config.xml File	Describes the Config.xml file and policies concerning its configuration.
Customize USMT XML Files	Describes how to customize USMT XML files.
Custom XML Examples	Gives examples of XML files for various migration scenarios.
Conflicts and Precedence	Describes the precedence of migration rules and how conflicts are handled.
General Conventions	Describes the XML helper functions.
XML File Requirements	Describes the requirements for custom XML files.
Recognized Environment Variables	Describes environment variables recognized by USMT.
XML Elements Library	Describes the XML elements and helper functions for authoring migration XML files to use with USMT.

Understanding Migration XML Files

6/14/2019 • 13 minutes to read • [Edit Online](#)

You can modify the behavior of a basic User State Migration Tool (USMT)10.0 migration by using XML files; these files provide instructions on where and how the USMT tools should gather and apply files and settings. USMT includes three XML files that you can use to customize a basic migration: the MigDocs.xml and MigUser.xml files, which modify how files are discovered on the source computer, and the MigApps.xml file, which is required in order to migrate supported application settings. You can also create and edit custom XML files and a Config.xml file to further customize your migration.

This topic provides an overview of the default and custom migration XML files and includes guidelines for creating and editing a customized version of the MigDocs.xml file. The MigDocs.xml file uses the new **GenerateDocPatterns** function available in USMT to automatically find user documents on a source computer.

In This Topic

[Overview of the Config.xml file](#)

[Overview of the MigApp.xml file](#)

[Overview of the MigDocs.xml file](#)

[Overview of the MigUser.xml file](#)

[Using multiple XML files](#)

[XML rules for migrating user files](#)

[The GenerateDocPatterns function](#)

[Understanding the system and user context](#)

[Sample migration rules for customized versions of XML files](#)

[Exclude rules usage examples](#)

[Include rules usage examples](#)

[Next Steps](#)

Overview of the Config.xml file

The Config.xml file is the configuration file created by the `/genconfig` option of the ScanState tool; it can be used to modify which operating-system components are migrated by USMT. The Config.xml file can be used in conjunction with other XML files, such as in the following example:

```
scanstate /i:migapps.xml /i:migdocs.xml /genconfig:c:\myFolder\config.xml
```

. When used this way, the Config.xml file tightly controls aspects of the migration, including user profiles, data, and settings, without modifying or creating other XML files. For more information about the Config.xml file, see [Customize USMT XML Files and Config.xml File](#).

Note When modifying the XML elements in the Config.xml file, you should edit an element and set the **migrate** property to **no**, rather than deleting the element from the file. If you delete the element instead of setting the property, the component may still be migrated by rules in other XML files.

Overview of the MigApp.xml file

The MigApp.xml file installed with USMT includes instructions to migrate the settings for the applications listed in [What Does USMT Migrate?](#). You must include the MigApp.xml file when using the ScanState and LoadState tools, by using the `/i` option in order to migrate application settings. The MigDocs.xml and MigUser.xml files do not migrate application settings. You can create a custom XML file to include additional applications. For more information, see [Customize USMT XML Files](#).

Important The MigApps.xml file will only detect and migrate .pst files that are linked to Microsoft Office Outlook. See the [Sample migration rules for customized versions of XML files](#) section of this document for more information about migrating .pst files that are not linked to Outlook.

Overview of the MigDocs.xml file

The MigDocs.xml file uses the new **GenerateDocPatterns** helper function to create instructions for USMT to migrate files from the source computer, based on the location of the files. You can use the MigDocs.xml file with the ScanState and LoadState tools to perform a more targeted migration than using USMT without XML instructions.

The default MigDocs.xml file migrates the following:

- All files on the root of the drive except %WINDIR%, %PROGRAMFILES%, %PROGRAMDATA%, or %USERS%.
- All folders in the root directory of all fixed drives. For example: c:\data_mail\[*]
- All files from the root of the Profiles folder, except for files in the system profile. For example: c:\users\name[mail.pst]
- All folders from the root of the Profiles folder, except for the system-profile folders. For example: c:\users\name\new folder\[*]
- Standard shared folders:
 - CSIDL_COMMON_DESKTOPDIRECTORY
 - CSIDL_COMMON_FAVORITES
 - CSIDL_COMMON_DOCUMENTS
 - CSIDL_COMMON_MUSIC
 - CSIDL_COMMON_PICTURES
 - CSIDL_COMMON_VIDEO
 - FOLDERID_PublicDownloads
- Standard user-profile folders for each user:
 - CSIDL_MYDOCUMENTS
 - CSIDL_MYPICTURES
 - FOLDERID_OriginalImages
 - CSIDL_MYMUSIC
 - CSIDL_MYVIDEO
 - CSIDL_FAVORITES
 - CSIDL_DESKTOP

- CSIDL_QUICKLAUNCH
- FOLDERID_Contacts
- FOLDERID_Libraries
- FOLDERID_Downloads
- FOLDERID_SavedGames
- FOLDERID_RecordedTV

The default MigDocs.xml file will not migrate the following:

- Files tagged with both the **hidden** and **system** attributes.
- Files and folders on removable drives.
- Data from the %WINDIR%, %PROGRAMDATA%, and %PROGRAMFILES% folders.
- Folders that contain installed applications.

You can also use the **/genmigxml** option with the ScanState tool to review and modify what files will be migrated.

Overview of the MigUser.xml file

The MigUser.xml file includes instructions for USMT to migrate user files based on file name extensions. You can use the MigUser.xml file with the ScanState and LoadState tools to perform a more targeted migration than using USMT without XML instructions. The MigUser.xml file will gather all files from the standard user-profile folders, as well as any files on the computer with the specified file name extensions.

The default MigUser.xml file migrates the following:

- All files from the standard user-profile folders which are described as:
 - CSIDL_MYVIDEO
 - CSIDL_MYMUSIC
 - CSIDL_DESKTOP
 - CSIDL_STARTMENU
 - CSIDL_PERSONAL
 - CSIDL_MYPICTURES
 - CSIDL_FAVORITES
 - CSIDL_QUICK LAUNCH
- Files with the following extensions:
 - .qdf, .qsd, .qel, .qph, .doc*, .dot*, .rtf, .mcw, .wps, .scd, .wri, .wpd, .xl*, .csv, .iqy, .dqy, .oqy, .rqy, .wk*, .wq1, .slk, .dif, .ppt*, .pps*, .pot*, .sh3, .ch3, .pre, .ppa, .txt, .pst, .one*, .vl*, .vsd, .mpp, .or6, .accdb, .mdb, .pub

The default MigUser.xml file does not migrate the following:

- Files tagged with both the **hidden** and **system** attributes.
- Files and folders on removable drives,
- Data from the %WINDIR%, %PROGRAMFILES%, %PROGRAMDATA% folders.

- ACLS for files in folders outside the user profile.

You can make a copy of the MigUser.xml file and modify it to include or exclude standard user-profile folders and file name extensions. If you know all of the extensions for the files you want to migrate from the source computer, use the MigUser.xml file to move all of your relevant data, regardless of the location of the files. However, this may result in a migration that contains more files than intended. For example, if you choose to migrate all .jpg files, you may migrate image files such as thumbnails and logos from legacy applications that are installed on the source computer.

Note Each file name extension you include in the rules within the MigUser.xml file increases the amount of time needed for the ScanState tool to gather the files for the migration. If you are migrating more than three hundred file types, you may experience a slow migration. For more information about other ways to organize the migration of your data, see the [Using multiple XML files](#) section of this document.

Using multiple XML files

You can use multiple XML files with the ScanState and LoadState tools. Each of the default XML files included with or generated by USMT is configured for a specific component of the migration. You can also use custom XML files to supplement these default files with additional migration rules.

XML MIGRATION FILE	MODIFIES THE FOLLOWING COMPONENTS:
Config.xml file	Operating-system components such as desktop wallpaper and background theme. You can also overload config.xml to include some application and document settings by generating the config.xml file with the other default XML files. For more information, see Customize USMT XML Files and Config.xml File .
MigApps.xml file	Applications settings.
MigUser.xml or MigDocs.xml files	User files and profile settings.
Custom XML files	Application settings, user profile settings, or user files, beyond the rules contained in the other XML files.

For example, you can use all of the XML migration file types for a single migration, as in the following example:

```
Scanstate <store> /config:c:\myFolder\config.xml /i:migapps.xml /i:migdocs.xml /i:customrules.xml
```

XML rules for migrating user files

Important You should not use the MigUser.xml and MigDocs.xml files together in the same command. Using both XML files can result in duplication of some migrated files. This occurs when conflicting target-location instructions are given in each XML file. The target file will be stored once during the migration, but will be applied by each XML file to a different location on the destination computer.

If your data set is unknown or if many files are stored outside of the standard user-profile folders, the MigDocs.xml is a better choice than the MigUser.xml file, because the MigDocs.xml file will gather a broader scope of data. The MigDocs.xml file migrates folders of data based on location. The MigUser.xml file migrates only the files with the specified file name extensions.

If you want more control over the migration, you can create custom XML files. See the [Creating and editing a custom ,xml file](#) section of this document.

Creating and editing a custom XML file

You can use the `/genmigxml` command-line option to determine which files will be included in your migration. The `/genmigxml` option creates a file in a location you specify, so that you can review the XML rules and make modifications as necessary.

Note If you reinstall USMT, the default migration XML files will be overwritten and any customizations you make directly to these files will be lost. Consider creating separate XML files for your custom migration rules and saving them in a secure location.

To generate the XML migration rules file for a source computer:

1. Click **Start**, click **All Programs**, click **Accessories**, right-click **Command Prompt**, and then click **Run as**.
2. Select an account with administrator privileges, supply a password, and then click **OK**.
3. At the command prompt, type:

```
cd /d <USMTpath>
scanstate.exe /genmigxml: <filepath.xml>
```

Where `<USMTpath>` is the location on your source computer where you have saved the USMT files and tools, and `<filepath.xml>` is the full path to a file where you can save the report. For example, type:

```
cd /d c:\USMT
scanstate.exe /genmigxml:"C:\Documents and Settings\USMT Tester\Desktop\genMig.xml"
```

The GenerateDocPatterns function

The MigDocs.xml file calls the **GenerateDocPatterns** function, which takes three Boolean values. You can change the settings to modify the way the MigDocs.xml file generates the XML rules for migration.

SETTING	VALUE	DEFAULT VALUE
---------	-------	---------------

SETTING	VALUE	DEFAULT VALUE
ScanProgramFiles	<p>The <i>ScanProgramFiles</i> argument is valid only when the GenerateDocPatterns function is called in a system context. This argument determines whether or not to scan the Program Files directory to gather registered file name extensions for known applications.</p> <p>For example, when set to TRUE, the function discovers and migrates .doc files under the Microsoft Office directory, because .doc is a file name extension registered to a Microsoft Office application. The GenerateDocPatterns function generates this inclusion pattern for .doc files:</p> <pre><pattern type="File">C:\Program Files\Microsoft Office[.doc] </pattern></pre> <p>If a child folder of an included folder contains an installed application, ScanProgramFiles will also create an exclusion rule for the child folder. All folders under the application folder will be scanned recursively for registered file name extensions.</p>	False
IncludePatterns	<p>The <i>IncludePatterns</i> argument determines whether to generate exclude or include patterns in the XML. When this argument is set to TRUE, the GenerateDocPatterns function generates include patterns and the function must be added under the <include> element. Changing this argument to FALSE generates exclude patterns and the function must be added under the <exclude> element.</p>	True
SystemDrive	<p>The <i>SystemDrive</i> argument determines whether to generate patterns for all fixed drives or only for the system drive. Changing this argument to TRUE restricts all patterns to the system drive.</p>	False

Usage:

```
MigXmlHelper.GenerateDocPatterns ("<ScanProgramFiles>", "<IncludePatterns>", "<SystemDrive>")
```

To create include data patterns for only the system drive:

```
<include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <script>MigXmlHelper.GenerateDocPatterns ("FALSE","TRUE","TRUE")</script>
  </objectSet>
</include>
```

To create an include rule to gather files for registered extensions from the %PROGRAMFILES% directory:

```
<include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <script>MigXmlHelper.GenerateDocPatterns ("TRUE","TRUE","FALSE")</script>
  </objectSet>
</include>
```

To create exclude data patterns:

```
<exclude filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <script>MigXmlHelper.GenerateDocPatterns ("FALSE","FALSE","FALSE")</script>
  </objectSet>
</exclude>
```

Understanding the system and user context

The migration XML files contain two <component> elements with different **context** settings. The system context applies to files on the computer that are not stored in the User Profiles directory, while the user context applies to files that are particular to an individual user.

System context

The system context includes rules for data outside of the User Profiles directory. For example, when called in a system context in the MigDocs.xml file, the **GenerateDocPatterns** function creates patterns for all common shell folders, files in the root directory of hard drives, and folders located at the root of hard drives. The following folders are included:

- CSIDL_COMMON_DESKTOPDIRECTORY
- CSIDL_COMMON_FAVORITES
- CSIDL_COMMON_DOCUMENTS
- CSIDL_COMMON_MUSIC
- CSIDL_COMMON_PICTURES
- CSIDL_COMMON_VIDEO
- FOLDERID_PublicDownloads

User context

The user context includes rules for data in the User Profiles directory. When called in a user context in the MigDocs.xml file, the **GenerateDocPatterns** function creates patterns for all user shell folders, files located at the root of the profile, and folders located at the root of the profile. The following folders are included:

- CSIDL_MYDOCUMENTS
- CSIDL_MYPICTURES

- FOLDERID_OriginalImages
- CSIDL_MYMUSIC
- CSIDL_MYVIDEO
- CSIDL_FAVORITES
- CSIDL_DESKTOP
- CSIDL_QUICKLAUNCH
- FOLDERID_Contacts
- FOLDERID_Libraries
- FOLDERID_Downloads
- FOLDERID_SavedGames
- FOLDERID_RecordedTV

Note Rules contained in a component that is assigned the user context will be run for each user profile on the computer. Files that are scanned multiple times by the MigDocs.xml files will only be copied to the migration store once; however, a large number of rules in the user context can slow down the migration. Use the system context when it is applicable.

Sample migration rules for customized versions of XML files

Note For best practices and requirements for customized XML files in USMT, see [Customize USMT XML Files](#) and [General Conventions](#).

Exclude rules usage examples

In the examples below, the source computer has a .txt file called "new text document" in a directory called "new folder". The default MigDocs.xml behavior migrates the new text document.txt file and all files contained in the "new folder" directory. The rules generated by the function are:

Rule 1	<pre><pattern type="File">d:\new folder[new text document.txt]</pattern></pre>
Rule 2	<pre><pattern type="File">d:\new folder[*]</pattern></pre>

To exclude the new text document.txt file as well as any .txt files in "new folder", you can do the following:

Example 1: Exclude all .txt files in a folder

To exclude Rule 1, there needs to be an exact match of the file name. However, for Rule 2, you can create a pattern to exclude files by using the file name extension.

```
<exclude>
  <objectSet>
    <pattern type="File">D:\Newfolder\[new text document.txt]</pattern>
    <pattern type="File">D:\New folder\[*.txt]</pattern>
  </objectSet>
</exclude>
```

Example 2: Use the `UnconditionalExclude` element to give a rule precedence over include rules

If you do not know the file name or location of the file, but you do know the file name extension, you can use the **GenerateDrivePatterns** function. However, the rule will be less specific than the default include rule generated by the MigDocs.xml file, so it will not have precedence. You must use the `<UnconditionalExclude>` element to give this rule precedence over the default include rule. For more information about the order of precedence for XML migration rules, see [Conflicts and Precedence](#).

```
<unconditionalExclude>
  <objectSet>
    <script>MigXmlHelper.GenerateDrivePatterns ("*[*].txt", "Fixed")</script>
  </objectSet>
</unconditionalExclude>
```

Example 3 : Use a `UserandSystem` context component to run rules in both contexts

If you want the `<UnconditionalExclude>` element to apply to both the system and user context, you can create a third component using the **UserandSystem** context. Rules in this component will be run in both contexts.

```
<component type="Documents" context="UserandSystem">
  <displayName>MigDocExcludes</displayName>
  <role role="Data">
    <rules>
      <unconditionalExclude>
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("*[*].txt", "Fixed")</script>
        </objectSet>
      </unconditionalExclude>
    </rules>
  </role>
</component>
```

For more examples of exclude rules that you can use in custom migration XML files, see [Exclude Files and Settings](#).

Include rules usage examples

The application data directory is the most common location that you would need to add an include rule for. The **GenerateDocPatterns** function excludes this location by default. If your company uses an application that saves important data to this location, you can create include rules to migrate the data. For example, the default location for .pst files is: `%CSIDL_LOCAL_APPDATA%\Microsoft\Outlook`. The Migapp.xml file contains migration rules to move only those .pst files that are linked to Microsoft Outlook. To include .pst files that are not linked, you can do the following:

Example 1: Include a file name extension in a known user folder

This rule will include .pst files that are located in the default location, but are not linked to Microsoft Outlook. Use the user context to run this rule for each user on the computer.

```
<include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <pattern type="File">%CSIDL_LOCAL_APPDATA%\Microsoft\Outlook\*[*].pst</pattern>
  </objectSet>
</include>
```

Example 2: Include a file name extension in Program Files

For locations outside the user profile, such as the Program Files folder, you can add the rule to the system context component.

```
<include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <pattern type="File">%CSIDL_PROGRAM_FILES%\*[*].pst</pattern>
  </objectSet>
</include>
```

For more examples of include rules that you can use in custom migration XML files, see [Include Files and Settings](#).

Note For more information about the order of precedence for XML migration rules, see [Conflicts and Precedence](#).

Next steps

You can include additional rules for the migration in the MigDocs.xml file or other XML migration files. For example, you can use the <locationModify> element to move files from the folder where they were gathered to a different folder, when they are applied to the destination computer.

You can use an XML schema (MigXML.xsd) file to validate the syntax of your customized XML files. For more information, see [USMT Resources](#).

Related topics

[Exclude Files and Settings](#)

[Include Files and Settings](#)

Config.xml File

6/26/2019 • 9 minutes to read • [Edit Online](#)

Config.xml File

The Config.xml file is an optional User State Migration Tool (USMT) 10.0 file that you can create using the **/genconfig** option with the ScanState.exe tool. If you want to include all of the default components, and do not want to change the default store-creation or profile-migration behavior, you do not need to create a Config.xml file.

However, if you are satisfied with the default migration behavior defined in the MigApp.xml, MigUser.xml and MigDocs.xml files, but you want to exclude certain components, you can create and modify a Config.xml file and leave the other .xml files unchanged. For example, you must create and modify the Config.xml file if you want to exclude any of the operating-system settings that are migrated. It is necessary to create and modify this file if you want to change any of the default store-creation or profile-migration behavior.

The Config.xml file has a different format than the other migration .xml files, because it does not contain any migration rules. It contains only a list of the operating-system components, applications, user documents that can be migrated, as well as user-profile policy and error-control policy. For this reason, excluding components using the Config.xml file is easier than modifying the migration .xml files, because you do not need to be familiar with the migration rules and syntax. However, you cannot use wildcard characters in this file.

For more information about using the Config.xml file with other migration files, such as the MigDocs.xml and MigApps.xml files, see [Understanding Migration XML Files](#).

Note To exclude a component from the Config.xml file, set the **migrate** value to **"no"**. Deleting the XML tag for the component from the Config.xml file will not exclude the component from your migration.

In This Topic

In USMT there are new migration policies that can be configured in the Config.xml file. For example, you can configure additional **<ErrorControl>**, **<ProfileControl>**, and **<HardLinkStoreControl>** options. The following elements and parameters are for use in the Config.xml file only.

[<Policies>](#)

[<ErrorControl>](#)

[<fatal>](#)

[<fileError>](#)

[<nonfatal>](#)

[<registryError>](#)

[<HardLinkStoreControl>](#)

[<fileLocked>](#)

[<createHardLink>](#)

[<errorHardLink>](#)

[<ProfileControl>](#)

<localGroups>

< mappings>

<changeGroup>

<include>

<exclude>

[Sample Config.xml File](#)

<Policies>

The **<Policies>** element contains elements that describe the policies that USMT follows while creating a migration store. Valid children of the **<Policies>** element are **<ErrorControl>** and **<HardLinkStoreControl>**. The **<Policies>** element is a child of **<Configuration>**.

Syntax: `<Policies> </Policies>`

<ErrorControl>

The **<ErrorControl>** element is an optional element you can configure in the Config.xml file. The configurable **<ErrorControl>** rules support only the environment variables for the operating system that is running and the currently logged-on user. As a workaround, you can specify a path using the (*) wildcard character.

- **Number of occurrences:** Once for each component
- **Parent elements:** The **<Policies>** element
- **Child elements:** The **<fileError>** and **<registryError>** element

Syntax: `<ErrorControl></ErrorControl>`

The following example specifies that all locked files, regardless of their location (including files in C:\Users), should be ignored. However, the migration fails if any file in C:\Users cannot be accessed because of any other reason. In the example below, the **<ErrorControl>** element ignores any problems in migrating registry keys that match the supplied pattern, and it resolves them to an **Access denied** error.

Additionally, the order in the **<ErrorControl>** section implies priority. In this example, the first **<nonFatal>** tag takes precedence over the second **<fatal>** tag. This precedence is applied, regardless of how many tags are listed.

```
<ErrorControl>
  <fileError>
    <nonFatal errorCode="33">* [*]</nonFatal>
    <fatal errorCode="any">C:\Users\* [*]</fatal>
  </fileError>
  <registryError>
    <nonFatal errorCode="5">HKCU\SOFTWARE\Microsoft\* [*]</nonFatal>
  </registryError>
</ErrorControl>
```

Important The configurable **<ErrorControl>** rules support only the environment variables for the operating system that is running and the currently logged-on user. As a workaround, you can specify a path using the (*) wildcard character.

<fatal>

The **<fatal>** element is not required.

- **Number of occurrences:** Once for each component
- **Parent elements:** `<fileError>` and `<registryError>`
- **Child elements:** None.

Syntax: `<fatal errorCode="any"> <pattern> </fatal>`

PARAMETER	REQUIRED	VALUE
errorCode	No	"any" or "specify system error message here"

You use the `<fatal>` element to specify that errors matching a specific pattern should cause USMT to halt the migration.

`<fileError>`

The `<fileError>` element is not required.

- **Number of occurrences:** Once for each component
- **Parent elements:** `<ErrorControl>`
- **Child elements:** `<nonFatal>` and `<fatal>`

Syntax: `<fileError></fileError>`

You use the `<fileError>` element to represent the behavior associated with file errors.

`<nonFatal>`

The `<nonFatal>` element is not required.

- **Number of occurrences:** Once for each component
- **Parent elements:** The `<fileError>` and `<registryError>` elements.
- **Child elements:** None.

Syntax: `<nonfatal errorCode="any"> <pattern> </nonFatal>`

PARAMETER	REQUIRED	VALUE
<code><errorCode></code>	No	"any" or "specify system error message here". If system error messages are not specified, the default behavior applies the parameter to all system error messages.

You use the `<nonFatal>` element to specify that errors matching a specific pattern should not cause USMT to halt the migration.

`<registryError>`

The `<registryError>` element is not required.

- **Number of occurrences:** Once for each component
- **Parent elements:** `<ErrorControl>`
- **Child elements:** `<nonfatal>` and `<fatal>`

Syntax: `<registryError></registryError>`

PARAMETER	REQUIRED	VALUE
<code><errorCode></code>	No	"any" or " <i>specify system error message here</i> ". If system error messages are not specified, the default behavior applies the parameter to all system error messages.

You use the `<registryError>` element to specify that errors matching a specific pattern should not cause USMT to halt the migration.

<HardLinkStoreControl>

The `<HardLinkStoreControl>` element contains elements that describe how to handle files during the creation of a hard-link migration store. Its only valid child is `<fileLocked>`.

Syntax: `<HardLinkStoreControl> </HardLinkStoreControl>`

- **Number of occurrences:** Once for each component
- **Parent elements:** `<Policies>`
- **Child elements:** `<fileLocked>`

Syntax: `<HardLinkStoreControl></HardLinkStoreControl>`

The `<HardLinkStoreControl>` sample code below specifies that hard links can be created to locked files only if the locked file resides somewhere under C:\Users\. Otherwise, a file-access error occurs when a locked file is encountered that cannot be copied, even though is technically possible for the link to be created.

Important The `<ErrorControl>` section can be configured to conditionally ignore file access errors, based on the file's location.

```
<Policy>
  <HardLinkStoreControl>
    <fileLocked>
      <createHardLink>C:\Users\*</createHardLink>
      <errorHardLink>C:\*</errorHardLink>
    </fileLocked>
  </HardLinkStoreControl>
  <ErrorControl>
    [...]
  </ErrorControl>
</Policy>
```

<fileLocked>

The `<fileLocked>` element contains elements that describe how to handle files that are locked for editing. The rules defined by the `<fileLocked>` element are processed in the order in which they appear in the XML file.

Syntax: `<fileLocked></fileLocked>`

<createHardLink>

The **<createHardLink>** element defines a standard MigXML pattern that describes file paths where hard links should be created, even if the file is locked for editing by another application.

Syntax: `<createHardLink> <pattern> </createHardLink>`

<errorHardLink>

The **<errorHardLink>** element defines a standard MigXML pattern that describes file paths where hard links should not be created if the file is locked for editing by another application. USMT will attempt to copy files under these paths into the migration store. However, if that is not possible, **Error_Locked** is thrown. This is a standard Windows application programming interface (API) error that can be captured by the **<ErrorControl>** section to either cause USMT to skip the file or abort the migration.

Syntax: `<errorHardLink> <pattern> </errorHardLink>`

<ProfileControl>

This element is used to contain other elements that establish rules for migrating profiles, users, and policies around local group membership during the migration. **<ProfileMigration>** is a child of **<Configuration>**.

Syntax: `< ProfileControl> </ProfileControl>`

<localGroups>

This element is used to contain other elements that establish rules for how to migrate local groups. **<localGroups>** is a child of **<ProfileControl>**.

Syntax: `<localGroups> </localGroups>`

<mappings>

This element is used to contain other elements that establish mappings between groups.

Syntax: `<mappings> </mappings>`

<changeGroup>

This element describes the source and destination groups for a local group membership change during the migration. It is a child of **<localGroups>**. The following parameters are defined:

PARAMETER	REQUIRED	VALUE
From	Yes	A valid local group on the source machine that contains users selected for migration on the command line.
To	Yes	A local group that the users are to be moved to during the migration.

PARAMETER	REQUIRED	VALUE
appliesTo	Yes	nonmigratedUsers, migratedUsers, AllUsers. This value defines which users the change group operation should apply to.

The valid and required children of **<changeGroup>** are **<include>** and **<exclude>**. Although both can be children at the same time, only one is required.

Syntax: `<changeGroup From="Group1" To= "Group2"> </changeGroup>`

<include>

This element specifies that its required child, *<pattern>*, should be included in the migration.

Syntax: `<include>` `</include>`

<exclude>

This element specifies that its required child, *<pattern>*, should be excluded from the migration.

Syntax: `<exclude>` `</exclude>`

Sample Config.xml File

Refer to the following sample Config.xml file for additional details about items you can choose to exclude from a migration.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <Applications/>
  <Documents/>
  <WindowsComponents>
    <component displayname="Tablet PC Settings" migrate="yes" ID="tablet_pc_settings">
      <component displayname="Accessories" migrate="yes" ID="tablet_pc_settings\tablet_pc_accessories">
        <component displayname="Microsoft-Windows-TabletPC-StickyNotes" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-stickynotes/microsoft-
windows-tabletpc-stickynotes/settings"/>
        <component displayname="Microsoft-Windows-TabletPC-SnippingTool" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-snippingtool/microsoft-
windows-tabletpc-snippingtool/settings"/>
        <component displayname="Microsoft-Windows-TabletPC-Journal" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-journal/microsoft-
windows-tabletpc-journal/settings"/>
      </component>
      <component displayname="Input Panel" migrate="yes" ID="tablet_pc_settings\tablet_pc_input_panel">
        <component displayname="Microsoft-Windows-TabletPC-InputPanel" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-inputpanel/microsoft-
windows-tabletpc-inputpanel/settings"/>
      </component>
      <component displayname="General Options" migrate="yes"
ID="tablet_pc_settings\tablet_pc_general_options">
        <component displayname="Microsoft-Windows-TabletPC-UIHub" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-uihub/microsoft-windows-
tabletpc-uihub/settings"/>
        <component displayname="Microsoft-Windows-TabletPC-Platform-Input-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-platform-input-
core/microsoft-windows-tabletpc-platform-input-core/settings"/>
      </component>
      <component displayname="Handwriting Recognition" migrate="yes">
```

```
ID="tablet_pc_settings\handwriting_recognition">
  <component displayname="Microsoft-Windows-TabletPC-InputPersonalization" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabletpc-
inputpersonalization/microsoft-windows-tabletpc-inputpersonalization/settings"/>
  </component>
</component>
  <component displayname="Sound and Speech Recognition" migrate="yes" ID="sound_and_speech_recognition">
  <component displayname="Speech Recognition" migrate="yes"
ID="sound_and_speech_recognition\speech_recognition">
  <component displayname="Microsoft-Windows-SpeechCommon" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-speechcommon/microsoft-windows-
speechcommon/settings"/>
  </component>
</component>
  <component displayname="Hardware" migrate="yes" ID="hardware">
  <component displayname="Phone and Modem" migrate="yes" ID="hardware\phone_and_modem">
  <component displayname="Microsoft-Windows-TapiSetup" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tapisetup/microsoft-windows-
tapisetup/settings"/>
  </component>
  <component displayname="Printers and Faxes" migrate="yes" ID="hardware\printers_and_faxes">
  <component displayname="Microsoft-Windows-Printing-Spooler-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-printing-spooler-core/microsoft-
windows-printing-spooler-core/settings"/>
  <component displayname="Microsoft-Windows-Printing-Spooler-NetworkClient" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-printing-spooler-
networkclient/microsoft-windows-printing-spooler-networkclient/settings"/>
  <component displayname="Microsoft-Windows-Printing-Spooler-Core-LocalSpl" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-printing-spooler-core-
localspl/microsoft-windows-printing-spooler-core-localspl/settings"/>
  </component>
</component>
  <component displayname="Programs" migrate="yes" ID="programs">
  <component displayname="Media Player Settings" migrate="yes" ID="programs\media_player_settings">
  <component displayname="Microsoft-Windows-MediaPlayer-Migration" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-mediaplayer-migration/microsoft-
windows-mediaplayer-migration/settings"/>
  </component>
</component>
  <component displayname="Communications and Sync" migrate="yes" ID="communications_and_sync">
  <component displayname="Windows Mail" migrate="yes" ID="communications_and_sync\windows_mail">
  <component displayname="Microsoft-Windows-WAB" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-wab/microsoft-windows-
wab/settings"/>
  <component displayname="Microsoft-Windows-Mail" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-mail/microsoft-windows-
mail/settings"/>
  </component>
</component>
  <component displayname="Microsoft-Windows-Migration-DisplayGroups" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-migration-displaygroups/microsoft-
windows-migration-displaygroups/settings"/>
  <component displayname="Performance and Maintenance" migrate="yes" ID="performance_and_maintenance">
  <component displayname="Diagnostics" migrate="yes" ID="performance_and_maintenance\diagnostics">
  <component displayname="Microsoft-Windows-RemoteAssistance-Exe" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-remoteassistance-exe/microsoft-
windows-remoteassistance-exe/settings"/>
  <component displayname="Microsoft-Windows-Feedback-Service" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-feedback-service/microsoft-
windows-feedback-service/settings"/>
  </component>
  <component displayname="Error Reporting" migrate="yes"
ID="performance_and_maintenance\error_reporting">
  <component displayname="Microsoft-Windows-ErrorReportingCore" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-errorreportingcore/microsoft-
windows-errorreportingcore/settings"/>
  </component>
</component>
  <component displayname="Network and Internet" migrate="yes" ID="network_and_internet">
```

```
<component displayname="Offline Files" migrate="yes" ID="network_and_internet\offline_files">
  <component displayname="Microsoft-Windows-OfflineFiles-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-offlinefiles-core/microsoft-
windows-offlinefiles-core/settings"/>
  </component>
  <component displayname="Internet Options" migrate="yes" ID="network_and_internet\internet_options">
    <component displayname="Microsoft-Windows-ieframe" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ieframe/microsoft-windows-
ieframe/settings"/>
    <component displayname="Microsoft-Windows-IE-InternetExplorer" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ie-internetexplorer/microsoft-
windows-ie-internetexplorer/settings"/>
    <component displayname="Microsoft-Windows-IE-Feeds-Platform" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ie-feeds-platform/microsoft-
windows-ie-feeds-platform/settings"/>
    <component displayname="Microsoft-Windows-IE-ClientNetworkProtocolImplementation" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ie-
clientnetworkprotocolimplementation/microsoft-windows-ie-clientnetworkprotocolimplementation/settings"/>
    </component>
    <component displayname="Networking Connections" migrate="yes"
ID="network_and_internet\networking_connections">
      <component displayname="Microsoft-Windows-Wlansvc" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-wlansvc/microsoft-windows-
wlansvc/settings"/>
      <component displayname="Microsoft-Windows-RasConnectionManager" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rasconnectionmanager/microsoft-
windows-rasconnectionmanager/settings"/>
      <component displayname="Microsoft-Windows-RasApi" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rasapi/microsoft-windows-
rasapi/settings"/>
      <component displayname="Microsoft-Windows-PeerToPeerCollab" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-peertopeercollab/microsoft-
windows-peertopeercollab/settings"/>
      <component displayname="Microsoft-Windows-MPR" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-mpr/microsoft-windows-
mpr/settings"/>
      <component displayname="Microsoft-Windows-Dot3svc" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-dot3svc/microsoft-windows-
dot3svc/settings"/>
    </component>
  </component>
  <component displayname="Date, Time, Language and Region" migrate="yes"
ID="date_time_language_and_region">
    <component displayname="Regional Language Options" migrate="yes"
ID="date_time_language_and_region\regional_language_options">
      <component displayname="Microsoft-Windows-TableDrivenTextService-Migration" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-tabledriventextservice-
migration/microsoft-windows-tabledriventextservice-migration/settings"/>
      <component displayname="Microsoft-Windows-TextServicesFramework-Migration" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-textservicesframework-
migration/microsoft-windows-textservicesframework-migration/settings"/>
      <component displayname="Microsoft-Windows-MUI-Settings" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-mui-settings/microsoft-windows-
mui-settings/settings"/>
      <component displayname="Microsoft-Windows-International-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-international-core/microsoft-
windows-international-core/settings"/>
      <component displayname="Microsoft-Windows-IME-Traditional-Chinese-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ime-traditional-chinese-
core/microsoft-windows-ime-traditional-chinese-core/settings"/>
      <component displayname="Microsoft-Windows-IME-Simplified-Chinese-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-ime-simplified-chinese-
core/microsoft-windows-ime-simplified-chinese-core/settings"/>
      <component displayname="Microsoft-Windows-Desktop_Technologies-Text_Input_Services-IME-Japanese-Core"
migrate="yes" ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-
desktop_technologies-text_input_services-ime-japanese-core/microsoft-windows-desktop_technologies-
text_input_services-ime-japanese-core/settings"/>
    </component>
  </component>
</component>
```

```
<component displayname="Security" migrate="yes" ID="security">
  <component displayname="Microsoft-Windows-Rights-Management-Client-v1-API" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-rights-management-client-v1-
api/microsoft-windows-rights-management-client-v1-api/settings"/>
  <component displayname="Security Options" migrate="yes" ID="security\security_options">
    <component displayname="Microsoft-Windows-Credential-Manager" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-credential-manager/microsoft-
windows-credential-manager/settings"/>
  </component>
</component>
<component displayname="Appearance and Display" migrate="yes" ID="appearance_and_display">
  <component displayname="Windows Games Settings" migrate="yes"
ID="appearance_and_display\windows_games_settings">
    <component displayname="Microsoft-Windows-GameExplorer" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-gameexplorer/microsoft-windows-
gameexplorer/settings"/>
  </component>
  <component displayname="Taskbar and Start Menu" migrate="yes"
ID="appearance_and_display\taskbar_and_start_menu">
    <component displayname="Microsoft-Windows-stobject" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-stobject/microsoft-windows-
stobject/settings"/>
    <component displayname="Microsoft-Windows-explorer" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-explorer/microsoft-windows-
explorer/settings"/>
  </component>
  <component displayname="Personalized Settings" migrate="yes"
ID="appearance_and_display\personalized_settings">
    <component displayname="Microsoft-Windows-uxtheme" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-uxtheme/microsoft-windows-
uxtheme/settings"/>
    <component displayname="Microsoft-Windows-themeui" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-themeui/microsoft-windows-
themeui/settings"/>
    <component displayname="Microsoft-Windows-shmig" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-shmig/microsoft-windows-
shmig/settings"/>
    <component displayname="Microsoft-Windows-shell32" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-shell32/microsoft-windows-
shell32/settings"/>
    <component displayname="Microsoft-Windows-CommandPrompt" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-commandprompt/microsoft-windows-
commandprompt/settings"/>
  </component>
</component>
  <component displayname="Additional Options" migrate="yes" ID="additional_options">
    <component displayname="Help Settings" migrate="yes" ID="additional_options\help_settings">
      <component displayname="Microsoft-Windows-Help-Client" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-help-client/microsoft-windows-
help-client/settings"/>
    </component>
    <component displayname="Windows Core Settings" migrate="yes"
ID="additional_options\windows_core_settings">
      <component displayname="Microsoft-Windows-Win32k-Settings" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-win32k-settings/microsoft-windows-
win32k-settings/settings"/>
      <component displayname="Microsoft-Windows-Web-Services-for-Management-Core" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-web-services-for-management-
core/microsoft-windows-web-services-for-management-core/settings"/>
      <component displayname="Microsoft-Windows-UPnPSSDP" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-upnpssdp/microsoft-windows-
upnpssdp/settings"/>
      <component displayname="Microsoft-Windows-UPnPDeviceHost" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-upnpdevicehost/microsoft-windows-
upnpdevicehost/settings"/>
      <component displayname="Microsoft-Windows-UPnPControlPoint" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlex/cmi/microsoft-windows-upnpcontrolpoint/microsoft-
windows-upnpcontrolpoint/settings"/>
      <component displayname="Microsoft-Windows-TerminalServices-RemoteConnectionManager" migrate="yes"
```

```

    <component displayname="Microsoft-Windows-TerminalServices-RemoteConnectionManager" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-terminalservices-
remoteconnectionmanager/microsoft-windows-terminalservices-remoteconnectionmanager/settings"/>
    <component displayname="Microsoft-Windows-TerminalServices-Drivers" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-terminalservices-
drivers/microsoft-windows-terminalservices-drivers/settings"/>
    <component displayname="Microsoft-Windows-SQMApi" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-sqmap/microsoft-windows-
sqmap/settings"/>
    <component displayname="Microsoft-Windows-RPC-Remote" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rpc-remote/microsoft-windows-rpc-
remote/settings"/>
    <component displayname="Microsoft-Windows-RPC-Local" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rpc-local/microsoft-windows-rpc-
local/settings"/>
    <component displayname="Microsoft-Windows-RPC-HTTP" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rpc-http/microsoft-windows-rpc-
http/settings"/>
    <component displayname="Microsoft-Windows-Rasppp" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rasppp/microsoft-windows-
rasppp/settings"/>
    <component displayname="Microsoft-Windows-RasMprDdm" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rasmprddm/microsoft-windows-
rasmprddm/settings"/>
    <component displayname="Microsoft-Windows-RasBase" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-rasbase/microsoft-windows-
rasbase/settings"/>
    <component displayname="Microsoft-Windows-Microsoft-Data-Access-Components-(MDAC)-ODBC-DriverManager-
Dll" migrate="yes" ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-microsoft-data-
access-components-(mdac)-odbc-drivermanager-dll/microsoft-windows-microsoft-data-access-components-(mdac)-
odbc-drivermanager-dll/settings"/>
    <component displayname="Microsoft-Windows-ICM-Profiles" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-icm-profiles/microsoft-windows-
icm-profiles/settings"/>
    <component displayname="Microsoft-Windows-feclient" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-feclient/microsoft-windows-
feclient/settings"/>
    <component displayname="Microsoft-Windows-dpapi-keys" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-dpapi-keys/microsoft-windows-
dpapi-keys/settings"/>
    <component displayname="Microsoft-Windows-Crypto-keys" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-crypto-keys/microsoft-windows-
crypto-keys/settings"/>
    <component displayname="Microsoft-Windows-COM-DTC-Setup" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-com-dtc-setup/microsoft-windows-
com-dtc-setup/settings"/>
    <component displayname="Microsoft-Windows-COM-ComPlus-Setup" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-com-complus-setup/microsoft-
windows-com-complus-setup/settings"/>
    <component displayname="Microsoft-Windows-COM-Base" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-com-base/microsoft-windows-com-
base/settings"/>
    <component displayname="Microsoft-Windows-CAPI2-certs" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-capi2-certs/microsoft-windows-
capi2-certs/settings"/>
  </component>
</component>
  <component displayname="Accessibility" migrate="yes" ID="accessibility">
    <component displayname="Accessibility Settings" migrate="yes"
ID="accessibility\accessibility_settings">
      <component displayname="Microsoft-Windows-accessibilitycpl" migrate="yes"
ID="http://www.microsoft.com/migration/1.0/migxmlext/cmi/microsoft-windows-accessibilitycpl/microsoft-
windows-accessibilitycpl/settings"/>
    </component>
  </component>
</WindowsComponents>
<Policies>
  <ErrorControl>
    <!-- Example:

```

```

    <fileError>
      <nonFatal errorCode="33">* [*]</nonFatal>
      <fatal errorCode="any">C:\Users\* [*]</fatal>
    </fileError>
  </registryError>
  <registryError>
    <nonFatal errorCode="5">* [*]</nonFatal>
  </registryError>
-->
</ErrorControl>
<HardLinkStoreControl>
  <!-- Example:

    <fileLocked>
      <createHardLink>c:\Users\* [*]</createHardLink>
      <errorHardLink>C:\* [*]</errorHardLink>
    </fileLocked>
  -->
</HardLinkStoreControl>
</Policies>
<ProfileControl>
  <!-- Example:

    <localGroups>
      <mappings>
        <changeGroup from="Administrators" to="Users" appliesTo="MigratedUsers">
          <include>
            <pattern>DomainName1\Username</pattern>
          </include>
          <exclude>
            <pattern>DomainName2\Username</pattern>
          </exclude>
        </changeGroup>
      </mappings>
    </localGroups>

  -->
</ProfileControl>
</Configuration>

```

Related topics

[USMT XML Reference](#)

Customize USMT XML Files

6/14/2019 • 7 minutes to read • [Edit Online](#)

In This Topic

[Overview](#)

[Migration .xml Files](#)

[Custom .xml Files](#)

[The Config.xml File](#)

[Examples](#)

[Additional Information](#)

Overview

If you want the **ScanState** and **LoadState** tools to use any of the migration .xml files, specify these files at the command line using the **/i** option. Because the **ScanState** and **LoadState** tools need the .xml files to control the migration, specify the same set of .xml files for both the **ScanState** and **LoadState** commands. However, you do not have to specify the Config.xml file with the **/config** option, unless you want to exclude some of the files and settings that you migrated to the store. For example, you might want to migrate the My Documents folder to the store but not to the destination computer. To do this, modify the Config.xml file and specify the updated file with the **LoadState** command. Then the **LoadState** command will migrate only the files and settings that you want to migrate.

If you leave out an .xml file from the **LoadState** command, all of the data in the store that was migrated with the missing .xml files will be migrated. However, the migration rules that were specified with the **ScanState** command will not apply. For example, if you leave out an .xml file, and it contains a rerouting rule such as:

```
MigsysHelperFunction.RelativeMove("c:\data", "%CSIDL_PERSONAL%")
```

USMT will not reroute the files, and they will be migrated to C:\data.

To modify the migration, do one or more of the following.

- **Modify the migration .xml files.** If you want to exclude a portion of a component—for example, you want to migrate C:\ but exclude all of the .mp3 files—or if you want to move data to a new location on the destination computer, modify the .xml files. To modify these files, you must be familiar with the migration rules and syntax. If you want **ScanState** and **LoadState** to use these files, specify them at the command line when each command is entered.
- **Create a custom .xml file.** You can also create a custom .xml file to migrate settings for another application, or to change the migration behavior to suit your needs. For **ScanState** and **LoadState** to use this file, specify them on both command lines.
- **Create and modify a Config.xml file.** Do this if you want to exclude an entire component from the migration. For example, you can use a Config.xml file to exclude the entire My Documents folder, or exclude the settings for an application. Excluding components using a Config.xml file is easier than modifying the migration .xml files because you do not need to be familiar with the migration rules and syntax. In addition, using a Config.xml file is the only way to exclude the operating system settings from being migrated.

For more information about excluding data, see the [Exclude Files and Settings](#) topic.

Migration .xml Files

This section describes the migration .xml files that are included with USMT. Each file contains migration rules that control which components are migrated and where they are migrated to on the destination computer.

Note You can use the asterisk (*) wildcard character in each of these files. However, you cannot use a question mark (?) as a wildcard character.

- **The MigApp.xml file.** Specify this file with both the **ScanState** and **LoadState** commands to migrate application settings.
- **The MigDocs.xml file.** Specify this file with both the **ScanState** and **LoadState** tools to migrate all user folders and files that are found by the **MigXmlHelper.GenerateDocPatterns** helper function. This helper function finds user data that resides on the root of any drive and in the Users directory. However, it does not find and migrate any application data, program files, or any files in the Windows directory. You can modify the MigDocs.xml file.
- **The MigUser.xml file.** Specify this file with both the **ScanState** and **LoadState** commands to migrate user folders, files, and file types. You can modify the MigUser.xml file. This file does not contain rules that migrate specific user accounts. The only way to specify which user accounts to migrate is on the command line using the **ScanState** and the **LoadState** user options.

Note Do not use the MigUser.xml and MigDocs.xml files together. For more information, see the [Identify File Types, Files, and Folders](#) and [USMT Best Practices](#) topics.

Custom .xml Files

You can create custom .xml files to customize the migration for your unique needs. For example, you may want to create a custom file to migrate a line-of-business application or to modify the default migration behavior. If you want **ScanState** and **LoadState** to use this file, specify it with both commands. For more information, see the [How to Create a Custom .xml File](#) topic.

The Config.xml File

The Config.xml file is an optional file that you create using the **/genconfig** option with the **ScanState** command. You should create and modify this file if you want to exclude certain components from the migration. In addition, you must create and modify this file if you want to exclude any of the operating system settings from being migrated. The Config.xml file format is different from that of the migration .xml files because it does not contain any migration rules. It contains only a list of the operating system components, applications, and the user documents that can be migrated. For an example, see the [Config.xml File](#) topic. For this reason, excluding components using this file is easier than modifying the migration .xml files because you do not need to be familiar with the migration rules and syntax. However, you cannot use wildcard characters in a Config.xml file.

If you want to include all of the default components, you do not need to create the Config.xml file. Alternatively, if you are satisfied with the default migration behavior defined in the MigApp.xml, MigDocs.xml, and MigUser.xml files, and you want to exclude only some components, you can create and modify a Config.xml file and leave the other .xml files in their original state.

When you run the **ScanState** command with the **/genconfig** option, **ScanState** reads the other .xml files that you specify using the **/i** option to create a custom list of components that can be migrated from the computer. This file will contain only operating system components, applications, and the user document sections that are in both of the .xml files and that are installed on the computer when you run the **ScanState** command with the **/genconfig** option. Therefore, you should create this file on a source computer that contains all of the components, applications, and settings that will be present on the destination computers. This will ensure that

this file contains every component that can be migrated. The components are organized into sections: <Applications>, <WindowsComponents>, and <Documents>. To choose not to migrate a component, change its entry to `migrate="no"`.

After you create this file, you need to specify it only with the **ScanState** command using the **/Config** option for it to affect the migration. However, if you want to exclude additional data that you migrated to the store, modify the Config.xml file and specify the updated file with the **LoadState** command. For example, if you collected the My Documents folder in the store, but you decide that you do not want to migrate the My Documents folder to a destination computer, you can modify the Config.xml file to indicate `migrate="no"` before you run the **LoadState** command, and the file will not be migrated. For more information about the precedence that takes place when excluding data, see the [Exclude Files and Settings](#) topic.

In addition, note the following functionality with the Config.xml file:

- If a parent component is removed from the migration in the Config.xml file by specifying `migrate="no"`, all of its child components will automatically be removed from the migration, even if the child component is set to `migrate="yes"`.
- If you mistakenly have two lines of code for the same component where one line specifies `migrate="no"` and the other line specifies `migrate="yes"`, the component will be migrated.
- In USMT there are several migration policies that can be configured in the Config.xml file. For example, you can configure additional **<ErrorControl>**, **<ProfileControl>**, and **<HardLinkStoreControl>** options. For more information, see the [Config.xml File](#) topic.

Note To exclude a component from the Config.xml file, set the **migrate** value to **"no"**. Deleting the XML tag for the component from the Config.xml file will not exclude the component from your migration.

Examples

- The following command creates a Config.xml file in the current directory, but it does not create a store:

```
scanstate /i:migapp.xml /i:migdocs.xml /genconfig:config.xml /v:5
```

- The following command creates an encrypted store using the Config.xml file and the default migration.xml files:

```
scanstate \\server\share\migration\mystore /i:migapp.xml /i:migdocs.xml /o /config:config.xml /v:5 /encrypt /key:"mykey"
```

- The following command decrypts the store and migrates the files and settings:

```
loadstate \\server\share\migration\mystore /i:migapp.xml /i:migdocs.xml /v:5 /decrypt /key:"mykey"
```

Additional Information

- For more information about how to change the files and settings that are migrated, see the [User State Migration Tool \(USMT\) How-to topics](#).
- For more information about each .xml element, see the [XML Elements Library](#) topic.
- For answers to common questions, see ".xml files" in the [Frequently Asked Questions](#) topic.

Related topics

[User State Migration Tool \(USMT\) Command-line Syntax](#)

[USMT Resources](#)

Custom XML Examples

6/26/2019 • 4 minutes to read • [Edit Online](#)

Note Because the tables in this topic are wide, you may need to adjust the width of its window.

In This Topic:

- [Example 1: Migrating an Unsupported Application](#)
- [Example 2: Migrating the My Videos Folder](#)
- [Example 3: Migrating Files and Registry Keys](#)
- [Example 4: Migrating Specific Folders from Various Locations](#)

Example 1: Migrating an Unsupported Application

The following is a template for the sections that you need to migrate your application. The template is not functional on its own, but you can use it to write your own .xml file.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/migttestapp">
  <component type="Application">
    <!-- Name of the application -->
    <displayName>Some Application</displayName>
    <!-- Specify whether the environment variables exist in the context of user or system or both -->
    <environment context="System">
      <!-- Create the environment variables -->
      <variable name="myVar1">
        <!-- Simple text value assignment to a variable -->
        <text>value</text>
      </variable>
      <variable name="myAppExePath">
        <!-- Make a call to in-built helper function to get a value from a reg key and assign that value to
the variable -->
        <script>MigXMLHelper.GetStringContent("Registry","HKLM\Software\MyApp\Installer [EXEPATH]")</script>
      </variable>
    </environment>
    <role role="Settings">
      <detects>
        <!-- All of these checks must be true for the component to be detected -->
        <detect>
          <!-- Make a call to in-built helper function to check if an object exists or not -->
          <condition>MigXMLHelper.DoesObjectExist("Registry","HKLM\Software\MyApp [win32_version]")
</condition>
        </detect>
        <detect>
          <!-- Either of these checks must be true for the component to be detected -->
          <!-- Make a call to in-built helper function to check if a file version matches or not -->
          <condition>MigXMLHelper.DoesFileVersionMatch("%MyAppExePath%","ProductVersion","8.*")</condition>
          <condition>MigXMLHelper.DoesFileVersionMatch("%MyAppExePath%","ProductVersion","9.*")</condition>
        </detect>
      </detects>
      <!-- Describe the rules that will be executed during migration of this component and the context,
whether user, system or both -->
      <rules context="User">
        <!-- Delete objects specified in the object set on the destination computer before applying source
objects -->
        <destinationCleanup>
          <!-- Describe the pattern for the list of objects to be deleted -->
          <objectSet>
            <pattern type="Registry">HKCU\Software\MyApp\Toolbar\* [*]</pattern>
            <pattern type="Registry">HKCU\Software\MyApp\ListView\* [*]</pattern>
            <pattern type="Registry">HKCU\Software\MyApp [ShowTips]</pattern>
          </objectSet>
        </destinationCleanup>
        <!-- Specify which set of objects should be migrated -->
        <include>
          <!-- Describe the pattern for the list of objects to be included -->
          <objectSet>
            <pattern type="Registry">HKCU\Software\MyApp\Toolbar\* [*]</pattern>
            <pattern type="Registry">HKCU\Software\MyApp\ListView\* [*]</pattern>
            <pattern type="Registry">HKCU\Software\MyApp [ShowTips]</pattern>
          </objectSet>
        </include>
        <!-- Specify which set of objects should not be migrated -->
        <exclude>
          <!-- Describe the pattern for the list of objects to be excluded from migration -->
          <objectSet>
            <pattern type="Registry">HKCU\Software\MyApp [Display]</pattern>
          </objectSet>
        </exclude>
      </rules>
    </role>
  </component>
</migration>

```

Example 2: Migrating the My Videos Folder

The following is a custom .xml file named CustomFile.xml that migrates My Videos for all users, if the folder exists on the source computer.

CODE	BEHAVIOR
<pre><condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_MYVIDE0%")</condition></pre>	Verifies that My Videos exists on the source computer.
<pre><include filter='MigXmlHelper.IgnoreIrrelevantLinks()'</pre>	Filters out the shortcuts in My Videos that do not resolve on the destination computer. This has no effect on files that are not shortcuts. For example, if there is a shortcut in My Videos on the source computer that points to C:\Folder1, that shortcut will be migrated only if C:\Folder1 exists on the destination computer. However, all other files, such as .mp3 files, migrate without any filtering.
<pre><pattern type="File">%CSIDL_MYVIDE0%* [*] </pattern></pre>	Migrates My Videos for all users.

```
<?xml version="1.0" encoding="UTF-8"?>
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/CustomFile">
<component type="Documents" context="User">
  <displayName>My Video</displayName>
  <role role="Data">
    <detects>
      <detect>
        <condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_MYVIDE0%")</condition>
      </detect>
    </detects>
    <rules>
      <include filter='MigXmlHelper.IgnoreIrrelevantLinks()''>
        <objectSet>
          <pattern type="File">%CSIDL_MYVIDE0%* [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>
```

Example 3: Migrating Files and Registry Keys

This table describes the behavior in the following example .xml file.

CODE	BEHAVIOR
<pre data-bbox="177 190 782 320"><pattern type="File">%ProgramFiles%\USMTTestFolder* [USMTTestFile.txt]</pattern></pre>	<p data-bbox="828 181 1385 237">Migrates all instances of the file Usmttestfile.txt from all sub-directories under %ProgramFiles%\USMTTestFolder.</p>
<pre data-bbox="177 439 782 517"><pattern type="File">%ProgramFiles%\USMTDIRTestFolder* [] </pattern></pre>	<p data-bbox="828 405 1198 461">Migrates the whole directory under %ProgramFiles%\USMTDIRTestFolder.</p>
<pre data-bbox="177 658 782 736"><pattern type="Registry">HKCU\Software\USMTTESTKEY* [MyKey]</pattern></pre>	<p data-bbox="828 624 1203 680">Migrates all instances of MyKey under HKCU\Software\USMTTESTKEY.</p>
<pre data-bbox="177 878 782 956"><pattern type="Registry">HKLM\Software\USMTTESTKEY* [] </pattern></pre>	<p data-bbox="828 844 1209 900">Migrates the entire registry hive under HKLM\Software\USMTTESTKEY.</p>

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/testfilemig">
  <component type="Application" context="System">
    <displayName>File Migration Test</displayName>
    <role role="Data">
      <rules context="System">
        <include>
          <objectSet>
            <pattern type="File">%ProgramFiles%\USMTTestFolder\* [USMTTestFile.txt]</pattern>
            <pattern type="File">%ProgramFiles%\USMTDIRTestFolder\* [*]</pattern>
          </objectSet>
        </include>
      </rules>
    </role>
  </component>
  <component type="System">
    <displayName>Registry Migration Test</displayName>
    <role role="Settings">
      <rules context="UserAndSystem">
        <include>
          <objectSet>
            <pattern type="Registry">HKCU\Software\USMTTESTKEY\* [MyKey]</pattern>
            <pattern type="Registry">HKLM\Software\USMTTESTKEY\* [*]</pattern>
          </objectSet>
        </include>
      </rules>
    </role>
  </component>
</migration>
```

Example 4: Migrating Specific Folders from Various Locations

The behavior for this custom .xml file is described within the < `displayName` > tags in the code.

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmltext/test">

<component type="Documents" context="System">
  <displayName>Component to migrate all Engineering Drafts subfolders without documents in this folder
</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </include>
      <exclude>
        <objectSet>
          <pattern type="File"> C:\EngineeringDrafts\* [*]</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>

<component type="Documents" context="System">
  <displayName>Component to migrate all user documents except Sample.doc</displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\UserDocuments\* [*]</pattern>
        </objectSet>
      </include>
      <exclude>
        <objectSet>
          <pattern type="File"> C:\UserDocuments\ [Sample.doc]</pattern>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>

<component type="Documents" context="System">
  <displayName>Component to migrate all Requests folders on any drive on the computer </displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <script>MigXmlHelper.GenerateDrivePatterns ("\\Requests\* [*] ", "Fixed")</script>
          <script>MigXmlHelper.GenerateDrivePatterns ("*\Requests\* [*] ", "Fixed")</script>
        </objectSet>
      </include>
    </rules>
  </role>
</component>

<component type="Documents" context="System">
  <displayName>Component to migrate all Presentations folder from any location on the C: drive </displayName>
  <role role="Data">
    <rules>
      <include>
        <objectSet>
          <pattern type="File"> C:\*\Presentations\* [*]</pattern>
          <pattern type="File"> C:\Presentations\* [*]</pattern>
        </objectSet>
      </include>
    </rules>
  </role>
</component>
</migration>

```

Related topics

[USMT XML Reference](#)

[Customize USMT XML Files](#)

Conflicts and Precedence

6/14/2019 • 10 minutes to read • [Edit Online](#)

When you include, exclude, and reroute files and settings, it is important to know how User State Migration Tool (USMT) 10.0 deals with conflicts and precedence. When working with USMT, the following are the most important conflicts and precedence guidelines to keep in mind.

- **If there are conflicting rules within a component, the most specific rule is applied.** However, the `<unconditionalExclude>` rule is an exception because it takes precedence over all others. Directory names take precedence over file extensions. For examples, see [What happens when there are conflicting include and exclude rules?](#) and the first example in [Include and exclude precedence examples](#)****later in this topic.
- **Only rules inside the same component can affect each other, depending on specificity.** Rules that are in different components do not affect each other, except for the `<unconditionalExclude>` rule.
- **If the rules are equally specific, `<exclude>` takes precedence over `<include>`.** For example, if you use the `<exclude>` rule to exclude a file and use the `<include>` rule to include the same file, the file will be excluded.
- **The ordering of components does not matter.** It does not matter which components are listed in which .xml file, because each component is processed independently of the other components across all of the .xml files.
- **The ordering of the `<include>` and `<exclude>` rules within a component does not matter.**
- **You can use the `<unconditionalExclude>` element to globally exclude data.** This element excludes objects, regardless of any other `<include>` rules that are in the .xml files. For example, you can use the `<unconditionalExclude>` element to exclude all MP3 files on the computer or to exclude all files from C:\UserData.

In This Topic

General

- [What is the relationship between rules that are located within different components?](#)
- [How does precedence work with the Config.xml file?](#)
- [How does USMT process each component in an .xml file with multiple components?](#)
- [How are rules processed?](#)
- [How does USMT combine all of the .xml files that I specify on the command line?](#)

The `<include>` and `<exclude>` rules

- [What happens when there are conflicting include and exclude rules?](#)
- [<include> and <exclude> precedence examples](#)

File collisions

- [What is the default behavior when there are file collisions?](#)
- [How does the `<merge>` rule work when there are file collisions?](#)

General

What is the relationship between rules that are located within different components?

Only rules inside the same component can affect each other, depending on specificity, except for the `<unconditionalExclude>` rule. Rules that are in different components do not affect each other. If there is an `<include>` rule in one component and an identical `<exclude>` rule in another component, the data will be migrated because the two rules are independent of each other.

If you have an `<include>` rule in one component and a `<locationModify>` rule in another component for the same file, the file will be migrated in both places. That is, it will be included based on the `<include>` rule, and it will be migrated based on the `<locationModify>` rule.

The following .xml file migrates all files from C:\Userdocs, including .mp3 files, because the `<exclude>` rule is specified in a separate component.

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlex/UserDocs">
  <component type="Documents" context="System">
    <displayName>User Documents</displayName>
    <role role="Data">
      <rules>
        <exclude>
          <objectSet>
            <pattern type="File">C:\Userdocs\* [*].mp3</pattern>
          </objectSet>
        </exclude>
      </rules>
    </role>
  </component>

  <component type="Documents" context="System">
    <displayName> User documents to include </displayName>
    <role role="Data">
      <rules>
        <include>
          <objectSet>
            <pattern type="File"> C:\Userdocs\ [*]</pattern>
          </objectSet>
        </include>
      </rules>
    </role>
  </component>
</migration>
```

How does precedence work with the Config.xml file?

Specifying `migrate="no"` in the Config.xml file is the same as deleting the corresponding component from the migration .xml file. However, if you set `migrate="no"` for My Documents, but you have a rule similar to the one shown below in a migration .xml file (which includes all of the .doc files from My Documents), then only the .doc files will be migrated, and all other files will be excluded.

```
<include>
  <objectSet>
    <pattern type="File">%CSIDL_PERSONAL%\* [*].doc </pattern>
  </objectSet>
</include>
```

How does USMT process each component in an .xml file with multiple components?

The ordering of components does not matter. Each component is processed independently of other components. For example, if you have an `<include>` rule in one component and a `<locationModify>` rule in

another component for the same file, the file will be migrated in both places. That is, it will be included based on the <include> rule, and it will be migrated based on the <locationModify> rule.

How are rules processed?

There are two broad categories of rules.

- **Rules that affect the behavior of both the ScanState and LoadState tools.** For example, the <include>, <exclude>, and <unconditionalExclude> rules are processed for each component in the .xml files. For each component, USMT creates an include list and an exclude list. Some of the rules in the component might be discarded due to specificity, but all of the remaining rules are processed. For each <include> rule, USMT iterates through the elements to see if any of the locations need to be excluded. USMT enumerates all of the objects and creates a list of objects it is going to collect for each user. Once the list is complete, each of the objects is stored or migrated to the destination computer.
- **Rules that affect the behavior of only the LoadState tool.** For example, the <locationModify>, <contentModify>, and <destinationCleanup> rules do not affect ScanState. They are processed only with LoadState. First, the LoadState tool determines the content and location of each component based on the <locationModify> and <contentModify> rules. Then, LoadState processes all of the <destinationCleanup> rules and deletes data from the destination computer. Lastly, LoadState applies the components to the computer.

How does USMT combine all of the .xml files that I specify on the command line?

USMT does not distinguish the .xml files based on their name or content. It processes each component within the files separately. USMT supports multiple .xml files only to make it easier to maintain and organize the components within them. Because USMT uses a urlid to distinguish each component from the others, be sure that each .xml file that you specify on the command line has a unique migration urlid.

The <include> and <exclude> rules

What happens when there are conflicting <include> and <exclude> rules?

If there are conflicting rules within a component, the most specific rule is applied, except with the <unconditionalExclude> rule, which takes precedence over all other rules. If the rules are equally specific, then the data will not be migrated. For example if you exclude a file, and include the same file, the file will not be migrated. If there are conflicting rules within different components, the rules do not affect each other because each component is processed independently.

In the following example, mp3 files will not be excluded from the migration. This is because directory names take precedence over the file extensions.

```
<include>
  <objectSet>
    <pattern type="File">C:\Data\* [*]</pattern>
  </objectSet>
</include>
<exclude>
  <objectSet>
    <pattern type="File"> C:\* [*mp3]</pattern>
  </objectSet>
</exclude>
```

<include> and <exclude> rules precedence examples

These examples explain how USMT deals with <include> and <exclude> rules. When the rules are in different components, the resulting behavior will be the same regardless of whether the components are in the same or in different migration .xml files.

- [Including and excluding files](#)

- Including and excluding registry objects

Including and excluding files

IF YOU HAVE THE FOLLOWING CODE IN THE SAME COMPONENT	RESULTING BEHAVIOR	EXPLANATION
<ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1* [] </pattern></code> • Exclude rule: <code><pattern type="File">C:* [.txt] </pattern></code> 	Migrates all files and subfolders in Dir1 (including all .txt files in C:).	The <code><exclude></code> rule does not affect the migration because the <code><include></code> rule is more specific.
<ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1* [] </pattern></code> • Exclude rule: <code><pattern type="File">C:\Dir1\Dir2* [.txt]</pattern></code> 	Migrates all files and subfolders in C:\Dir1, except the .txt files in C:\Dir1\Dir2 and its subfolders.	Both rules are processed as intended.
<ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1* [] </pattern></code> • Exclude rule: <code><pattern type="File">C:\Dir1\ * [.txt] </pattern></code> 	Migrates all files and subfolders in C:\Dir1, except the .txt files in C:\Dir1 and its subfolders.	Both rules are processed as intended.
<ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1\Dir2* [.txt]</pattern></code> • Exclude rule: <code><pattern type="File">C:\Dir1\Dir2* [.txt]</pattern></code> 	Nothing will be migrated.	The rules are equally specific, so the <code><exclude></code> rule takes precedence over the <code><include></code> rule.
<ul style="list-style-type: none"> • Include rule: <code>C:\Dir1* [.txt]</code> • Exclude rule: <code>C:\Dir1\Dir2* []</code> 	Migrates the .txt files in Dir1 and the .txt files from subfolders other than Dir2. No files are migrated from Dir2 or its subfolders.	Both rules are processed as intended.
<ul style="list-style-type: none"> • Include rule: <code>C:\Dir1\Dir2* []</code> • Exclude rule: <code>C:\Dir1* [.txt]</code> 	Migrates all files and subfolders of Dir2, except the .txt files from Dir1 and any subfolders of Dir1 (including Dir2).	Both rules are processed as intended.

IF YOU HAVE THE FOLLOWING CODE IN DIFFERENT COMPONENTS	RESULTING BEHAVIOR	EXPLANATION
<p>Component 1:</p> <ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1* [] </pattern></code> • Exclude rule: <code><pattern type="File">C:\Dir1\Dir2* [.txt]</pattern></code> <p>Component 2:</p> <ul style="list-style-type: none"> • Include rule: <code><pattern type="File">C:\Dir1\Dir2* [.txt]</pattern></code> • Exclude rule: <code><pattern type="File">C:\Dir1* [] </pattern></code> 	Migrates all files and subfolders of C:\Dir1\ (including C:\Dir1\Dir2).	Rules that are in different components do not affect each other, except for the <code><unconditionalExclude></code> rule. Therefore, in this example, although some .txt files were excluded when Component 1 was processed, they were included when Component 2 was processed.
<p>Component 1:</p> <ul style="list-style-type: none"> • Include rule: <code>C:\Dir1\Dir2* []</code> <p>Component 2:</p> <ul style="list-style-type: none"> • Exclude rule: <code>C:\Dir1* [.txt]</code> 	Migrates all files and subfolders from Dir2 except the .txt files in C:\Dir1 and its subfolders.	Both rules are processed as intended.
<p>Component 1:</p> <ul style="list-style-type: none"> • Exclude rule: <code>C:\Dir1\Dir2* []</code> <p>Component 2:</p> <ul style="list-style-type: none"> • Include rule: <code>C:\Dir1* [.txt]</code> 	Migrates all .txt files in Dir1 and any subfolders.	Component 1 does not contain an <code><include></code> rule, so the <code><exclude></code> rule is not processed.

Including and excluding registry objects

IF YOU HAVE THE FOLLOWING CODE IN THE SAME COMPONENT	RESULTING BEHAVIOR	EXPLANATION
<ul style="list-style-type: none"> • Include rule: <code>HKLM\Software\Microsoft\Command Processor* []</code> • Exclude Rule: <code>HKLM\Software\Microsoft\Command Processor [DefaultColor]</code> 	Migrates all keys in HKLM\Software\Microsoft\Command Processor except DefaultColor.	Both rules are processed as intended.

IF YOU HAVE THE FOLLOWING CODE IN THE SAME COMPONENT	RESULTING BEHAVIOR	EXPLANATION
<ul style="list-style-type: none"> • Include rule: HKLM\Software\Microsoft\Command Processor [DefaultColor] • Exclude Rule: HKLM\Software\Microsoft\Command Processor* [] 	Migrates only DefaultColor in HKLM\Software\Microsoft\Command Processor.	DefaultColor is migrated because the <include> rule is more specific than the <exclude> rule.
<ul style="list-style-type: none"> • Include rule: HKLM\Software\Microsoft\Command Processor [DefaultColor] • Exclude rule: HKLM\Software\Microsoft\Command Processor [DefaultColor] 	Does not migrate DefaultColor.	The rules are equally specific, so the <exclude> rule takes precedence over the <include> rule.
IF YOU HAVE THE FOLLOWING CODE IN DIFFERENT COMPONENTS	RESULTING BEHAVIOR	EXPLANATION
<p>Component 1:</p> <ul style="list-style-type: none"> • Include rule: HKLM\Software\Microsoft\Command Processor [DefaultColor] • Exclude rule: HKLM\Software\Microsoft\Command Processor* [] <p>Component 2:</p> <ul style="list-style-type: none"> • Include rule: HKLM\Software\Microsoft\Command Processor* [] • Exclude rule: HKLM\Software\Microsoft\Command Processor [DefaultColor] 	Migrates all the keys/values under HKLM\Software\Microsoft\Command Processor.	Rules that are in different components do not affect each other, except for the <unconditionalExclude> rule. Therefore, in this example, the objects that were excluded when Component 1 was processed were included when Component 2 was processed.

File collisions

What is the default behavior when there are file collisions?

If there is not a <merge> rule, the default behavior for the registry is for the source to overwrite the destination. The default behavior for files is for the source to be renamed incrementally: for example, OriginalFileName(1).OriginalExtension, OriginalFileName(2).OriginalExtension, and so on.

How does the <merge> rule work when there are file collisions?

When a collision is detected, USMT will select the most specific <merge> rule and apply it to resolve the conflict. For example, if you have a <merge> rule for C:* [*] set to **sourcePriority()** and another <merge> rule for C:\subfolder* [*] set to **destinationPriority()**, then USMT uses the destinationPriority() rule because it is

the most specific.

Example scenario

The source computer contains the following files:

- C:\Data\SampleA.txt
- C:\Data\SampleB.txt
- C:\Data\Folder\SampleB.txt

The destination computer contains the following files:

- C:\Data\SampleB.txt
- C:\Data\Folder\SampleB.txt

You have a custom .xml file that contains the following code:

```
<include>
  <objectSet>
    <pattern type="File">c:\data\* [*]</pattern>
  </objectSet>
</include>
```

For this example, the following table describes the resulting behavior if you add the code in the first column to your custom .xml file.

IF YOU SPECIFY THE FOLLOWING CODE	RESULTING BEHAVIOR
<pre><merge script="MigXmlHelper.DestinationPriority()"> <objectSet> <pattern type="File">c:\data* [] </pattern> </objectSet> </merge></pre>	During ScanState, all the files will be added to the store. During LoadState, only C:\Data\SampleA.txt will be restored.
<pre><merge script="MigXmlHelper.SourcePriority()"> <objectSet> <pattern type="File">c:\data* [] </pattern> </objectSet> </merge></pre>	During ScanState, all the files will be added to the store. During LoadState, all the files will be restored, overwriting the existing files on the destination computer.
<pre><merge script="MigXmlHelper.SourcePriority()"> <objectSet> <pattern type="File">c:\data\ [*] </pattern> </objectSet> </merge></pre>	During ScanState, all the files will be added to the store. During LoadState, the following will occur: <ul style="list-style-type: none">• C:\Data\SampleA.txt will be restored.• C:\Data\SampleB.txt will be restored, overwriting the existing file on the destination computer.• C:\Data\Folder\SampleB.txt will not be restored.

Related topics

General Conventions

6/6/2019 • 2 minutes to read • [Edit Online](#)

This topic describes the XML helper functions.

In This Topic

[General XML Guidelines](#)

[Helper Functions](#)

General XML Guidelines

Before you modify the .xml files, become familiar with the following guidelines:

- **XML schema**

You can use the User State Migration Tool (USMT) 10.0 XML schema, MigXML.xsd, to write and validate migration .xml files.

- **Conflicts**

In general, when there are conflicts within the XML schema, the most specific pattern takes precedence. For more information, see [Conflicts and Precedence](#).

- **Required elements**

The required elements for a migration .xml file are **<migration>**, **<component>**, **<role>**, and **<rules>**.

- **Required child elements**

- USMT does not fail with an error if you do not specify the required child elements. However, you must specify the required child elements for the parent element to affect the migration.
- The required child elements apply only to the first definition of the element. If these elements are defined and then referred to using their name, the required child elements do not apply. For example, if you define `<detects name="Example">` in **<namedElements>**, and you specify `<detects name="Example"/>` in **<component>** to refer to this element, the definition inside **<namedElements>** must have the required child elements, but the **<component>** element does not need to have the required child elements.

- **File names with brackets**

If you are migrating a file that has a bracket character ([or]) in the file name, you must insert a caret (^) character directly before the bracket for the bracket character to be valid. For example, if there is a file named File.txt, you must specify `<pattern type="File">c:\documents\mydocs [file^].txt</pattern>` instead of `<pattern type="File">c:\documents\mydocs [file].txt</pattern>`.

- **Using quotation marks**

When you surround code in quotation marks, you can use either double (") or single (') quotation marks.

Helper Functions

You can use the XML helper functions in the [XML Elements Library](#) to change migration behavior. Before you use

these functions in an .xml file, note the following:

- **All of the parameters are strings**
- **You can leave NULL parameters blank**

As with parameters with a default value convention, if you have a NULL parameter at the end of a list, you can leave it out. For example, the following function:

```
SomeFunction("My String argument",NULL,NULL)
```

is equivalent to:

```
SomeFunction("My String argument")
```

- **The encoded location used in all the helper functions is an unambiguous string representation for the name of an object**

It is composed of the node part, optionally followed by the leaf enclosed in square brackets. This makes a clear distinction between nodes and leaves.

For example, specify the file C:\Windows\notepad.exe: **c:\Windows[Notepad.exe]**. Similarly, specify the directory C:\Windows\System32 like this: **c:\Windows\System32**; note the absence of the [] characters.

The registry is represented in a similar way. The default value of a registry key is represented as an empty [] construct. For example, the default value for the HKLM\SOFTWARE\MyKey registry key is **HKLM\SOFTWARE\MyKey[]**.

- **You specify a location pattern in a way that is similar to how you specify an actual location**

The exception is that both the node and leaf part accept patterns. However, a pattern from the node does not extend to the leaf.

For example, the pattern **c:\Windows*** will match the \Windows directory and all subdirectories, but it will not match any of the files in those directories. To match the files as well, you must specify **c:\Windows*[*]**.

Related topics

[USMT XML Reference](#)

XML File Requirements

5/31/2019 • 2 minutes to read • [Edit Online](#)

When creating custom .xml files, note the following requirements:

- **The file must be in Unicode Transformation Format-8 (UTF-8).** You must save the file in this format, and you must specify the following syntax at the beginning of each .xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
```

- **The file must have a unique migration urlid.** The urlid of each file that you specify on the command line must be different. If two migration .xml files have the same urlid, the second .xml file that is specified on the command line will not be processed. This is because USMT uses the urlid to define the components within the file. For example, you must specify the following syntax at the beginning of each file:

```
<?xml version="1.0" encoding="UTF-8"?>  
<migration urlid="http://www.microsoft.com/migration/1.0/migxmltext/<CustomFileName">
```

- **Each component in the file must have a display name in order for it to appear in the Config.xml file.** This is because the Config.xml file defines the components by the display name and the migration urlid. For example, specify the following syntax:

```
<displayName>My Application</displayName>
```

For examples of custom .xml files, see [Custom XML Examples](#).

Recognized Environment Variables

5/31/2019 • 7 minutes to read • [Edit Online](#)

When using the XML files MigDocs.xml, MigApp.xml, and MigUser.xml, you can use environment variables to identify folders that may be different on different computers. Constant special item ID list (CSIDL) values provide a way to identify folders that applications use frequently but may not have the same name or location on any given computer. For example, the documents folder may be C:\Users\<<Username>\My Documents on one computer and C:\Documents and Settings on another. You can use the asterisk (*) wildcard character in MigUser.xml, MigApp.xml and MigDoc.xml files. However, you cannot use the asterisk (*) wildcard characters in the Config.xml file.

In This Topic

- [Variables that are processed for the operating system and in the context of each user](#)
- [Variables that are recognized only in the user context](#)

Variables that are processed for the operating system and in the context of each user

You can use these variables within sections in the .xml files with `context=UserAndSystem`, `context=User`, and `context=System`.

VARIABLE	EXPLANATION
ALLUSERSAPPDATA	Same as CSIDL_COMMON_APPDATA .
ALLUSERSPROFILE	Refers to % PROFILESFOLDER %\Public or % PROFILESFOLDER %\all users.
COMMONPROGRAMFILES	Same as CSIDL_PROGRAM_FILES_COMMON .
COMMONPROGRAMFILES(X86)	Refers to the C:\Program Files (x86)\Common Files folder on 64-bit systems.
CSIDL_COMMON_ADMINTOOLS	Version 10.0. The file-system directory that contains administrative tools for all users of the computer.
CSIDL_COMMON_ALTSTARTUP	The file-system directory that corresponds to the non-localized Startup program group for all users.
CSIDL_COMMON_APPDATA	The file-system directory that contains application data for all users. A typical path Windows is C:\ProgramData.

VARIABLE	EXPLANATION
CSIDL_COMMON_DESKTOPDIRECTORY	The file-system directory that contains files and folders that appear on the desktop for all users. A typical Windows® XP path is C:\Documents and Settings\All Users\Desktop. A typical path is C:\Users\Public\Desktop.
CSIDL_COMMON_DOCUMENTS	The file-system directory that contains documents that are common to all users. A typical path in Windows XP is C:\Documents and Settings\All Users\Documents. A typical path is C:\Users\Public\Documents.
CSIDL_COMMON_FAVORITES	The file-system directory that serves as a common repository for favorites common to all users. A typical path is C:\Users\Public\Favorites.
CSIDL_COMMON_MUSIC	The file-system directory that serves as a repository for music files common to all users. A typical path is C:\Users\Public\Music.
CSIDL_COMMON_PICTURES	The file-system directory that serves as a repository for image files common to all users. A typical path is C:\Users\Public\Pictures.
CSIDL_COMMON_PROGRAMS	The file-system directory that contains the directories for the common program groups that appear on the Start menu for all users. A typical path is C:\ProgramData\Microsoft\Windows\Start Menu\Programs.
CSIDL_COMMON_STARTMENU	The file-system directory that contains the programs and folders which appear on the Start menu for all users. A typical path in Windows is C:\ProgramData\Microsoft\Windows\Start Menu.
CSIDL_COMMON_STARTUP	The file-system directory that contains the programs that appear in the Startup folder for all users. A typical path in Windows XP is C:\Documents and Settings\All Users\Start Menu\Programs\Startup. A typical path is C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup.
CSIDL_COMMON_TEMPLATES	The file-system directory that contains the templates that are available to all users. A typical path is C:\ProgramData\Microsoft\Windows\Templates.
CSIDL_COMMON_VIDEO	The file-system directory that serves as a repository for video files common to all users. A typical path is C:\Users\Public\Videos.

VARIABLE	EXPLANATION
CSIDL_DEFAULT_APPDATA	Refers to the Appdata folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_LOCAL_APPDATA	Refers to the local Appdata folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_COOKIES	Refers to the Cookies folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_CONTACTS	Refers to the Contacts folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_DESKTOP	Refers to the Desktop folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_DOWNLOADS	Refers to the Downloads folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_FAVORITES	Refers to the Favorites folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_HISTORY	Refers to the History folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_INTERNET_CACHE	Refers to the Internet Cache folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_PERSONAL	Refers to the Personal folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_MYDOCUMENTS	Refers to the My Documents folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_MYPICTURES	Refers to the My Pictures folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_MYMUSIC	Refers to the My Music folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_MYVIDEO	Refers to the My Videos folder inside % DEFAULTUSERPROFILE %.

VARIABLE	EXPLANATION
CSIDL_DEFAULT_RECENT	Refers to the Recent folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_SENDTO	Refers to the Send To folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_STARTMENU	Refers to the Start Menu folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_PROGRAMS	Refers to the Programs folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_STARTUP	Refers to the Startup folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_TEMPLATES	Refers to the Templates folder inside % DEFAULTUSERPROFILE %.
CSIDL_DEFAULT_QUICKLAUNCH	Refers to the Quick Launch folder inside % DEFAULTUSERPROFILE %.
CSIDL_FONTS	A virtual folder containing fonts. A typical path is C:\Windows\Fonts.
CSIDL_PROGRAM_FILESX86	The Program Files folder on 64-bit systems. A typical path is C:\Program Files(86).
CSIDL_PROGRAM_FILES_COMMONX86	A folder for components that are shared across applications on 64-bit systems. A typical path is C:\Program Files(86)\Common.
CSIDL_PROGRAM_FILES	The Program Files folder. A typical path is C:\Program Files.
CSIDL_PROGRAM_FILES_COMMON	A folder for components that are shared across applications. A typical path is C:\Program Files\Common.
CSIDL_RESOURCES	The file-system directory that contains resource data. A typical path is C:\Windows\Resources.
CSIDL_SYSTEM	The Windows System folder. A typical path is C:\Windows\System32.

VARIABLE	EXPLANATION
CSIDL_WINDOWS	The Windows directory or system root. This corresponds to the % WINDIR % or % SYSTEMROOT % environment variables. A typical path is C:\Windows.
DEFAULTUSERPROFILE	Refers to the value in HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList [DefaultUserProfile] .
PROFILESFOLDER	Refers to the value in HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList [ProfilesDirectory] .
PROGRAMFILES	Same as CSIDL_PROGRAM_FILES .
PROGRAMFILES(X86)	Refers to the C:\Program Files (x86) folder on 64-bit systems.
SYSTEM	Refers to % WINDIR %\system32.
SYSTEM16	Refers to % WINDIR %\system.
SYSTEM32	Refers to % WINDIR %\system32.
SYSTEMPROFILE	Refers to the value in HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-18 [ProfileImagePath] .
SYSTEMROOT	Refers to the root of the system drive.
WINDIR	Refers to the Windows folder located on the system drive.

Variables that are recognized only in the user context

You can use these variables in the .xml files within sections with `context=User` and `context=UserAndSystem`.

VARIABLE	EXPLANATION
APPDATA	Same as CSIDL_APPDATA .

VARIABLE	EXPLANATION
CSIDL_ADMINTOOLS	The file-system directory that is used to store administrative tools for an individual user. The Microsoft® Management Console (MMC) saves customized consoles to this directory, which roams with the user profile.
CSIDL_ALTSTARTUP	The file-system directory that corresponds to the user's non-localized Startup program group.
CSIDL_APPDATA	The file-system directory that serves as a common repository for application-specific data. A typical path is C:\Documents and Settings\username\Application Data or C:\Users\username\AppData\Roaming.
CSIDL_BITBUCKET	The virtual folder that contains the objects in the user's Recycle Bin.
CSIDL_CDBURN_AREA	The file-system directory acting as a staging area for files waiting to be written to CD. A typical path is C:\Users\username\AppData\Local\Microsoft\Windows\MasteredBurning\Disc Burning.
CSIDL_CONNECTIONS	The virtual folder representing Network Connections that contains network and dial-up connections.
CSIDL_CONTACTS	This refers to the Contacts folder in % CSIDL_PROFILE %.
CSIDL_CONTROLS	The virtual folder that contains icons for the Control Panel items.
CSIDL_COOKIES	The file-system directory that serves as a common repository for Internet cookies. A typical path is C:\Users\username\AppData\Roaming\Microsoft\Windows\Cookies.
CSIDL_DESKTOP	The virtual folder representing the Windows desktop.
CSIDL_DESKTOPDIRECTORY	The file-system directory used to physically store file objects on the desktop, which should not be confused with the desktop folder itself. A typical path is C:\Users\username\Desktop.
CSIDL_DRIVES	The virtual folder representing My Computer that contains everything on the local computer: storage devices, printers, and Control Panel. The folder may also contain mapped network drives.

VARIABLE	EXPLANATION
CSIDL_FAVORITES	The file-system directory that serves as a common repository for the user's favorites. A typical path is C:\Users\Username\Favorites.
CSIDL_HISTORY	The file-system directory that serves as a common repository for Internet history items.
CSIDL_INTERNET	A virtual folder for Internet Explorer.
CSIDL_INTERNET_CACHE	The file-system directory that serves as a common repository for temporary Internet files. A typical path is C:\Users\username\AppData\Local\Microsoft\Windows\Temporary Internet Files
CSIDL_LOCAL_APPDATA	The file-system directory that serves as a data repository for local, non-roaming applications. A typical path is C:\Users\username\AppData\Local.
CSIDL_MYDOCUMENTS	The virtual folder representing My Documents. A typical path is C:\Users\Username\Documents.
CSIDL_MYMUSIC	The file-system directory that serves as a common repository for music files. A typical path is C:\Users\Username\Music.
CSIDL_MYPICTURES	The file-system directory that serves as a common repository for image files. A typical path is C:\Users\Username\Pictures.
CSIDL_MYVIDEO	The file-system directory that serves as a common repository for video files. A typical path is C:\Users\Username\Videos.
CSIDL_NETHOOD	A file-system directory that contains the link objects that may exist in the My Network Places virtual folder. It is not the same as CSIDL_NETWORK, which represents the network namespace root. A typical path is C:\Users\Username\AppData\Roaming\Microsoft\Windows\Network Shortcuts.
CSIDL_NETWORK	A virtual folder representing My Network Places, the root of the network namespace hierarchy.

VARIABLE	EXPLANATION
CSIDL_PERSONAL	<p>The virtual folder representing the My Documents desktop item. This is equivalent to CSIDL_MYDOCUMENTS.</p> <p>A typical path is C:\Documents and Settings\username\My Documents.</p>
CSIDL_PLAYLISTS	<p>The virtual folder used to store play albums, typically C:\Users\username\My Music\Playlists.</p>
CSIDL_PRINTERS	<p>The virtual folder that contains installed printers.</p>
CSIDL_PRINTHOOD	<p>The file-system directory that contains the link objects that can exist in the Printers virtual folder. A typical path is C:\Users\username\AppData\Roaming\Microsoft\Windows\Printer Shortcuts.</p>
CSIDL_PROFILE	<p>The user's profile folder. A typical path is C:\Users\Username.</p>
CSIDL_PROGRAMS	<p>The file-system directory that contains the user's program groups, which are themselves file-system directories. A typical path is C:\Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs.</p>
CSIDL_RECENT	<p>The file-system directory that contains shortcuts to the user's most recently used documents. A typical path is C:\Users\Username\AppData\Roaming\Microsoft\Windows\Recent.</p>
CSIDL_SENDTO	<p>The file-system directory that contains Send To menu items. A typical path is C:\Users\username\AppData\Roaming\Microsoft\Windows\SendTo.</p>
CSIDL_STARTMENU	<p>The file-system directory that contains Start menu items. A typical path in Windows XP is C:\Documents and Settings\username\Start Menu. A typical path in Windows Vista, Windows 7, or Windows 8 is C:\Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu.</p>
CSIDL_STARTUP	<p>The file-system directory that corresponds to the user's Startup program group. A typical path is C:\Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup.</p>

VARIABLE	EXPLANATION
CSIDL_TEMPLATES	The file-system directory that serves as a common repository for document templates. A typical path is C:\Users\username\AppData\Roaming\Microsoft\Windows\Templates.
HOMEPATH	Same as the standard environment variable.
TEMP	The temporary folder on the computer. A typical path is % USERPROFILE %\AppData\Local\Temp.
TMP	The temporary folder on the computer. A typical path is % USERPROFILE %\AppData\Local\Temp.
USERPROFILE	Same as CSIDL_PROFILE .
USERSID	Represents the current user-account security identifier (SID). For example, S-1-5-21-1714567821-1326601894-715345443-1026.

Related topics

[USMT XML Reference](#)

XML Elements Library

6/6/2019 • 66 minutes to read • [Edit Online](#)

Overview

This topic describes the XML elements and helper functions that you can employ to author migration .xml files to use with User State Migration Tool (USMT). It is assumed that you understand the basics of XML. .

In This Topic

In addition to XML elements and helper functions, this topic describes how to specify encoded locations and locations patterns, functions that are for internal USMT use only, and the version tags that you can use with helper functions.

- [Elements and helper functions](#)
- [Appendix](#)
 - [Specifying locations](#)
 - [Internal USMT functions](#)
 - [Valid version tags](#)

Elements and Helper Functions

The following table describes the XML elements and helper functions you can use with USMT.

ELEMENTS A-K	ELEMENTS L-Z	HELPER FUNCTIONS
<addObjects> <attributes> <bytes> <commandLine> <component> <condition> <conditions> <content> <contentModify> <description> <destinationCleanup> <detect> <detects> <detection> <displayName> <environment> <exclude> <excludeAttributes> <extensions> <extension> <externalProcess> <icon> <include> <includeAttribute>	<library> <location> <locationModify> <_locDefinition> <manufacturer> <merge> <migration> <namedElements> <object> <objectSet> <path> <paths> <pattern> <processing> <plugin> <role> <rules> <script> <text> <unconditionalExclude> <variable> <version> <windowsObjects>	<condition> functions <content> functions <contentModify> functions <include> and <exclude> filter functions <locationModify> functions <merge> functions <script> functions Internal USMT functions

<addObjects>

The `<addObjects>` element emulates the existence of one or more objects on the source computer. The child `<object>` elements provide the details of the emulated objects. If the content is a `<script>` element, the result of the invocation will be an array of objects.

- **Number of occurrences:** unlimited
- **Parent elements:** [<rules>](#)
- **Required child elements:** [<object>](#) In addition, you must specify [<location>](#) and [<attribute>](#) as child elements of this `<object>` element.
- **Optional child elements:** [<conditions>](#), [<condition>](#), [<script>](#)

Syntax:

```
<addObjects>
```

```
</addObjects>
```

The following example is from the MigApp.xml file:

```
<addObjects>
  <object>
    <location type="Registry"%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office
[UpgradeVersion]</location>
    <attributes>DWORD</attributes>
    <bytes>0B000000</bytes>
  </object>
  <object>
    <location type="Registry"%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office [Lang]
</location>
    <attributes>DWORD</attributes>
    <bytes>00000000</bytes>
  </object>
</addObjects>
```

<attributes>

The <attributes> element defines the attributes for a registry key or file.

- **Number of occurrences:** once for each <object>
- **Parent elements:** <object>
- **Child elements:** none

Syntax:

<attributes> *Content* </attributes>

SETTING	REQUIRED?	VALUE
<i>Content</i>	Yes	<p>The content depends on the type of object specified.</p> <ul style="list-style-type: none">• For files, the content can be a string containing any of the following attributes separated by commas:<ul style="list-style-type: none">◦ Archive◦ Read-only◦ System◦ Hidden• For registry keys, the content can be one of the following types:<ul style="list-style-type: none">◦ None◦ String◦ ExpandString◦ Binary◦ Dword◦ REG_SZ

The following example is from the MigApp.xml file:

```

<object>
  <location type="Registry"%HkImWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office [Lang]
</location>
  <attributes>DWORD</attributes>
  <bytes>00000000</bytes>
</object>

```

<bytes>

You must specify the <bytes> element only for files because, if <location> corresponds to a registry key or a directory, then <bytes> will be ignored.

- **Number of occurrences:** zero or one
- **Parent elements:** <object>
- **Child elements:** none

Syntax:

```
<bytes string="Yes|No" expand="Yes|No">Content</bytes>
```

SETTING	REQUIRED?	VALUE
string	No, default is No	Determines whether <i>Content</i> should be interpreted as a string or as bytes.
expand	No (default = Yes	When the expand parameter is Yes, the content of the <bytes> element is first expanded in the context of the source computer and then interpreted.
<i>Content</i>	Yes	<p>Depends on the value of the string.</p> <ul style="list-style-type: none"> • When the string is Yes: the content of the <bytes> element is interpreted as a string. • When the string is No: the content of the <bytes> element is interpreted as bytes. Each two characters represent the hexadecimal value of a byte. For example, "616263" is the representation for the "abc" ANSI string. A complete representation of the UNICODE string "abc" including the string terminator would be: "6100620063000000".

The following example is from the MigApp.xml file:

```

<object>
  <location type="Registry"%>%HklmWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office [Lang]
</location>
  <attributes>DWORD</attributes>
  <bytes>00000000</bytes>
</object>

```

<commandLine>

You might want to use the <commandLine> element if you want to start or stop a service or application before or after you run the ScanState and LoadState tools.

- **Number of occurrences:** unlimited
- **Parent elements:** <externalProcess>
- **Child elements:** none****

Syntax:

```
<commandLine>CommandLineString</commandLine>
```

SETTING	REQUIRED?	VALUE
CommandLineString	Yes	A valid command line.

<component>

The <component> element is required in a custom .xml file. This element defines the most basic construct of a migration .xml file. For example, in the MigApp.xml file, "Microsoft® Office 2003" is a component that contains another component, "Microsoft Office Access® 2003". You can use the child elements to define the component.

A component can be nested inside another component; that is, the <component> element can be a child of the <role> element within the <component> element in two cases: 1) when the parent <component> element is a container or 2) if the child <component> element has the same role as the parent <component> element.

- **Number of occurrences:** Unlimited
- **Parent elements:** <migration>, <role>
- **Required child elements:** <role>, <displayName>
- **Optional child elements:** <manufacturer>, <version>, <description>, <paths>, <icon>, <environment>, <extensions>

Syntax:

```
<component type="System|Application|Device|Documents" context="User|System|UserAndSystem"
defaultSupported="TRUE|FALSE|YES|NO"
```

```
hidden="Yes|No">
```

```
</component>
```

SETTING	REQUIRED?	VALUE
type	Yes	<p>You can use the following to group settings, and define the type of the component.</p> <ul style="list-style-type: none"> • System: Operating system settings. All Windows® components are defined by this type. <p>When type="System" and defaultSupported="FALSE" the settings will not migrate unless there is an equivalent component in the .xml files that is specified on the LoadState command line. For example, the default MigSys.xml file contains components with type="System" and defaultSupported="FALSE". If you specify this file on the ScanState command line, you must also specify the file on the LoadState command line for the settings to migrate. This is because the LoadState tool must detect an equivalent component. That is, the component must have the same migration urlid of the .xml file and an identical display name. Otherwise, the LoadState tool will not migrate those settings from the store. This is helpful when the source computer is running Windows XP, and you are migrating to both Windows Vista and Windows XP because you can use the same store for both destination computers.</p> <ul style="list-style-type: none"> • Application: Settings for an application. • Device: Settings for a device. • Documents: Specifies files.

SETTING	REQUIRED?	VALUE
context	<p>No</p> <p>Default = UserAndSystem</p>	<p>Defines the scope of this parameter; that is, whether to process this component in the context of the specific user, across the entire operating system, or both.</p> <p>The largest possible scope is set by the <component> element. For example, if a <component> element has a context of User and a <rules> element had a context of UserAndSystem, then the <rules> element would act as though it has a context of User. If a <rules> element has a context of System, it would act as though the <rules> element is not there.</p> <ul style="list-style-type: none"> • User. Evaluates the component for each user. • System. Evaluates the component only once for the system. • UserAndSystem. Evaluates the component for the entire operating system and each user.

SETTING	REQUIRED?	VALUE
defaultSupported	No (default = TRUE)	<p>Can be any of TRUE, FALSE, YES or NO. If this parameter is FALSE (or NO), the component will not be migrated unless there is an equivalent component on the destination computer.</p> <p>When type="System" and defaultSupported="FALSE" the settings will not migrate unless there is an equivalent component in the .xml files that are specified on the LoadState command line. For example, the default MigSys.xml file contains components with type="System" and defaultSupported="FALSE". If you specify this file on the ScanState command line, you must also specify the file on the LoadState command line for the settings to migrate. This is because the LoadState tool must detect an equivalent component. That is, the component must have the same migration urlid of the .xml file and an identical display name or the LoadState tool will not migrate those settings from the store. This is helpful when the source computer is running Windows XP, and you are migrating to both Windows Vista and Windows XP because you can use the same store for both destination computers.</p>
hidden		This parameter is for internal USMT use only.

For an example, see any of the default migration .xml files.

<condition>

Although the <condition> element under the <detect>, <objectSet>, and <addObjects> elements is supported, we recommend that you do not use it. This element might be deprecated in future versions of USMT, requiring you to rewrite your scripts. We recommend that, if you need to use a condition within the <objectSet> and <addObjects> elements, you use the more powerful [<conditions>](#) element, which allows you to formulate complex Boolean statements.

The <condition> element has a Boolean result. You can use this element to specify the conditions in which the parent element will be evaluated. If any of the present conditions return FALSE, the parent element will not be evaluated.

- **Number of occurrences:** unlimited.
- **Parent elements:** [<conditions>](#), <detect>, <objectSet>, <addObjects>

- **Child elements:** none
- **Helper functions:** You can use the following [<condition> functions](#) with this element: DoesOSMatch, IsNative64Bit(), IsOSLaterThan, IsOSEarlierThan, DoesObjectExist, DoesFileVersionMatch, IsFileVersionAbove, IsFileVersionBelow, IsSystemContext, DoesStringContentEqual, DoesStringContentContain, IsSameObject, IsSameContent, and IsSameStringContent.

Syntax:

```
<condition negation="Yes|No">ScriptName</condition>
```

SETTING	REQUIRED?	VALUE
negation	No Default = No	"Yes" reverses the True/False value of the condition.
<i>ScriptName</i>	Yes	A script that has been defined within this migration section.

For example,

In the code sample below, the <condition> elements, A and B, are joined together by the AND operator because they are in separate <conditions> sections. For example:

```
<detection>
  <conditions>
    <condition>A</condition>
  </conditions>
  <conditions operation="AND">
    <condition>B</condition>
  </conditions>
</detection>
```

However, in the code sample below, the <condition> elements, A and B, are joined together by the OR operator because they are in the same <conditions> section.

```
<detection>
  <conditions>
    <condition>A</condition>
    <condition>B</condition>
  </conditions>
</detection>
```

<condition> functions

The <condition> functions return a Boolean value. You can use these elements in <addObjects> conditions.

- [Operating system version functions](#)
- [Object content functions](#)

Operating system version functions

- **DoesOSMatch**

All matches are case insensitive.

Syntax: DoesOSMatch("OSType","OSVersion")

SETTING	REQUIRED?	VALUE
<i>OSType</i>	Yes	The only valid value for this setting is NT . Note, however, that you must set this setting for the <condition> functions to work correctly.
<i>OSVersion</i>	Yes	The major version, minor version, build number and corrected service diskette version separated by periods. For example, <code>5.0.2600.Service Pack 1</code> . You can also specify partial specification of the version with a pattern. For example, <code>5.0.*</code> .

For example:

```
<condition>MigXmlHelper.DoesOSMatch("NT","*")</condition>
```

- **IsNative64Bit**

The IsNative64Bit function returns TRUE if the migration process is running as a native 64-bit process; that is, a process running on a 64-bit system without Windows on Windows (WOW). Otherwise, it returns FALSE.

- **IsOSLaterThan**

All comparisons are case insensitive.

Syntax: IsOSLaterThan("*OSType*","*OSVersion*")

SETTING	REQUIRED?	VALUE
<i>OSType</i>	Yes	Can be 9x or NT . If <i>OSType</i> does not match the type of the current operating system, then it returns FALSE. For example, if the current operating system is Windows NT-based and <i>OSType</i> is "9x", the result will be FALSE.

SETTING	REQUIRED?	VALUE
<i>OSVersion</i>	Yes	<p>The major version, minor version, build number, and corrected service diskette version separated by periods. For example, <code>5.0.2600.Service Pack 1</code>.</p> <p>You can also specify partial specification of the version but no pattern is allowed. For example, <code>5.0</code>.</p> <p>The <code>IsOSLaterThan</code> function returns TRUE if the current operating system is later than or equal to <i>OSVersion</i>.</p>

For example:

```
<condition negation="Yes">MigXmlHelper.IsOSLaterThan("NT","6.0")</condition>
```

- **IsOSEarlierThan**

All comparisons are case insensitive.

Syntax: `IsOSEarlierThan("OSType","OSVersion")`

SETTING	REQUIRED?	VALUE
<i>OSType</i>	Yes	<p>Can be 9x or NT. If <i>OSType</i> does not match the type of the current operating system, then it returns FALSE. For example, if the current operating system is Windows NT-based and <i>OSType</i> is "9x" the result will be FALSE.</p>
<i>OSVersion</i>	Yes	<p>The major version, minor version, build number, and corrected service diskette version separated by periods. For example, <code>5.0.2600.Service Pack 1</code>.</p> <p>You can also specify partial specification of the version but no pattern is allowed. For example, <code>5.0</code>.</p> <p>The <code>IsOSEarlierThan</code> function returns TRUE if the current operating system is earlier than <i>OSVersion</i>.</p>

Object content functions

- **DoesObjectExist**

The `DoesObjectExist` function returns TRUE if any object exists that matches the location pattern.

Otherwise, it returns FALSE. The location pattern is expanded before attempting the enumeration.

Syntax: DoesObjectExist("Object Type","EncodedLocationPattern")

SETTING	REQUIRED?	VALUE
<i>ObjectType</i>	Yes	Defines the object type. Can be File or Registry.
<i>EncodedLocationPattern</i>	Yes	The location pattern . Environment variables are allowed.

For an example of this element, see the MigApp.xml file.

• DoesFileVersionMatch

The pattern check is case insensitive.

Syntax: DoesFileVersionMatch("EncodedFileLocation","VersionTag","VersionValue")

SETTING	REQUIRED?	VALUE
<i>EncodedFileLocation</i>	Yes	The location pattern for the file that will be checked. Environment variables are allowed.
<i>VersionTag</i>	Yes	The version tag value that will be checked.
<i>VersionValue</i>	Yes	A string pattern. For example, "Microsoft*".

For example:

```
<condition>MigXmlHelper.DoesFileVersionMatch("%MSNMessengerInstPath%\msnmsgr.exe", "ProductVersion", "6.*")</condition>
```

```
<condition>MigXmlHelper.DoesFileVersionMatch("%MSNMessengerInstPath%\msnmsgr.exe", "ProductVersion", "7.*")</condition>
```

• IsFileVersionAbove

The IsFileVersionAbove function returns TRUE if the version of the file is higher than *VersionValue*.

Syntax: IsFileVersionAbove("EncodedFileLocation","VersionTag","VersionValue")

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
<i>EncodedFileLocation</i>	Yes	The location pattern for the file that will be checked. Environment variables are allowed.
<i>VersionTag</i>	Yes	The version tag value that will be checked.
<i>VersionValue</i>	Yes	The value to compare to. You cannot specify a pattern.

- **IsFileVersionBelow**

Syntax: `IsFileVersionBelow("EncodedFileLocation","VersionTag","VersionValue")`

SETTING	REQUIRED?	VALUE
<i>EncodedFileLocation</i>	Yes	The location pattern for the file that will be checked. Environment variables are allowed.
<i>VersionTag</i>	Yes	The version tag value that will be checked.
<i>VersionValue</i>	Yes	The value to compare to. You cannot specify a pattern.

- **IsSystemContext**

The `IsSystemContext` function returns TRUE if the current context is "System". Otherwise, it returns FALSE.

Syntax: `IsSystemContext()`

- **DoesStringContentEqual**

The `DoesStringContentEqual` function returns TRUE if the string representation of the given object is identical to `StringContent`.

Syntax: `DoesStringContentEqual("ObjectType","EncodedLocation","StringContent")`

SETTING	REQUIRED?	VALUE
<i>ObjectType</i>	Yes	Defines the type of object. Can be File or Registry.

SETTING	REQUIRED?	VALUE
<i>EncodedLocationPattern</i>	Yes	The encoded location for the object that will be examined. You can specify environment variables.
StringContent	Yes	The string that will be checked against.

For example:

```

``` syntax
<condition negation="Yes">MigXmlHelper.DoesStringContentEqual("File", "%USERNAME%", "")</condition>
```

```

- **DoesStringContentContain**

The DoesStringContentContain function returns TRUE if there is at least one occurrence of *StrToFind* in the string representation of the object.

Syntax: DoesStringContentContain("*ObjectType*", "*EncodedLocation*", "*StrToFind*")

| SETTING | REQUIRED? | VALUE |
|-------------------------------|-----------|---|
| <i>ObjectType</i> | Yes | Defines the type of object. Can be File or Registry. |
| <i>EncodedLocationPattern</i> | Yes | The encoded location for the object that will be examined. You can specify environment variables. |
| <i>StrToFind</i> | Yes | A string that will be searched inside the content of the given object. |

- **IsSameObject**

The IsSameObject function returns TRUE if the given encoded locations resolve to the same physical object. Otherwise, it returns FALSE.

Syntax: IsSameObject("*ObjectType*", "*EncodedLocation1*", "*EncodedLocation2*")

| SETTING | REQUIRED? | VALUE |
|-------------------------|-----------|---|
| <i>ObjectType</i> | Yes | Defines the type of object. Can be File or Registry. |
| <i>EncodedLocation1</i> | Yes | The encoded location for the first object. You can specify environment variables. |

| SETTING | REQUIRED? | VALUE |
|-------------------------|-----------|--|
| <i>EncodedLocation2</i> | Yes | The encoded location for the second object. You can specify environment variables. |

For example:

```

` `` syntax
<objectSet>
  <condition
negation="Yes">MigXmlHelper.IsSameObject("File", "%CSIDL_FAVORITES%", "%CSIDL_COMMON_FAVORITES%")</condition>
  <pattern type="File">%CSIDL_FAVORITES%\* [*]</pattern>
</objectSet>
` ``

```

- **IsSameContent**

The IsSameContent function returns TRUE if the given objects have the same content. Otherwise, it returns FALSE. The content will be compared byte by byte.

Syntax: IsSameContent("Object Type1", "EncodedLocation1", "Object Type2", "EncodedLocation2")

| SETTING | REQUIRED? | VALUE |
|-------------------------|-----------|--|
| <i>Object Type1</i> | Yes | Defines the type of the first object. Can be File or Registry. |
| <i>EncodedLocation1</i> | Yes | The encoded location for the first object. You can specify environment variables. |
| <i>Object Type2</i> | Yes | Defines the type of the second object. Can be File or Registry. |
| <i>EncodedLocation2</i> | Yes | The encoded location for the second object. You can specify environment variables. |

- **IsSameStringContent**

The IsSameStringContent function returns TRUE if the given objects have the same content. Otherwise, it returns FALSE. The content will be interpreted as a string.

Syntax: IsSameStringContent("Object Type1", "EncodedLocation1", "Object Type2", "EncodedLocation2")

| SETTING | REQUIRED? | VALUE |
|---------------------|-----------|--|
| <i>Object Type1</i> | Yes | Defines the type of the first object. Can be File or Registry. |

| SETTING | REQUIRED? | VALUE |
|-------------------------|-----------|--|
| <i>EncodedLocation1</i> | Yes | The encoded location for the first object. You can specify environment variables. |
| <i>ObjectType2</i> | Yes | Defines the type of the second object. Can be File or Registry. |
| <i>EncodedLocation2</i> | Yes | The encoded location for the second object. You can specify environment variables. |

<conditions>

The <conditions> element returns a Boolean result that is used to specify the conditions in which the parent element is evaluated. USMT evaluates the child elements, and then joins their results using the operators AND or OR according to the **operation** parameter.

- **Number of occurrences:** Unlimited inside another <conditions> element. Limited to one occurrence in [<detection>](#), [<rules>](#), [<addObjects>](#), and [<objectSet>](#)
- **Parent elements:** [<conditions>](#), [<detection>](#), [<environment>](#), [<rules>](#), [<addObjects>](#), and [<objectSet>](#)
- **Child elements:** [<conditions>](#), [<condition>](#)

Syntax:

```
<conditions operation="AND|OR">
```

```
</conditions>
```

SETTING	REQUIRED?	VALUE
operation	No, default = AND	Defines the Boolean operation that is performed on the results that are obtained from the child elements.

The following example is from the MigApp.xml file:

```
<environment name="GlobalEnv">
  <conditions>
    <condition negation="Yes">MigXmlHelper.IsNative64Bit()</condition>
  </conditions>
  <variable name="HkLMWowSoftware">
    <text>HKLM\Software</text>
  </variable>
</environment>
```

<content>

You can use the <content> element to specify a list of object patterns to obtain an object set from the source computer. Each <objectSet> within a <content> element is evaluated. For each resulting object pattern list, the

objects that match it are enumerated and their content is filtered by the filter parameter. The resulting string array is the output for the <content> element. The filter script returns an array of locations. The parent <objectSet> element can contain multiple child <content> elements.

- **Number of occurrences:** unlimited
- **Parent elements:** <objectSet>
- **Child elements:** <objectSet>
- **Helper functions:** You can use the following <content> functions with this element: ExtractSingleFile, ExtractMultipleFiles, and ExtractDirectory.

Syntax:

```
<content filter="ScriptInvocation">
```

```
</content>
```

SETTING	REQUIRED?	VALUE
filter	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2")</pre> <p>The script is called for each object that is enumerated by the object sets in the <include> rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

<content> functions

The following functions generate patterns out of the content of an object. These functions are called for every object that the parent <ObjectSet> element is enumerating.

- **ExtractSingleFile**

If the registry value is a MULTI-SZ, only the first segment is processed. The returned pattern is the encoded location for a file that must exist on the system. If the specification is correct in the registry value, but the file does not exist, this function returns NULL.

Syntax: ExtractSingleFile(*Separators*,*PathHints*)

SETTING	REQUIRED?	VALUE
<i>Separators</i>	Yes	<p>A list of possible separators that might follow the file specification in this registry value name. For example, if the content is "C:\Windows\notepad.exe,-2", the separator is a comma. You can specify NULL.</p>

SETTING	REQUIRED?	VALUE
<i>PathHints</i>	Yes	A list of extra paths, separated by colons (:), where the function will look for a file matching the current content. For example, if the content is "Notepad.exe" and the path is the %Path% environment variable, the function will find Notepad.exe in %windir% and returns "c:\Windows [Notepad.exe]". You can specify NULL.

For example:

```

` `` syntax
<content filter="MigXmlHelper.ExtractSingleFile('','%system%')">
` ``

and

` `` syntax
<content filter="MigXmlHelper.ExtractSingleFile(NULL,'%CSIDL_COMMON_FONTS%')">
` ``

```

- **ExtractMultipleFiles**

The ExtractMultipleFiles function returns multiple patterns, one for each file that is found in the content of the given registry value. If the registry value is a MULTI-SZ, the MULTI-SZ separator is considered a separator by default. therefore, for MULTI-SZ, the <Separators> argument must be NULL.

The returned patterns are the encoded locations for files that must exist on the source computer. If the specification is correct in the registry value but the file does not exist, it will not be included in the resulting list.

Syntax: ExtractMultipleFiles(*Separators*,*PathHints*)

SETTING	REQUIRED?	VALUE
<i>Separators</i>	Yes	A list of possible separators that might follow the file specification in this registry value name. For example, if the content is "C:\Windows\Notepad.exe,-2", the separator is a comma. This parameter must be NULL when processing MULTI-SZ registry values.

SETTING	REQUIRED?	VALUE
<i>PathHints</i>	Yes	A list of extra paths, separated by colons (:), where the function will look for a file matching the current content. For example, if the content is "Notepad.exe" and the path is the %Path% environment variable, the function will find Notepad.exe in %windir% and returns "c:\Windows [Notepad.exe]". You can specify NULL.

- **ExtractDirectory**

The ExtractDirectory function returns a pattern that is the encoded location for a directory that must exist on the source computer. If the specification is correct in the registry value, but the directory does not exist, this function returns NULL. If it is processing a registry value that is a MULTI-SZ, only the first segment will be processed.

Syntax: ExtractDirectory(*Separators,LevelsTo Trim,PatternSuffix*)

SETTING	REQUIRED?	VALUE
<i>Separators</i>	No	A list of possible separators that might follow the file specification in this registry value name. For example, if the content is "C:\Windows\Notepad.exe,-2", the separator is a comma. You must specify NULL when processing MULTI-SZ registry values.
<i>LevelsTo Trim</i>	Yes	The number of levels to delete from the end of the directory specification. Use this function to extract a root directory when you have a registry value that points inside that root directory in a known location.
<i>PatternSuffix</i>	Yes	The pattern to add to the directory specification. For example, * [*] .

For example:

```
``` syntax
<objectSet>
 <content filter='MigXmlHelper.ExtractDirectory (NULL, "1")'>
 <objectSet>
 <pattern
type="Registry">%HklmWowSoftware%\Classes\Software\RealNetworks\Preferences\DT_Common []</pattern>
 </objectSet>
 </content>
</objectSet>
```
```

<contentModify>

The <contentModify> element modifies the content of an object before it is written to the destination computer. For each <contentModify> element there can be multiple <objectSet> elements. This element returns the new content of the object that is being processed.

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Required child elements:** <objectSet>
- **Helper functions:** You can use the following <contentModify> functions with this element: ConvertToDWORD, ConvertToString, ConvertToBinary, KeepExisting, OffsetValue, SetValueByTable, MergeMultiSzContent, and MergeDelimitedContent.

Syntax:

```
<contentModify script="ScriptInvocation">
</contentModify>
```

SETTING	REQUIRED?	VALUE
script	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the include rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

<contentModify> functions

The following functions change the content of objects as they are migrated. These functions are called for every object that the parent <ObjectSet> element is enumerating.

- **ConvertToDWORD**

The ConvertToDWORD function converts the content of registry values that are enumerated by the

parent <ObjectSet> element to a DWORD. For example, ConvertToDWORD will convert the string "1" to the DWORD 0x00000001. If the conversion fails, then the value of DefaultValueOnError will be applied.

Syntax: ConvertToDWORD(*DefaultValueOnError*)

SETTING	REQUIRED?	VALUE
<i>DefaultValueOnError</i>	No	The value that will be written into the value name if the conversion fails. You can specify NULL, and 0 will be written if the conversion fails.

- **ConvertToString**

The ConvertToString function converts the content of registry values that match the parent <ObjectSet> element to a string. For example, it will convert the DWORD 0x00000001 to the string "1". If the conversion fails, then the value of DefaultValueOnError will be applied.

Syntax: ConvertToString(*DefaultValueOnError*)

SETTING	REQUIRED?	VALUE
<i>DefaultValueOnError</i>	No	The value that will be written into the value name if the conversion fails. You can specify NULL, and 0 will be written if the conversion fails.

For example:

```

` `` syntax
<contentModify script="MigXmlHelper.ConvertToString('1')">
  <objectSet>
    <pattern type="Registry">HKCU\Control Panel\Desktop [ScreenSaveUsePassword]</pattern>
  </objectSet>
</contentModify>
` ``

```

- **ConvertToBinary**

The ConvertToBinary function converts the content of registry values that match the parent <ObjectSet> element to a binary type.

Syntax: ConvertToBinary ()

- **OffsetValue**

The OffsetValue function adds or subtracts *Value* from the value of the migrated object, and then writes the result back into the registry value on the destination computer. For example, if the migrated object is a DWORD with a value of 14, and the *Value* is "-2", the registry value will be 12 on the destination computer.

Syntax: OffsetValue(*Value*)

SETTING	REQUIRED?	VALUE
<i>Value</i>	Yes	The string representation of a numeric value. It can be positive or negative. For example, <code>offsetValue(2)</code> .

- **SetValueByTable**

The SetValueByTable function matches the value from the source computer to the source table. If the value is there, the equivalent value in the destination table will be applied. If the value is not there, or if the destination table has no equivalent value, the *DefaultValueOnError* will be applied.

Syntax: `SetValueByTable(SourceTable, DestinationTable, DefaultValueOnError)`

SETTING	REQUIRED?	VALUE
<i>SourceTable</i>	Yes	A list of values separated by commas that are possible for the source registry values.
<i>DestinationTable</i>	No	A list of translated values separated by commas.
<i>DefaultValueOnError</i>	No	The value that will be applied to the destination computer if either 1) the value for the source computer does not match <i>SourceTable</i> , or 2) <i>DestinationTable</i> has no equivalent value. If DefaultValueOnError is NULL, the value will not be changed on the destination computer.

- **KeepExisting**

You can use the KeepExisting function when there are conflicts on the destination computer. This function will keep (not overwrite) the specified attributes for the object that is on the destination computer.

Syntax: `KeepExisting("OptionString", "OptionString", "OptionString", ...)`

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
<i>OptionString</i>	Yes	<p><i>OptionString</i> can be Security, TimeFields, or FileAttrib:Letter. You can specify one of each type of <i>OptionStrings</i>. Do not specify multiple <i>OptionStrings</i> with the same value. If you do, the right-most option of that type will be kept. For example, do not specify ("FileAttrib:H", "FileAttrib:R") because only Read-only will be evaluated. Instead specify ("FileAttrib:HR") and both Hidden and Read-only attributes will be kept on the destination computer.</p> <ul style="list-style-type: none"> • Security. Keeps the destination object's security descriptor if it exists. • TimeFields. Keeps the destination object's time stamps. This parameter is for files only. • FileAttrib:Letter. Keeps the destination object's attribute value, either On or OFF, for the specified set of file attributes. This parameter is for files only. The following are case-insensitive, but USMT will ignore any values that are invalid, repeated, or if there is a space after "FileAttrib:". You can specify any combination of the following attributes: <ul style="list-style-type: none"> ◦ A = Archive ◦ C = Compressed ◦ E = Encrypted ◦ H = Hidden ◦ I = Not Content Indexed ◦ O = Offline ◦ R = Read-Only ◦ S = System ◦ T = Temporary

- **MergeMultiSzContent**

The MergeMultiSzContent function merges the MULTI-SZ content of the registry values that are

enumerated by the parent <ObjectSet> element with the content of the equivalent registry values that already exist on the destination computer. `Instruction` and `String` either remove or add content to the resulting MULTI-SZ. Duplicate elements will be removed.

Syntax: MergeMultiSzContent (*Instruction,String,Instruction,String,...*)

SETTING	REQUIRED?	VALUE
<i>Instruction</i>	Yes	Can be one of the following: <ul style="list-style-type: none"> • Add. Adds the corresponding String to the resulting MULTI-SZ if it is not already there. • Remove. Removes the corresponding String from the resulting MULTI-SZ.
<i>String</i>	Yes	The string to be added or removed.

• **MergeDelimitedContent**

The MergeDelimitedContent function merges the content of the registry values that are enumerated by the parent <ObjectSet> element with the content of the equivalent registry values that already exist on the destination computer. The content is considered a list of elements separated by one of the characters in the Delimiters parameter. Duplicate elements will be removed.

Syntax: MergeDelimitedContent(*Delimiters,Instruction,String,...*)

SETTING	REQUIRED?	VALUE
<i>Delimiters</i>	Yes	A single character that will be used to separate the content of the object that is being processed. The content will be considered as a list of elements that is separated by the <i>Delimiters</i> . For example, "." will separate the string based on a period.
<i>Instruction</i>	Yes	Can one of the following: <ul style="list-style-type: none"> • Add. Adds <i>String</i> to the resulting MULTI-SZ if it is not already there. • Remove. Removes <i>String</i> from the resulting MULTI-SZ.
<i>String</i>	Yes	The string to be added or removed.

<description>

The <description> element defines a description for the component but does not affect the migration.

- **Number of occurrences:** zero or one
- **Parent elements:** <component>
- **Child elements:** none

Syntax:

```
<description>ComponentDescription</description>
```

SETTING	REQUIRED?	VALUE
ComponentDescription	Yes	The description of the component.

The following code sample shows how the <description> element defines the "My custom component" description.:

```
<description>My custom component</description>
```

<destinationCleanup>

The <destinationCleanup> element deletes objects, such as files and registry keys, from the destination computer before applying the objects from the source computer. This element is evaluated only when the LoadState tool is run on the destination computer. That is, this element is ignored by the ScanState tool.

Important

Use this option with extreme caution because it will delete objects from the destination computer.

For each <destinationCleanup> element there can be multiple <objectSet> elements. A common use for this element is if there is a missing registry key on the source computer and you want to ensure that a component is migrated. In this case, you can delete all of the component's registry keys before migrating the source registry keys. This will ensure that if there is a missing key on the source computer, it will also be missing on the destination computer.

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Child elements:** <objectSet> (Note that the destination computer will delete all child elements.)

Syntax:

```
<destinationCleanup filter=ScriptInvocation>
```

```
</destinationCleanup>
```

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
filter	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the include rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

For example:

```
<destinationCleanup>
  <objectSet>
    <pattern type="Registry">HKCU\Software\Lotus\123\99.0\DDE Preferences\* [*]</pattern>
    <pattern type="Registry">HKCU\Software\Lotus\123\99.0\Find Preferences\* [*]</pattern>
  </objectSet>
</destinationCleanup>
```

<detect>

Although the <detect> element is still supported, we do not recommend using it because it may be deprecated in future versions of USMT. In that case, you would have to rewrite your scripts. Instead, we recommend that you use the [<detection> element](#).

You use the <detect> element to determine if the component is present on a system. If all child <detect> elements within a <detect> element resolve to TRUE, then the <detect> element resolves to TRUE. If any child <detect> elements resolve to FALSE, then their parent <detect> element resolves to FALSE. If there is no <detect> element section, then USMT will assume that the component is present.

For each <detect> element there can be multiple child <condition> or <objectSet> elements, which will be logically joined by an OR operator. If at least one <condition> or <objectSet> element evaluates to TRUE, then the <detect> element evaluates to TRUE.

- **Number of occurrences:** unlimited
- **Parent elements:** <detects>, [<namedElements>](#)
- **Required child elements:** [<condition>](#)
- **Optional child elements:** [<objectSet>](#)

Syntax:

```
<detect name="/ID" context="User|System|UserAndSystem">
```

```
</detect>
```

SETTING	REQUIRED?	VALUE
name	Yes, when <detect> is a child to <namedElements> No, when <detect> is a child to <detects>	When <i>ID</i> is specified, any child elements are not processed. Instead, any other <detect> elements with the same name that are declared within the <namedElements> element are processed.
context	No (default = UserAndSystem)	<p>Defines the scope of this parameter: whether to process this component in the context of the specific user, across the entire operating system, or both.</p> <p>The largest possible scope is set by the component element. For example, if a <component> element has a context of User, and a <rules> element had a context of UserAndSystem, then the <rules> element would act as though it had a context of User. If the <rules> element had a context of System, it would act as though the <rules> element were not there.</p> <ul style="list-style-type: none"> • User. Evaluates the variables for each user. • System. Evaluates the variables only once for the system. • UserAndSystem. Evaluates the variables for the entire operating system and each user.

For examples, see the examples for [<detection>](#).

<detects>

Although the <detects> element is still supported, we recommend that you do not use it because it may be deprecated in future versions of USMT, which would require you to rewrite your scripts. Instead, we recommend that you use the [<detection>](#) element if the parent element is <role> or <namedElements>, and we recommend that you use the <conditions> element if the parent element is <rules>. Using <detection> allows you to more clearly formulate complex Boolean statements.

The <detects> element is a container for one or more <detect> elements. If all of the child <detect> elements within a <detects> element resolve to TRUE, then <detects> resolves to TRUE. If any of the child <detect> elements resolve to FALSE, then <detects> resolves to FALSE. If you do not want to write the <detects> elements within a component, then you can create the <detects> element under the <namedElements> element, and then refer to it. If there is no <detects> element section, then USMT will assume that the component is present. The results from each <detects> element are joined together by the OR operator to form the rule used to detect the parent element.

Syntax:

<detects name="/ID" context="User|System|UserAndSystem">

</detects>

- **Number of occurrences:** Unlimited.
- **Parent elements:** <role>, <rules>, <namedElements>
- **Required child elements:** <detect>

SETTING	REQUIRED?	VALUE
name	Yes, when <detects> is a child to <namedElements> No, when <detects> is a child to <role> or <rules>	When <i>ID</i> is specified, no child <detect> elements are processed. Instead, any other <detects> elements with the same name that are declared within the <namedElements> element are processed.
context	No (default = UserAndSystem)	Defines the scope of this parameter: whether to process this component in the context of the specific user, across the entire operating system, or both. The largest possible scope is set by the <component element>. For example, if a <component> element has a context of User and a <rules> element had a context of UserAndSystem, then the <rules> element would act as though it had a context of User. If the <rules> element had a context of System, it would act as though the <rules> element were not there. <ul style="list-style-type: none">• User. Evaluates the variables for each user.• System. Evaluates the variables only once for the system.• UserAndSystem. Evaluates the variables for the entire operating system and each user. The context parameter is ignored for <detects> elements that are inside <rules> elements.

The following example is from the MigApp.xml file.

```

<detects>
  <detect>
    <condition>MigXmlHelper.DoesFileVersionMatch("%Lotus123InstPath%\123w.exe","ProductVersion","9.*")
  </condition>
</detect>
  <detect>

  <condition>MigXmlHelper.DoesFileVersionMatch("%SmartSuiteInstPath%\smartctr.exe","ProductVersion","99.*")
  </condition>
</detect>
</detects>

```

<detection>

The <detection> element is a container for one <conditions> element. The result of the child <condition> elements, located underneath the <conditions> element, determines the result of this element. For example, if all of the child <conditions> elements within the <detection> element resolve to TRUE, then the <detection> element resolves to TRUE. If any of the child <conditions> elements resolve to FALSE, then the <detection> element resolves to FALSE.

In addition, the results from each <detection> section within the <role> element are joined together by the OR operator to form the detection rule of the parent element. That is, if one of the <detection> sections resolves to TRUE, then the <role> element will be processed. Otherwise, the <role> element will not be processed.

Use the <detection> element under the <namedElements> element if you do not want to write it within a component. Then include a matching <detection> section under the <role> element to control whether the component is migrated. If there is not a <detection> section for a component, then USMT will assume that the component is present.

- **Number of occurrences:** Unlimited.
- **Parent elements:** <role>, <namedElements>
- **Child elements:** <conditions>

Syntax:

```
<detection name="/D" context="User|System|UserAndSystem">
```

```
</detection>
```

SETTING	REQUIRED?	VALUE
name	<ul style="list-style-type: none"> • Yes, when <detection> is declared under <namedElements> • Optional, when declared under <role> 	If declared, the content of the <detection> element is ignored and the content of the <detection> element with the same name that is declared in the <namedElements> element will be evaluated.

SETTING	REQUIRED?	VALUE
context	No, default = UserAndSystem	<p>Defines the scope of this parameter: whether to process this component in the context of the specific user, across the entire operating system, or both.</p> <ul style="list-style-type: none"> • User. Evaluates the component for each user. • System. Evaluates the component only once for the system. • UserAndSystem. Evaluates the component for the entire operating system and each user.

For example:

```
<detecion name="AdobePhotoshopCS">
  <conditions>
    <condition>MigXmlHelper.DoesObjectExist("Registry", "HKCU\Software\Adobe\Photoshop\8.0")</condition>

    <condition>MigXmlHelper.DoesFileVersionMatch("%PhotoshopSuite8Path%\Photoshop.exe", "FileVersion", "8.*")
  </condition>
  </conditions>
</detecion>
```

and

```
<role role="Settings">
  <detecion>
    <conditions>
      <condition>MigXmlHelper.DoesFileVersionMatch("%QuickTime5Exe%", "ProductVersion", "QuickTime 5.*")
    </condition>
      <condition>MigXmlHelper.DoesFileVersionMatch("%QuickTime5Exe%", "ProductVersion", "QuickTime 6.*")
    </condition>
    </conditions>
  </detecion>
```

<displayName>

The <displayName> element is a required field within each <component> element.

- **Number of occurrences:** once for each component
- **Parent elements:** <component>
- **Child elements:** none

Syntax:

```
<displayName _locID="ID">ComponentName</displayName>
```

SETTING	REQUIRED?	VALUE
locID	No	This parameter is for internal USMT use. Do not use this parameter.
<i>ComponentName</i>	Yes	The name for the component.

For example:

```
<displayName>Command Prompt settings</displayName>
```

<environment>

The <environment> element is a container for <variable> elements in which you can define variables to use in your .xml file. All environment variables defined this way will be private. That is, they will be available only for their child components and the component in which they were defined. For two example scenarios, see [Examples](#).

- **Number of occurrences:** unlimited
- **Parent elements:** <role>, <component>, <namedElements>
- **Required child elements:** <variable>
- **Optional child elements:** <conditions>

Syntax:

```
<environment name="ID" context="User|System|UserAndSystem">
</environment>
```

SETTING	REQUIRED?	VALUE
name	Yes, when <environment> is a child of <namedElements> No, when <environment> is a child of <role> or <component>	When declared as a child of the <role> or <component> elements, if <i>ID</i> is declared, USMT ignores the content of the <environment> element and the content of the <environment> element with the same name declared in the <namedElements> element is processed.

SETTING	REQUIRED?	VALUE
context	No (default = UserAndSystem)	<p>Defines the scope of this parameter: whether to process this component in the context of the specific user, across the entire operating system, or both.</p> <p>The largest possible scope is set by the <component> element. For example, if a <component> element has a context of User and a <rules> element had a context of UserAndSystem, then the <rules> element would act as though it had a context of User. If the <rules> element had a context of System, it would act as though <rules> were not there.</p> <ul style="list-style-type: none"> • User. Evaluates the variables for each user. • System. Evaluates the variables only once for the system. • UserAndSystem. Evaluates the variables for the entire operating system and each user.

Example scenario 1

In this scenario, you want to generate the location of objects at run time depending on the configuration of the destination computer. For example, you must do this if an application writes data in the directory where it is installed, and users can install the application anywhere on the computer. If the application writes a registry value `hklm\software\companyname\install [path]` and then updates this value with the location where the application is installed, then the only way for you to migrate the required data correctly is to define an environment variable. For example:

```
<environment>
  <variable name="INSTALLPATH">
    <script>MigXmlHelper.GetStringContent("Registry", "\software\companyname\install [path]")</script>
  </variable>
</environment>
```

Then you can use an include rule as follows. You can use any of the [<script> functions](#) to perform similar tasks.

```
<include>
  <objectSet>
    <pattern type="File">%INSTALLPATH%\ [*.*xyz]</pattern>
  </objectSet>
</include>
```

Second, you can also filter registry values that contain data that you need. The following example extracts the first string (before the separator ",") in the value of the registry `Hklm\software\companyname\application\ [Path]`.

```

<environment>
  <variable name="APPPATH">
    <objectSet>
      <content filter='MigXmlHelper.ExtractDirectory ("", "1")'>
        <objectSet>
          <pattern type="Registry">Hklm\software\companyname\application\ [Path]</pattern>
        </objectSet>
      </content>
    </objectSet>
  </variable>
</environment>

```

Example scenario 2:

In this scenario, you want to migrate five files named File1.txt, File2.txt, and so on, from %SYSTEMDRIVE%\data\userdata\dir1\dir2\. To do this you must have the following <include> rule in an .xml file:

```

<include>
  <objectSet>
    <pattern type="File">%SYSTEMDRIVE%\data\userdata\dir1\dir2 [File1.txt]</pattern>
    <pattern type="File">%SYSTEMDRIVE%\data\userdata\dir1\dir2 [File2.txt]</pattern>
    <pattern type="File">%SYSTEMDRIVE%\data\userdata\dir1\dir2 [File3.txt]</pattern>
    <pattern type="File">%SYSTEMDRIVE%\data\userdata\dir1\dir2 [File4.txt]</pattern>
    <pattern type="File">%SYSTEMDRIVE%\data\userdata\dir1\dir2 [File5.txt]</pattern>
  </objectSet>
</include>

```

Instead of typing the path five times, you can create a variable for the location as follows:

```

<environment>
  <variable name="DATAPATH">
    <text>%SYSTEMDRIVE%\data\userdata\dir1\dir2 </text>
  </variable>
</environment>

```

Then, you can specify the variable in an <include> rule as follows:

```

<include>
  <objectSet>
    <pattern type="File">%DATAPATH% [File1.txt]</pattern>
    <pattern type="File">%DATAPATH% [File2.txt]</pattern>
    <pattern type="File">%DATAPATH% [File3.txt]</pattern>
    <pattern type="File">%DATAPATH% [File4.txt]</pattern>
    <pattern type="File">%DATAPATH% [File5.txt]</pattern>
  </objectSet>
</include>

```

<exclude>

The <exclude> element determines what objects will not be migrated, unless there is a more specific <include> element that migrates an object. If there is an <include> and <exclude> element for the same object, the object will be included. For each <exclude> element there can be multiple child <objectSet> elements.

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Child elements:** <objectSet>

- **Helper functions:** You can use the following [<exclude> filter functions](#) with this element: CompareStringContent, IgnoreIrrelevantLinks, AnswerNo, NeverRestore, and SameRegContent.

Syntax:

```
<exclude filter="ScriptInvocation">
```

```
</exclude>
```

SETTING	REQUIRED?	VALUE
filter	No (default = No)	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the include rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

For example, from the MigUser.xml file:

```
<exclude>
  <objectSet>
    <pattern type="File">%CSIDL_MYMUSIC%\* [*]</pattern>
    <pattern type="File">%CSIDL_MYPICTURES%\* [*]</pattern>
    <pattern type="File">%CSIDL_MYVIDEO%\* [*]</pattern>
  </objectSet>
</exclude>
```

<excludeAttributes>

You can use the <excludeAttributes> element to determine which parameters associated with an object will not be migrated. If there are conflicts between the <includeAttributes> and <excludeAttributes> elements, the most specific pattern determines the patterns that will not be migrated. If an object does not have an <includeAttributes> or <excludeAttributes> element, then all of its parameters will be migrated.

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Child elements:** <objectSet>

Syntax:

```
<excludeAttributes attributes="Security|TimeFields|Security,TimeFields">
```

```
</excludeAttributes>
```

PARAMETER	REQUIRED?	VALUE
attributes	Yes	<p>Specifies the attributes to be excluded. You can specify one of the following, or both separated by quotes; for example, "Security", "TimeFields" :</p> <ul style="list-style-type: none">• Security can be one of Owner, Group, DACL, or SACL.• TimeFields can be one of CreationTime, LastAccessTime and LastWrittenTime

Example:

```

<migration urlid="http://www.microsoft.com/migration/1.0/migxmlex/miguser">
<!-- This component migrates My Video files -->
  <component type="System" context="System">
    <displayName>System Data</displayName>
    <role role="Data">
      <rules>
<!-- Include all of the text files, which are immediately in the drive where the operating system is
installed -->
        <include>
          <objectSet>
            <pattern type="File">%SYSTEMDRIVE%\ [*.*txt]</pattern>
          </objectSet>
        </include>
<!-- Exclude the time stamps from the text file starting with the letter a -->
        <excludeAttributes attributes="TimeFields">
          <objectSet>
            <pattern type="File">%SYSTEMDRIVE%\ [a*.*txt]</pattern>
          </objectSet>
        </excludeAttributes>
<!-- include the time stamps from the text file aa.txt -->
        <includeAttributes attributes="TimeFields">
          <objectSet>
            <pattern type="File">%SYSTEMDRIVE%\ [aa.*txt]</pattern>
          </objectSet>
        </includeAttributes>
<!-- Logoff the user after loadstate successfully completed. -->
        <externalProcess when="post-apply">
          <commandLine>
            logoff
          </commandLine>
        </externalProcess>
      </rules>
    </role>
  <!-- Migrate
  all doc files from the system
  all power point files
  all visio design files
  all my c++ program files -->
  <extensions>
    <extension>DOC</extension>
    <extension>PPT</extension>
    <extension>VXD</extension>
    <extension>PST</extension>
    <extension>CPP</extension>
  </extensions>
</component>
</migration>

```

<extensions>

The <extensions> element is a container for one or more <extension> elements.

- **Number of occurrences:** zero or one
- **Parent elements:** <component>
- **Required child elements:** <extension>

Syntax:

```
<extensions>
```

```
</extensions>
```

<extension>

You can use the <extension> element to specify documents of a specific extension.

- **Number of occurrences:** unlimited
- **Parent elements:** <extensions>
- **Child elements:** none

Syntax:

```
<extension>FilenameExtension</extension>
```

SETTING	REQUIRED?	VALUE
FilenameExtension	Yes	A file name extension.

For example, if you want to migrate all *.doc files from the source computer, specifying the following code under the <component> element:

```
<extensions>
  <extension>doc</extension>
</extensions>
```

is the same as specifying the following code below the <rules> element:

```
<include>
  <objectSet>
    <script>MigXmlHelper.GenerateDrivePatterns ("* [*].doc", "Fixed")</script>
  </objectSet>
</include>
```

For another example of how to use the <extension> element, see the example for <excludeAttributes>.

<externalProcess>

You can use the <externalProcess> element to run a command line during the migration process. For example, you may want to run a command after the LoadState process completes.

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Required child elements:** <commandLine>

Syntax:

```
<externalProcess when="pre-scan|scan-success|post-scan|pre-apply|apply-success|post-apply">
</externalProcess>
```

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
when	Yes	<p>Indicates when the command line should be run. This value can be one of the following:</p> <ul style="list-style-type: none"> • pre-scan before the scanning process begins. • scan-success after the scanning process has finished successfully. • post-scan after the scanning process has finished, whether it was successful or not. • pre-apply before the apply process begins. • apply-success after the apply process has finished successfully. • post-apply after the apply process has finished, whether it was successful or not.

For an example of how to use the `<externalProcess>` element, see the example for [<excludeAttributes>](#).

<icon>

This is an internal USMT element. Do not use this element.

<include>

The `<include>` element determines what to migrate, unless there is a more specific [<exclude>](#) rule. You can specify a script to be more specific to extend the definition of what you want to collect. For each `<include>` element there can be multiple `<objectSet>` elements.

- **Number of occurrences:** Unlimited
- **Parent elements:** [<rules>](#)
- **Required child element:** [<objectSet>](#)
- **Helper functions:** You can use the following [<include>](#) [filter functions](#) with this element: CompareStringContent, IgnoreIrrelevantLinks, AnswerNo, and NeverRestore.

Syntax:

```
<include filter="ScriptInvocation">
```

```
</include>
```

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
filter	No. If this parameter is not specified, then all patterns that are inside the child <ObjectSet> element will be processed.	A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example <pre style="border: 1px solid gray; padding: 5px;">, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the <include> rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

The following example is from the MigUser.xml file:

```
<component type="Documents" context="User">
  <displayName _locID="miguser.myvideo">My Video</displayName>
  <paths>
    <path type="File">%CSIDL_MYVIDEO%</path>
  </paths>
  <role role="Data">
    <detects>
      <detect>
        <condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_MYVIDEO%")</condition>
      </detect>
    </detects>
    <rules>
      <include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
        <objectSet>
          <pattern type="File">%CSIDL_MYVIDEO%\* [*]</pattern>
        </objectSet>
      </include>
      <merge script="MigXmlHelper.DestinationPriority() ">
        <objectSet>
          <pattern type="File">%CSIDL_MYVIDEO% [desktop.ini]</pattern>
        </objectSet>
      </merge>
    </rules>
  </role>
</component>
```

<include> and <exclude> filter functions

The following functions return a Boolean value. You can use them to migrate certain objects based on when certain conditions are met.

- **AnswerNo**

This filter always returns FALSE.

Syntax: AnswerNo ()

- **CompareStringContent**

Syntax: CompareStringContent("StringContent", "CompareType")

SETTING	REQUIRED?	VALUE
<i>StringContent</i>	Yes	The string to check against.
<i>CompareType</i>	Yes	<p>A string. Use one of the following values:</p> <ul style="list-style-type: none"> • Equal (case insensitive). The function returns TRUE if the string representation of the current object that is processed by the migration engine is identical to <code>StringContent</code>. • NULL or any other value. The function returns TRUE if the string representation of the current object that is processed by the migration engine does not match <code>StringContent</code>.

- **IgnoreIrrelevantLinks**

This filter screens out the .lnk files that point to an object that is not valid on the destination computer. Note that the screening takes place on the destination computer, so all .lnk files will be saved to the store during ScanState. Then they will be screened out when you run the LoadState tool.

Syntax: IgnoreIrrelevantLinks ()

For example:

```
<include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
  <objectSet>
    <pattern type="File"%CSIDL_COMMON_VIDEOS% \* [*]</pattern>
  </objectSet>
</include>
```

- **NeverRestore**

You can use this function to collect the specified objects from the source computer but then not migrate the objects to the destination computer. When run with the ScanState tool, this function evaluates to TRUE. When run with the LoadState tool, this function evaluates to FALSE. You may want to use this function when you want to check an object's value on the destination computer but do not intend to migrate the object to the destination.

Syntax: NeverRestore()

In the following example, HKCU\Control Panel\International [Locale] will be included in the store, but it will not be migrated to the destination computer:

```
<include filter="MigXmlHelper.NeverRestore()">
  <objectSet>
    <pattern type="Registry">HKCU\Control Panel\International [Locale]</pattern>
  </objectSet>
</include>
```

<includeAttributes>

You can use the <includeAttributes> element to determine whether certain parameters associated with an object will be migrated along with the object itself. If there are conflicts between the <includeAttributes> and <excludeAttributes> elements, the most specific pattern will determine which parameters will be migrated. If an object does not have an <includeAttributes> or <excludeAttributes> element, then all of its parameters will be migrated.

- **Number of occurrences:** unlimited
- **Parent elements:** <rules>
- **Child elements:** <objectSet>

Syntax:

```
<includeAttributes attributes="Security|TimeFields|Security,TimeFields">
```

```
</includeAttributes>
```

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
attributes	Yes	<p>Specifies the attributes to be included with a migrated object. You can specify one of the following, or both separated by quotes; for example, <code>"Security", "TimeFields"</code> :</p> <ul style="list-style-type: none"> • Security can be one of the following values: <ul style="list-style-type: none"> ◦ Owner. The owner of the object (SID). ◦ Group. The primary group for the object (SID). ◦ DAACL (discretionary access control list). An access control list that is controlled by the owner of an object and that specifies the access particular users or groups can have to the object. ◦ SACL (system access control list). An ACL that controls the generation of audit messages for attempts to access a securable object. The ability to get or set an object's SACL is controlled by a privilege typically held only by system administrators. • TimeFields can be one of the following: <ul style="list-style-type: none"> ◦ CreationTime. Specifies when the file or directory was created. ◦ LastAccessTime. Specifies when the file is last read from, written to, or, in the case of executable files, run. ◦ LastWrittenTime. Specifies when the file is last written to, truncated, or overwritten.

For an example of how to use the `<includeAttributes>` element, see the example for [<excludeAttributes>](#).

<library>

This is an internal USMT element. Do not use this element.

<location>

The `<location>` element defines the location of the `<object>` element.

- **Number of occurrences:** once for each `<object>`
- **Parent elements:** [<object>](#)
- **Child elements:** [<script>](#)

Syntax:

```
<location type="typeID">ObjectLocation</location>
```

SETTING	REQUIRED?	VALUE
type	Yes	<i>typeID</i> can be Registry or File.
<i>ObjectLocation</i>	Yes	The location of the object.

The following example is from the MigApp.xml file:

```
<addObjects>
  <object>
    <location type="Registry">%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office
[UpgradeVersion]</location>
    <attributes>DWORD</attributes>
    <bytes>0B000000</bytes>
  </object>
  <object>
    <location type="Registry">%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office [Lang]
</location>
    <attributes>DWORD</attributes>
    <bytes>00000000</bytes>
  </object>
</addObjects>
```

<locationModify>

You can use the `<locationModify>` element to change the location and name of an object before it is migrated to the destination computer. The `<locationModify>` element is processed only when the LoadState tool is run on the destination computer. In other words, this element is ignored by the ScanState tool. The `<locationModify>` element will create the appropriate folder on the destination computer if it does not already exist.

Number of occurrences: Unlimited

- **Parent elements:** [<rules>](#)
- **Required child element:** [<objectSet>](#)
- **Helper functions:** You can use the following [<locationModify> functions](#) with this element: ExactMove, RelativeMove, and Move.

Syntax:

```
<locationModify script="ScriptInvocation">
```

```
</locationModify>
```

SETTING	REQUIRED?	VALUE
script	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the include rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

The following example is from the MigApp.xml file:

```
<locationModify script="MigXmlHelper.RelativeMove('%CSIDL_APPDATA%\Microsoft\Office', '%CSIDL_APPDATA%')">  
  <objectSet>  
    <pattern type="File">%CSIDL_APPDATA%\Microsoft\Office\ [Access10.pip]</pattern>  
  </objectSet>  
</locationModify>
```

<locationModify> functions

The following functions change the location of objects as they are migrated when using the <locationModify> element. These functions are called for every object that the parent <ObjectSet> element is enumerating. The <locationModify> element will create the appropriate folder on the destination computer if it does not already exist.

- **ExactMove**

The ExactMove function moves all of the objects that are matched by the parent <ObjectSet> element into the given *ObjectEncodedLocation*. You can use this function when you want to move a single file to a different location on the destination computer. If the destination location is a node, all of the matching source objects will be written to the node without any subdirectories. If the destination location is a leaf, the migration engine will migrate all of the matching source objects to the same location. If a collision occurs, the normal collision algorithms will apply.

Syntax: ExactMove(*ObjectEncodedLocation*)

SETTING	REQUIRED?	VALUE
<i>ObjectEncodedLocation</i>	Yes	The destination location for all of the source objects.

For example:

```
``` syntax
<locationModify script="MigXmlHelper.ExactMove('HKCU\Keyboard Layout\Toggle [HotKey]')">
 <objectSet>
 <pattern type="Registry">HKCU\Keyboard Layout\Toggle []</pattern>
 </objectSet>
</locationModify>
```
```

- **Move**

The Move function moves objects to a different location on the destination computer. In addition, this function creates subdirectories that were above the longest CSIDL in the source object name.

Syntax: Move(*DestinationRoot*)

| SETTING | REQUIRED? | VALUE |
|------------------------|-----------|---|
| <i>DestinationRoot</i> | Yes | The location where the source objects will be moved. If needed, this function will create any subdirectories that were above the longest CSIDL in the source object name. |

- **RelativeMove**

You can use the RelativeMove function to collect and move data. Note that you can use environment variables in source and destination roots, but they may be defined differently on the source and destination computers.

Syntax: RelativeMove(*SourceRoot*,*DestinationRoot*)

| SETTING | REQUIRED? | VALUE |
|------------------------|-----------|---|
| <i>SourceRoot</i> | Yes | The location from where the objects will be moved. Any source objects that are enumerated by the parent <ObjectSet> element that are not in this location will not be moved. |
| <i>DestinationRoot</i> | Yes | The location where the source objects will be moved to on the destination computer. If needed, this function will create any subdirectories that were above <i>SourceRoot</i> . |

For example:

```
```` syntax
<include>
 <objectSet>
 <pattern type="File">%CSIDL_COMMON_FAVORITES%* [*]</pattern>
 </objectSet>
</include>
<locationModify script="MigXmlHelper.RelativeMove('%CSIDL_COMMON_FAVORITES%', '%CSIDL_COMMON_FAVORITES%')">
 <objectSet>
 <pattern type="File">%CSIDL_COMMON_FAVORITES%* [*]</pattern>
 </objectSet>
</locationModify>
````
```

<_locDefinition>

This is an internal USMT element. Do not use this element.

<manufacturer>

The <manufacturer> element defines the manufacturer for the component, but does not affect the migration.

- **Number of occurrences:** zero or one
- **Parent elements:** <component>
- **Child elements:** none

Syntax:

```
<manufacturer>Name</manufacturer>
```

| SETTING | REQUIRED? | VALUE |
|-------------|-----------|---|
| <i>Name</i> | Yes | The name of the manufacturer for the component. |

<merge>

The <merge> element determines what will happen when a collision occurs. A collision is when an object that is migrated is already present on the destination computer. If you do not specify this element, the default behavior for the registry is for the source object to overwrite the destination object. The default behavior for files is for the source file to be renamed to "OriginalFileName(1).OriginalExtension". This element specifies only what should be done when a collision occurs. It does not include objects. Therefore, for your objects to migrate, you must specify <include> rules along with the <merge> element. When an object is processed and a collision is detected, USMT will select the most specific merge rule and apply it to resolve the conflict. For example, if you have a <merge> rule C:* [*] set to <sourcePriority> and a <merge> rule C:\subfolder* [*] set to <destinationPriority>, then USMT would use the <destinationPriority> rule because it is the more specific.

For an example of this element, see [Conflicts and Precedence](#).

- **Number of occurrences:** Unlimited
- **Parent elements:** <rules>
- **Required child element:** <objectSet>

- **Helper functions:** You can use the following [merge functions](#) with this element: SourcePriority, DestinationPriority, FindFilePlaceByPattern, LeafPattern, NewestVersion, HigherValue(), and LowerValue().

Syntax:

```
<merge script="ScriptInvocation">
```

```
</merge>
```

SETTING	REQUIRED?	VALUE
script	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre>, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the <include> rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p>

The following example is from the MigUser.xml file:

```
<rules>
  <include filter='MigXmlHelper.IgnoreIrrelevantLinks()'\>
    <objectSet>
      <pattern type="File">%CSIDL_MYVIDEO%* [*]</pattern>
    </objectSet>
  </include>
  <merge script="MigXmlHelper.DestinationPriority()"\>
    <objectSet>
      <pattern type="File">%CSIDL_MYVIDEO% [desktop.ini]</pattern>
    </objectSet>
  </merge>
</rules>
```

<merge> functions

These functions control how collisions are resolved.

- **DestinationPriority**

Specifies to keep the object that is on the destination computer and not migrate the object from the source computer.

For example:

```

<merge script="MigXmlHelper.DestinationPriority(">
  <objectSet>
    <pattern type="Registry">HKCU\Software\Microsoft\Office\9.0\PhotoDraw\ [MyPictures]
  </pattern>
    <pattern type="Registry">HKCU\Software\Microsoft\Office\9.0\PhotoDraw\Settings\
  [PicturesPath]</pattern>
    <pattern type="Registry">HKCU\Software\Microsoft\Office\9.0\PhotoDraw\Settings\
  [AdditionalPlugInPath]</pattern>
  </objectSet>
</merge>

```

● **FindFilePlaceByPattern**

The FindFilePlaceByPattern function saves files with an incrementing counter when a collision occurs. It is a string that contains one of each constructs: <F>, <E>, <N> in any order.

Syntax: FindFilePlaceByPattern(*FilePattern*)

SETTING	REQUIRED?	VALUE
<i>FilePattern</i>	Yes	<ul style="list-style-type: none"> • <F> will be replaced by the original file name. • <N> will be replaced by an incrementing counter until there is no collision with the objects on the destination computer. • <E> will be replaced by the original file name extension. <p>For example, <F> (<N>).<E> will change the source file MyDocument.doc into MyDocument (1).doc on the destination computer.</p>

● **NewestVersion**

The NewestVersion function will resolve conflicts on the destination computer based on the version of the file.

Syntax: NewestVersion(*VersionTag*)

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
<i>VersionTag</i>	Yes	The version field that will be checked. This can be "FileVersion" or "ProductVersion". The file with the highest <i>VersionTag</i> version determines which conflicts will be resolved based on the file's version. For example, if Myfile.txt contains FileVersion 1 and the same file on the destination computer contains FileVersion 2, the file on destination will remain.

- **HigherValue()**

You can use this function for merging registry values. The registry values will be evaluated as numeric values, and the one with the higher value will determine which registry values will be merged.

- **LowerValue()**

You can use this function for merging registry values. The registry values will be evaluated as numeric values and the one with the lower value will determine which registry values will be merged.

- **SourcePriority**

Specifies to migrate the object from the source computer, and to delete the object that is on the destination computer.

For example:

```
<merge script="MigXmlHelper.SourcePriority()">
  <objectSet>
    <pattern type="Registry">%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Publisher
[UpgradeVersion]</pattern>
    <pattern type="Registry">%Hk1mWowSoftware%\Microsoft\Office\11.0\Common\Migration\Publisher
[UpgradeVersion]</pattern>
    <pattern type="Registry">%Hk1mWowSoftware%\Microsoft\Office\10.0\Common\Migration\Publisher
[UpgradeVersion]</pattern>
  </objectSet>
</merge>
```

<migration>

The <migration> element is the single root element of a migration .xml file and is required. Each .xml file must have a unique migration urlid. The urlid of each file that you specify on the command line must be unique. This is because USMT uses the urlid to define the components within the file. For example, you must specify the following at the beginning of each file: <CustomFileName> is the name of the file; for example, "CustomApp".

- **Number of occurrences:** one
- **Parent elements:** none
- **Required child elements:** <component>
- **Optional child elements:** <library>, <namedElements>

Syntax:

<migration urlid="UrlID/Name">

</migration>

SETTING	REQUIRED?	VALUE
urlid	Yes	<i>UrlID</i> is a string identifier that uniquely identifies this .xml file. This parameter must be a no-colon-name as defined by the XML Namespaces specification. Each migration .xml file must have a unique urlid. If two migration .xml files have the same urlid, the second .xml file that is specified on the command line will not be processed. For more information about XML Namespaces, see Use XML Namespaces .
Name	No	Although not required, it is good practice to use the name of the .xml file.

The following example is from the MigApp.xml file:

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/migapp">
</migration>
```

MigXMLHelper.FileProperties

This filter helper function can be used to filter the migration of files based on file size and date attributes.

HELPER FUNCTION	MIGXMLHELPER.FILEPROPERTIES (PROPERTY, OPERATOR, VALUETOCOMPARE)
Property	filesize, dateCreated, dateModified, dateAccessed
Operator	range, neq, lte, lt, eq, gte, gt
valueToCompare	The value we are comparing. For example: Date: "2008/05/15-2005/05/17", "2008/05/15" Size: A numeral with B, KB, MB, or GB at the end. "5GB", "1KB-1MB"

```
<component context="System" type="Application">
<displayName>File_size</displayName>
<role role="Data">

  <rules>
    <include filter='MigXmlHelper.FileProperties("dateAccessed","range","2008/05/15-2008/05/17")'>
      <objectSet>
        <pattern type="File">%SYSTEMDRIVE%\DOCS\* [*]</pattern>
      </objectSet>
    </include>
  </rules>
</role>
</component>
```

<namedElements>

You can use the **<namedElements>** element to define named elements. You can use these elements in any component throughout your .xml file. For an example of how to use this element, see the MigApp.xml file.

Syntax:

```
<namedElements>
</namedElements>
```

- **Number of occurrences:** Unlimited
- **Parent elements:** [<migration>](#)
- **Child elements:** [<environment>](#), [<rules>](#), [<conditions>](#), [<detection>](#), [<detects>](#), [<detect>](#)

For an example of this element, see the MigApp.xml file.

<object>

The **<object>** element represents a file or registry key.

- **Number of occurrences:** Unlimited
- **Parent elements:** [<addObjects>](#)
- **Required child elements:** [<location>](#), [<attributes>](#)
- **Optional child elements:** [<bytes>](#)

Syntax:

```
<object>
</object>
```

The following example is from the MigApp.xml file:

```

<addObjects>
  <object>
    <location type="Registry">%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office
[UpgradeVersion]</location>
    <attributes>DWORD</attributes>
    <bytes>0B000000</bytes>
  </object>
  <object>
    <location type="Registry">%Hk1mWowSoftware%\Microsoft\Office\12.0\Common\Migration\Office [Lang]
</location>
    <attributes>DWORD</attributes>
    <bytes>00000000</bytes>
  </object>
</addObjects>

```

<objectSet>

The <objectSet> element contains a list of object patterns ; for example, file paths, registry locations, and so on. Any child <conditions> elements will be evaluated first. If all child <conditions> elements return FALSE, the <objectSet> element will evaluate to an empty set. For each parent element, there can be only multiple <objectSet> elements.

- **Number of occurrences:** Unlimited
- **Parent elements:** <variable>, <content>, <include>, <exclude>, <merge>, <contentModify>, <locationModify>, <destinationCleanup>, <includeAttributes>, <excludeAttributes>, <unconditionalExclude>, <detect>
- **Required child elements:** either <script> or <pattern>
- **Optional child elements:** <content>, <conditions>, <condition>

Syntax:

```
<objectSet>
```

```
</objectSet>
```

The following example is from the MigUser.xml file:

```

<component type="Documents" context="User">
  <displayName _locID="miguser.mymusic">My Music</displayName>
  <paths>
    <path type="File">%CSIDL_MYMUSIC%</path>
  </paths>
  <role role="Data">
    <detects>
      <detect>
        <condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_MYMUSIC%")</condition>
      </detect>
    </detects>
  </role>
  <rules>
    <include filter='MigXmlHelper.IgnoreIrrelevantLinks()''>
      <objectSet>
        <pattern type="File">%CSIDL_MYMUSIC%\* [*]</pattern>
      </objectSet>
    </include>
    <merge script="MigXmlHelper.DestinationPriority()">
      <objectSet>
        <pattern type="File">%CSIDL_MYMUSIC%\ [desktop.ini]</pattern>
      </objectSet>
    </merge>
  </rules>
</component>

```

<path>

This is an internal USMT element. Do not use this element.

<paths>

This is an internal USMT element. Do not use this element.

<pattern>

You can use this element to specify multiple objects. You can specify multiple <pattern> elements for each <objectSet> element and they will be combined. If you are specifying files, you may want to use GenerateDrivePatterns with <script> instead. GenerateDrivePatterns is basically the same as a <pattern> rule, without the drive letter specification. For example, the following two lines of code are similar:

```

<pattern type="File">C:\Folder\* [Sample.doc]</pattern>
<script>MigXmlHelper.GenerateDrivePatterns("\Folder\* [Sample.doc]", "Fixed"</script>

```

- **Number of occurrences:** Unlimited
- **Parent elements:** <objectSet>
- **Child elements:** none but *Path [object]* must be valid.

Syntax:

```
<pattern type="typeID">Path [object]</pattern>
```

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
type	Yes	<p><i>typeID</i> can be Registry, File, or Ini. If <i>typeID</i> is Ini, then you cannot have a space between <i>Path</i> and <i>object</i>. For example, the following is correct when type="Ini":</p> <pre><pattern type="Ini">%WinAmp5InstPath %\Winamp.ini WinAmp[keepon screen]</pattern></pre>
<i>Path</i> [<i>object</i>]	Yes	<p>A valid registry or file path pattern, followed by at least one space, followed by brackets [] that contain the object to be migrated.</p> <ul style="list-style-type: none"> <i>Path</i> can contain the asterisk () wildcard character or can be an Recognized Environment Variables. You cannot use the question mark as a wildcard character. You can use HKCU and HKLM to refer to HKEY_CURRENT_USER and HKEY_LOCAL_MACHINE respectively. <i>Object</i> can contain the asterisk () wildcard character. However, you cannot use the question mark as a wildcard character. For example: <ul style="list-style-type: none"> C:\Folder\ [] enumerates all files in C:Path but no subfolders of C:\Folder. C:\Folder* [] enumerates all files and subfolders of C:\Folder. C:\Folder\ [*.*mp3] enumerates all .mp3 files in C:\Folder. C:\Folder\ [Sample.doc] enumerates only the Sample.doc file located in C:\Folder.

SETTING	REQUIRED?	VALUE
		<div data-bbox="1082 76 1396 721" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>If you are migrating a file that has a square bracket character ([or]) in the file name, you must insert the caret (^) character directly before the bracket for it to be valid. For example, if there is a file named "file.txt", you must specify</p> <pre data-bbox="1107 488 1369 555"><pattern type="File">c:\documents\mydocs [file^].txt</pattern></pre> <p>instead of</p> <pre data-bbox="1107 591 1369 658"><pattern type="File">c:\documents\mydocs [file].txt</pattern></pre> </div>

For example:

- To migrate a single registry key:

```
<pattern type="Registry">HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache [Persistent]</pattern>
```

- To migrate the EngineeringDrafts folder and any subfolders from the C: drive:

```
<pattern type="File">C:\EngineeringDrafts\* [*]</pattern>
```

- To migrate only the EngineeringDrafts folder, excluding any subfolders, from the C: drive:

[Reroute Files and Settings](#)

- To migrate the Sample.doc file from C:\EngineeringDrafts:

```
<pattern type="File"> C:\EngineeringDrafts\ [Sample.doc]</pattern>
```

- To migrate the Sample.doc file from where ever it exists on the C: drive use pattern in the following way. If multiple files exist with the same name on the C: drive, then all of these files will be migrated.

```
<pattern type="File"> C:\* [Sample.doc] </pattern>
```

- For more examples of how to use this element, see [Exclude Files and Settings](#), [Reroute Files and Settings](#), [Include Files and Settings](#), and [Custom XML Examples](#).

<processing>

You can use this element to run a script during a specific point within the migration process. Return values are not expected from the scripts that you specify, and if there are return values, they will be ignored.

- **Number of occurrences:** unlimited
- **Parent elements:** [<rules>](#)

- **Required child element:** [<script>](#)

Syntax:

```
<processing when="pre-scan|scan-success|post-scan|pre-apply|apply-success|post-apply">
</processing>
```

SETTING	REQUIRED?	VALUE
when	Yes	<p>Indicates when the script should be run. This value can be one of the following:</p> <ul style="list-style-type: none"> • pre-scan means before the scanning process begins. • scan-success means after the scanning process has finished successfully. • post-scan means after the scanning process has finished, whether it was successful or not. • pre-apply means before the apply process begins. • apply-success means after the apply process has finished successfully. • post-apply means after the apply process has finished, whether it was successful or not.

<plugin>

This is an internal USMT element. Do not use this element.

<role>

The <role> element is required in a custom .xml file. By specifying the <role> element, you can create a concrete component. The component will be defined by the parameters specified at the <component> level, and with the role that you specify here.

- **Number of occurrences:** Each <component> can have one, two or three child <role> elements.
- **Parent elements:** [<component>](#), [<role>](#)
- **Required child elements:** [<rules>](#)
- **Optional child elements:** [<environment>](#), [<detection>](#), [<component>](#), [<role>](#), [<detects>](#), [<plugin>](#),

Syntax:

```
<role role="Container|Binaries|Settings|Data">
</role>
```

SETTING	REQUIRED?	VALUE
role	Yes	<p>Defines the role for the component. Role can be one of:</p> <ul style="list-style-type: none"> • Container • Binaries • Settings • Data <p>You can either:</p> <ol style="list-style-type: none"> 1. Specify up to three <role> elements within a <component> — one "Binaries" role element, one "Settings" role element and one "Data" role element. These parameters do not change the migration behavior — their only purpose is to help you categorize the settings that you are migrating. You can nest these <role> elements, but each nested element must be of the same role parameter. 2. Specify one "Container" <role> element within a <component> element. In this case, you cannot specify any child <rules> elements, only other <component> elements. And each child <component> element must have the same type as that of parent <component> element. For example: <pre data-bbox="1050 1482 1437 2107"> <component context="UserAndSystem" type="Application"> <displayName _locID="migapp.msoffice2003" >Microsoft Office 2003</displayName> <environment name="GlobalEnv" /> <role role="Container"> <detection name="AnyOffice2003Version" /> <detection name="FrontPage2003" /> <!-- Office 2003 Common Settings --> </component> </component> </pre>

The following example is from the MigUser.xml file. For more examples, see the MigApp.xml file:

```

<component type="System" context="User">
  <displayName _locID="miguser.startmenu">Start Menu</displayName>
  <paths>
    <path type="File">%CSIDL_STARTMENU%</path>
  </paths>
  <role role="Settings">
    <detects>
      <detect>
        <condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_STARTMENU%")</condition>
      </detect>
    </detects>
    <rules>
      <include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
        <objectSet>
          <pattern type="File">%CSIDL_STARTMENU%\* [*]</pattern>
        </objectSet>
      </include>
      <merge script="MigXmlHelper.DestinationPriority() ">
        <objectSet>
          <pattern type="File">%CSIDL_STARTMENU% [desktop.ini]</pattern>
          <pattern type="File">%CSIDL_STARTMENU%\* [*]</pattern>
        </objectSet>
      </merge>
    </rules>
  </role>
</component>

```

<rules>

The <rules> element is required in a custom .xml file. This element contains rules that will run during the migration if the parent <component> element is selected, unless the child <conditions> element, if present, evaluates to FALSE. For each <rules> element there can be multiple child <rules> elements.

- **Number of occurrences:** unlimited
- **Parent elements:** <role>, <rules>, <namedElements>
- **Required child elements:** <include>
- **Optional child elements:** <rules>, <exclude>, <unconditionalExclude>, <merge>, <contentModify>, <locationModify>, <destinationCleanup>, <addObjects>, <externalProcess>, <processing>, <includeAttributes>, <excludeAttributes>, <conditions>, <detects>

Syntax:

```
<rules name="/ID" context="User|System|UserAndSystem">
```

```
</rules>
```

SETTING	REQUIRED?	VALUE
name	Yes, when <rules> is a child to <namedElements> No, when <rules> is a child to any other element	When <i>ID</i> is specified, any child elements are not processed. Instead, any other <rules> elements with the same name that are declared within <namedElements> are processed.

SETTING	REQUIRED?	VALUE
context	No (default = UserAndSystem)	<p>Defines the scope of this parameter — whether to process this component in the context of the specific user, across the entire operating system, or both.</p> <p>The largest possible scope is set by the component element. For example, if a <component> element has a context of User and a <rules> element had a context of UserAndSystem, then the <rules> element would act as though it has a context of User. If <rules> had a context of System, it would act as though <rules> was not there.</p> <ul style="list-style-type: none"> • User. Evaluates the variables for each user. • System. Evaluates the variables only once for the system. • UserAndSystem. Evaluates the variables for the entire operating system and each user.

The following example is from the MigUser.xml file:

```
<component type="Documents" context="User">
  <displayName _locID="miguser.mymusic">My Music</displayName>
  <paths>
    <path type="File">%CSIDL_MYMUSIC%</path>
  </paths>
  <role role="Data">
    <detects>
      <detect>
        <condition>MigXmlHelper.DoesObjectExist("File", "%CSIDL_MYMUSIC%")</condition>
      </detect>
    </detects>
    <rules>
      <include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
        <objectSet>
          <pattern type="File">%CSIDL_MYMUSIC%\* [*]</pattern>
        </objectSet>
      </include>
      <merge script="MigXmlHelper.DestinationPriority() ">
        <objectSet>
          <pattern type="File">%CSIDL_MYMUSIC%\ [desktop.ini]</pattern>
        </objectSet>
      </merge>
    </rules>
  </role>
</component>
```

<script>

The return value that is required by <script> depends on the parent element.

Number of occurrences: Once for `<variable>`, unlimited for `<objectSet>` and `<processing>`

Parent elements: `<objectSet>`, `<variable>`, `<processing>`

Child elements: none

Syntax and helper functions:

- General Syntax: `<script>ScriptWithArguments</script>`
- You can use [GetStringContent](#) when `<script>` is within `<variable>`.

Syntax: `<script>MigXmlHelper.GetStringContent("ObjectType","EncodedLocationPattern",
"ExpandContent")</script>`

Example:

```
<script>MigXMLHelper.GetStringContent("Registry","HKLM\Software\MyApp\Installer [EXEPATH]")</script>
```

- You can use [GenerateUserPatterns](#) when `<script>` is within `<objectSet>`.

Syntax:

```
<script>MigXmlHelper.GenerateUserPatterns("ObjectType","EncodedLocationPattern","ProcessCurrentU  
ser")</script>
```

Example:

```
<script>MigXmlHelper.GenerateUserPatterns ("File", "%USERPROFILE%\* [* .doc]", "FALSE")</script>
```

- You can use [GenerateDrivePatterns](#) when `<script>` is within `<objectSet>`.

Syntax: `<script>MigXmlHelper.GenerateDrivePatterns("PatternSegment","DriveType")</script>`

Example: `<script>MigXmlHelper.GenerateDrivePatterns("* [sample.doc]", "Fixed")</script>`

- You can use the [Simple executing scripts](#) with `<script>` elements that are within `<processing>` elements:
AskForLogoff, ConvertToShortFileName, KillExplorer, RemoveEmptyDirectories, RestartExplorer,
RegisterFonts, StartService, StopService, SyncSCM.

Syntax: `<script>MigXmlHelper.ExecutingScript</script>`

Example: `<script>MigXmlHelper.KillExplorer()</script>`

SETTING	REQUIRED?	VALUE
---------	-----------	-------

SETTING	REQUIRED?	VALUE
<i>ScriptWithArguments</i>	Yes	<p>A script followed by any number of string arguments that are separated by a comma and enclosed in parenthesis. For example</p> <pre data-bbox="1050 338 1398 394">, MyScripts.AScript ("Arg1", "Arg2").</pre> <p>The script will be called for each object that is enumerated by the object sets in the <include> rule. The filter script returns a Boolean value. If the return value is TRUE, the object will be migrated. If it is FALSE, it will not be migrated.</p> <p>The return value that is required by <script> depends on the parent element.</p> <ul data-bbox="1050 775 1362 1379" style="list-style-type: none"> • When used within <variable>, the return value must be a string. • When used within <objectSet>, the return value must be a two-dimensional array of strings. • When used within <location>, the return value must be a valid location that aligns with the type attribute of <location>. For example, if <location type="File">, the child script element, if specified, must be a valid file location. <div data-bbox="1082 1402 1398 2018" style="border: 1px solid black; padding: 5px;"> <p>Note</p> <p>If you are migrating a file that has a bracket character ([or]) in the file name, insert the carrot (^) character directly before the bracket for it to be valid. For example, if there is a file named "file].txt", specify</p> <pre data-bbox="1107 1783 1369 1850"><pattern type="File">c:\documents\mydocs [file^].txt</pattern></pre> <p>instead of</p> <pre data-bbox="1107 1883 1369 1951"><pattern type="File">c:\documents\mydocs [file].txt</pattern></pre> </div>

Examples:

To migrate the Sample.doc file from any drive on the source computer, use <script> as follows. If multiple files exist with the same name, all such files will get migrated.

```
<script>MigXmlHelper.GenerateDrivePatterns("* [sample.doc]", "Fixed")</script>
```

For more examples of how to use this element, see [Exclude Files and Settings](#), [Reroute Files and Settings](#), and [Custom XML Examples](#).

<script> functions

You can use the following functions with the <script> element

- [String and pattern generating functions](#)
- [Simple executing scripts](#)

String and pattern generating functions

These functions return either a string or a pattern.

- **GetStringContent**

You can use GetStringContent with <script> elements that are within <variable> elements. If possible, this function returns the string representation of the given object. Otherwise, it returns NULL. For file objects this function always returns NULL.

Syntax: GetStringContent("ObjectType","EncodedLocationPattern", "ExpandContent")

SETTING	REQUIRED?	VALUE
<i>ObjectType</i>	Yes	The type of object. Can be Registry or Ini (for an .ini file).
<i>EncodedLocationPattern</i>	Yes	<ul style="list-style-type: none">• If type of object is Registry, EncodedLocationPattern must be a valid registry path. For example, HKLM\SOFTWARE\MyKey[].• If the type of object is Ini, then EncodedLocationPattern must be in the following format: IniFilePath SectionName[SettingName]
<i>ExpandContent</i>	No (default=TRUE)	Can be TRUE or FALSE. If FALSE, then the given location will not be expanded before it is returned.

For example:

```
``` syntax
<variable name="MSNMessengerInstPath">
<script>MigXmlHelper.GetStringContent("Registry", "%HkLmWowSoftware%\Microsoft\MSNMessenger
[InstallationDirectory]")</script>
</variable>
```
```

• GenerateDrivePatterns

The GenerateDrivePatterns function will iterate all of the available drives and select the ones that match the requested drive type. It will then concatenate the selected drives with the end part of *PatternSegment* to form a full encoded file pattern. For example, if *PatternSegment* is `Path [file.txt]` and *DriveType* is `Fixed`, then the function will generate `C:\Path [file.txt]`, and other patterns if there are fixed drives other than C:. You cannot specify environment variables with this function. You can use GenerateDrivePatterns with `<script>` elements that are within `<objectSet>` that are within `<include>/<exclude>`.

Syntax: GenerateDrivePatterns("PatternSegment","DriveType")

| SETTING | REQUIRED? | VALUE |
|-----------------------|-----------|--|
| <i>PatternSegment</i> | Yes | The suffix of an encoded pattern. It will be concatenated with a drive specification, such as "c:", to form a complete encoded file pattern . For example, "* [*.doc]". <i>PatternSegment</i> cannot be an environment variable. |
| <i>DriveType</i> | Yes | The drive type for which the patterns are to be generated. You can specify one of: <ul style="list-style-type: none">• Fixed• CDROM• Removable• Remote |

See the last component in the MigUser.xml file for an example of this element.

• GenerateUserPatterns

The function will iterate through all users that are being migrated, excluding the currently processed user if `<ProcessCurrentUser>` is FALSE, and will expand the specified pattern in the context of each user. For example, if users A, B and C have profiles in C:\Documents and Settings), by calling `GenerateUserPattens('File', '%userprofile% [*.doc]', 'TRUE')`, the helper function will generate the following three patterns:

- o "C:\Documents and Settings\A* [*.doc]"
- o "C:\Documents and Settings\B* [*.doc]"

- o "C:\Documents and Settings\C* [* .doc]"

Syntax: `GenerateUserPatterns("ObjectType","EncodedLocationPattern","ProcessCurrentUser")`

| SETTING | REQUIRED? | VALUE |
|-------------------------------|-----------|---|
| <i>ObjectType</i> | Yes | Defines the object type. Can be File or Registry. |
| <i>EncodedLocationPattern</i> | Yes | The location pattern . Environment variables are allowed. |
| <i>ProcessCurrentUser</i> | Yes | Can be TRUE or FALSE. Indicates if the patterns should be generated for the current user. |

****Example:****

If `GenerateUserPatterns('File','%userprofile% \[* .doc\'],'FALSE')` is called while USMT is processing user A, then this function will only generate patterns for users B and C. You can use this helper function to build complex rules. For example, to migrate all .doc files from the source computer – but if user X is not migrated, then do not migrate any of the .doc files from user X’s profile.

The following is example code for this scenario. The first `<rules>` element migrates all.doc files on the source computer with the exception of those inside C:\Documents and Settings. The second `<rules>` elements will migrate all .doc files from C:\Documents and Settings with the exception of the .doc files in the profiles of the other users. Because the second `<rules>` element will be processed in each migrated user context, the end result will be the desired behavior. The end result is the one we expected.

```


` ` ` syntax
<rules context="System">
  <include>
    <objectSet>
      <script>MigXmlHelper.GenerateDrivePatterns ("* [* .doc]", "Fixed")</script>
    </objectSet>
  </include>
  <exclude>
    <objectSet>
      <pattern type="File">%ProfilesFolder%\* [* .doc]</pattern>
    </objectSet>
  </exclude>
</rules>
<rules context="User">
  <include>
    <objectSet>
      <pattern type="File">%ProfilesFolder%\* [* .doc]</pattern>
    </objectSet>
  </include>
  <exclude>
    <objectSet>
      <script>MigXmlHelper.GenerateUserPatterns ("File","%userprofile%\* [* .doc]", "FALSE")</script>
    </objectSet>
  </exclude>
</rules>
` ` `


```

MigXmlHelper.GenerateDocPatterns

This helper function invokes the document finder to scan the system for all files that can be migrated. It can be invoked in either System or User context to focus the scan.

| SETTING | REQUIRED? | VALUE |
|-------------------------|----------------------|---|
| <i>ScanProgramFiles</i> | No (default = FALSE) | Can be TRUE or FALSE. The <i>ScanProgramFiles</i> parameter determines whether or not the document finder scans the Program Files directory to gather registered file extensions for known applications. For example, when set to TRUE it will discover and migrate .jpg files under the Photoshop directory, if .jpg is a file extension registered to Photoshop. |
| <i>IncludePatterns</i> | No (default = TRUE) | Can be TRUE or FALSE. TRUE will generate include patterns and can be added under the <include> element. FALSE will generate exclude patterns and can be added under the <exclude> element. |
| <i>SystemDrive</i> | No (default = FALSE) | Can be TRUE or FALSE. If TRUE, restricts all patterns to the system drive. |

```

<!-- This component migrates data in user context -->
<component type="Documents" context="User">
  <displayName>MigDocUser</displayName>
  <role role="Data">
    <rules>
      <include filter='MigXmlHelper.IgnoreIrrelevantLinks() '>
        <objectSet>
          <script>MigXmlHelper.GenerateDocPatterns ("false")</script>
        </objectSet>
      </include>
      <exclude>
        <objectSet>
          <script>MigXmlHelper.GenerateDocPatterns ("false", "false", "false")</script>
        </objectSet>
      </exclude>
    </rules>
  </role>
</component>

```

Simple executing scripts

The following scripts have no return value. You can use the following errors with <script> elements that are within <processing> elements

- **AskForLogoff()**. Prompts the user to log off at the end of the migration. For example:

```

<processing when="apply-success">
  <script>MigXmlHelper.AskForLogoff()</script>
</processing>

```

- **ConvertToShortFileName(RegistryEncodedLocation)**. If *RegistryEncodedLocation* is the full path of an existing file, this function will convert the file to its short file name and then it will update the registry value.

- **KillExplorer()**. Stops Explorer.exe for the current user context. This allows access to certain keys and files that are kept open when Explorer.exe is running. For example:

```
<processing when="pre-apply">
  <script>MigXmlHelper.KillExplorer()</script>
</processing>
```

- **RegisterFonts(FileEncodedLocation)**. Registers the given font or all of the fonts in the given directory. For example:

```
<processing when="apply-success">
<script>MigXmlHelper.RegisterFonts("%CSIDL_COMMON_FONTS%")</script>
</processing>
```

- **RemoveEmptyDirectories (DirectoryEncodedPattern)**. Deletes any empty directories that match *DirectoryEncodedPattern* on the destination computer.
- **RestartExplorer()**. Restarts Explorer.exe at the end of the migration. For example:

```
<processing when="post-apply">
  <script>MigXmlHelper.RestartExplorer()</script>
</processing>
```

- **StartService (ServiceName, OptionalParam1, OptionalParam2,...)**. Starts the service identified by *ServiceName*. *ServiceName* is the subkey in HKLM\System\CurrentControlSet\Services that holds the data for the given service. The optional parameters, if any, will be passed to the StartService API. For more information, see [this Microsoft Web site](#).
- **StopService (ServiceName)**. Stops the service that is identified by *ServiceName*. *ServiceName* is the subkey in HKLM\System\CurrentControlSet\Services that holds the data for the given service.
- **SyncSCM(ServiceShortName)**. Reads the Start type value from the registry (HKLM\System\CurrentControlSet\Services\ServiceShortName [Start]) after it is changed by the migration engine, and then synchronizes Service Control Manager (SCM) with the new value.

<text>

You can use the <text> element to set a value for any environment variables that are inside one of the migration .xml files.

- **Number of occurrences:** Once in each <variable> element.
- **Parent elements:** <variable>
- **Child elements:** None.

Syntax:

```
<text>NormalText</text>
```

SETTING	VALUE
<i>NormalText</i>	This is interpreted as normal text.

For example:

```
<variable name="QuickTime5or6DataSys">
  <text>%CSIDL_COMMON_APPDATA%\QuickTime</text>
</variable>
```

<unconditionalExclude>

The <unconditionalExclude> element excludes the specified files and registry values from the migration, regardless of the other include rules in any of the migration .xml files or in the Config.xml file. The objects declared here will not be migrated because this element takes precedence over all other rules. For example, even if there are explicit <include> rules to include .mp3 files, if you specify to exclude them with this option, then they will not be migrated.

Use this element if you want to exclude all .mp3 files from the source computer. Or, if you are backing up C:\UserData using another method, you can exclude the entire folder from the migration. Use this element with caution, however, because if an application needs a file that you exclude, the application may not function properly on the destination computer.

- **Number of occurrences:** Unlimited.
- **Parent elements:** <rules>
- **Child elements:** <objectSet>

Syntax:

```
<unconditionalExclude> </unconditionalExclude>
```

The following .xml file excludes all .mp3 files from migration. For additional examples of how to use this element, see the [Exclude Files and Settings](#).

```
<migration urlid="http://www.microsoft.com/migration/1.0/migxmlext/excludefiles">
  <component context="System" type="Documents">
    <displayName>Test</displayName>
    <role role="Data">
      <rules>
        <unconditionalExclude>
          <objectSet>
            <script>MigXmlHelper.GenerateDrivePatterns ("* [*.mp3]", "Fixed")</script>
          </objectSet>
        </unconditionalExclude>
      </rules>
    </role>
  </component>
</migration>
```

<variable>

The <variable> element is required in an <environment> element. For each <variable> element there must be one <objectSet>, <script>, or <text> element. The content of the <variable> element assigns a text value to the environment variable. This element has the following three options:

1. If the <variable> element contains a <text> element, then the value of the variable element will be the value of the <text> element.
2. If the <variable> element contains a <script> element and the invocation of the script produces a non-null string, then the value of the <variable> element will be the result of the script invocation.
3. If the <variable> element contains an <objectSet> element and the evaluation of the <objectSet>

element produces at least one object pattern, then the value of the first object to match the resulting object pattern will be the value of the variable element.

- **Number of occurrences:** Unlimited
- **Parent elements:** <environment>
- **Required child elements:** either <text>, or <script>, or <objectSet>

Syntax:

```
<variable name="ID" remap=TRUE|FALSE>
</variable>
```

SETTING	REQUIRED?	VALUE
name	Yes	<i>ID</i> is a string value that is the name used to reference the environment variable. We recommend that <i>ID</i> start with the component's name to avoid namespace collisions. For example, if your component's name is MyComponent, and you want a variable that is your component's install path, you could specify <code>MyComponent.InstallPath</code> .
remap	No, default = FALSE	Specifies whether to evaluate this environment variable as a remapping environment variable. Objects that are located in a path that is underneath this environment variable's value are automatically moved to where the environment variable points on the destination computer.

The following example is from the MigApp.xml file:

```
<environment>
  <variable name="Hk1mWowSoftware">
    <text>HKLM\Software</text>
  </variable>
  <variable name="WinZip8or9or10Exe">

  <script>MigXmlHelper.GetStringContent("Registry", "%Hk1mWowSoftware%\Microsoft\Windows\CurrentVersion\App
Paths\winzip32.exe [ ]")</script>
  </variable>
</environment>
```

<version>

The <version> element defines the version for the component, but does not affect the migration.

- **Number of occurrences:** zero or one
- **Parent elements:** <component>

- **Child elements:** none

Syntax:

```
<version>ComponentVersion</version>
```

SETTING	REQUIRED?	VALUE
<i>ComponentVersion</i>	Yes	The version of the component, which can contain patterns.

For example:

```
<version>4.*</version>
```

<windowsObjects>

The <windowsObjects> element is for USMT internal use only. Do not use this element.

Appendix

Specifying locations

- **Specifying encoded locations.** The encoded location used in all of the helper functions is an unambiguous string representation for the name of an object. It is composed of the node part, optionally followed by the leaf enclosed in square brackets. This makes a clear distinction between nodes and leaves.

For example, specify the file C:\Windows\notepad.exe like this: `c:\Windows[Notepad.exe]`. Similarly, specify the directory C:\Windows\System32 like this: `c:\Windows\System32`. (Notice the absence of the [] construct.)

Representing the registry is very similar. The default value of a registry key is represented as an empty [] construct. For example, the default value for the HKLM\SOFTWARE\MyKey registry key will be

```
HKLM\SOFTWARE\MyKey[]
```

- **Specifying location patterns.** You specify a location pattern in a way that is similar to how you specify an actual location. The exception is that both the node and leaf part accept patterns. However, a pattern from the node does not extend to the leaf.

For example, the pattern `c:\Windows*` will match the Windows directory and all subdirectories. But it will not match any of the files in those directories. To match the files as well, you must specify

```
c:\Windows\*[*]
```

Internal USMT functions

The following functions are for internal USMT use only. Do not use them in an .xml file.

- AntiAlias
- ConvertScreenSaver
- ConvertShowLEOnDesktop
- ConvertToOfficeLangID
- MigrateActiveDesktop
- MigrateAppearanceUPM

- MigrateDisplayCS
- MigrateDisplaySS
- MigrateIEAutoSearch
- MigrateMouseUPM
- MigrateSoundSysTray
- MigrateTaskBarSS
- SetPstPathInMapiStruc

Valid version tags

You can use the following version tags with various helper functions:

- "CompanyName"
- "FileDescription"
- "FileVersion"
- "InternalName"
- "LegalCopyright"
- "OriginalFilename"
- "ProductName"
- "ProductVersion"

The following version tags contain values that can be compared:

- "FileVersion"
- "ProductVersion"

Related topics

[USMT XML Reference](#)

Offline Migration Reference

6/14/2019 • 5 minutes to read • [Edit Online](#)

Offline migration enables the ScanState tool to run inside a different Windows® operating system than the Windows operating system from which ScanState is gathering files and settings. There are two primary offline scenarios:

- **Windows PE.** The ScanState tool can be run from within Windows PE, gathering files and settings from the offline Windows operating system on that machine.
- **Windows.old.** The ScanState tool can now gather files and settings from the Windows.old directory that is created during Windows installation on a partition that contains a previous installation of Windows. For example, the ScanState tool can run in Windows 10, gathering files from a previous Windows 7 or Windows 8 installation contained in the Windows.old directory.

When you use User State Migration Tool (USMT) 10.0 to gather and restore user state, offline migration reduces the cost of deployment by:

- **Reducing complexity.** In computer-refresh scenarios, migrations from the Windows.old directory reduce complexity by eliminating the need for the ScanState tool to be run before the operating system is deployed. Also, migrations from the Windows.old directory enable ScanState and LoadState to be run successively.
- **Improving performance.** When USMT runs in an offline Windows Preinstallation Environment (WinPE) environment, it has better access to the hardware resources. This may increase performance on older machines with limited hardware resources and numerous installed software applications.
- **New recovery scenario.** In scenarios where a machine no longer restarts properly, it might be possible to gather user state with the ScanState tool from within WinPE.

In This Topic

- [What Will Migrate Offline?](#)
- [What Offline Environments are Supported?](#)
- [User-Group Membership and Profile Control](#)
- [Command-Line Options](#)
- [Environment Variables](#)
- [Offline.xml Elements](#)

What Will Migrate Offline?

The following user data and settings migrate offline, similar to an online migration:

- Data and registry keys specified in MigXML
- User accounts
- Application settings
- Limited set of operating-system settings

- EFS files
- Internet Explorer® Favorites

For exceptions to what you can migrate offline, see [What Does USMT Migrate?](#)

What Offline Environments are Supported?

The following table defines the supported combination of online and offline operating systems in USMT.

RUNNING OPERATING SYSTEM	OFFLINE OPERATING SYSTEM
WinPE 5.0 or greater, with the MSXML library	Windows Vista, Windows 7, Windows 8, Windows 10
Windows 7, Windows 8, Windows 10	Windows.old directory

Note It is possible to run the ScanState tool while the drive remains encrypted by suspending Windows BitLocker Drive Encryption before booting into WinPE. For more information, see [this Microsoft site](#).

User-Group Membership and Profile Control

User-group membership is not preserved during offline migrations. You must configure a **<ProfileControl>** section in the Config.xml file to specify the groups that the migrated users should be made members of. The following example places all migrated users into the Users group:

```
<Configuration>
<ProfileControl>
  <localGroups>
    <mappings>
      <changeGroup from="*" to="Users" appliesTo="MigratedUsers">
        <include>
          <pattern>*</pattern>
        </include>
      </changeGroup>
    </mappings>
  </localGroups>
</ProfileControl>
</Configuration>
```

For information about the format of a Config.xml file, see [Config.xml File](#).

Command-Line Options

An offline migration can either be enabled by using a configuration file on the command line, or by using one of the following command line options:

COMPONENT	OPTION	DESCRIPTION
ScanState.exe	/offline: <i><path to offline.xml></i>	This command-line option enables the offline-migration mode and requires a path to an Offline.xml configuration file.

COMPONENT	OPTION	DESCRIPTION
ScanState.exe	/offlineWinDir: < Windows directory >	This command-line option enables the offline-migration mode and starts the migration from the location specified. It is only for use in WinPE offline scenarios where the migration is occurring from a Windows directory.
ScanState.exe	/OfflineWinOld: < Windows.old directory >	This command-line option enables the offline migration mode and starts the migration from the location specified. It is only intended to be used in Windows.old migration scenarios, where the migration is occurring from a Windows.old directory.

You can use only one of the **/offline**, **/offlineWinDir** , or **/OfflineWinOld** command-line options at a time; USMT does not support using more than one together.

Environment Variables

The following system environment variables are necessary in the scenarios outlined below.

VARIABLE	VALUE	SCENARIO
USMT_WORKING_DIR	Full path to a working directory	<p>Required when USMT binaries are located on read-only media, which does not support the creation of log files or temporary storage. To set the system environment variable, at a command prompt type the following:</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <pre>Set USMT_WORKING_DIR=[path to working directory]</pre> </div>

VARIABLE	VALUE	SCENARIO
MIG_OFFLINE_PLATFORM_ARCH	32 or 64	<p>While operating offline, this environment variable defines the architecture of the offline system, if the system does not match the WinPE and Scanstate.exe architecture. This environment variable enables the 32-bit ScanState application to gather data from a computer with 64-bit architecture, or the 64-bit ScanState application to gather data from a computer with 32-bit architecture. This is required when auto-detection of the offline architecture doesn't function properly, for example, when the source system is running a 64-bit version of Windows XP. For example, to set this system environment variable for a 32-bit architecture, at a command prompt type the following:</p> <div style="border: 1px solid gray; padding: 5px; width: fit-content; margin: 10px auto;"> <pre>Set MIG_OFFLINE_PLATFORM_ARCH=32</pre> </div>

Offline.xml Elements

Use an offline.xml file when running the ScanState tool on a computer that has multiple Windows directories. The offline.xml file specifies which directories to scan for windows files. An offline.xml file can be used with the /offline option as an alternative to specifying a single Windows directory path with the /offlineDir option.

<offline>

This element contains other elements that define how an offline migration is to be performed.

Syntax: <offline> </offline>

<winDir>

This element is a required child of <offline> and contains information about how the offline volume can be selected. The migration will be performed from the first element of <winDir> that contains a valid Windows system volume.

Syntax: < winDir > </ winDir >

<path>

This element is a required child of <winDir> and contains a file path pointing to a valid Windows directory. Relative paths are interpreted from the ScanState tool's working directory.

Syntax: <path> c:\windows </path>

-or-

Syntax, when used with the <mappings> element: <path> C:\, D:\ </path>

<mappings>

This element is an optional child of <offline>. When specified, the <mappings> element will override the

automatically detected WinPE drive mappings. Each child **<path>** element will provide a mapping from one system volume to another. Additionally, mappings between folders can be provided, since an entire volume can be mounted to a specific folder.

Syntax: <mappings> </mappings>

<failOnMultipleWinDir>

This element is an optional child of **<offline>**. The **<failOnMultipleWinDir>** element allows the user to specify that the migration should fail when USMT detects that there are multiple instances of Windows installed on the source machine. When the **<failOnMultipleWinDir>** element isn't present, the default behavior is that the migration does not fail.

Syntax: <failOnMultipleWinDir>1</failOnMultipleWinDir> or Syntax:
<failOnMultipleWinDir>0</failOnMultipleWinDir>

Offline .xml Example

The following XML example illustrates some of the elements discussed earlier in this topic.

```
<offline>
  <winDir>
    <path>C:\Windows</path>
    <path>D:\Windows</path>
    <path>E:\</path>
  </winDir>
  <failOnMultipleWinDir>1</failOnMultipleWinDir>
</offline>
```

Related topics

[Plan Your Migration](#)

How to install fonts that are missing after upgrading to Windows 10

6/18/2019 • 3 minutes to read • [Edit Online](#)

Applies to: Windows 10

When you upgrade from the Windows 7, Windows 8, or Windows 8.1 operating system to Windows 10, certain fonts are no longer available by default post-upgrade. To reduce the operating system footprint, improve performance, and optimize disk space usage, we moved many of the fonts that were previously shipped with prior versions of Windows to the optional features of Windows 10. If you install a fresh instance of Windows 10, or upgrade an older version of Windows to Windows 10, these optional features are not enabled by default. As a result, these fonts appear to be missing from the system.

If you have documents created using the missing fonts, these documents might display differently on Windows 10.

For example, if you have an English (or French, German, or Spanish) version of Windows 10 installed, you might notice that fonts such as the following appear to be missing:

- Gautami
- Meiryo
- Narkism/Batang
- BatangChe
- Dotum
- DotumChe
- Gulim
- GulimChe
- Gungsuh
- GungsuhChe

If you want to use these fonts, you can enable the optional feature to add these back to your system. Be aware that this is a permanent change in behavior for Windows 10, and it will remain this way in future releases.

Installing language-associated features via language settings:

If you want to use the fonts from the optional feature and you know that you will want to view Web pages, edit documents, or use apps in the language associated with that feature, add that language into your user profile. You do this the Settings app.

For example, here are the steps to install the fonts associated with the Hebrew language:

1. Click **Start > Settings**.
2. In Settings, click **Time & language**, and then click **Region & language**.
3. If Hebrew is not included in the list of languages, click the plus sign (+) to add a language.
4. Find Hebrew, and then click it to add it to your language list.

Once you have added Hebrew to your language list, then the optional Hebrew font feature and other optional features for Hebrew language support are installed. This should only take a few minutes.

Note: The optional features are installed by Windows Update. This means you need to be online for the

Install optional fonts manually without changing language settings:

If you want to use fonts in an optional feature but don't need to search web pages, edit documents, or use apps in the associated language, you can install the optional font features manually without changing your language settings.

For example, here are the steps to install the fonts associated with the Hebrew language without adding the Hebrew language itself to your language preferences:

1. Click **Start > Settings**.
2. In Settings, click **Apps**, click **Apps & features**, and then click **Manage optional features**.
3. If you don't see **Hebrew Supplemental Fonts** in the list of installed features, click the plus sign (+) to add a feature.
4. Select **Hebrew Supplemental Fonts** in the list, and then click **Install**.

Note: The optional features are installed by Windows Update. You need to be online for the Windows Update service to work.

Fonts included in optional font features

Here is a comprehensive list of the font families in each of the optional features. Some font families might include multiple fonts for different weights and styles.

- Arabic Script Supplemental Fonts: Aldhabi, Andalus, Arabic Typesetting, Microsoft Uighur, Sakkal Majalla, Simplified Arabic, Traditional Arabic, Urdu Typesetting
- Bangla Script Supplemental Fonts: Shonar Bangla, Vrinda
- Canadian Aboriginal Syllabics Supplemental Fonts: Euphemia
- Cherokee Supplemental Fonts: Plantagenet Cherokee
- Chinese (Simplified) Supplemental Fonts: DengXian, FangSong, KaiTi, SimHei
- Chinese (Traditional) Supplemental Fonts: DFKai-SB, MingLiU, MingLiU_HKSCS, PMingLiU
- Devanagari Supplemental Fonts: Aparajita, Kokila, Mangal, Sanskrit Text, Utsaah
- Ethiopic Supplemental Fonts: Nyala
- Gujarati Supplemental Fonts: Shruti
- Gurmukhi Supplemental Fonts: Raavi
- Hebrew Supplemental Fonts: Aharoni Bold, David, FrankRuehl, Gisha, Levanim MT, Miriam, Miriam Fixed, Narkism, Rod
- Japanese Supplemental Fonts: Meiryo, Meiryo UI, MS Gothic, MS PGothic, MS UI Gothic, MS Mincho, MS PMincho, Yu Mincho
- Kannada Supplemental Fonts: Tunga
- Khmer Supplemental Fonts: DaunPenh, Khmer UI, MoolBoran
- Korean Supplemental Fonts: Batang, BatangChe, Dotum, DotumChe, Gulim, GulimChe, Gungsuh, GungsuhChe
- Lao Supplemental Fonts: DokChampa, Lao UI
- Malayalam Supplemental Fonts: Karthika
- Odia Supplemental Fonts: Kalinga
- Pan-European Supplemental Fonts: Arial Nova, Georgia Pro, Gill Sans Nova, Neue Haas Grotesk, Rockwell Nova, Verdana Pro
- Sinhala Supplemental Fonts: Iskoola Pota

- Syriac Supplemental Fonts: Estrangelo Edessa
- Tamil Supplemental Fonts: Latha, Vijaya
- Telugu Supplemental Fonts: Gautami, Vani
- Thai Supplemental Fonts: Angsana New, AngsanaUPC, Browallia New, BrowalliaUPC, Cordia New, CordiaUPC, DilleniaUPC, EucrosiaUPC, FreesiaUPC, IrisUPC, JasmineUPC, KodchiangUPC, Leelawadee, LilyUPC

Related Topics

[Download the list of all available language FODs](#)

[Features On Demand V2 \(Capabilities\)](#)

[Add Language Packs to Windows](#)

Update Windows 10 in enterprise deployments

4/5/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

Windows as a service provides a new way to think about building, deploying, and servicing the Windows operating system. The Windows as a service model is focused on continually providing new capabilities and updates while maintaining a high level of hardware and software compatibility. Deploying new versions of Windows is simpler than ever before: Microsoft releases new features two to three times per year rather than the traditional upgrade cycle where new features are only made available every few years. Ultimately, this model replaces the need for traditional Windows deployment projects, which can be disruptive and costly, and spreads the required effort out into a continuous updating process, reducing the overall effort required to maintain Windows 10 devices in your environment. In addition, with the Windows 10 operating system, organizations have the chance to try out “flighted” builds of Windows as Microsoft develops them, gaining insight into new features and the ability to provide continual feedback about them.

TIP

See [Windows 10 update history](#) for details about each Windows 10 update released to date.

In this section

TOPIC	DESCRIPTION
Quick guide to Windows as a service	Provides a brief summary of the key points for the new servicing model for Windows 10.
Overview of Windows as a service	Explains the differences in building, deploying, and servicing Windows 10; introduces feature updates, quality updates, and the different servicing branches; compares servicing tools.
Prepare servicing strategy for Windows 10 updates	Explains the decisions you need to make in your servicing strategy.
Build deployment rings for Windows 10 updates	Explains how to make use of servicing branches and update deferrals to manage Windows 10 updates.
Assign devices to servicing branches for Windows 10 updates	Explains how to assign devices to Current Branch (CB) or Current Branch for Business (CBB) for feature and quality updates, and how to enroll devices in Windows Insider.
Monitor Windows Updates with Update Compliance	Explains how to use Windows Analytics: Update Compliance to monitor and manage Windows Updates on devices in your organization.

TOPIC	DESCRIPTION
Optimize update delivery for Windows 10 updates	Explains the benefits of using Delivery Optimization or BranchCache for update distribution.
Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile	Explains updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile.
Deploy updates using Windows Update for Business	Explains how to use Windows Update for Business to manage when devices receive updates directly from Windows Update. Includes walkthroughs for configuring Windows Update for Business using Group Policy and Microsoft Intune.
Deploy Windows 10 updates using Windows Server Update Services (WSUS)	Explains how to use WSUS to manage Windows 10 updates.
Deploy Windows 10 updates using System Center Configuration Manager	Explains how to use Configuration Manager to manage Windows 10 updates.
Manage device restarts after updates	Explains how to manage update related device restarts.
Manage additional Windows Update settings	Provides details about settings available to control and configure Windows Update
Windows Insider Program for Business	Explains how the Windows Insider Program for Business works and how to become an insider.

TIP

Windows servicing is changing, but for disaster recovery scenarios and bare-metal deployments of Windows 10, you still can use traditional imaging software such as System Center Configuration Manager or the Microsoft Deployment Toolkit. Using these tools to deploy Windows 10 images is similar to deploying previous versions of Windows. With each release of a new feature update for CB, Microsoft makes available new .iso files for use in updating your custom images. Each Windows 10 build has a finite servicing lifetime, so it's important that images stay up to date with the latest build. For detailed information about how to deploy Windows 10 to bare-metal machines or to upgrade to Windows 10 from previous builds of Windows, see [Deploy Windows 10 with System Center 2012 R2 Configuration Manager](#). Additionally, Windows 10 clients can move from any supported version of Windows 10 (i.e. Version 1511) to the latest version directly (i.e 1709).

Find the tools and resources you need to help deploy and support Windows as a service in your organization.

Latest news, videos, & podcasts

Find the latest and greatest news on Windows 10 deployment and servicing.

Discovering the Windows 10 Update history pages

<https://www.youtube-nocookie.com/embed/mTnAb9XjMPY>

Everyone wins when transparency is a top priority. We want you to know when updates are available, as well as alert you to any potential issues you may encounter during or after you install an update. Bookmark the [Windows release health dashboard](#) for near real-time information on known issues, workarounds, and resolutions--as well as the current status of the latest feature update rollout.

The latest news:

[Updating Windows 10, version 1903 using Configuration Manager or WSUS](#) - May 23, 2019

[What's new in Windows Update for Business in Windows 10, version 1903](#) - May 21, 2019

[What's new for IT pros in Windows 10, version 1903](#) - May 21, 2019

[How to get the Windows 10 May 2019 Update](#) - May 21, 2019

[The benefits of Windows 10 Dynamic Update](#) - April 17, 2019

[Improving the Windows 10 update experience with control, quality and transparency](#) - April 4, 2019

[Call to action: review your Windows Update for Business deferral values](#) - April 3, 2019

[See more news.](#) You can also check out the [Windows 10 blog](#).

IT pro champs corner

Written by IT pros for IT pros, sharing real world examples and scenarios for Windows 10 deployment and servicing.



NEW [Deployment rings: The hidden \[strategic\] gem of Windows as a service](#)

[Classifying Windows updates in common deployment tools](#)

[Express updates for Windows Server 2016 re-enabled for November 2018 update](#)

[2019 SHA-2 Code Signing Support requirement for Windows and WSUS](#)

[Deploying Windows 10 Feature Updates to 24/7 Mission Critical Devices](#)

Discover

Learn more about Windows as a service and its value to your organization.



[Overview of Windows as a service](#)

[Quick guide to Windows as a service](#)

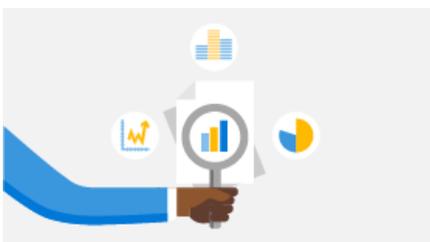
[Windows Analytics overview](#)

[What's new in Windows 10 deployment](#)

[How Microsoft IT deploys Windows 10](#)

Plan

Prepare to implement Windows as a service effectively using the right tools, products, and strategies.



[Simplified updates](#)

[Windows 10 end user readiness](#)

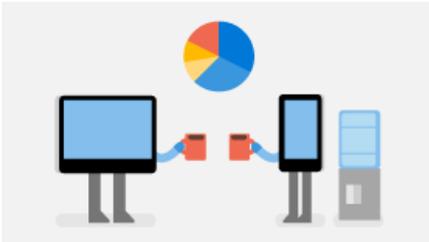
[Ready for Windows](#)

[Manage Windows upgrades with Upgrade Readiness](#)

[Preparing your organization for a seamless Windows 10 deployment](#)

Deploy

Secure your organization's deployment investment.



[Update Windows 10 in the enterprise](#)

[Deploying as an in-place upgrade](#)

[Configure Windows Update for Business](#)

[Express update delivery](#)

[Windows 10 deployment considerations](#)

Microsoft Ignite 2018



Looking to learn more? These informative session replays from Microsoft Ignite 2018 (complete with downloadable slide decks) can provide some great insights on Windows as a service.

[BRK2417: What's new in Windows Analytics: An Intro to Desktop Analytics](#)

[BRK3018: Deploying Windows 10 in the enterprise using traditional and modern techniques](#)

[BRK3019: Delivery Optimization deep dive: How to reduce internet bandwidth impact on your network](#)

BRK3020: Using AI to automate Windows and Office update staging with Windows Update for Business

BRK3027: Deploying Windows 10: Making the update experience smooth and seamless

BRK3039: Windows 10 and Microsoft Office 365 ProPlus lifecycle and servicing update

BRK3211: Ask the Experts: Successfully deploying, servicing, managing Windows 10

THR2234: Windows servicing and delivery fundamentals

THR3006: The pros and cons of LTSC in the enterprise

Quick guide to Windows as a service

6/14/2019 • 4 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile
- Windows 10 IoT Mobile

Windows as a service is a new concept, introduced with the release of Windows 10. While [an extensive set of documentation](#) is available explaining all the specifics and nuances, here is a quick guide to the most important concepts.

Definitions

Some new terms have been introduced as part of Windows as a service, so you should know what these terms mean.

- **Feature updates** will be released twice per year, around March and September. As the name suggests, these will add new features to Windows 10, delivered in bite-sized chunks compared to the previous practice of Windows releases every 3-5 years.
- **Quality updates** deliver both security and non-security fixes. They are typically released on the second Tuesday of each month ("Patch Tuesday"), though they can be released at any time. Quality updates include security updates, critical updates, servicing stack updates, and driver updates. Quality updates are cumulative, so installing the latest quality update is sufficient to get all the available fixes for a specific Windows 10 feature update. The "servicing stack" is the code that installs other updates, so they are important to keep current. For more information, see [Servicing stack updates](#).
- **Insider Preview** builds are made available during the development of the features that will be shipped in the next feature update, enabling organizations to validate new features as well as compatibility with existing apps and infrastructure, providing feedback to Microsoft on any issues encountered.
- **Servicing channels** allow organizations to choose when to deploy new features.
 - The **Semi-Annual Channel** receives feature updates twice per year.
 - The **Long Term Servicing Channel**, which is designed to be used only for specialized devices (which typically don't run Office) such as those that control medical equipment or ATM machines, receives new feature releases every two to three years.
- **Deployment rings** are groups of devices used to initially pilot, and then to broadly deploy, each feature update in an organization.

See [Overview of Windows as a service](#) for more information.

For some interesting in-depth information about how cumulative updates work, see [Windows Updates using forward and reverse differentials](#).

Key Concepts

Windows 10 gains new functionality with twice-per-year feature update releases. Initially, organizations will use these feature update releases for pilot deployments to ensure compatibility with existing apps and infrastructure. After a period of time, typically about four months after the feature update release, broad deployment throughout the organization can begin. The exact timeframe is determined by feedback from customers, ISVs, OEMs, and others, with an explicit "ready for broad deployment" declaration signaling this to customers.

Each Windows 10 feature update will be serviced with quality updates for 18 months from the date of the feature update release.

Windows 10 Enterprise LTSB is a separate **Long Term Servicing Channel** version. Each release is supported for a total of 10 years (five years standard support, five years extended support). New releases are expected about every three years.

See [Assign devices to servicing channels for Windows 10 updates](#) for more information.

Staying up to date

The process for keeping Windows 10 up to date involves deploying a feature update, at an appropriate time after its release. A variety of tools management and patching tools such as Windows Update, Windows Update for Business, Windows Server Update Services, System Center Configuration Manager, and third-party products) can be used to help with this process. [Windows Analytics Upgrade Readiness](#), a free tool to streamline Windows upgrade projects, is another important tool to help.

Because app compatibility, both for desktop apps and web apps, is outstanding with Windows 10, extensive advanced testing isn't required. Instead, only business-critical apps need to be tested, with the remaining apps validated through a series of pilot deployment rings. Once these pilot deployments have validated most apps, broad deployment can begin.

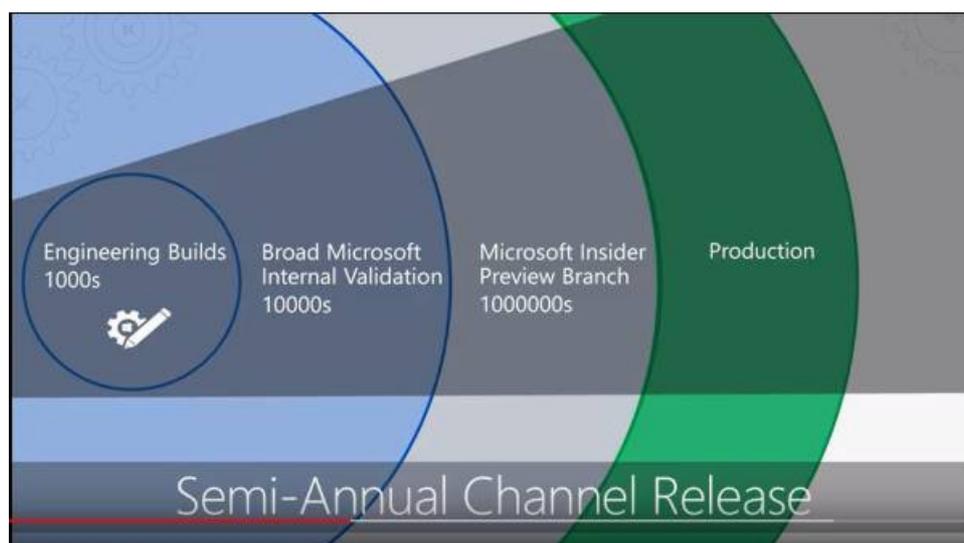
This process repeats with each new feature update, twice per year. These are small deployment projects, compared to the big projects that were necessary with the old three-to-five-year Windows release cycles.

Additional technologies such as BranchCache and Delivery Optimization, both peer-to-peer distribution tools, can help with the distribution of the feature update installation files.

See [Build deployment rings for Windows 10 updates](#) and [Optimize update delivery for Windows 10 updates](#) for more information.

Video: An overview of Windows as a service

Click the following Microsoft Mechanics video for an overview of the updated release model, particularly the Semi-Annual Channel.



Learn more

- [Adopting Windows as a service at Microsoft](#)
- [Windows lifecycle fact sheet](#)

Related topics

- [Update Windows 10 in the enterprise](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Servicing stack updates

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10, Windows 8.1, Windows 8, Windows 7

What is a servicing stack update?

Servicing stack updates provide fixes to the servicing stack, the component that installs Windows updates. Additionally, it contains the "component-based servicing stack" (CBS), which is a key underlying component for several elements of Windows deployment, such as DISM, SFC, changing Windows features or roles, and repairing components. The CBS is a small component that typically does not have updates released every month.

Why should servicing stack updates be installed and kept up to date?

Servicing stack updates improve the reliability of the update process to mitigate potential issues while installing the latest quality updates and feature updates. If you don't install the latest servicing stack update, there's a risk that your device can't be updated with the latest Microsoft security fixes.

When are they released?

Servicing stack updates are released depending on new issues or vulnerabilities. In rare occasions a servicing stack update may need to be released on demand to address an issue impacting systems installing the monthly security update. Starting in November 2018 new servicing stack updates will be classified as "Security" with a severity rating of "Critical."

NOTE

You can find a list of servicing stack updates at [Latest servicing stack updates](#).

What's the difference between a servicing stack update and a cumulative update?

Both Windows 10 and Windows Server use the cumulative update mechanism, in which many fixes to improve the quality and security of Windows are packaged into a single update. Each cumulative update includes the changes and fixes from all previous updates.

Servicing stack updates must ship separately from the cumulative updates because they modify the component that installs Windows updates. The servicing stack is released separately because the servicing stack itself requires an update. For example, the cumulative update [KB4284880](#) requires the [May 17, 2018 servicing stack update](#), which includes updates to Windows Update.

Is there any special guidance?

Microsoft recommends you install the latest servicing stack updates for your operating system before installing the latest cumulative update.

Typically, the improvements are reliability and performance improvements that do not require any specific special guidance. If there is any significant impact, it will be present in the release notes.

Installation notes

- Servicing stack updates contain the full servicing stack; as a result, typically administrators only need to install the latest servicing stack update for the operating system.
- Installing servicing stack update does not require restarting the device, so installation should not be disruptive.
- Servicing stack update releases are specific to the operating system version (build number), much like quality updates.
- Search to install latest available [Servicing stack update for Windows 10](#).

Overview of Windows as a service

6/14/2019 • 17 minutes to read • [Edit Online](#)

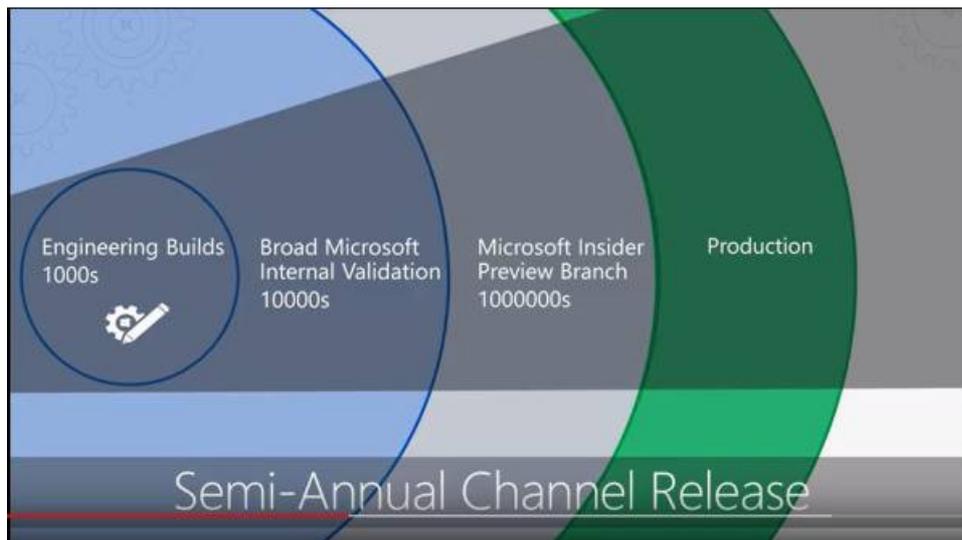
Applies to

- Windows 10
- Windows 10 Mobile
- Windows 10 IoT Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

The Windows 10 operating system introduces a new way to build, deploy, and service Windows: Windows as a service. Microsoft has reimagined each part of the process, to simplify the lives of IT pros and maintain a consistent Windows 10 experience for its customers. These improvements focus on maximizing customer involvement in Windows development, simplifying the deployment and servicing of Windows client computers, and leveling out the resources needed to deploy and maintain Windows over time.

Click the following Microsoft Mechanics video for an overview of the release model, particularly the Semi-Annual Channel.



Building

Prior to Windows 10, Microsoft released new versions of Windows every few years. This traditional deployment schedule imposed a training burden on users because the feature revisions were often significant. That schedule also meant waiting long periods without new features — a scenario that doesn't work in today's rapidly changing world, a world in which new security, management, and deployment capabilities are necessary to address challenges. Windows as a service will deliver smaller feature updates two times per year, around March and September, to help address these issues.

In the past, when Microsoft developed new versions of Windows, it typically released technical previews near the end of the process, when Windows was nearly ready to ship. With Windows 10, new features will be delivered to the [Windows Insider community](#) as soon as possible — during the development cycle, through a process called *flighting* — so that organizations can see exactly what Microsoft is developing and start their testing as soon as possible.

Microsoft also depends on receiving feedback from organizations throughout the development process so that

it can make adjustments as quickly as possible rather than waiting until after release. For more information about the Windows Insider Program and how to sign up, see the section [Windows Insider](#).

Of course Microsoft also performs extensive internal testing, with engineering teams installing new builds daily, and larger groups of employees installing builds frequently, all before those builds are ever released to the Windows Insider Program.

Deploying

Deploying Windows 10 is simpler than with previous versions of Windows. When migrating from earlier versions of Windows, an easy in-place upgrade process can be used to automatically preserve all apps, settings, and data. And once running Windows 10, deployment of Windows 10 feature updates will be equally simple.

One of the biggest challenges for organizations when it comes to deploying a new version of Windows is compatibility testing. Whereas compatibility was previously a concern for organizations upgrading to a new version of Windows, Windows 10 is compatible with most hardware and software capable of running on Windows 7 or later. Because of this high level of compatibility, the app compatibility testing process can be greatly simplified.

Application compatibility

Application compatibility testing has historically been a burden when approaching a Windows deployment or upgrade. With Windows 10, application compatibility from the perspective of desktop applications, websites, and apps built on the Universal Windows Platform (UWP) has improved tremendously. Microsoft understands the challenges organizations experienced when they migrated from the Windows XP operating system to Windows 7 and has been working to make Windows 10 upgrades a much better experience.

Most Windows 7-compatible desktop applications will be compatible with Windows 10 straight out of the box. Windows 10 achieved such high compatibility because the changes in the existing Win32 application programming interfaces were minimal. Combined with valuable feedback via the Windows Insider Program and diagnostic data, this level of compatibility can be maintained through each feature update. As for websites, Windows 10 includes Internet Explorer 11 and its backward-compatibility modes for legacy websites. Finally, UWP apps follow a compatibility story similar to desktop applications, so most of them will be compatible with Windows 10.

For the most important business-critical applications, organizations should still perform testing on a regular basis to validate compatibility with new builds. For remaining applications, consider validating them as part of a pilot deployment process to reduce the time spent on compatibility testing. If it's unclear whether an application is compatible with Windows 10, IT pros can either consult with the ISV or check the supported software directory at <http://www.readyforwindows.com>.

Device compatibility

Device compatibility in Windows 10 is also very strong; new hardware is not needed for Windows 10 as any device capable of running Windows 7 or later can run Windows 10. In fact, the minimum hardware requirements to run Windows 10 are the same as those required for Windows 7. Most hardware drivers that functioned in Windows 8.1, Windows 8, or Windows 7 will continue to function in Windows 10.

Servicing

Traditional Windows servicing has included several release types: major revisions (e.g., the Windows 8.1, Windows 8, and Windows 7 operating systems), service packs, and monthly updates. With Windows 10, there are two release types: feature updates that add new functionality twice per year, and quality updates that provide security and reliability fixes at least once a month.

With Windows 10, organizations will need to change the way they approach deploying updates. Servicing channels are the first way to separate users into deployment groups for feature and quality updates. With the

introduction of servicing channels comes the concept of a [deployment ring](#), which is simply a way to categorize the combination of a deployment group and a servicing channel to group devices for successive waves of deployment. For more information about developing a deployment strategy that leverages servicing channels and deployment rings, see [Plan servicing strategy for Windows 10 updates](#).

For information about each servicing tool available for Windows 10, see [Servicing tools](#).

To align with this new update delivery model, Windows 10 has three servicing channels, each of which provides different levels of flexibility over when these updates are delivered to client computers. For information about the servicing channels available in Windows 10, see [Servicing channels](#).

Naming changes

As part of the alignment with Windows 10 and Office 365 ProPlus, we are adopting common terminology to make it as easy as possible to understand the servicing process. Going forward, these are the new terms we will be using:

- Semi-Annual Channel - We will be referring to Current Branch (CB) as "Semi-Annual Channel (Targeted)", while Current Branch for Business (CBB) will simply be referred to as "Semi-Annual Channel".
- Long-Term Servicing Channel - The Long-Term Servicing Branch (LTSB) will be referred to as Long-Term Servicing Channel (LTSC).

IMPORTANT

With each Semi-Annual Channel release, we recommend beginning deployment right away to devices selected for early adoption (targeted validation) and ramp up to full deployment at your discretion, regardless of the "Targeted" designation. This will enable you to gain access to new features, experiences, and integrated security as soon as possible. For more information, see the blog post [Windows 10 and the "disappearing" SAC-T](#).

NOTE

For additional information, see the section about [Servicing Channels](#).

You can also read the blog post [Waas simplified and aligned](#), with details on this change.

IMPORTANT

Devices on the Semi-Annual Channel (formerly called Current Branch for Business) must have their diagnostic data set to **1 (Basic)** or higher, in order to ensure that the service is performing at the expected quality. If diagnostic data is set to **0**, the device will be treated as if it were in the Semi-Annual Channel (Targeted)(formerly called Current Branch or CB) branch. For instructions to set the diagnostic data level, see [Configure the operating system diagnostic data level](#).

Feature updates

With Windows 10, Microsoft will package new features into feature updates that can be deployed using existing management tools. Because feature updates are delivered more frequently than with previous Windows releases — twice per year, around March and September, rather than every 3–5 years — changes will be in bite-sized chunks rather than all at once and end user readiness time much shorter.

TIP

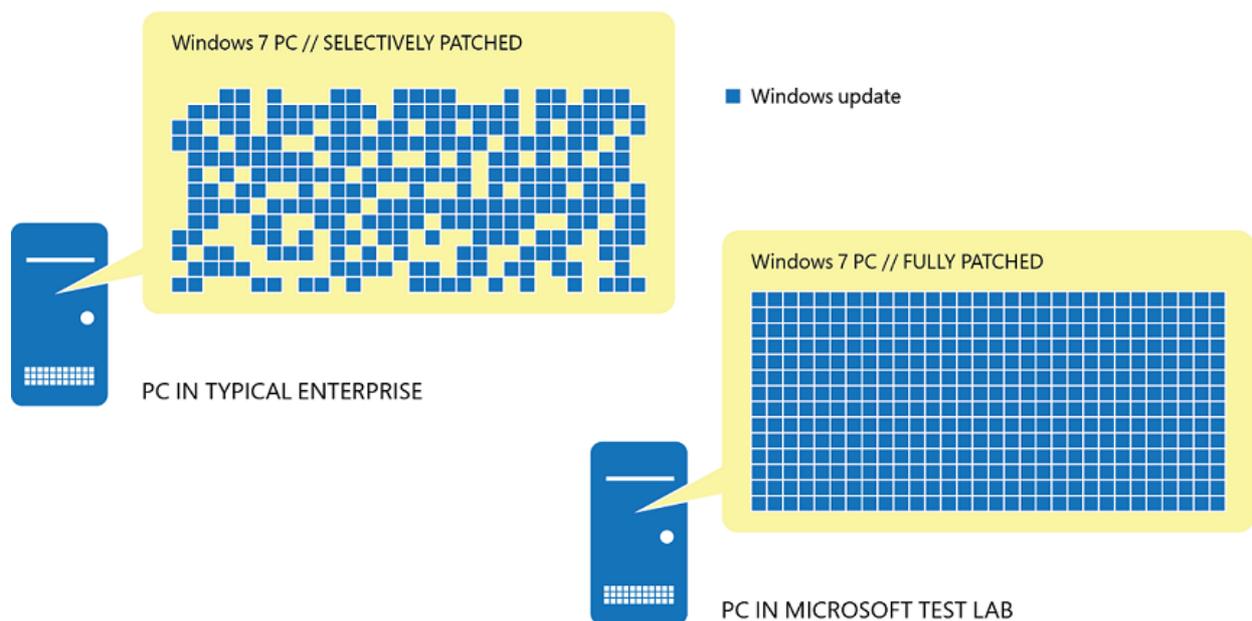
The feature update cadence has been aligned with Office 365 ProPlus updates. Starting with this falls' update, both Windows and Office will deliver their major updates semi-annually, around March and September. See [upcoming changes to Office 365 ProPlus update management](#) for more information about changes to Office update management.

Quality updates

Monthly updates in previous Windows versions were often overwhelming because of the sheer number of updates available each month. Many organizations selectively chose which updates they wanted to install and which they didn't, and this created countless scenarios in which organizations deployed essential security updates but picked only a subset of non-security fixes.

In Windows 10, rather than receiving several updates each month and trying to figure out which the organization needs, which ultimately causes platform fragmentation, administrators will see one cumulative monthly update that supersedes the previous month's update, containing both security and non-security fixes. This approach makes patching simpler and ensures that customers' devices are more closely aligned with the testing done at Microsoft, reducing unexpected issues resulting from patching. The left side of Figure 1 provides an example of Windows 7 devices in an enterprise and what their current patch level might look like. On the right is what Microsoft's test environment PCs contain. This drastic difference is the basis for many compatibility issues and system anomalies related to Windows updates.

Figure 1



Servicing channels

To align with the new method of delivering feature updates and quality updates in Windows 10, Microsoft introduced the concept of servicing channels to allow customers to designate how frequently their individual devices are updated. For example, an organization may have test devices that the IT department can update with new features as soon as possible, and then specialized devices that require a longer feature update cycle to ensure continuity.

With that in mind, Windows 10 offers 3 servicing channels. The [Windows Insider Program](#) provides organizations with the opportunity to test and provide feedback on features that will be shipped in the next feature update. The [Semi-Annual Channel](#) provides new functionality with twice-per-year feature update releases. Organizations can choose when to deploy updates from the Semi-Annual Channel. The [Long Term Servicing Channel](#), which is designed to be used only for specialized devices (which typically don't run Office) such as those that control medical equipment or ATM machines, receives new feature releases every two to three years. For details about the versions in each servicing channel, see [Windows 10 release information](#).

The concept of servicing channels is new, but organizations can use the same management tools they used to manage updates and upgrades in previous versions of Windows. For more information about the servicing tool options for Windows 10 and their capabilities, see [Servicing tools](#).

NOTE

Servicing channels are not the only way to separate groups of devices when consuming updates. Each channel can contain subsets of devices, which staggers servicing even further. For information about the servicing strategy and ongoing deployment process for Windows 10, including the role of servicing channels, see [Plan servicing strategy for Windows 10 updates](#).

Semi-Annual Channel

In the Semi-Annual servicing channel, feature updates are available as soon as Microsoft releases them. Windows 10, version 1511, had few servicing tool options to delay feature updates, limiting the use of the Semi-Annual servicing channel. Windows 10, version 1607 and onward, includes more servicing tools that can delay feature updates for up to 365 days. This servicing model is ideal for pilot deployments and testing of Windows 10 feature updates and for users such as developers who need to work with the latest features immediately. Once the latest release has gone through pilot deployment and testing, you will be able to choose the timing at which it goes into broad deployment.

When Microsoft officially releases a feature update for Windows 10, it is made available to any PC not configured to defer feature updates so that those devices can immediately install it. Organizations that use Windows Server Update Services (WSUS), Microsoft System Center Configuration Manager, or Windows Update for Business, however, can defer feature updates to selective devices by withholding their approval and deployment. In this scenario, the content available for the Semi-Annual Channel will be available but not necessarily immediately mandatory, depending on the policy of the management system. For more details about Windows 10 servicing tools, see [Servicing tools](#).

Organizations are expected to initiate targeted deployment on Semi-Annual Channel releases. All customers, independent software vendors (ISVs), and partners should use this time for testing and piloting within their environments. After 2-4 months, we will transition to broad deployment and encourage customers and partners to expand and accelerate the deployment of the release. For customers using Windows Update for Business, the Semi-Annual Channel provides three months of additional total deployment time before being required to update to the next release.

NOTE

All releases of Windows 10 have 18 months of servicing for all editions--these updates provide security and feature updates for the release. Customers running Enterprise and Education editions have an additional 12 months of servicing for specific Windows 10 releases, for a total of 30 months from initial release. These versions include Enterprise and Education editions for Windows 10, versions 1607, 1703, 1709 and 1803. Starting in October 2018, all Semi-Annual Channel releases in the September/October timeframe will also have the additional 12 months of servicing for a total of 30 months from the initial release. The Semi-Annual Channel versions released in March/April timeframe will continue to have an 18 month lifecycle.

NOTE

Organizations can electively delay feature updates into as many phases as they wish by using one of the servicing tools mentioned in the section Servicing tools.

Long-term Servicing Channel

Specialized systems—such as PCs that control medical equipment, point-of-sale systems, and ATMs—often require a longer servicing option because of their purpose. These devices typically perform a single important task and don't need feature updates as frequently as other devices in the organization. It's more important that these devices be kept as stable and secure as possible than up to date with user interface changes. The LTSC servicing model prevents Windows 10 Enterprise LTSB devices from receiving the usual feature updates and

provides only quality updates to ensure that device security stays up to date. With this in mind, quality updates are still immediately available to Windows 10 Enterprise LTSC clients, but customers can choose to defer them by using one of the servicing tools mentioned in the section Servicing tools.

NOTE

Windows 10 Enterprise LTSC is a separate Long Term Servicing Channel version.

Long-term Servicing channel is not intended for deployment on most or all the PCs in an organization; it should be used only for special-purpose devices. As a general guideline, a PC with Microsoft Office installed is a general-purpose device, typically used by an information worker, and therefore it is better suited for the Semi-Annual servicing channel.

Microsoft never publishes feature updates through Windows Update on devices that run Windows 10 Enterprise LTSC. Instead, it typically offers new LTSC releases every 2–3 years, and organizations can choose to install them as in-place upgrades or even skip releases over a 10-year life cycle.

NOTE

Windows 10 LTSC will support the currently released processors and chipsets at the time of release of the LTSC. As future CPU generations are released, support will be created through future Windows 10 LTSC releases that customers can deploy for those systems. For more information, see **Supporting the latest processor and chipsets on Windows** in [Lifecycle support policy FAQ - Windows Products](#).

The Long-term Servicing Channel is available only in the Windows 10 Enterprise LTSC edition. This edition of Windows doesn't include a number of applications, such as Microsoft Edge, Microsoft Store, Cortana (though limited search capabilities remain available), Microsoft Mail, Calendar, OneNote, Weather, News, Sports, Money, Photos, Camera, Music, and Clock. These apps are not supported in Windows 10 Enterprise LTSC edition, even if you install by using sideloading.

NOTE

If an organization has devices currently running Windows 10 Enterprise LTSC that it would like to change to the Semi-Annual Channel, it can make the change without losing user data. Because LTSC is its own SKU, however, an upgrade is required from Windows 10 Enterprise LTSC to Windows 10 Enterprise, which supports the Semi-Annual Channel.

Windows Insider

For many IT pros, gaining visibility into feature updates early—before they're available to the Semi-Annual Channel—can be both intriguing and valuable for future end user communications as well as provide the means to test for any issues on the next Semi-Annual Channel release. With Windows 10, feature flighting enables Windows Insiders to consume and deploy preproduction code to their test machines, gaining early visibility into the next build. Testing the early builds of Windows 10 helps both Microsoft and its customers because they have the opportunity to discover possible issues before the update is ever publicly available and can report it to Microsoft.

Microsoft recommends that all organizations have at least a few PCs enrolled in the Windows Insider Program and provide feedback on any issues they encounter. For information about the Windows Insider Program for Business, go to [Windows Insider Program for Business](#).

NOTE

Microsoft recommends that all organizations have at least a few PCs enrolled in the Windows Insider Program, to include the Windows Insider Program in their deployment plans and to provide feedback on any issues they encounter to Microsoft via our Feedback Hub app.

The Windows Insider Program isn't intended to replace Semi-Annual Channel deployments in an organization. Rather, it provides IT pros and other interested parties with pre-release Windows builds that they can test and ultimately provide feedback on to Microsoft.

Servicing tools

There are many tools with which IT pros can service Windows as a service. Each option has its pros and cons, ranging from capabilities and control to simplicity and low administrative requirements. The following are examples of the servicing tools available to manage Windows as a service updates:

- **Windows Update (stand-alone)** provides limited control over feature updates, with IT pros manually configuring the device to be in the Semi-Annual Channel. Organizations can target which devices defer updates by selecting the Defer upgrades check box in Start\Settings\Update & Security\Advanced Options on a Windows 10 client.
- **Windows Update for Business** is the second option for servicing Windows as a service. This servicing tool includes control over update deferral and provides centralized management using Group Policy. Windows Update for Business can be used to defer updates by up to 365 days, depending on the version. These deployment options are available to clients in the Semi-Annual Channel. In addition to being able to use Group Policy to manage Windows Update for Business, either option can be configured without requiring any on-premises infrastructure by using Intune.
- **Windows Server Update Services (WSUS)** provides extensive control over Windows 10 updates and is natively available in the Windows Server operating system. In addition to the ability to defer updates, organizations can add an approval layer for updates and choose to deploy them to specific computers or groups of computers whenever ready.
- **System Center Configuration Manager** provides the greatest control over servicing Windows as a service. IT pros can defer updates, approve them, and have multiple options for targeting deployments and managing bandwidth usage and deployment times.

With all these options, which an organization chooses depends on the resources, staff, and expertise its IT organization already has. For example, if IT already uses System Center Configuration Manager to manage Windows updates, it can continue to use it. Similarly, if IT is using WSUS, it can continue to use that. For a consolidated look at the benefits of each tool, see Table 1.

Table 1

SERVICING TOOL	CAN UPDATES BE DEFERRED?	ABILITY TO APPROVE UPDATES	PEER-TO-PEER OPTION	ADDITIONAL FEATURES
Windows Update	Yes (manual)	No	Delivery Optimization	None
Windows Update for Business	Yes	No	Delivery Optimization	Other Group Policy objects
WSUS	Yes	Yes	BranchCache or Delivery Optimization	Upstream/downstream server scalability

SERVICING TOOL	CAN UPDATES BE DEFERRED?	ABILITY TO APPROVE UPDATES	PEER-TO-PEER OPTION	ADDITIONAL FEATURES
Configuration Manager	Yes	Yes	BranchCache, Client Peer Cache	Distribution points, multiple deployment options

NOTE

Due to [naming changes](#), older terms like CB,CBB and LTSB may still be displayed in some of our products.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels (this topic)
<input type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input type="checkbox"/>	Build deployment rings for Windows 10 updates
<input type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Quick guide to Windows as a service](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Prepare servicing strategy for Windows 10 updates

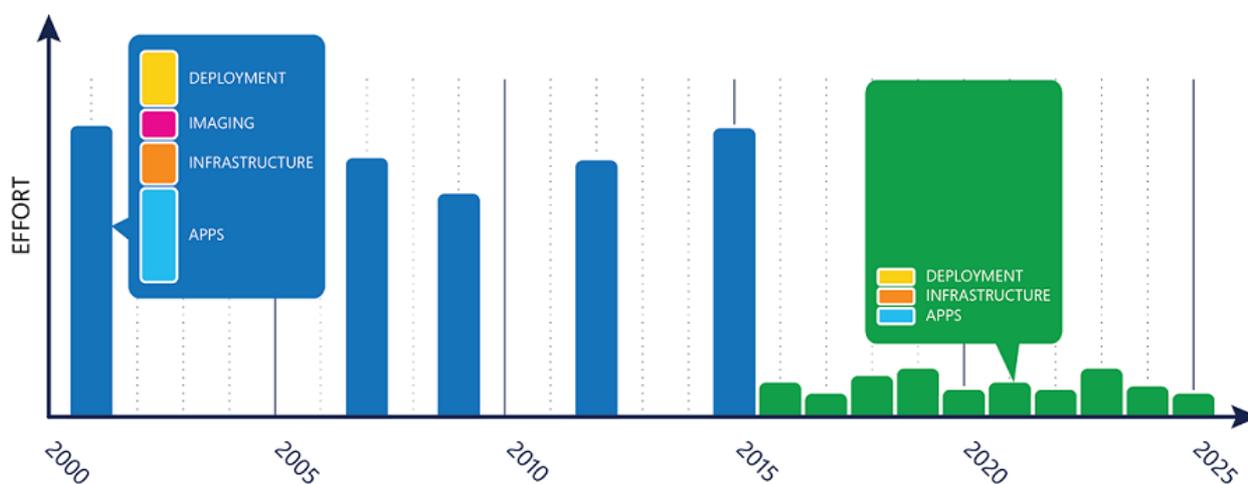
6/14/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

In the past, traditional Windows deployments tended to be large, lengthy, and expensive. Windows 10 offers a new approach to deploying both quality and feature updates, making the process much simpler and therefore the planning much more straightforward. With Windows as a service, the methodology around updating Windows has completely changed, moving away from major upgrades every few years to iterative updates twice per year. Each iteration contains a smaller subset of changes so that they won't seem like substantial differences, like they do today. This image illustrates the level of effort needed for traditional Windows deployments versus servicing Windows 10 and how it is now spread evenly over time versus spiking every few years.



Windows 10 spreads the traditional deployment effort of a Windows upgrade, which typically occurred every few years, over smaller, continuous updates. With this change, you must approach the ongoing deployment and servicing of Windows differently. A strong Windows 10 deployment strategy begins with establishing a simple, repeatable process for testing and deploying each feature update. Here's an example of what this process might look like:

- **Configure test devices.** Configure test devices in the Windows Insider Program so that Insiders can test feature updates before they're available to the Semi-Annual Channel. Typically, this would be a small number of test devices that IT staff members use to evaluate pre-releas builds of Windows. Microsoft provides current development builds to Windows Insider members approximately every week so that interested users can see the functionality Microsoft is adding. See the section Windows Insider for details on how to enroll in the Windows Insider Program on a Windows 10 device.
- **Identify excluded devices.** For some organizations, special-purpose devices such as those used to control factory or medical equipment or run ATMs require a stricter, less frequent feature update cycle than the Semi-annual Channel can offer. For those machines, you must install Windows 10 Enterprise LTSB to avoid feature updates for up to 10 years. Identify these devices, and separate them from the phased deployment and servicing cycles to help remove confusion for your administrators and ensure that devices are handled correctly.

- **Recruit volunteers.** The purpose of testing a deployment is to receive feedback. One effective way to recruit pilot users is to request volunteers. When doing so, clearly state that you're looking for feedback rather than people to just "try it out" and that there could be occasional issues involved with accepting feature updates right away. With Windows as a service, the expectation is that there should be few issues, but if an issue does arise, you want testers to let you know as soon as possible. When considering whom to recruit for pilot groups, be sure to include members who provide the broadest set of applications and devices to validate the largest number of apps and devices possible.
- **Update Group Policy.** Each feature update includes new group policies to manage new features. If you use Group Policy to manage devices, the Group Policy Admin for the Active Directory domain will need to download a .admx package and copy it to their [Central Store](#) (or to the [PolicyDefinitions](#) directory in the SYSVOL of a domain controller if not using a Central Store). Always manage new group policies from the version of Windows 10 they shipped with by using the Remote Server Administration Tools. The ADMX download package is created at the end of each development cycle and then posted for download. To find the ADMX download package for a given Windows build, search for "ADMX download for Windows build xxxx". For details about Group Policy management, see [How to create and manage the Central Store for Group Policy Administrative Templates in Windows](#)
- **Choose a servicing tool.** Decide which product you'll use to manage the Windows updates in your environment. If you're currently using Windows Server Update Services (WSUS) or System Center Configuration Manager to manage your Windows updates, you can continue using those products to manage Windows 10 updates. Alternatively, you can use Windows Update for Business. In addition to which product you'll use, consider how you'll deliver the updates. With Windows 10, multiple peer-to-peer options are available to make update distribution faster. For a comparison of tools, see [Servicing tools](#).
- **Prioritize applications.** First, create an application portfolio. This list should include everything installed in your organization and any webpages your organization hosts. Next, prioritize this list to identify those that are the most business critical. Because the expectation is that application compatibility with Windows 10 will be high, only the most business critical applications should be tested before the pilot phase; everything else can be tested afterwards. For more information about identifying compatibility issues with applications, see [Manage Windows upgrades with Upgrade Analytics](#).

NOTE

This strategy is applicable to approaching an environment in which Windows 10 already exists. For information about how to deploy or upgrade to Windows 10 where another version of Windows exists, see [Plan for Windows 10 deployment](#).

Windows 10 Enterprise LTSB is a separate Long Term Servicing Channel version.

Each time Microsoft releases a Windows 10 feature update, the IT department should use the following high-level process to help ensure that the broad deployment is successful:

1. **Validate compatibility of business critical apps.** Test your most important business-critical applications for compatibility with the new Windows 10 feature update running on your Windows Insider machines identified in the earlier "Configure test machines" step of the Predeployment strategy section. The list of applications involved in this validation process should be small because most applications can be tested during the pilot phase. For more information about device and application compatibility in Windows 10, see the section [Compatibility](#).
2. **Target and react to feedback.** With Windows 10, Microsoft expects application and device compatibility to be high, but it's still important to have targeted groups within both the IT department and business units to verify application compatibility for the remaining applications in your application portfolio. Because only the most business-critical applications are tested beforehand, this will represent the majority of application compatibility testing in your environment. This should not necessarily be a formal process but rather user validation through the use of a particular application. So, the next step is to deploy the feature update to

early-adopting IT users and your targeted groups running in the Semi-annual channel that you identified in the “Recruit volunteers” step of the Predeployment strategy section. Be sure to communicate clearly that you’re looking for feedback as soon as possible, and state exactly how users can submit feedback to you. Should an issue arise, have a remediation plan in place to address it.

3. **Deploy broadly.** Finally, focus on the large-scale deployment using deployment rings, like the ones discussed in Table 1. Build deployment rings that target groups of computers in your selected update-management product. To reduce risk as much as possible, construct your deployment rings in a way that splits individual departments into multiple rings. This way, if you were to encounter an issue, you don’t prevent any critical business from continuing. By using this method, each deployment ring reduces risk as more and more people have been updated in any particular department.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates (this topic)
<input type="checkbox"/>	Build deployment rings for Windows 10 updates
<input type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Build deployment rings for Windows 10 updates

5/31/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

For Windows as a service, maintenance is ongoing and iterative. Deploying previous versions of Windows required organizations to build sets of users to roll out the changes in phases. Typically, these users ranged (in order) from the most adaptable and least risky to the least adaptable or riskiest. With Windows 10, a similar methodology exists, but construction of the groups is a little different.

Deployment rings in Windows 10 are similar to the deployment groups most organizations constructed for previous major revision upgrades. They are simply a method by which to separate machines into a deployment timeline. With Windows 10, you construct deployment rings a bit differently in each servicing tool, but the concepts remain the same. Each deployment ring should reduce the risk of issues derived from the deployment of the feature updates by gradually deploying the update to entire departments. As previously mentioned, consider including a portion of each department's employees in several deployment rings.

Defining deployment rings is generally a one-time event (or at least infrequent), but IT should revisit these groups to ensure that the sequencing is still correct. Also, there are times in which client computers could move between different deployment rings when necessary.

Table 1 provides an example of the deployment rings you might use.

Table 1

DEPLOYMENT RING	SERVICING CHANNEL	DEFERRAL FOR FEATURE UPDATES	DEFERRAL FOR QUALITY UPDATES	EXAMPLE
Preview	Windows Insider Program	None	None	A few machines to evaluate early builds prior to their arrival to the semi-annual channel
Targeted	Semi-annual channel (Targeted)	None	None	Select devices across various teams used to evaluate the major release prior to broad deployment
Broad	Semi-annual channel	120 days	7-14 days	Broadly deployed to most of the organization and monitored for feedback Pause updates if there are critical issues

DEPLOYMENT RING	SERVICING CHANNEL	DEFERRAL FOR FEATURE UPDATES	DEFERRAL FOR QUALITY UPDATES	EXAMPLE
Critical	Semi-annual channel	180 days	30 days	Devices that are critical and will only receive updates once they've been vetted for a period of time by the majority of the organization

NOTE

In this example, there are no rings made up of the long-term servicing channel (LTSC). The LTSC does not receive feature updates.

As Table 1 shows, each combination of servicing channel and deployment group is tied to a specific deployment ring. As you can see, the associated groups of devices are combined with a servicing channel to specify which deployment ring those devices and their users fall into. The naming convention used to identify the rings is completely customizable as long as the name clearly identifies the sequence. Deployment rings represent a sequential deployment timeline, regardless of the servicing channel they contain. Deployment rings will likely rarely change for an organization, but they should be periodically assessed to ensure that the deployment cadence still makes sense.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates (this topic)
<input type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)

- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Manage software updates in Intune](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Assign devices to servicing channels for Windows 10 updates

6/14/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

TIP

If you're not familiar with the Windows 10 servicing or release channels, read [Servicing Channels](#) first.

Due to [naming changes](#), older terms like CB, CBB and LTSB may still be displayed in some of our products.

Semi-Annual Channel is the default servicing channel for all Windows 10 devices except those with the LTSB edition installed. The following table shows the servicing channels available to each Windows 10 edition.

WINDOWS 10 EDITION	SEMI-ANNUAL CHANNEL (TARGETED)	SEMI-ANNUAL CHANNEL	LONG-TERM SERVICING CHANNEL	INSIDER PROGRAM
Home	✓	✗	✗	✓
Pro	✓	✓	✗	✓
Enterprise	✓	✓	✗	✓
Enterprise LTSB	✗	✗	✓	✗
Pro Education	✓	✓	✗	✓
Education	✓	✓	✗	✓
Mobile	✓	✗	✗	✓
Mobile Enterprise	✓	✓	✗	✓

NOTE

The LTSB edition of Windows 10 is only available through the [Microsoft Volume Licensing Center](#).

NOTE

Semi-Annual Channel (Targeted) should be used only by the customers that are using [Windows Update for Business](#). For those who don't use Windows Update for Business, Semi-Annual Channel (Targeted) would be the same as Semi-Annual Channel.

Assign devices to Semi-Annual Channel

IMPORTANT

Due to [naming changes](#), older terms like CB, CBB and LTSB may still be displayed in some of our products.

In the following settings CB refers to Semi-Annual Channel (Targeted), while CBB refers to Semi-Annual Channel.

To assign a single PC locally to CBB

1. Go to **Settings > Update & security > Windows Update > Advanced options**.
2. Select **Defer feature updates**.

To assign PCs to CBB using Group Policy

- In Windows 10, version 1511:

Computer Configuration > Administrative Templates > Windows Components > Windows Update > **Defer Upgrades and Updates**

- In Windows 10, version 1607:

Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > **Select when Feature Updates are received** - enable policy and set branch readiness level to CBB

To assign PCs to CBB using MDM

- In Windows 10, version 1511:

../Vendor/MSFT/Policy/Config/Update/**RequireDeferUpgrade**

- In Windows 10, version 1607:

../Vendor/MSFT/Policy/Config/Update/**BranchReadinessLevel**

To assign Windows 10 Mobile Enterprise to CBB using MDM

- In Windows 10 Mobile Enterprise, version 1511:

../Vendor/MSFT/Policy/Config/Update/RequireDeferUpgrade

- In Windows 10 Mobile Enterprise, version 1607:

../Vendor/MSFT/Policy/Config/Update/BranchReadinessLevel

Enroll devices in the Windows Insider Program

To get started with the Windows Insider Program for Business, you will need to follow a few simple steps:

1. On the [Windows Insider](#) website, go to **For Business > Getting Started** to [register your organizational Azure AD account](#).
2. **Register your domain**. Rather than have each user register individually for Insider Preview builds,

administrators can simply [register their domain](#) and control settings centrally.

Note: The signed-in user needs to be a **Global Administrator** of the Azure AD domain in order to be able to register the domain.

3. Make sure the **Allow Telemetry** setting is set to **2** or higher.
4. Starting with Windows 10, version 1709, set policies to manage preview builds and their delivery:

The **Manage preview builds** setting gives administrators control over enabling or disabling preview build installation on a device. You can also decide to stop preview builds once the release is public.

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Windows Update/Windows Update for Business** - *Manage preview builds*
- MDM: **Update/ManagePreviewBuilds**

The **Branch Readiness Level** settings allows you to choose between preview flight rings, and allows you to defer or pause the delivery of updates.

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Windows Update/ Windows Update for Business** - *Select when Preview Builds and Feature Updates are received*
- MDM: **Update/BranchReadinessLevel**

For more information, see [Windows Insider Program for Business](#)

Block access to Windows Insider Program

To prevent devices in your enterprise from being enrolled in the Insider Program for early releases of Windows 10:

- Group Policy: Computer Configuration\Administrative Templates\Windows Components\Data Collection and Preview Builds**Toggle user control over Insider builds**
- MDM: Policy CSP - [System/AllowBuildPreview](#)

IMPORTANT

Starting with Windows 10, version 1709, this policy is replaced by **Manage preview builds** policy.

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Windows Update/Windows Update for Business** - *Manage preview builds*
- MDM: **Update/ManagePreviewBuilds**

Switching channels

During the life of a device, it may be necessary or desirable to switch between the available channels. Depending on the channel you are using, the exact mechanism for doing this can be different; some will be simple, others more involved.

FROM THIS CHANNEL	TO THIS CHANNEL	YOU NEED TO
Windows Insider Program	Semi-Annual Channel (Targeted)	Wait for the final Semi-Annual Channel release.
	Semi-Annual Channel	Not directly possible, because Windows Insider Program devices are automatically upgraded to the Semi-Annual Channel (Targeted) release at the end of the development cycle.

FROM THIS CHANNEL	TO THIS CHANNEL	YOU NEED TO
	Long-Term Servicing Channel	Not directly possible (requires wipe-and-load).
Semi-Annual Channel (Targeted)	Insider	Use the Settings app to enroll the device in the Windows Insider Program.
	Semi-Annual Channel	Select the Defer upgrade setting, or move the PC to a target group or flight that will not receive the next upgrade until it is business ready. Note that this change will not have any immediate impact; it only prevents the installation of the next Semi-Annual Channel release.
	Long-Term Servicing Channel	Not directly possible (requires wipe-and-load).
Semi-Annual Channel	Insider	Use the Settings app to enroll the device in the Windows Insider Program.
	Semi-Annual Channel (Targeted)	Disable the Defer upgrade setting, or move the device to a target group or flight that will receive the latest Current Semi-Annual Channel release.
	Long-Term Servicing Channel	Not directly possible (requires wipe-and-load).
Long-Term Servicing Channel	Insider	Use media to upgrade to the latest Windows Insider Program build.
	Semi-Annual Channel (Targeted)	Use media to upgrade. Note that the Semi-Annual Channel build must be a later build.
	Semi-Annual Channel	Use media to upgrade. Note that the Semi-Annual Channel build must be a later build.

Block user access to Windows Update settings

In Windows 10, administrators can control user access to Windows Update. By enabling the Group Policy setting under **Computer Configuration\Administrative Templates\Windows Components\Windows update\Remove access to use all Windows update features**, administrators can disable the "Check for updates" option for users. Any background update scans, downloads and installations will continue to work as configured.

NOTE

In Windows 10, any Group Policy user configuration settings for Windows Update were deprecated and are no longer supported on this platform.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates
<input checked="" type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates (this topic)
<input type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Get started with Windows Update

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 10

With the release of Windows 10, we moved the update model to the Unified Update Platform. Unified Update Platform (UUP) is a single publishing, hosting, scan and download model for all types of OS updates, desktop and mobile for all Windows-based operating systems, for everything from monthly quality updates to new feature updates.

Use the following information to get started with Windows Update:

- Understand the UUP architecture
- Understand [how Windows Update works](#)
- Find [Windows Update log files](#)
- Learn how to [troubleshoot Windows Update](#)
- Review [common Windows Update errors](#) and check out the [error code reference](#)
- Review [other resources](#) to help you use Windows Update

Unified Update Platform (UUP) architecture

To understand the changes to the Windows Update architecture that UUP introduces let's start with some new key terms.

Unified Update Platform	Client Upgrades	Client Updates
User Experience	Update UI	
Update Scheduling	Orchestrator	
Check for Updates and Downloads	WU Client	
Install Handler	WU Arbiter Handler	
Deployment Manager	Deployment Arbiter	
Installer	Upgrade Engine	CBS

- **Update UI** – The user interface to initiate Windows Update check and history. Available under **Settings --> Update & Security --> Windows Update**.
- **Update Session Orchestrator (USO)**- A Windows OS component that orchestrates the sequence of downloading and installing various update types from Windows Update.

Update types-

- OS Feature updates
- OS Security updates
- Device drivers

- Defender definition updates

NOTE

Other types of updates, like Office desktop updates, are installed if the user opts into Microsoft Update.

Store apps aren't installed by USO, today they are separate.

- **WU Client/ UpdateAgent** - The component running on your PC. It's essentially a DLL that is downloaded to the device when an update is applicable. It surfaces the APIs needed to perform an update, including those needed to generate a list of payloads to download, as well as starts stage and commit operations. It provides a unified interface that abstracts away the underlying update technologies from the caller.
- **WU Arbiter handle**- Code that is included in the UpdateAgent binary. The arbiter gathers information about the device, and uses the CompDB(s) to output an action list. It is responsible for determining the final "composition state" of your device, and which payloads (like ESDs or packages) are needed to get your device up to date.
- **Deployment Arbiter**- A deployment manager that calls different installers. For example, CBS.

Additional components include the following-

- **CompDB** – A generic term to refer to the XML describing information about target build composition, available diff packages, and conditional rules.
- **Action List** – The payload and additional information needed to perform an update. The action list is consumed by the UpdateAgent, as well as other installers to determine what payload to download. It's also consumed by the "Install Agent" to determine what actions need to be taken, such as installing or removing packages.

How does Windows Update work?

5/31/2019 • 5 minutes to read • [Edit Online](#)

Applies to: Windows 10

The Windows Update workflow has four core areas of functionality:

Scan

1. Orchestrator schedules the scan.
2. Orchestrator verifies admin approvals and policies for download.

Download

1. Orchestrator initiates downloads.
2. Windows Update downloads manifest files and provides them to the arbiter.
3. The arbiter evaluates the manifest and tells the Windows Update client to download files.
4. Windows Update client downloads files in a temporary folder.
5. The arbiter stages the downloaded files.

Install

1. Orchestrator initiates the installation.
2. The arbiter calls the installer to install the package.

Commit

1. Orchestrator initiates a restart.
2. The arbiter finalizes before the restart.

How updating works

During the updating process, the Windows Update Orchestrator operates in the background to scan, download, and install updates. It does this automatically, according to your settings, and in a silent manner that doesn't disrupt your computer usage.

Scanning updates



The Windows Update Orchestrator on your PC checks the Microsoft Update server or your WSUS endpoint for new updates at random intervals. The randomization ensures that the Windows Update server isn't overloaded with requests all at the same time. The Update Orchestrator searches only for updates that have been added since the last time updates were searched, allowing it to find updates quickly and efficiently.

When checking for updates, the Windows Update Orchestrator evaluates whether the update is appropriate for your computer using guidelines defined by the publisher of the update, for example, Microsoft Office including

enterprise group policies.

Make sure you're familiar with the following terminology related to Windows Update scan:

TERM	DEFINITION
Update	We use this term to mean a lot of different things, but in this context it's the actual patch or change.
Bundle update	An update that contains 1-N child updates; doesn't contain payload itself.
Child update	Leaf update that's bundled by another update; contains payload.
Detectoid update	A special 'update' that contains "IsInstalled" applicability rule only and no payload. Used for prereq evaluation.
Category update	A special 'detectoid' that has always true IsInstalled rule. Used for grouping updates and for client to filter updates.
Full scan	Scan with empty datastore.
Delta scan	Scan with updates from previous scan already cached in datastore.
Online scan	Scan that hits network and goes against server on cloud.
Offline scan	Scan that doesn't hit network and goes against local datastore. Only useful if online scan has been performed before.
CatScan	Category scan where caller can specify a categoryId to get updates published under the categoryId.
AppCatScan	Category scan where caller can specify an AppCategoryId to get apps published under the appCategoryId.
Software sync	Part of the scan that looks at software updates only (OS and apps).
Driver sync	Part of the scan that looks at Driver updates only. This is run after Software sync and is optional.
ProductSync	Attributes based sync, where client provides a list of device, product and caller attributes ahead of time to allow service to evaluate applicability in the cloud.

How Windows Update scanning works

Windows Update takes the following sets of actions when it runs a scan.

Starts the scan for updates

When users start scanning in Windows Update through the Settings panel, the following occurs:

- The scan first generates a "ComApi" message. The caller (Windows Defender Antivirus) tells the WU engine to scan for updates.
- "Agent" messages: queueing the scan, then actually starting the work:

- Updates are identified by the different IDs ("Id = 10", "Id = 11") and from the different thread ID numbers.
- Windows Update uses the thread ID filtering to concentrate on one particular task.

```

05/26/2017-11:30:25.982 0A1C 41F0 ComApi * START * Search ClientId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), ServiceId = 7971F918-A847-4430-9279-4A52D1EFE18D (cV = IGqGElKdWEyr9okh.1.0.0)
05/26/2017-11:30:26.037 3A68 4A14 Agent * START * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10]
05/26/2017-11:30:26.037 0A1C 41F0 ComApi * START * Search ClientId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), ServiceId = 8B24B027-1DEE-BABB-9A95-3517DFB9C552 (cV = IGqGElKdWEyr9okh.1.0.0)
05/26/2017-11:30:26.045 3A68 4918 Agent * END * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10]
05/26/2017-11:30:26.052 3A68 4918 Agent * START * Finding updates CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10
05/26/2017-11:30:26.052 3A68 4918 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and Category)Ds contains '8c3cc84-7410-4a95-8b89-a166a0190486' and Category)Ds contains 'e0789628-ce08-4437-be74-2495b84243b'"
05/26/2017-11:30:26.052 3A68 4918 Agent ServiceId = (7971F918-A847-4430-9279-4A52D1EFE18D) Third party service
05/26/2017-11:30:26.062 3A68 4A14 Agent * START * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11]
05/26/2017-11:30:26.070 3A68 39F8 Agent * END * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11]
05/26/2017-11:30:26.077 3A68 39F8 Agent * START * Finding updates CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11
05/26/2017-11:30:26.077 3A68 39F8 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and Category)Ds contains '8c3cc84-7410-4a95-8b89-a166a0190486' and Category)Ds contains 'e0789628-ce08-4437-be74-2495b84243b'"
05/26/2017-11:30:26.077 3A68 39F8 Agent ServiceId = (8B24B027-1DEE-BABB-9A95-3517DFB9C552) Third party service

```

Identifies service IDs

- Service IDs indicate which update source is being scanned. Note The next screen shot shows Microsoft Update and the Flighting service.
- The Windows Update engine treats every service as a separate entity, even though multiple services may contain the same updates.

```

05/26/2017-11:30:25.982 0A1C 41F0 ComApi * START * Search ClientId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), ServiceId = 7971F918-A847-4430-9279-4A52D1EFE18D (cV = IGqGElKdWEyr9okh.1.0.0)
05/26/2017-11:30:26.037 3A68 4A14 Agent * START * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10]
05/26/2017-11:30:26.037 0A1C 41F0 ComApi * START * Search ClientId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), ServiceId = 8B24B027-1DEE-BABB-9A95-3517DFB9C552 (cV = IGqGElKdWEyr9okh.1.0.0)
05/26/2017-11:30:26.045 3A68 4918 Agent * END * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10]
05/26/2017-11:30:26.052 3A68 4918 Agent * START * Finding updates CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 10
05/26/2017-11:30:26.052 3A68 4918 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and Category)Ds contains '8c3cc84-7410-4a95-8b89-a166a0190486' and Category)Ds contains 'e0789628-ce08-4437-be74-2495b84243b'"
05/26/2017-11:30:26.052 3A68 4918 Agent ServiceId = (7971F918-A847-4430-9279-4A52D1EFE18D) Third party service
05/26/2017-11:30:26.062 3A68 39F8 Agent * END * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11]
05/26/2017-11:30:26.070 3A68 39F8 Agent * END * Queueing Finding updates [CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11]
05/26/2017-11:30:26.077 3A68 39F8 Agent * START * Finding updates CallerId = Windows Defender Antivirus (77BDAF73-B396-481F-9042-AD358843EC24), Id = 11
05/26/2017-11:30:26.077 3A68 39F8 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and Category)Ds contains '8c3cc84-7410-4a95-8b89-a166a0190486' and Category)Ds contains 'e0789628-ce08-4437-be74-2495b84243b'"
05/26/2017-11:30:26.077 3A68 39F8 Agent ServiceId = (8B24B027-1DEE-BABB-9A95-3517DFB9C552) Third party service

```

- Common service IDs

IMPORTANT

ServiceId here identifies a client abstraction, not any specific service in the cloud. No assumption should be made of which server a serviceId is pointing to, it's totally controlled by the SLS responses.

SERVICE	SERVICEID
Unspecified / Default	WU, MU or WSUS 00000000-0000-0000-0000-000000000000
WU	9482F4B4-E343-43B6-B170-9A65BC822C77
MU	7971F918-a847-4430-9279-4a52d1efe18d
Store	855E8A7C-ECB4-4CA3-B045-1DFA50104289
OS Flighting	8B24B027-1DEE-BABB-9A95-3517DFB9C552
WSUS or SCCM	Via ServerSelection::ssManagedServer 3DA21691-E39D-4da6-8A4B-B43877BCB1B7
Offline scan service	Via IUpdateServiceManager::AddScanPackageService

Finds network faults

Common update failure is caused due to network issues. To find the root of the issue:

- Look for "ProtocolTalker" messages to see client-server sync network traffic.
- "SOAP faults" can be either client- or server-side issues; read the message.
- The WU client uses SLS (Service Locator Service) to discover the configurations and endpoints of Microsoft network update sources – WU, MU, Flighting.

NOTE

Warning messages for SLS can be ignored if the search is against WSUS/SCCM.

- On sites that only use WSUS/SCCM, the SLS may be blocked at the firewall. In this case the SLS request will fail, and can't scan against Windows Update or Microsoft Update but can still scan against WSUS/SCCM, since it's locally configured.

```
5/26/2017-17:30:38.38.8383104.7368.1504.ProtocolTalker.[0]1CC8.05E0:06/22/2017-22:02:38.838.[agent]ServiceId = {8B24B027-1DEE-BABB-9A95-3517DFB9C552}.Server URL = https://fe3.delivery.mp.microsoft.com/ClientWebService/client.asmx
5/26/2017-17:30:38.38.8383129.7368.1504.ProtocolTalker.[0]1CC8.05E0:06/22/2017-22:02:38.838.[agent]OK to reuse existing configuration
5/26/2017-17:30:38.38.8383166.7368.1504.ProtocolTalker.[0]1CC8.05E0:06/22/2017-22:02:38.838.[agent]Existing cookie is valid, just use it
```

Downloading updates



Once the Windows Update Orchestrator determines which updates apply to your computer, it will begin downloading the updates, if you have selected the option to automatically download updates. It does this in the background without interrupting your normal use of the computer.

To ensure that your other downloads aren't affected or slowed down because updates are downloading, Windows Update uses the Delivery Optimization (DO) technology which downloads updates and reduces bandwidth consumption.

For more information see [Configure Delivery Optimization for Windows 10 updates](#).

Installing updates



When an update is applicable, the "Arbiter" and metadata are downloaded. Depending on your Windows Update settings, when downloading is complete, the Arbiter will gather details from the device, and compare that with the downloaded metadata to create an "action list".

The action list describes all the files needed from WU, and what the install agent (such as CBS or Setup) should do with them. The action list is provided to the install agent along with the payload to begin the installation.

Committing Updates



When the option to automatically install updates is configured, the Windows Update Orchestrator, in most cases, automatically restarts the PC for you after installing the updates. This is necessary because your PC may be

insecure, or not fully updated, until a restart is completed. You can use Group Policy settings, mobile device management (MDM), or the registry (not recommended) to configure when devices will restart after a Windows 10 update is installed.

For more information see [Manage device restarts after updates](#).

Windows Update log files

6/14/2019 • 5 minutes to read • [Edit Online](#)

Applies to: Windows 10

The following table describes the log files created by Windows Update.

LOG FILE	LOCATION	DESCRIPTION	WHEN TO USE
windowsupdate.log	C:\Windows\Logs\Windows Update	Starting in Windows 8.1 and continuing in Windows 10, Windows Update client uses Event Tracing for Windows (ETW) to generate diagnostic logs.	If you receive an error message when you run Windows Update (WU), you can use the information that is included in the Windowsupdate.log log file to troubleshoot the issue.
UpdateSessionOrchestration.etl	C:\ProgramData\USOShared\Logs	Starting Windows 10, the Update Orchestrator is responsible for sequence of downloading and installing various update types from Windows Update. And the events are logged to these etl files.	When you see that the updates are available but download is not getting triggered. When Updates are downloaded but installation is not triggered. When Updates are installed but reboot is not triggered.
NotificationUxBroker.etl	C:\ProgramData\USOShared\Logs	Starting Windows 10, the notification toast or the banner is triggered by this NotificationUxBroker.exe . And the logs to check its working is this etl.	When you want to check whether the Notification was triggered or not for reboot or update availability etc.
CBS.log	%systemroot%\Logs\CBS	This logs provides insight on the update installation part in the servicing stack.	To troubleshoot the issues related to WU installation.

Generating WindowsUpdate.log

To merge and convert WU trace files (.etl files) into a single readable WindowsUpdate.log file, see [Get-WindowsUpdateLog](#).

NOTE

When you run the **Get-WindowsUpdateLog** cmdlet, an copy of WindowsUpdate.log file is created as a static log file. It does not update as the old WindowsUpdate.log unless you run **Get-WindowsUpdateLog** again.

Windows Update log components

The WU engine has different component names. The following are some of the most common components that appear in the WindowsUpdate.log file:

- AGENT- Windows Update agent

- AU - Automatic Updates is performing this task
- AUCLNT- Interaction between AU and the logged-on user
- CDM- Device Manager
- CMPRESS- Compression agent
- COMAPI- Windows Update API
- DRIVER- Device driver information
- DTASTOR- Handles database transactions
- EEHANDLER- Expression handler that's used to evaluate update applicability
- HANDLER- Manages the update installers
- MISC- General service information
- OFFLSNC- Detects available updates without network connection
- PARSER- Parses expression information
- PT- Synchronizes updates information to the local datastore
- REPORT- Collects reporting information
- SERVICE- Startup/shutdown of the Automatic Updates service
- SETUP- Installs new versions of the Windows Update client when it is available
- SHUTDWN- Install at shutdown feature
- WUREDIR- The Windows Update redirector files
- WUWEB- The Windows Update ActiveX control
- ProtocolTalker - Client-server sync
- DownloadManager - Creates and monitors payload downloads
- Handler, Setup - Installer handlers (CBS, and so on)
- EEHandler - Evaluating update applicability rules
- DataStore - Caching update data locally
- IdleTimer - Tracking active calls, stopping a service

NOTE

Many component log messages are invaluable if you are looking for problems in that specific area. However, they can be useless if you don't filter to exclude irrelevant components so that you can focus on what's important.

Windows Update log structure

The Windows update log structure is separated into four main identities:

- Time Stamps
- Process ID and Thread ID
- Component Name
- Update Identifiers
 - Update ID and Revision Number
 - Revision ID
 - Local ID
 - Inconsistent terminology

The WindowsUpdate.log structure is discussed in the following sections.

Time stamps

The time stamp indicates the time at which the logging occurs.

- Messages are usually in chronological order, but there may be exceptions.
- A pause during a sync can indicate a network problem, even if the scan succeeds.

- A long pause near the end of a scan can indicate a supersedence chain issue.

```

05/26/2017-11:30:26.077 368 39F8 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and C
05/26/2017-11:30:26.077 368 39F8 Agent ServiceID = {8B24B027-1DEE-BABB-9A95-3517
05/26/2017-11:30:26.159 368 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.363 368 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.428 368 4918 Agent update {9E4D4E24-B06F-4930-BC70-77C563FC
05/26/2017-11:30:26.428 368 4918 Agent update {79D6195D-89A6-4391-B550-EC5A7165
05/26/2017-11:30:26.428 368 4918 Agent update {CAF1D4A2-0881-46D0-8552-170349A1;
05/26/2017-11:30:26.428 368 4918 Agent update {DE25AECF-4BF6-4094-98C3-7896860D
05/26/2017-11:30:26.428 368 4918 Agent update {AD2BC8ED-5A55-4C2F-A891-F886FE29
05/26/2017-11:30:26.428 368 4918 Agent update {228A2AF3-2B2A-4A32-AEA6-1BA26C92

```

Process ID and thread ID

The Process IDs and Thread IDs are random, and they can vary from log to log and even from service session to service session within the same log.

- The first four hex digits are the process ID.
- The next four hex digits are the thread ID.
- Each component, such as the USO, WU engine, COM API callers, and WU installer handlers, has its own process ID.

```

05/26/2017-11:30:26.077 3A68 39F8 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and C
05/26/2017-11:30:26.077 3A68 39F8 Agent ServiceID = {8B24B027-1DEE-BABB-9A95-3517
05/26/2017-11:30:26.159 3A68 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.363 3A68 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.428 3A68 4918 Agent update {9E4D4E24-B06F-4930-BC70-77C563FC
05/26/2017-11:30:26.428 3A68 4918 Agent update {79D6195D-89A6-4391-B550-EC5A7165
05/26/2017-11:30:26.428 3A68 4918 Agent update {CAF1D4A2-0881-46D0-8552-170349A1;
05/26/2017-11:30:26.428 3A68 4918 Agent update {DE25AECF-4BF6-4094-98C3-7896860D
05/26/2017-11:30:26.428 3A68 4918 Agent update {AD2BC8ED-5A55-4C2F-A891-F886FE29
05/26/2017-11:30:26.428 3A68 4918 Agent update {228A2AF3-2B2A-4A32-AEA6-1BA26C92

```

Component name

Search for and identify the components that are associated with the IDs. Different parts of the WU engine have different component names. Some of them are as follows:

- ProtocolTalker - Client-server sync
- DownloadManager - Creates and monitors payload downloads
- Handler, Setup - Installer handlers (CBS, etc.)
- EEHandler - Evaluating update applicability rules
- DataStore - Caching update data locally
- IdleTimer - Tracking active calls, stopping service

```

05/26/2017-11:30:26.077 3A68 39F8 Agent Criteria = "(IsInstalled = 0 and IsHidden = 0 and C
05/26/2017-11:30:26.077 3A68 39F8 Agent ServiceID = {8B24B027-1DEE-BABB-9A95-3517
05/26/2017-11:30:26.159 3A68 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.363 3A68 4918 ProtocolTalker ServiceId = {7971F918-A847-4430-9279
05/26/2017-11:30:26.428 3A68 4918 Agent update {9E4D4E24-B06F-4930-BC70-77C563FC
05/26/2017-11:30:26.428 3A68 4918 Agent update {79D6195D-89A6-4391-B550-EC5A7165
05/26/2017-11:30:26.428 3A68 4918 Agent update {CAF1D4A2-0881-46D0-8552-170349A1;
05/26/2017-11:30:26.428 3A68 4918 Agent update {DE25AECF-4BF6-4094-98C3-7896860D
05/26/2017-11:30:26.428 3A68 4918 Agent update {AD2BC8ED-5A55-4C2F-A891-F886FE29
05/26/2017-11:30:26.428 3A68 4918 Agent update {228A2AF3-2B2A-4A32-AEA6-1BA26C92

```

Update identifiers

Update ID and revision number

There are different identifiers for the same update in different contexts. It's important to know the identifier schemes.

- Update ID: A GUID (indicated in the previous screen shot) that's assigned to a given update at publication time
- Revision number: A number incremented every time that a given update (that has a given update ID) is modified and republished on a service
- Revision numbers are reused from one update to another (not a unique identifier).

- The update ID and revision number are often shown together as "{GUID}.revision."

```
05/26/2017-11:30:26.428 3A68 4918 Agent updat {9E4D4E24-B06F-4930-BC70-77C563FC8C48}.100 is
```

Revision ID

- A Revision ID (do not confuse this with "revision number") is a serial number that's issued when an update is initially published or revised on a given service.
- An existing update that's revised keeps the same update ID (GUID), has its revision number incremented (for example, from 100 to 101), but gets a completely new revision ID that is not related to the previous ID.
- Revision IDs are unique on a given update source, but not across multiple sources.
- The same update revision may have completely different revision IDs on WU and WSUS.
- The same revision ID may represent different updates on WU and WSUS.

Local ID

- Local ID is a serial number issued when an update is received from a service by a given WU client
- Usually seen in debug logs, especially involving the local cache for update info (Datastore)
- Different client PCs will assign different Local IDs to the same update
- You can find the local IDs that a client is using by getting the client's %WINDIR%\SoftwareDistribution\Datastore\Datastore.edb file

Inconsistent terminology

- Sometimes the logs use terms inconsistently. For example, the InstalledNonLeafUpdateIDs list actually contains revision IDs, not update IDs.
- Recognize IDs by form and context:
 - GUIDs are update IDs
 - Small integers that appear alongside an update ID are revision numbers
 - Large integers are typically revision IDs
 - Small integers (especially in Datastore) can be local IDs

```
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker SyncUpdateParameters (Loop 1):
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker ExpressQuery: FALSE
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker InstalledNonLeafUpdateIDs (22):
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 1
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 2
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 3
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 5169044
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 23110993
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 59830006
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 60484010
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 62450019
05/26/2017-11:30:27.101 3A68 39F8 ProtocolTalker 98959023
```

Windows Setup log files analysis using SetupDiag tool

SetupDiag is a diagnostic tool that can be used for analysis of logs related to installation of Windows Updates. For detailed information, see [SetupDiag](#).

Windows Update troubleshooting

6/14/2019 • 11 minutes to read • [Edit Online](#)

Applies to: Windows 10

If you run into problems when using Windows Update, start with the following steps:

1. Run the built-in Windows Update troubleshooter to fix common issues. Navigate to **Settings > Update & Security > Troubleshoot > Windows Update**.
2. Install the most recent Servicing Stack Update (SSU) that matches your version of Windows from the Microsoft Update Catalog. See [Servicing stack updates](#) for more details on SSU.
3. Make sure that you install the latest Windows updates, cumulative updates, and rollup updates. To verify the update status, refer to the appropriate update history for your system:
 - [Windows 10, version 1809 and Windows Server 2019](#)
 - [Windows 10, version 1803](#)
 - [Windows 10, version 1709](#)
 - [Windows 10, version 1703](#)
 - [Windows 10 and Windows Server 2016](#)
 - [Windows 8.1 and Windows Server 2012 R2](#)
 - [Windows Server 2012](#)
 - [Windows 7 SP1 and Windows Server 2008 R2 SP1](#)

Advanced users can also refer to the [log](#) generated by Windows Update for further investigation.

You might encounter the following scenarios when using Windows Update.

Why am I offered an older update/upgrade?

The update that is offered to a device depends on several factors. Some of the most common attributes include the following:

- OS Build
- OS Branch
- OS Locale
- OS Architecture
- Device update management configuration

If the update you're offered isn't the most current available, it might be because your device is being managed by a WSUS server, and you're being offered the updates available on that server. It's also possible, if your device is part of a Windows as a Service deployment ring, that your admin is intentionally slowing the rollout of updates. Since the WaaS rollout is slow and measured to begin with, all devices will not receive the update on the same day.

My machine is frozen at scan. Why?

The Settings UI is talking to the Update Orchestrator service which in turn is talking to Windows Update service. If these services stop unexpectedly then you might see this behavior. In such cases, do the following:

1. Close the Settings app and reopen it.

2. Launch Services.msc and check if the following services are running:

- Update State Orchestrator
- Windows Update

Feature updates are not being offered while other updates are

On computers running [Windows 10 1709 or higher](#) configured to update from Windows Update (usually WUfB scenario) servicing and definition updates are being installed successfully, but feature updates are never offered.

Checking the WindowsUpdate.log reveals the following error:

```
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent * START * Finding updates CallerId = Update;taskhostw Id =
25
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent Online = Yes; Interactive = No; AllowCachedResults = No;
Ignore download priority = No
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent ServiceID = {855E8A7C-ECB4-4CA3-B045-1DFA50104289} Third
party service
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent Search Scope = {Current User}
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent Caller SID for Applicability: S-1-12-1-2933642503-
1247987907-1399130510-4207851353
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc Got 855E8A7C-ECB4-4CA3-B045-1DFA50104289 redir Client/Server
URL: https://fe3.delivery.mp.microsoft.com/ClientWebService/client.asmx""
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc Token Requested with 0 category IDs.
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc GetUserTickets: No user tickets found. Returning
WU_E_NO_USERTOKEN.
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] Method failed
[AuthTicketHelper::GetDeviceTickets:570]
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] Method failed
[AuthTicketHelper::GetDeviceTickets:570]
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] GetDeviceTickets
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] Method failed
[AuthTicketHelper::AddTickets:1092]
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] Method failed
[CUpdateEndpointProvider::GenerateSecurityTokenWithAuthTickets:1587]
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] GetAgentTokenFromServer
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] GetAgentToken
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] EP:Call to GetEndpointToken
YYYY/MM/DD HH:mm:ss:SSS PID TID Misc *FAILED* [80070426] Failed to obtain service 855E8A7C-ECB4-
4CA3-B045-1DFA50104289 plugin Client/Server auth token of type 0x00000001
YYYY/MM/DD HH:mm:ss:SSS PID TID ProtocolTalker *FAILED* [80070426] Method failed
[CAgentProtocolTalkerContext::DetermineServiceEndpoint:377]
YYYY/MM/DD HH:mm:ss:SSS PID TID ProtocolTalker *FAILED* [80070426] Initialization failed for Protocol
Talker Context
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent Exit code = 0x80070426
YYYY/MM/DD HH:mm:ss:SSS PID TID Agent * END * Finding updates CallerId = Update;taskhostw Id = 25
```

The 0x80070426 error code translates to:

```
ERROR_SERVICE_NOT_ACTIVE - # The service has not been started.
```

Microsoft Account Sign In Assistant (MSA or wlidsvc) is the service in question. The DCAT Flighting service (ServiceId: 855E8A7C-ECB4-4CA3-B045-1DFA50104289) relies on the Microsoft Account Sign In Assistant (MSA) to get the Global Device ID for the device. Without the MSA service running, the global device ID will not be generated and sent by the client and the search for feature updates never completes successfully.

In order to solve this issue, we need to reset the MSA service to the default StartType of manual.

Issues related to HTTP/Proxy

Windows Update uses WinHttp with Partial Range requests (RFC 7233) to download updates and applications

from Windows Update servers or on-premises WSUS servers. Because of this proxy servers configured on the network must support HTTP RANGE requests. If a proxy was configured in Internet Explorer (User level) but not in WinHTTP (System level), connections to Windows Update will fail.

To fix this issue, configure a proxy in WinHTTP by using the following netsh command:

```
netsh winhttp set proxy ProxyServerName:PortNumber
```

NOTE

You can also import the proxy settings from Internet Explorer by using the following command: netsh winhttp import proxy source=ie

If downloads through a proxy server fail with a 0x80d05001 DO_E_HTTP_BLOCKSIZE_MISMATCH error, or if you notice high CPU usage while updates are downloading, check the proxy configuration to permit HTTP RANGE requests to run.

You may choose to apply a rule to permit HTTP RANGE requests for the following URLs:

*.download.windowsupdate.com

*.dl.delivery.mp.microsoft.com

*.emdl.ws.microsoft.com

If you cannot permit RANGE requests, keep in mind that this means you are downloading more content than needed in updates (as delta patching will not work).

The update is not applicable to your computer

The most common reasons for this error are described in the following table:

CAUSE	EXPLANATION	RESOLUTION
Update is superseded	As updates for a component are released, the updated component will supersede an older component that is already on the system. When this occurs, the previous update is marked as superseded. If the update that you're trying to install already has a newer version of the payload on your system, you may encounter this error message.	Check that the package that you are installing contains newer versions of the binaries. Or, check that the package is superseded by another new package.
Update is already installed	If the update that you're trying to install was previously installed, for example, by another update that carried the same payload, you may encounter this error message.	Verify that the package that you are trying to install was not previously installed.

CAUSE	EXPLANATION	RESOLUTION
Wrong update for architecture	Updates are published by CPU architecture. If the update that you're trying to install does not match the architecture for your CPU, you may encounter this error message.	Verify that the package that you're trying to install matches the Windows version that you are using. The Windows version information can be found in the "Applies To" section of the article for each update. For example, Windows Server 2012-only updates cannot be installed on Windows Server 2012 R2-based computers. Also, verify that the package that you are installing matches the processor architecture of the Windows version that you are using. For example, an x86-based update cannot be installed on x64-based installations of Windows.
Missing prerequisite update	Some updates require a prerequisite update before they can be applied to a system. If you are missing a prerequisite update, you may encounter this error message. For example, KB 2919355 must be installed on Windows 8.1 and Windows Server 2012 R2 computers before many of the updates that were released after April 2014 can be installed.	Check the related articles about the package in the Microsoft Knowledge Base (KB) to make sure that you have the prerequisite updates installed. For example, if you encounter the error message on Windows 8.1 or Windows Server 2012 R2, you may have to install the April 2014 update 2919355 as a prerequisite and one or more prerequisite servicing updates (KB 2919442 and KB 3173424). Note: To determine if these prerequisite updates are installed, run the following PowerShell command: get-hotfix KB3173424,KB2919355,KB2919442 If the updates are installed, the command will return the installed date in the "InstalledOn" section of the output.

Issues related to firewall configuration

Error that may be seen in the WU logs:

```
DownloadManager Error 0x800706d9 occurred while downloading update; notifying dependent calls.
```

Or

```
[DownloadManager] BITS job {A4AC06DD-D6E6-4420-8720-7407734FDAF2} hit a transient error, updateId = {D053C08A-6250-4C43-A111-56C5198FE142}.200 <NULL>, error = 0x800706D9
```

Or

```
DownloadManager [0]12F4.1FE8::09/29/2017-13:45:08.530 [agent]DO job {C6E2F6DC-5B78-4608-B6F1-0678C23614BD} hit a transient error, updateId = 5537BD35-BB74-40B2-A8C3-B696D3C97CBA.201 <NULL>, error = 0x80D0000A
```

Go to Services.msc and ensure that Windows Firewall Service is enabled. Stopping the service associated with Windows Firewall with Advanced Security is not supported by Microsoft. For more information, see [I need to](#)

[disable Windows Firewall](#) or [Windows Update stuck at 0 percent on Windows 10 or Windows Server 2016](#).

Issues arising from configuration of conflicting policies

Windows Update provides a wide range configuration policies to control the behavior of WU service in a managed environment. While these policies let you configure the settings at a granular level, misconfiguration or setting conflicting policies may lead to unexpected behaviors.

See [How to configure automatic updates by using Group Policy or registry settings](#) for more information.

Updates aren't downloading from the intranet endpoint (WSUS/SCCM)

Windows 10 devices can receive updates from a variety of sources, including Windows Update online, a Windows Server Update Services server, and others. To determine the source of Windows Updates currently being used on a device, follow these steps:

1. Start Windows PowerShell as an administrator
2. Run `$MUSM = New-Object -ComObject "Microsoft.Update.ServiceManager"`.
3. Run `$MUSM.Services`.

Check the output for the Name and OffersWindowsUpdates parameters, which you can interpret according to this table.

OUTPUT	INTERPRETATION
- Name: Microsoft Update -OffersWindowsUpdates: True	- The update source is Microsoft Update, which means that updates for other Microsoft products besides the operating system could also be delivered. - Indicates that the client is configured to receive updates for all Microsoft Products (Office, etc.)
- Name: DCat Flighting Prod - OffersWindowsUpdates: True	- Starting with Windows 10 1709, feature updates are always delivered through the DCAT service. - Indicates that the client is configured to receive feature updates from Windows Update.
- Name: Windows Store (DCat Prod) - OffersWindowsUpdates: False	-The update source is Insider Updates for Store Apps. - Indicates that the client will not receive or is not configured to receive these updates.
- Name: Windows Server Update Service - OffersWindowsUpdates: True	- The source is a Windows Server Updates Services server. - The client is configured to receive updates from WSUS.
- Name: Windows Update - OffersWindowsUpdates: True	- The source is Windows Update. - The client is configured to receive updates from Windows Update Online.

You have a bad setup in the environment

If we look at the GPO being set through registry, the system is configured to use WSUS to download updates:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU
"UseWUServer"=dword:00000001
says use WSUS server. =====> it
```

From the WU logs:

```

2018-08-06 09:33:31:085 480 1118 Agent ** START ** Agent: Finding updates [CallerId = OperationalInsight Id = 49]
2018-08-06 09:33:31:085 480 1118 Agent *****
2018-08-06 09:33:31:085 480 1118 Agent * Include potentially superseded updates
2018-08-06 09:33:31:085 480 1118 Agent * Online = No; Ignore download priority = No
2018-08-06 09:33:31:085 480 1118 Agent * Criteria = "IsHidden = 0 AND DeploymentAction="
2018-08-06 09:33:31:085 480 1118 Agent * ServiceID = {00000000-0000-0000-0000-000000000000} Third party service
2018-08-06 09:33:31:085 480 1118 Agent * Search Scope = {Machine}
2018-08-06 09:33:32:554 480 1118 Agent * Found 83 updates and 83 categories in search; evaluated appl. rules of 517 out of 1473 deployed entities
2018-08-06 09:33:32:554 480 1118 Agent *****
2018-08-06 09:33:32:554 480 1118 Agent ** END ** Agent: Finding updates [CallerId = OperationalInsight Id = 49]

```

In the above log snippet, we see that the Criteria = "IsHidden = 0 AND DeploymentAction=". "" means there is nothing specified from the server. So, the scan happens but there is no direction to download or install to the agent. So it just scans the update and provides the results.

Now if you look at the below logs, the Automatic update runs the scan and finds no update approved for it. So it reports there are 0 updates to install or download. This is due to bad setup or configuration in the environment. The WSUS side should approve the patches for WU so that it fetches the updates and installs it on the specified time according to the policy. Since this scenario doesn't include SCCM, there's no way to install unapproved updates. And that is the problem you are facing. You expect that the scan should be done by the operational insight agent and automatically trigger download and install but that won't happen here.

```

2018-08-06 10:58:45:992 480 5d8 Agent ** START ** Agent: Finding updates [CallerId = AutomaticUpdates Id = 57]
2018-08-06 10:58:45:992 480 5d8 Agent *****
2018-08-06 10:58:45:992 480 5d8 Agent * Online = Yes; Ignore download priority = No
2018-08-06 10:58:45:992 480 5d8 Agent * Criteria = "IsInstalled=0 and DeploymentAction='Installation' or IsPresent=1 and DeploymentAction='Uninstallation' or IsInstalled=1 and DeploymentAction='Installation' and RebootRequired=1 or IsInstalled=0 and DeploymentAction='Uninstallation' and RebootRequired=1"

2018-08-06 10:58:46:617 480 5d8 PT + SyncUpdates round trips: 2
2018-08-06 10:58:47:383 480 5d8 Agent * Found 0 updates and 83 categories in search; evaluated appl. rules of 617 out of 1473 deployed entities
2018-08-06 10:58:47:383 480 5d8 Agent Reporting status event with 0 installable, 83 installed, 0 installed pending, 0 failed and 0 downloaded updates
2018-08-06 10:58:47:383 480 5d8 Agent *****
2018-08-06 10:58:47:383 480 5d8 Agent ** END ** Agent: Finding updates [CallerId = AutomaticUpdates Id = 57]

```

High bandwidth usage on Windows 10 by Windows Update

Users may see that Windows 10 is consuming all the bandwidth in the different offices under the system context. This behavior is by design. Components that may consume bandwidth expand beyond Windows Update components.

The following group policies can help mitigate this:

- Blocking access to Windows Update servers: [Policy Turn off access to all Windows Update features](#) (Set to enabled)
- Driver search: [Policy Specify search order for device driver source locations](#) (Set to "Do not search Windows Update")
- Windows Store automatic update: [Policy Turn off Automatic Download and Install of updates](#) (Set to enabled)

Other components that reach out to the internet:

- Windows Spotlight: [Policy Configure Windows spotlight on lock screen](#) (Set to disabled)
- Consumer experiences: [Policy Turn off Microsoft consumer experiences](#) (Set to enabled)
- Background traffic from Windows apps: [Policy Let Windows apps run in the background](#)

Windows Update common errors and mitigation

6/26/2019 • 3 minutes to read • [Edit Online](#)

Applies to: Windows 10

The following table provides information about common errors you might run into with Windows Update, as well as steps to help you mitigate them.

ERROR CODE	MESSAGE	DESCRIPTION	MITIGATION
0x8024402F	WU_E_PT_ECP_SUCCEEDED_WITH_ERRORS	External cab file processing completed with some errors	One of the reasons we see this issue is due to the design of a software called Lightspeed Rocket for Web filtering. The IP addresses of the computers you want to get updates successfully on, should be added to the exceptions list of Lightspeed
0x80242006	WU_E_UH_INVALIDMETADATA	A handler operation could not be completed because the update contains invalid metadata.	Rename Software Redistribution Folder and attempt to download the updates again: Rename the following folders to *.BAK: - %systemroot%\system32\cactroot2 To do this, type the following commands at a command prompt. Press ENTER after you type each command. - Ren %systemroot%\SoftwareDistribution\DataStore *.bak - Ren %systemroot%\SoftwareDistribution\Download *.bak Ren %systemroot%\system32\cactroot2 *.bak
0x80070BC9	ERROR_FAIL_REBOOT_REQUIRED	The requested operation failed. A system reboot is required to roll back changes made.	Ensure that we do not have any policies that control the start behavior for the Windows Module Installer. This service should not be hardened to any start value and should be managed by the OS.

ERROR CODE	MESSAGE	DESCRIPTION	MITIGATION
0x80200053	BG_E_VALIDATION_FAILED	NA	<p>Ensure that there is no Firewalls that filter downloads. The Firewall filtering may lead to invalid responses being received by the Windows Update Client.</p> <p>If the issue still persists, run the WU reset script.</p>
0x80072EE2	WININET_E_TIMEOUT	The operation timed out	<p>This error message can be caused if the computer isn't connected to Internet. To fix this issue, following these steps: make sure these URLs are not blocked: http://update.microsoft.com https://update.microsoft.com http://download.windowsupdate.com</p> <p>Additionally , you can take a network trace and see what is timing out. <Refer to Firewall Troubleshooting scenario></p>
0x80072EFD 0x80072EFE 0x80D02002	TIME OUT ERRORS	The operation timed out	<p>Make sure there are no firewall rules or proxy to block Microsoft download URLs. Take a network monitor trace to understand better. <Refer to Firewall Troubleshooting scenario></p>
0X8007000D	ERROR_INVALID_DATA	Indicates invalid data downloaded or corruption occurred.	Attempt to re-download the update and initiate installation.
0x8024A10A	USO_E_SERVICE_SHUTTING_DOWN	Indicates that the WU Service is shutting down.	<p>This may happen due to a very long period of time of inactivity, a system hang leading to the service being idle and leading to the shutdown of the service. Ensure that the system remains active and the connections remain established to complete the upgrade.</p>
0x80240020	WU_E_NO_INTERACTIVE_USER	Operation did not complete because there is no logged-on interactive user.	Please login to the system to initiate the installation and allow the system to be rebooted.

ERROR CODE	MESSAGE	DESCRIPTION	MITIGATION
0x80242014	WU_E_UH_POSTREBOOTSTALLPENDING	The post-reboot operation for the update is still in progress.	Some Windows Updates require the system to be restarted. Reboot the system to complete the installation of the Updates.
0x80246017	WU_E_DM_UNAUTHORIZED_LOCAL_USER	The download failed because the local user was denied authorization to download the content.	Ensure that the user attempting to download and install updates has been provided with sufficient privileges to install updates (Local Administrator).
0x8024000B	WU_E_CALL_CANCELLED	Operation was cancelled.	This indicates that the operation was cancelled by the user/service. You may also encounter this error when we are unable to filter the results. Run the Decline Superseded PowerShell script to allow the filtering process to complete.
0x8024000E	WU_E_XML_INVALID	Windows Update Agent found invalid information in the update's XML data.	Certain drivers contain additional metadata information in the update.xml, which could lead Orchestrator to understand it as invalid data. Ensure that you have the latest Windows Update Agent installed on the machine.
0x8024D009	WU_E_SETUP_SKIP_UPDATE	An update to the Windows Update Agent was skipped due to a directive in the wuident.cab file.	You may encounter this error when WSUS is not sending the Self-update to the clients. Review KB920659 for instructions to resolve the issue.
0x80244007	WU_E_PT_SOAPCLIENT_SOAPFAULT	SOAP client failed because there was a SOAP fault for reasons of WU_E_PT_SOAP_* error codes.	This issue occurs because Windows cannot renew the cookies for Windows Update. Review KB2883975 for instructions to resolve the issue.

Windows Update error codes by component

6/14/2019 • 19 minutes to read • [Edit Online](#)

Applies to: Windows 10

This section lists the error codes for Microsoft Windows Update.

Automatic Update Errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80243FFF	WU_E_AUCLIENT_UNEXPECTED	There was a user interface error not covered by another WU_E_AUCLIENT_* error code.
0x8024A000	WU_E_AU_NOSERVICE	Automatic Updates was unable to service incoming requests.
0x8024A002	WU_E_AU_NONLEGACYSERVER	The old version of the Automatic Updates client has stopped because the WSUS server has been upgraded.
0x8024A003	WU_E_AU_LEGACYCLIENTDISABLED	The old version of the Automatic Updates client was disabled.
0x8024A004	WU_E_AU_PAUSED	Automatic Updates was unable to process incoming requests because it was paused.
0x8024A005	WU_E_AU_NO_REGISTERED_SERVICE	No unmanaged service is registered with AU.
0x8024AFFF	WU_E_AU_UNEXPECTED	An Automatic Updates error not covered by another WU_E_AU * code.

Windows Update UI errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80243001	WU_E_INSTALLATION_RESULTS_UNKNO WN_VERSION	The results of download and installation could not be read from the registry due to an unrecognized data format version.
0x80243002	WU_E_INSTALLATION_RESULTS_INVALI D_DATA	The results of download and installation could not be read from the registry due to an invalid data format.
0x80243003	WU_E_INSTALLATION_RESULTS_NOT_F OUND	The results of download and installation are not available; the operation may have failed to start.

ERROR CODE	MESSAGE	DESCRIPTION
0x80243004	WU_E_TRAYICON_FAILURE	A failure occurred when trying to create an icon in the taskbar notification area.
0x80243FFD	WU_E_NON_UI_MODE	Unable to show UI when in non-UI mode; WU client UI modules may not be installed.
0x80243FFE	WU_E_WUCLTUI_UNSUPPORTED_VERSION	Unsupported version of WU client UI exported functions.
0x80243FFF	WU_E_AUCLIENT_UNEXPECTED	There was a user interface error not covered by another WU_E_AUCLIENT_* error code.

Inventory errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80249001	WU_E_INVENTORY_PARSEFAILED	Parsing of the rule file failed.
0x80249002	WU_E_INVENTORY_GET_INVENTORY_TYPE_FAILED	Failed to get the requested inventory type from the server.
0x80249003	WU_E_INVENTORY_RESULT_UPLOAD_FAILED	Failed to upload inventory result to the server.
0x80249004	WU_E_INVENTORY_UNEXPECTED	There was an inventory error not covered by another error code.
0x80249005	WU_E_INVENTORY_WMI_ERROR	A WMI error occurred when enumerating the instances for a particular class.

Expression evaluator errors

ERROR CODE	MESSAGE	DESCRIPTION
0x8024E001	WU_E_EE_UNKNOWN_EXPRESSION	An expression evaluator operation could not be completed because an expression was unrecognized.
0x8024E002	WU_E_EE_INVALID_EXPRESSION	An expression evaluator operation could not be completed because an expression was invalid.
0x8024E003	WU_E_EE_MISSING_METADATA	An expression evaluator operation could not be completed because an expression contains an incorrect number of metadata nodes.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024E004	WU_E_EE_INVALID_VERSION	An expression evaluator operation could not be completed because the version of the serialized expression data is invalid.
0x8024E005	WU_E_EE_NOT_INITIALIZED	The expression evaluator could not be initialized.
0x8024E006	WU_E_EE_INVALID_ATTRIBUTEDATA	An expression evaluator operation could not be completed because there was an invalid attribute.
0x8024E007	WU_E_EE_CLUSTER_ERROR	An expression evaluator operation could not be completed because the cluster state of the computer could not be determined.
0x8024EFFF	WU_E_EE_UNEXPECTED	There was an expression evaluator error not covered by another WU_E_EE_* error code.

Reporter errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80247001	WU_E_OL_INVALID_SCANFILE	An operation could not be completed because the scan package was invalid.
0x80247002	WU_E_OL_NEWCLIENT_REQUIRED	An operation could not be completed because the scan package requires a greater version of the Windows Update Agent.
0x80247FFF	WU_E_OL_UNEXPECTED	Search using the scan package failed.
0x8024F001	WU_E_REPORTER_EVENTCACHECORRUPT	The event cache file was defective.
0x8024F002	WU_E_REPORTER_EVENTNAMESPACEPARSEFAILED	The XML in the event namespace descriptor could not be parsed.
0x8024F003	WU_E_INVALID_EVENT	The XML in the event namespace descriptor could not be parsed.
0x8024F004	WU_E_SERVER_BUSY	The server rejected an event because the server was too busy.
0x8024FFFF	WU_E_REPORTER_UNEXPECTED	There was a reporter error not covered by another error code.

Redirector errors

The components that download the Wuredir.cab file and then parse the Wuredir.cab file generate the following

errors.

ERROR CODE	MESSAGE	DESCRIPTION
0x80245001	WU_E_REDIRECTOR_LOAD_XML	The redirector XML document could not be loaded into the DOM class.
0x80245002	WU_E_REDIRECTOR_S_FALSE	The redirector XML document is missing some required information.
0x80245003	WU_E_REDIRECTOR_ID_SMALLER	The redirectorld in the downloaded redirector cab is less than in the cached cab.
0x80245FFF	WU_E_REDIRECTOR_UNEXPECTED	The redirector failed for reasons not covered by another WU_E_REDIRECTOR_* error code.

Protocol Talker errors

The following errors map to SOAPCLIENT_ERRORS through the Atlsoap.h file. These errors are obtained when the CClientWebService object calls the GetClientError() method.

ERROR CODE	MESSAGE	DESCRIPTION
0x80244000	WU_E_PT_SOAPCLIENT_BASE	WU_E_PT_SOAPCLIENT_* error codes map to the SOAPCLIENT_ERROR enum of the ATL Server Library.
0x80244001	WU_E_PT_SOAPCLIENT_INITIALIZE	Same as SOAPCLIENT_INITIALIZE_ERROR - initialization of the SOAP client failed possibly because of an MSXML installation failure.
0x80244002	WU_E_PT_SOAPCLIENT_OUTOFMEMORY	Same as SOAPCLIENT_OUTOFMEMORY - SOAP client failed because it ran out of memory.
0x80244003	WU_E_PT_SOAPCLIENT_GENERATE	Same as SOAPCLIENT_GENERATE_ERROR - SOAP client failed to generate the request.
0x80244004	WU_E_PT_SOAPCLIENT_CONNECT	Same as SOAPCLIENT_CONNECT_ERROR - SOAP client failed to connect to the server.
0x80244005	WU_E_PT_SOAPCLIENT_SEND	Same as SOAPCLIENT_SEND_ERROR - SOAP client failed to send a message for reasons of WU_E_WINHTTP_* error codes.
0x80244006	WU_E_PT_SOAPCLIENT_SERVER	Same as SOAPCLIENT_SERVER_ERROR - SOAP client failed because there was a server error.

ERROR CODE	MESSAGE	DESCRIPTION
0x80244007	WU_E_PT_SOAPCLIENT_SOAPFAULT	Same as SOAPCLIENT_SOAPFAULT - SOAP client failed because there was a SOAP fault for reasons of WU_E_PT_SOAP_* error codes.
0x80244008	WU_E_PT_SOAPCLIENT_PARSEFAULT	Same as SOAPCLIENT_PARSEFAULT_ERROR - SOAP client failed to parse a SOAP fault.
0x80244009	WU_E_PT_SOAPCLIENT_READ	Same as SOAPCLIENT_READ_ERROR - SOAP client failed while reading the response from the server.
0x8024400A	WU_E_PT_SOAPCLIENT_PARSE	Same as SOAPCLIENT_PARSE_ERROR - SOAP client failed to parse the response from the server.

Other Protocol Talker errors

The following errors map to SOAP_ERROR_CODEs from the Atlsoap.h file. These errors are obtained from the m_fault.m_soapErrCode member of the CClientWebService object when GetClientError() returns SOAPCLIENT_SOAPFAULT.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024400B	WU_E_PT_SOAP_VERSION	Same as SOAP_E_VERSION_MISMATCH - SOAP client found an unrecognizable namespace for the SOAP envelope.
0x8024400C	WU_E_PT_SOAP_MUST_UNDERSTAND	Same as SOAP_E_MUST_UNDERSTAND - SOAP client was unable to understand a header.
0x8024400D	WU_E_PT_SOAP_CLIENT	Same as SOAP_E_CLIENT - SOAP client found the message was malformed; fix before resending.
0x8024400E	WU_E_PT_SOAP_SERVER	Same as SOAP_E_SERVER - The SOAP message could not be processed due to a server error; resend later.
0x8024400F	WU_E_PT_WMI_ERROR	There was an unspecified Windows Management Instrumentation (WMI) error.
0x80244010	WU_E_PT_EXCEEDED_MAX_SERVER_TRIPS	The number of round trips to the server exceeded the maximum limit.
0x80244011	WU_E_PT_SUS_SERVER_NOT_SET	WU Server policy value is missing in the registry.
0x80244012	WU_E_PT_DOUBLE_INITIALIZATION	Initialization failed because the object was already initialized.

ERROR CODE	MESSAGE	DESCRIPTION
0x80244013	WU_E_PT_INVALID_COMPUTER_NAME	The computer name could not be determined.
0x80244015	WU_E_PT_REFRESH_CACHE_REQUIRED	The reply from the server indicates that the server was changed or the cookie was invalid; refresh the state of the internal cache and retry.
0x80244016	WU_E_PT_HTTP_STATUS_BAD_REQUEST	Same as HTTP status 400 - the server could not process the request due to invalid syntax.
0x80244017	WU_E_PT_HTTP_STATUS_DENIED	Same as HTTP status 401 - the requested resource requires user authentication.
0x80244018	WU_E_PT_HTTP_STATUS_FORBIDDEN	Same as HTTP status 403 - server understood the request but declined to fulfill it.
0x80244019	WU_E_PT_HTTP_STATUS_NOT_FOUND	Same as HTTP status 404 - the server cannot find the requested URI (Uniform Resource Identifier).
0x8024401A	WU_E_PT_HTTP_STATUS_BAD_METHOD	Same as HTTP status 405 - the HTTP method is not allowed.
0x8024401B	WU_E_PT_HTTP_STATUS_PROXY_AUTH_REQ	Same as HTTP status 407 - proxy authentication is required.
0x8024401C	WU_E_PT_HTTP_STATUS_REQUEST_TIMEOUT	Same as HTTP status 408 - the server timed out waiting for the request.
0x8024401D	WU_E_PT_HTTP_STATUS_CONFLICT	Same as HTTP status 409 - the request was not completed due to a conflict with the current state of the resource.
0x8024401E	WU_E_PT_HTTP_STATUS_GONE	Same as HTTP status 410 - requested resource is no longer available at the server.
0x8024401F	WU_E_PT_HTTP_STATUS_SERVER_ERROR	Same as HTTP status 500 - an error internal to the server prevented fulfilling the request.
0x80244020	WU_E_PT_HTTP_STATUS_NOT_SUPPORTED	Same as HTTP status 500 - server does not support the functionality required to fulfill the request.
0x80244021	WU_E_PT_HTTP_STATUS_BAD_GATEWAY	Same as HTTP status 502 - the server while acting as a gateway or a proxy received an invalid response from the upstream server it accessed in attempting to fulfill the request.

ERROR CODE	MESSAGE	DESCRIPTION
0x80244022	WU_E_PT_HTTP_STATUS_SERVICE_UNAVAILABLE	Same as HTTP status 503 - the service is temporarily overloaded.
0x80244023	WU_E_PT_HTTP_STATUS_GATEWAY_TIMEOUT	Same as HTTP status 503 - the request was timed out waiting for a gateway.
0x80244024	WU_E_PT_HTTP_STATUS_VERSION_NOT_SUPPORTED	Same as HTTP status 505 - the server does not support the HTTP protocol version used for the request.
0x80244025	WU_E_PT_FILE_LOCATIONS_CHANGED	Operation failed due to a changed file location; refresh internal state and resend.
0x80244026	WU_E_PT_REGISTRATION_NOT_SUPPORTED	Operation failed because Windows Update Agent does not support registration with a non-WSUS server.
0x80244027	WU_E_PT_NO_AUTH_PLUGINS_REQUESTED	The server returned an empty authentication information list.
0x80244028	WU_E_PT_NO_AUTH_COOKIES_CREATED	Windows Update Agent was unable to create any valid authentication cookies.
0x80244029	WU_E_PT_INVALID_CONFIG_PROP	A configuration property value was wrong.
0x8024402A	WU_E_PT_CONFIG_PROP_MISSING	A configuration property value was missing.
0x8024402B	WU_E_PT_HTTP_STATUS_NOT_MAPPED	The HTTP request could not be completed and the reason did not correspond to any of the WU_E_PT_HTTP_* error codes.
0x8024402C	WU_E_PT_WINHTTP_NAME_NOT_RESOLVED	Same as ERROR_WINHTTP_NAME_NOT_RESOLVED - the proxy server or target server name cannot be resolved.
0x8024402F	WU_E_PT_ECP_SUCCEEDED_WITH_ERRORS	External cab file processing completed with some errors.
0x80244030	WU_E_PT_ECP_INIT_FAILED	The external cab processor initialization did not complete.
0x80244031	WU_E_PT_ECP_INVALID_FILE_FORMAT	The format of a metadata file was invalid.
0x80244032	WU_E_PT_ECP_INVALID_METADATA	External cab processor found invalid metadata.
0x80244033	WU_E_PT_ECP_FAILURE_TO_EXTRACT_DIGEST	The file digest could not be extracted from an external cab file.

ERROR CODE	MESSAGE	DESCRIPTION
0x80244034	WU_E_PT_ECP_FAILURE_TO_DECOMPRESS_CAB_FILE	An external cab file could not be decompressed.
0x80244035	WU_E_PT_ECP_FILE_LOCATION_ERROR	External cab processor was unable to get file locations.
0x80244FFF	WU_E_PT_UNEXPECTED	A communication error not covered by another WU_E_PT_* error code.
0x8024502D	WU_E_PT_SAME_REDIR_ID	Windows Update Agent failed to download a redirector cabinet file with a new redirectorId value from the server during the recovery.
0x8024502E	WU_E_PT_NO_MANAGED_RECOVER	A redirector recovery action did not complete because the server is managed.

Download Manager errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80246001	WU_E_DM_URLNOTAVAILABLE	A download manager operation could not be completed because the requested file does not have a URL.
0x80246002	WU_E_DM_INCORRECTFILEHASH	A download manager operation could not be completed because the file digest was not recognized.
0x80246003	WU_E_DM_UNKNOWNALGORITHM	A download manager operation could not be completed because the file metadata requested an unrecognized hash algorithm.
0x80246004	WU_E_DM_NEEDDOWNLOADREQUEST	An operation could not be completed because a download request is required from the download handler.
0x80246005	WU_E_DM_NONETWORK	A download manager operation could not be completed because the network connection was unavailable.
0x80246006	WU_E_DM_WRONGBITSVERSION	A download manager operation could not be completed because the version of Background Intelligent Transfer Service (BITS) is incompatible.
0x80246007	WU_E_DM_NOTDOWNLOADED	The update has not been downloaded.
0x80246008	WU_E_DM_FAILTOCONNECTTOBITS	A download manager operation failed because the download manager was unable to connect the Background Intelligent Transfer Service (BITS).

ERROR CODE	MESSAGE	DESCRIPTION
0x80246009	WU_E_DM_BITSTRANSFERERROR	A download manager operation failed because there was an unspecified Background Intelligent Transfer Service (BITS) transfer error.
0x8024600A	WU_E_DM_DOWNLOADLOCATIONCHANGED	A download must be restarted because the location of the source of the download has changed.
0x8024600B	WU_E_DM_CONTENTCHANGED	A download must be restarted because the update content changed in a new revision.
0x80246FFF	WU_E_DM_UNEXPECTED	There was a download manager error not covered by another WU_E_DM_* error code.

Update Handler errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80242000	WU_E_UH_REMOTEUNAVAILABLE	9A request for a remote update handler could not be completed because no remote process is available.
0x80242001	WU_E_UH_LOCALONLY	A request for a remote update handler could not be completed because the handler is local only.
0x80242002	WU_E_UH_UNKNOWNHANDLER	A request for an update handler could not be completed because the handler could not be recognized.
0x80242003	WU_E_UH_REMOTEALREADYACTIVE	A remote update handler could not be created because one already exists.
0x80242004	WU_E_UH_DOESNOTSUPPORTACTION	A request for the handler to install (uninstall) an update could not be completed because the update does not support install (uninstall).
0x80242005	WU_E_UH_WRONGHANDLER	An operation did not complete because the wrong handler was specified.
0x80242006	WU_E_UH_INVALIDMETADATA	A handler operation could not be completed because the update contains invalid metadata.
0x80242007	WU_E_UH_INSTALLERHUNG	An operation could not be completed because the installer exceeded the time limit.
0x80242008	WU_E_UH_OPERATIONCANCELLED	An operation being done by the update handler was cancelled.

ERROR CODE	MESSAGE	DESCRIPTION
0x80242009	WU_E_UH_BADHANDLERXML	An operation could not be completed because the handler-specific metadata is invalid.
0x8024200A	WU_E_UH_CANREQUIREINPUT	A request to the handler to install an update could not be completed because the update requires user input.
0x8024200B	WU_E_UH_INSTALLERFAILURE	The installer failed to install (uninstall) one or more updates.
0x8024200C	WU_E_UH_FALLBACKTOSELFCONTAINED	The update handler should download self-contained content rather than delta-compressed content for the update.
0x8024200D	WU_E_UH_NEEDANOTHERDOWNLOAD	The update handler did not install the update because it needs to be downloaded again.
0x8024200E	WU_E_UH_NOTIFYFAILURE	The update handler failed to send notification of the status of the install (uninstall) operation.
0x8024200F	WU_E_UH_INCONSISTENT_FILE_NAMES	The file names contained in the update metadata and in the update package are inconsistent.
0x80242010	WU_E_UH_FALLBACKERROR	The update handler failed to fall back to the self-contained content.
0x80242011	WU_E_UH_TOOMANYDOWNLOADREQUESTS	The update handler has exceeded the maximum number of download requests.
0x80242012	WU_E_UH_UNEXPECTEDCBSRESPONSE	The update handler has received an unexpected response from CBS.
0x80242013	WU_E_UH_BADCBSPACKAGEID	The update metadata contains an invalid CBS package identifier.
0x80242014	WU_E_UH_POSTREBOOTSTILLPENDING	The post-reboot operation for the update is still in progress.
0x80242015	WU_E_UH_POSTREBOOTRESULTUNKNOWN	The result of the post-reboot operation for the update could not be determined.
0x80242016	WU_E_UH_POSTREBOOTUNEXPECTEDSTATE	The state of the update after its post-reboot operation has completed is unexpected.
0x80242017	WU_E_UH_NEW_SERVICING_STACK_REQUIRED	The OS servicing stack must be updated before this update is downloaded or installed.

ERROR CODE	MESSAGE	DESCRIPTION
0x80242FFF	WU_E_UH_UNEXPECTED	An update handler error not covered by another WU_E_UH_* code.

Data Store errors

ERROR CODE	MESSAGE	DESCRIPTION
0x80248000	WU_E_DS_SHUTDOWN	An operation failed because Windows Update Agent is shutting down.
0x80248001	WU_E_DS_INUSE	An operation failed because the data store was in use.
0x80248002	WU_E_DS_INVALID	The current and expected states of the data store do not match.
0x80248003	WU_E_DS_TABLEMISSING	The data store is missing a table.
0x80248004	WU_E_DS_TABLEINCORRECT	The data store contains a table with unexpected columns.
0x80248005	WU_E_DS_INVALIDTABLENAME	A table could not be opened because the table is not in the data store.
0x80248006	WU_E_DS_BADVERSION	The current and expected versions of the data store do not match.
0x80248007	WU_E_DS_NODATA	The information requested is not in the data store.
0x80248008	WU_E_DS_MISSINGDATA	The data store is missing required information or has a NULL in a table column that requires a non-null value.
0x80248009	WU_E_DS_MISSINGREF	The data store is missing required information or has a reference to missing license terms file localized property or linked row.
0x8024800A	WU_E_DS_UNKNOWNHANDLER	The update was not processed because its update handler could not be recognized.
0x8024800B	WU_E_DS_CANTDELETE	The update was not deleted because it is still referenced by one or more services.
0x8024800C	WU_E_DS_LOCKTIMEOUTEXPIRED	The data store section could not be locked within the allotted time.
0x8024800D	WU_E_DS_NOCATEGORIES	The category was not added because it contains no parent categories and is not a top-level category itself.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024800E	WU_E_DS_ROWEXISTS	The row was not added because an existing row has the same primary key.
0x8024800F	WU_E_DS_STOREFILELOCKED	The data store could not be initialized because it was locked by another process.
0x80248010	WU_E_DS_CANNOTREGISTER	The data store is not allowed to be registered with COM in the current process.
0x80248011	WU_E_DS_UNABLETOSTART	Could not create a data store object in another process.
0x80248013	WU_E_DS_DUPLICATEUPDATEID	The server sent the same update to the client with two different revision IDs.
0x80248014	WU_E_DS_UNKNOWNSERVICE	An operation did not complete because the service is not in the data store.
0x80248015	WU_E_DS_SERVICEEXPIRED	An operation did not complete because the registration of the service has expired.
0x80248016	WU_E_DS_DECLINENOTALLOWED	A request to hide an update was declined because it is a mandatory update or because it was deployed with a deadline.
0x80248017	WU_E_DS_TABLESESSIONMISMATCH	A table was not closed because it is not associated with the session.
0x80248018	WU_E_DS_SESSIONLOCKMISMATCH	A table was not closed because it is not associated with the session.
0x80248019	WU_E_DS_NEEDWINDOWSSERVICE	A request to remove the Windows Update service or to unregister it with Automatic Updates was declined because it is a built-in service and/or Automatic Updates cannot fall back to another service.
0x8024801A	WU_E_DS_INVALIDOPERATION	A request was declined because the operation is not allowed.
0x8024801B	WU_E_DS_SCHEMAMISMATCH	The schema of the current data store and the schema of a table in a backup XML document do not match.
0x8024801C	WU_E_DS_RESETREQUIRED	The data store requires a session reset; release the session and retry with a new session.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024801D	WU_E_DS_IMPERSONATED	A data store operation did not complete because it was requested with an impersonated identity.
0x80248FFF	WU_E_DS_UNEXPECTED	A data store error not covered by another WU_E_DS_* code.

Driver Util errors

The PnP enumerated device is removed from the System Spec because one of the hardware IDs or the compatible IDs matches an installed printer driver. This is not a fatal error, and the device is merely skipped.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024C001	WU_E_DRV_PRUNED	A driver was skipped.
0x8024C002	WU_E_DRV_NOPROP_OR_LEGACY	A property for the driver could not be found. It may not conform with required specifications.
0x8024C003	WU_E_DRV_REG_MISMATCH	The registry type read for the driver does not match the expected type.
0x8024C004	WU_E_DRV_NO_METADATA	The driver update is missing metadata.
0x8024C005	WU_E_DRV_MISSING_ATTRIBUTE	The driver update is missing a required attribute.
0x8024C006	WU_E_DRV_SYNC_FAILED	Driver synchronization failed.
0x8024C007	WU_E_DRV_NO_PRINTER_CONTENT	Information required for the synchronization of applicable printers is missing.
0x8024CFFF	WU_E_DRV_UNEXPECTED	A driver error not covered by another WU_E_DRV_* code.

Windows Update error codes

ERROR CODE	MESSAGE	DESCRIPTION
0x80240001	WU_E_NO_SERVICE	Windows Update Agent was unable to provide the service.
0x80240002	WU_E_MAX_CAPACITY_REACHED	The maximum capacity of the service was exceeded.
0x80240003	WU_E_UNKNOWN_ID	An ID cannot be found.
0x80240004	WU_E_NOT_INITIALIZED	The object could not be initialized.

ERROR CODE	MESSAGE	DESCRIPTION
0x80240005	WU_E_RANGEOVERLAP	The update handler requested a byte range overlapping a previously requested range.
0x80240006	WU_E_TOOMANYRANGES	The requested number of byte ranges exceeds the maximum number ($2^{31} - 1$).
0x80240007	WU_E_INVALIDINDEX	The index to a collection was invalid.
0x80240008	WU_E_ITEMNOTFOUND	The key for the item queried could not be found.
0x80240009	WU_E_OPERATIONINPROGRESS	Another conflicting operation was in progress. Some operations such as installation cannot be performed twice simultaneously.
0x8024000A	WU_E_COULDNOTCANCEL	Cancellation of the operation was not allowed.
0x8024000B	WU_E_CALL_CANCELLED	Operation was cancelled.
0x8024000C	WU_E_NOOP	No operation was required.
0x8024000D	WU_E_XML_MISSINGDATA	Windows Update Agent could not find required information in the update's XML data.
0x8024000E	WU_E_XML_INVALID	Windows Update Agent found invalid information in the update's XML data.
0x8024000F	WU_E_CYCLE_DETECTED	Circular update relationships were detected in the metadata.
0x80240010	WU_E_TOO_DEEP_RELATION	Update relationships too deep to evaluate were evaluated.
0x80240011	WU_E_INVALID_RELATIONSHIP	An invalid update relationship was detected.
0x80240012	WU_E_REG_VALUE_INVALID	An invalid registry value was read.
0x80240013	WU_E_DUPLICATE_ITEM	Operation tried to add a duplicate item to a list.
0x80240016	WU_E_INSTALL_NOT_ALLOWED	Operation tried to install while another installation was in progress or the system was pending a mandatory restart.
0x80240017	WU_E_NOT_APPLICABLE	Operation was not performed because there are no applicable updates.

ERROR CODE	MESSAGE	DESCRIPTION
0x80240018	WU_E_NO_USERTOKEN	Operation failed because a required user token is missing.
0x80240019	WU_E_EXCLUSIVE_INSTALL_CONFLICT	An exclusive update cannot be installed with other updates at the same time.
0x8024001A	WU_E_POLICY_NOT_SET	A policy value was not set.
0x8024001B	WU_E_SELFUPDATE_IN_PROGRESS	The operation could not be performed because the Windows Update Agent is self-updating.
0x8024001D	WU_E_INVALID_UPDATE	An update contains invalid metadata.
0x8024001E	WU_E_SERVICE_STOP	Operation did not complete because the service or system was being shut down.
0x8024001F	WU_E_NO_CONNECTION	Operation did not complete because the network connection was unavailable.
0x80240020	WU_E_NO_INTERACTIVE_USER	Operation did not complete because there is no logged-on interactive user.
0x80240021	WU_E_TIME_OUT	Operation did not complete because it timed out.
0x80240022	WU_E_ALL_UPDATES_FAILED	Operation failed for all the updates.
0x80240023	WU_E_EULAS_DECLINED	The license terms for all updates were declined.
0x80240024	WU_E_NO_UPDATE	There are no updates.
0x80240025	WU_E_USER_ACCESS_DISABLED	Group Policy settings prevented access to Windows Update.
0x80240026	WU_E_INVALID_UPDATE_TYPE	The type of update is invalid.
0x80240027	WU_E_URL_TOO_LONG	The URL exceeded the maximum length.
0x80240028	WU_E_UNINSTALL_NOT_ALLOWED	The update could not be uninstalled because the request did not originate from a WSUS server.
0x80240029	WU_E_INVALID_PRODUCT_LICENSE	Search may have missed some updates before there is an unlicensed application on the system.
0x8024002A	WU_E_MISSING_HANDLER	A component required to detect applicable updates was missing.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024002B	WU_E_LEGACYSERVER	An operation did not complete because it requires a newer version of server.
0x8024002C	WU_E_BIN_SOURCE_ABSENT	A delta-compressed update could not be installed because it required the source.
0x8024002D	WU_E_SOURCE_ABSENT	A full-file update could not be installed because it required the source.
0x8024002E	WU_E_WU_DISABLED	Access to an unmanaged server is not allowed.
0x8024002F	WU_E_CALL_CANCELLED_BY_POLICY	Operation did not complete because the DisableWindowsUpdateAccess policy was set.
0x80240030	WU_E_INVALID_PROXY_SERVER	The format of the proxy list was invalid.
0x80240031	WU_E_INVALID_FILE	The file is in the wrong format.
0x80240032	WU_E_INVALID_CRITERIA	The search criteria string was invalid.
0x80240033	WU_E_EULA_UNAVAILABLE	License terms could not be downloaded.
0x80240034	WU_E_DOWNLOAD_FAILED	Update failed to download.
0x80240035	WU_E_UPDATE_NOT_PROCESSED	The update was not processed.
0x80240036	WU_E_INVALID_OPERATION	The object's current state did not allow the operation.
0x80240037	WU_E_NOT_SUPPORTED	The functionality for the operation is not supported.
0x80240038	WU_E_WINHTTP_INVALID_FILE	The downloaded file has an unexpected content type.
0x80240039	WU_E_TOO_MANY_RESYNC	Agent is asked by server to resync too many times.
0x80240040	WU_E_NO_SERVER_CORE_SUPPORT	WUA API method does not run on Server Core installation.
0x80240041	WU_E_SYSPREP_IN_PROGRESS	Service is not available while sysprep is running.
0x80240042	WU_E_UNKNOWN_SERVICE	The update service is no longer registered with AU.
0x80240043	WU_E_NO_UI_SUPPORT	There is no support for WUA UI.

ERROR CODE	MESSAGE	DESCRIPTION
0x80240FFF	WU_E_UNEXPECTED	An operation failed due to reasons not covered by another error code.

Windows Update success codes

ERROR CODE	MESSAGE	DESCRIPTION
0x00240001	WU_S_SERVICE_STOP	Windows Update Agent was stopped successfully.
0x00240002	WU_S_SELFUPDATE	Windows Update Agent updated itself.
0x00240003	WU_S_UPDATE_ERROR	Operation completed successfully but there were errors applying the updates.
0x00240004	WU_S_MARKED_FOR_DISCONNECT	A callback was marked to be disconnected later because the request to disconnect the operation came while a callback was executing.
0x00240005	WU_S_REBOOT_REQUIRED	The system must be restarted to complete installation of the update.
0x00240006	WU_S_ALREADY_INSTALLED	The update to be installed is already installed on the system.
0x00240007	WU_S_ALREADY_UNINSTALLED	The update to be removed is not installed on the system.
0x00240008	WU_S_ALREADY_DOWNLOADED	The update to be downloaded has already been downloaded.

Windows Installer minor errors

The following errors are used to indicate that part of a search fails because of Windows Installer problems. Another part of the search may successfully return updates. All Windows Installer minor codes must share the same error code range so that the caller can tell that they are related to Windows Installer.

ERROR CODE	MESSAGE	DESCRIPTION
0x80241001	WU_E_MSI_WRONG_VERSION	Search may have missed some updates because the Windows Installer is less than version 3.1.
0x80241002	WU_E_MSI_NOT_CONFIGURED	Search may have missed some updates because the Windows Installer is not configured.
0x80241003	WU_E_MSP_DISABLED	Search may have missed some updates because policy has disabled Windows Installer patching.

ERROR CODE	MESSAGE	DESCRIPTION
0x80241004	WU_E_MSI_WRONG_APP_CONTEXT	An update could not be applied because the application is installed per-user.
0x80241FFF	WU_E_MSP_UNEXPECTED	Search may have missed some updates because there was a failure of the Windows Installer.

Windows Update Agent update and setup errors

ERROR CODE	MESSAGE	DESCRIPTION
0x8024D001	WU_E_SETUP_INVALID_INFDATA	Windows Update Agent could not be updated because an INF file contains invalid information.
0x8024D002	WU_E_SETUP_INVALID_IDENTDATA	Windows Update Agent could not be updated because the wuident.cab file contains invalid information.
0x8024D003	WU_E_SETUP_ALREADY_INITIALIZED	Windows Update Agent could not be updated because of an internal error that caused setup initialization to be performed twice.
0x8024D004	WU_E_SETUP_NOT_INITIALIZED	Windows Update Agent could not be updated because setup initialization never completed successfully.
0x8024D005	WU_E_SETUP_SOURCE_VERSION_MISMATCH	Windows Update Agent could not be updated because the versions specified in the INF do not match the actual source file versions.
0x8024D006	WU_E_SETUP_TARGET_VERSION_GREATER	Windows Update Agent could not be updated because a WUA file on the target system is newer than the corresponding source file.
0x8024D007	WU_E_SETUP_REGISTRATION_FAILED	Windows Update Agent could not be updated because regsvr32.exe returned an error.
0x8024D009	WU_E_SETUP_SKIP_UPDATE	An update to the Windows Update Agent was skipped due to a directive in the wuident.cab file.
0x8024D00A	WU_E_SETUP_UNSUPPORTED_CONFIGURATION	Windows Update Agent could not be updated because the current system configuration is not supported.
0x8024D00B	WU_E_SETUP_BLOCKED_CONFIGURATION	Windows Update Agent could not be updated because the system is configured to block the update.

ERROR CODE	MESSAGE	DESCRIPTION
0x8024D00C	WU_E_SETUP_REBOOT_TO_FIX	Windows Update Agent could not be updated because a restart of the system is required.
0x8024D00D	WU_E_SETUP_ALREADYRUNNING	Windows Update Agent setup is already running.
0x8024D00E	WU_E_SETUP_REBOOTREQUIRED	Windows Update Agent setup package requires a reboot to complete installation.
0x8024D00F	WU_E_SETUP_HANDLER_EXEC_FAILURE	Windows Update Agent could not be updated because the setup handler failed during execution.
0x8024D010	WU_E_SETUP_INVALID_REGISTRY_DATA	Windows Update Agent could not be updated because the registry contains invalid information.
0x8024D013	WU_E_SETUP_WRONG_SERVER_VERSION	Windows Update Agent could not be updated because the server does not contain update information for this version.
0x8024DFFF	WU_E_SETUP_UNEXPECTED	Windows Update Agent could not be updated because of an error not covered by another WU_E_SETUP_* error code.

Windows Update - additional resources

6/14/2019 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 10

The following resources provide additional information about using Windows Update.

WSUS Troubleshooting

[Troubleshooting issues with WSUS client agents](#)

[How to troubleshoot WSUS](#)

[Error 80244007 when WSUS client scans for updates](#)

[Updates may not be installed with Fast Startup in Windows 10](#)

How do I reset Windows Update components?

[This script](#) will completely reset the Windows Update client settings. It has been tested on Windows 7, 8, 10, and Windows Server 2012 R2. It will configure the services and registry keys related to Windows Update for default settings. It will also clean up files related to Windows Update, in addition to BITS related data.

[This script](#) allow reset the Windows Update Agent resolving issues with Windows Update.

Reset Windows Update components manually

1. Open a Windows command prompt. To open a command prompt, click **Start > Run**. Copy and paste (or type) the following command and then press ENTER:

```
cmd
```

2. Stop the BITS service and the Windows Update service. To do this, type the following commands at a command prompt. Press ENTER after you type each command.

```
net stop bits  
net stop wuauerv
```

3. Delete the qmgr*.dat files. To do this, type the following command at a command prompt, and then press ENTER:

```
Del "%ALLUSERSPROFILE%\Application Data\Microsoft\Network\Downloader\qmgr*.dat"
```

4. If this is your first attempt at resolving your Windows Update issues by using the steps in this article, go to step 5 without carrying out the steps in step 4. The steps in step 4 should only be performed at this point in the troubleshooting if you cannot resolve your Windows Update issues after following all steps but step 4. The steps in step 4 are also performed by the "Aggressive" mode of the Fix it Solution above.
 - a. Rename the following folders to *.BAK:

- %systemroot%\SoftwareDistribution\DataStore

- %systemroot%\SoftwareDistribution\Download
- %systemroot%\system32\catroot2

To do this, type the following commands at a command prompt. Press ENTER after you type each command.

- Ren %systemroot%\SoftwareDistribution\DataStore *.bak
- Ren %systemroot%\SoftwareDistribution\Download *.bak
- Ren %systemroot%\system32\catroot2 *.bak

b. Reset the BITS service and the Windows Update service to the default security descriptor. To do this, type the following commands at a command prompt. Press ENTER after you type each command.

- sc.exe sdset bits D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;AU)
(A;;CCLCSWRPWPDTLOCRRC;;;PU)
- sc.exe sdset wuau serv D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;AU)
(A;;CCLCSWRPWPDTLOCRRC;;;PU)

5. Type the following command at a command prompt, and then press ENTER:

```
cd /d %windir%\system32
```

6. Reregister the BITS files and the Windows Update files. To do this, type the following commands at a command prompt. Press ENTER after you type each command.

- regsvr32.exe atl.dll
- regsvr32.exe urlmon.dll
- regsvr32.exe mshtml.dll
- regsvr32.exe shdocvw.dll
- regsvr32.exe browseui.dll
- regsvr32.exe jscript.dll
- regsvr32.exe vbscript.dll
- regsvr32.exe scrrun.dll
- regsvr32.exe msxml.dll
- regsvr32.exe msxml3.dll
- regsvr32.exe msxml6.dll
- regsvr32.exe actxprxy.dll
- regsvr32.exe softpub.dll
- regsvr32.exe wintrust.dll
- regsvr32.exe dssenh.dll
- regsvr32.exe rsaenh.dll
- regsvr32.exe gpkcsp.dll
- regsvr32.exe sccbase.dll
- regsvr32.exe slbcsp.dll
- regsvr32.exe cryptdlg.dll
- regsvr32.exe oleaut32.dll
- regsvr32.exe ole32.dll
- regsvr32.exe shell32.dll
- regsvr32.exe initpki.dll
- regsvr32.exe wuapi.dll

- regsvr32.exe wuaueng.dll
- regsvr32.exe wuaueng1.dll
- regsvr32.exe wucltui.dll
- regsvr32.exe wups.dll
- regsvr32.exe wups2.dll
- regsvr32.exe wuweb.dll
- regsvr32.exe qmgr.dll
- regsvr32.exe qmgrprxy.dll
- regsvr32.exe wucltux.dll
- regsvr32.exe muweb.dll
- regsvr32.exe wuwebv.dll

7. Reset Winsock. To do this, type the following command at a command prompt, and then press ENTER:

```
netsh winsock reset
```

8. If you are running Windows XP or Windows Server 2003, you have to set the proxy settings. To do this, type the following command at a command prompt, and then press ENTER:

```
proxycfg.exe -d
```

9. Restart the BITS service and the Windows Update service. To do this, type the following commands at a command prompt. Press ENTER after you type each command.

```
net start bits  
  
net start wuauerv
```

10. If you are running Windows Vista or Windows Server 2008, clear the BITS queue. To do this, type the following command at a command prompt, and then press ENTER:

```
bitsadmin.exe /reset /allusers
```

Optimize Windows 10 update delivery

6/14/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

When considering your content distribution strategy for Windows 10, think about enabling a form of peer-to-peer content sharing to reduce bandwidth issues during updates. Windows 10 offers two peer-to-peer options for update content distribution: Delivery Optimization and BranchCache. These technologies can be used with several of the servicing tools for Windows 10.

Two methods of peer-to-peer content distribution are available in Windows 10.

- **Delivery Optimization** is a new peer-to-peer distribution method in Windows 10. Windows 10 clients can source content from other devices on their local network that have already downloaded the updates or from peers over the internet. Using the settings available for Delivery Optimization, clients can be configured into groups, allowing organizations to identify devices that are possibly the best candidates to fulfill peer-to-peer requests.

Windows Update, Windows Update for Business, and Windows Server Update Services (WSUS) can use Delivery Optimization. Delivery Optimization can significantly reduce the amount of network traffic to external Windows Update sources as well as the time it takes for clients to retrieve the updates.

- **BranchCache** is a bandwidth optimization technology that is included in some editions of Windows Server 2016 and Windows 10 operating systems, as well as in some editions of Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2, and Windows 7.

NOTE

Full BranchCache functionality is supported in Windows 10 Enterprise and Education; Windows 10 Pro supports some BranchCache functionality, including BITS transfers used for servicing operations.

Windows Server Update Services (WSUS) and System Center Configuration Manager can use BranchCache to allow peers to source content from each other versus always having to contact a server. Using BranchCache, files are cached on each individual client, and other clients can retrieve them as needed. This approach distributes the cache rather than having a single point of retrieval, saving a significant amount of bandwidth while drastically reducing the time that it takes for clients to receive the requested content.

METHOD	WINDOWS UPDATE	WINDOWS UPDATE FOR BUSINESS	WSUS	CONFIGURATION MANAGER
Delivery Optimization	✓	✓	✓	✓
BranchCache	✗	✗	✓	✓

NOTE

System Center Configuration Manager has an additional feature called Client Peer Cache that allows peer-to-peer content sharing between clients you use System Center Configuration Manager to manage, in the same Configuration Manager boundary Group. For more information, see [Client Peer Cache](#).

In addition to Client Peer Cache, similar functionality is available in the Windows Preinstallation Environment (Windows PE) for imaging-related content. Using this technology, clients imaging with System Center Configuration Manager task sequences can source operating system images, driver packages, boot images, packages, and programs from peers instead of distribution points. For detailed information about how Windows PE Peer Cache works and how to configure it, see [Prepare Windows PE peer cache to reduce WAN traffic in System Center Configuration Manager](#).

Express update delivery

Windows 10 quality update downloads can be large because every package contains all previously released fixes to ensure consistency and simplicity. Windows has been able to reduce the size of Windows Update downloads with a feature called Express.

NOTE

Express update delivery applies to quality update downloads. Starting with Windows 10, version 1709, Express update delivery also applies to feature update downloads for clients connected to Windows Update and Windows Update for Business.

How Microsoft supports Express

- **Express on System Center Configuration Manager** starting with version 1702 of Configuration Manager and Windows 10, version 1703 or later, or Windows 10, version 1607 with the April 2017 cumulative update.
- **Express on WSUS Standalone**
Express update delivery is available on [all support versions of WSUS](#).
- **Express on devices directly connected to Windows Update**
- **Enterprise devices managed using Windows Update for Business** also get the benefit of Express update delivery support without any change in configuration.

How Express download works

For OS updates that support Express, there are two versions of the file payload stored on the service:

1. **Full-file version** - essentially replacing the local versions of the update binaries.
2. **Express version** - containing the deltas needed to patch the existing binaries on the device.

Both the full-file version and the Express version are referenced in the update's metadata, which has been downloaded to the client as part of the scan phase.

Express download works as follows:

The Windows Update client will try to download Express first, and under certain situations fall back to full-file if needed (for example, if going through a proxy that doesn't support byte range requests).

1. When the Windows Update client initiates an Express download, **Windows Update first downloads a stub**, which is part of the Express package.
2. **The Windows Update client passes this stub to the Windows installer**, which uses the stub to do a local inventory, comparing the deltas of the file on the device with what is needed to get to the latest version

of the file being offered.

3. **The Windows installer then requests the Windows Update client to download the ranges**, which have been determined to be required.
4. **The client downloads these ranges and passes them to the Windows Installer**, which applies the ranges and then determines if additional ranges are needed. This repeats until the Windows installer tells the Windows Update client that all necessary ranges have been downloaded.

At this point, the download is complete and the update is ready to be installed.

TIP

Express will **always** be leveraged if your machines are updated regularly with the latest cumulative updates.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates
<input checked="" type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input checked="" type="checkbox"/>	Optimize update delivery for Windows 10 updates (this topic)
<input type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

Delivery Optimization for Windows 10 updates

5/31/2019 • 6 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

Windows updates, upgrades, and applications can contain packages with very large files. Downloading and distributing updates can consume quite a bit of network resources on the devices receiving them. You can use Delivery Optimization to reduce bandwidth consumption by sharing the work of downloading these packages among multiple devices in your deployment. Delivery Optimization can accomplish this because it is a self-organizing distributed cache that allows clients to download those packages from alternate sources (such as other peers on the network) in addition to the traditional Internet-based servers. You can use Delivery Optimization in conjunction with Windows Update, Windows Server Update Services (WSUS), Windows Update for Business, or System Center Configuration Manager (when installation of Express Updates is enabled).

Delivery Optimization is a cloud-managed solution. Access to the Delivery Optimization cloud services is a requirement. This means that in order to use the peer-to-peer functionality of Delivery Optimization, devices must have access to the internet.

NOTE

WSUS can also use [BranchCache](#) for content sharing and caching. If Delivery Optimization is enabled on devices that use BranchCache, Delivery Optimization will be used instead.

Requirements

The following table lists the minimum Windows 10 version that supports Delivery Optimization:

DEVICE TYPE	MINIMUM WINDOWS VERSION
Computers running Windows 10	1511
Computers running Server Core installations of Windows Server	1709
IoT devices	1803
HoloLens devices	1803

Types of download packages supported by Delivery Optimization

DOWNLOAD PACKAGE	MINIMUM WINDOWS VERSION
Windows 10 updates (feature updates and quality updates)	1511

DOWNLOAD PACKAGE	MINIMUM WINDOWS VERSION
Windows 10 drivers	1511
Windows Store files	1511
Windows Store for Business files	1511
Windows Defender definition updates	1511
Office Click-to-Run updates	1709
Win32 apps for Intune	1709
SCCM Express Updates	1709 + Configuration Manager version 1711

By default in Windows 10 Enterprise and Education editions, Delivery Optimization allows peer-to-peer sharing on the organization's own network only (specifically, all of the devices must be behind the same NAT), but you can configure it differently in Group Policy and mobile device management (MDM) solutions such as Microsoft Intune.

For more details, see "Download mode" in [Delivery optimization reference](#).

Set up Delivery Optimization

See [Set up Delivery Optimization](#) for suggested values for a number of common scenarios.

You can use Group Policy or an MDM solution like Intune to configure Delivery Optimization.

You will find the Delivery Optimization settings in Group Policy under **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization**. In MDM, the same settings are under **.Vendor/MSFT/Policy/Config/DeliveryOptimization/**.

Starting with Microsoft Intune version 1902, you can set many Delivery Optimization policies as a profile which you can then apply to groups of devices. For more information, see [Delivery Optimization settings in Microsoft Intune](#)

Starting with Windows 10, version 1903, you can use the Azure Active Directory (AAD) Tenant ID as a means to define groups. To do this set the value for DOGroupIDSource to its new maximum value of 5.

Reference

For complete list of every possible Delivery Optimization setting, see [Delivery Optimization reference](#).

How Microsoft uses Delivery Optimization

At Microsoft, to help ensure that ongoing deployments weren't affecting our network and taking away bandwidth for other services, Microsoft IT used a couple of different bandwidth management strategies. Delivery Optimization, peer-to-peer caching enabled through Group Policy, was piloted and then deployed to all managed devices using Group Policy. Based on recommendations from the Delivery Optimization team, we used the "group" configuration to limit sharing of content to only the devices that are members of the same Active Directory domain. The content is cached for 24 hours. More than 76 percent of content came from peer devices versus the Internet.

For more details, check out the [Adopting Windows as a Service at Microsoft](#) technical case study.

Frequently asked questions

Does Delivery Optimization work with WSUS?: Yes. Devices will obtain the update payloads from the WSUS server, but must also have an internet connection as they communicate with the Delivery Optimization cloud service for coordination.

Which ports does Delivery Optimization use?: For peer-to-peer traffic, it uses 7680 for TCP/IP or 3544 for NAT traversal (optionally Teredo). For client-service communication, it uses HTTP or HTTPS over port 80/443.

What are the requirements if I use a proxy?: You must allow Byte Range requests. See [Proxy requirements for Windows Update](#) for details.

What hostnames should I allow through my firewall to support Delivery Optimization?:

For communication between clients and the Delivery Optimization cloud service: ***.do.dsp.mp.microsoft.com**.

For Delivery Optimization metadata:

- *.dl.delivery.mp.microsoft.com
- *.emdl.ws.microsoft.com

For the payloads (optional):

- *.download.windowsupdate.com
- *.windowsupdate.com

Does Delivery Optimization use multicast?: No. It relies on the cloud service for peer discovery, resulting in a list of peers and their IP addresses. Client devices then connect to their peers to obtain download files over TCP/IP.

How does Delivery Optimization deal with congestion on the router from peer-to-peer activity on the LAN?: Starting in Windows 10, version 1903, Delivery Optimization uses LEDBAT to relieve such congestion. For more details see this post on the [Networking Blog](#).

Troubleshooting

This section summarizes common problems and some solutions to try.

If you don't see any bytes from peers

If you don't see any bytes coming from peers the cause might be one of the following issues:

- Clients aren't able to reach the Delivery Optimization cloud services.
- The cloud service doesn't see other peers on the network.
- Clients aren't able to connect to peers that are offered back from the cloud service.

Clients aren't able to reach the Delivery Optimization cloud services.

If you suspect this is the problem, try these steps:

1. Start a download of an app that is larger than 50 MB from the Store (for example "Candy Crush Saga").
2. Run `Get-DeliveryOptimizationStatus` from an elevated Powershell window and observe the DownloadMode setting. For peering to work, DownloadMode should be 1, 2, or 3.
3. If **DownloadMode** is 99 it could indicate your device is unable to reach the Delivery Optimization cloud services. Ensure that the Delivery Optimization hostnames are allowed access: most importantly ***.do.dsp.mp.microsoft.com**.

The cloud service doesn't see other peers on the network.

If you suspect this is the problem, try these steps:

1. Download the same app on two different devices on the same network, waiting 10 – 15 minutes between downloads.
2. Run `Get-DeliveryOptimizationStatus` from an elevated Powershell window and ensure that **DownloadMode** is 1 or 2 on both devices.
3. Run `Get-DeliveryOptimizationPerfSnap` from an elevated Powershell window on the second device. The **NumberOfPeers** field should be non-zero.
4. If the number of peers is zero and you have **DownloadMode** = 1, ensure that both devices are using the same public IP address to reach the internet. To do this, open a browser Windows and search for "what is my IP". You can **DownloadMode 2** (Group) and a custom GroupID (Guid) to fix this if the devices aren't reporting the same public IP address.

Clients aren't able to connect to peers offered by the cloud service

If you suspect this is the problem, try a Telnet test between two devices on the network to ensure they can connect using port 7680. To do this, follow these steps:

1. Install Telnet by running **dism /online /Enable-Feature /FeatureName:TelnetClient** from an elevated command prompt.
2. Run the test. For example, if you are on device with IP 192.168.8.12 and you are trying to test the connection to 192.168.9.17 run **telnet 192.168.9.17 7680** (the syntax is *telnet [destination IP] [port]*). You will either see a connection error or a blinking cursor like this `_`. The blinking cursor means success.

Learn more

[Windows 10, Delivery Optimization, and WSUS](#)

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)
- [Manage device restarts after updates](#)

Set up Delivery Optimization for Windows 10 updates

6/6/2019 • 7 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

Recommended Delivery Optimization settings

Delivery Optimization offers a great many settings to fine-tune its behavior (see [Delivery Optimization reference](#) for a comprehensive list), but for the most efficient performance, there are just a few key parameters that will have the greatest impact if particular situations exist in your deployment:

- Does your topology include multiple breakouts to the internet (i.e., a "hybrid WAN") or are there only a few connections to the internet, so that all requests appear to come from a single external IP address (a "hub and spoke" topology)?
- If you use boundary groups in your topology, how many devices are present in a given group?
- What percentage of your devices are mobile?
- Do your devices have a lot of free space on their drives?
- Do you have a lab scenario with many devices on AC power?

NOTE

These scenarios (and the recommended settings for each) are not mutually exclusive. It's possible that your deployment might involve more than one of these scenarios, in which case you can employ the related settings in any combination as needed. In all cases, however, "download mode" is the most important one to set.

Quick-reference table:

USE CASE	POLICY	RECOMMENDED VALUE	REASON
Hub & spoke topology	Download mode	1 or 2	Automatic grouping of peers to match your topology
Sites with > 30 devices	Minimum file size to cache	10 MB (or 1 MB)	Leverage peers-to-peer capability in more downloads
Large number of mobile devices	Allow uploads on battery power	60%	Increase # of devices that can upload while limiting battery drain
Labs with AC-powered devices	Content Expiration	7 (up to 30) days	Leverage devices that can upload more for a longer period

Hybrid WAN scenario

For this scenario, grouping devices by domain allows devices to be included in peer downloads and uploads across VLANs. **Set Download Mode to 2 - Group**. The default group is the authenticated domain or Active Directory site. If your domain-based group is too wide, or your Active Directory sites aren't aligned with your site network topology, then you should consider additional options for dynamically creating groups, for example by using the GroupIDSrc parameter.

To do this in Group Policy go to **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization** and set **Download mode** to **2**.

To do this with MDM, go to **.Vendor/MSFT/Policy/Config/DeliveryOptimization/** and set **DODownloadMode** to 1 or 2.

Hub and spoke topology with boundary groups

The default download mode setting is **1**; this means all devices breaking out to the internet using the same public IP will be considered as a single peer group. To prevent peer-to-peer activity across groups, you should set the download mode to **2**. If you have already defined Active Directory sites per hub or branch office, then you don't need to do anything else. If you're not using Active Directory sites, you should set *RestrictPeerSelectionBy* policies to restrict the activity to the subnet or set a different source for Groups by using the GroupIDSrc parameter. See [Select a method to restrict peer selection](#).

To do this in Group Policy go to **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization** and set **Download mode** to **2**.

To do this with MDM, go to **.Vendor/MSFT/Policy/Config/DeliveryOptimization/** and set **DODownloadMode** to **2**.

Large number of mobile devices

If you have a mobile workforce with a great many mobile devices, set Delivery Optimization to allow uploads on battery power, while limiting the use to prevent battery drain. A setting for **DOMinBatteryPercentageAllowedToUpload** of 60% is a good starting point, though you might want to adjust it later.

To do this in Group Policy, go to **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization** and set **Allow uploads while the device is on battery while under set Battery level** to 60.

To do this with MDM, go to **.Vendor/MSFT/Policy/Config/DeliveryOptimization/** and set **DOMinBatteryPercentageAllowedToUpload** to 60.

Plentiful free space and large numbers of devices

Many devices now come with large internal drives. You can set Delivery Optimization to take better advantage of this space (especially if you have large numbers of devices) by changing the minimum file size to cache. If you have more than 30 devices in your local network or group, change it from the default 50 MB to 10 MB. If you have more than 100 devices (and are running Windows 10, version 1803 or later), set this value to 1 MB.

To do this in Group Policy, go to **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization** and set **Minimum Peer Caching Content File Size** to 100 (if you have more than 30 devices) or 1 (if you have more than 100 devices).

To do this with MDM, go to **.Vendor/MSFT/Policy/Config/DeliveryOptimization/** and set **DOMinFileSizeToCache** to 100 (if you have more than 30 devices) or 1 (if you have more than 100 devices).

Lab scenario

In a lab situation, you typically have a large number of devices that are plugged in and have a lot of free disk space. By increasing the content expiration interval, you can take advantage of these devices, using them as excellent upload sources in order to upload much more content over a longer period.

To do this in Group Policy, go to **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization** and set **Max Cache Age** to **6048000** (7 days) or more (up to 30 days).

To do this with MDM, go to **.Vendor/MSFT/Policy/Config/DeliveryOptimization/** and set **DOMaxCacheAge** to 7 or more (up to 30 days).

Monitor Delivery Optimization

Windows PowerShell cmdlets

Starting in Windows 10, version 1703, you can use new PowerShell cmdlets to check the performance of Delivery Optimization.

Analyze usage

`Get-DeliveryOptimizationStatus` returns a real-time snapshot of all current Delivery Optimization jobs.

KEY	VALUE
File ID	A GUID that identifies the file being processed
Priority	Priority of the download; values are foreground or background
FileSize	Size of the file
TotalBytesDownloaded	The number of bytes from any source downloaded so far
PercentPeerCaching	The percentage of bytes downloaded from peers versus over HTTP
BytesFromPeers	Total bytes downloaded from peer devices (sum of bytes downloaded from LAN, Group, and Internet Peers)
BytesfromHTTP	Total number of bytes received over HTTP
DownloadDuration	Total download time in seconds
Status	Current state of the operation. Possible values are: Downloading (download in progress); Complete (download completed, but is not uploading yet); Caching (download completed successfully and is ready to upload or uploading); Paused (download/upload paused by caller)
NumPeers	Indicates the total number of peers returned from the service.
PredefinedCallerApplication	Indicates the last caller that initiated a request for the file.
ExpireOn	The target expiration date and time for the file.
Pinned	A yes/no value indicating whether an item has been "pinned" in the cache (see <code>setDeliveryOptimizationStatus</code>).

`Get-DeliveryOptimizationPerfSnap` returns a list of key performance data:

- Number of files downloaded
- Number of files uploaded

- Total bytes downloaded
- Total bytes uploaded
- Average transfer size (download); that is, the number bytes downloaded divided by the number of files
- Average transfer size (upload); the number of bytes uploaded divided by the number of files
- Peer efficiency; same as PercentPeerCaching

Using the `-Verbose` option returns additional information:

- Bytes from peers (per type)
- Bytes from CDN (the number of bytes received over HTTP)
- Average number of peer connections per download

Starting in Windows 10, version 1903, `Get-DeliveryOptimizationPerfSnap` has a new option `-CacheSummary` which provides a summary of the cache status.

Starting in Windows 10, version 1803, `Get-DeliveryOptimizationPerfSnapThisMonth` returns data similar to that from `Get-DeliveryOptimizationPerfSnap` but limited to the current calendar month.

Manage the Delivery Optimization cache

Starting in Windows 10, version 1903:

`set-DeliveryOptimizationStatus -ExpireOn [date time]` extends the expiration of all files in the cache. You can set the expiration immediately for all files that are in the "caching" state. For files in progress ("downloading"), the expiration is applied once the download is complete. You can set the expiration up to one year from the current date and time.

`set-DeliveryOptimizationStatus -ExpireOn [date time] -FileID [FileID]` extends expiration for a single specific file in the cache.

You can now "pin" files to keep them persistent in the cache. You can only do this with files that are downloaded in modes 1, 2, or 3.

`set-DeliveryOptimizationStatus -Pin [True] -File ID [FileID]` keeps a specific file in the cache such that it won't be deleted until the expiration date and time (which you set with `set-DeliveryOptimizationStatus -ExpireOn [date time] -FileID [FileID]`). The file is also excluded from the cache quota calculation.

`set-DeliveryOptimizationStatus -Pin [False] -File ID [FileID]` "unpins" a file, so that it will be deleted when the expiration date and time are reached. The file is included in the cache quota calculation.

`delete-DeliveryOptimizationCache` lets you clear files from the cache and remove all persisted data related to them. You can use these options with this cmdlet:

- `-FileID` specifies a particular file to delete.
- `-IncludePinnedFiles` deletes all files that are pinned.
- `-Force` deletes the cache with no prompts.

Work with Delivery Optimization logs

Starting in Windows 10, version 1803:

`Get-DeliveryOptimizationLog [-Path <etl file path, supports wildcards>] [-Flush]`

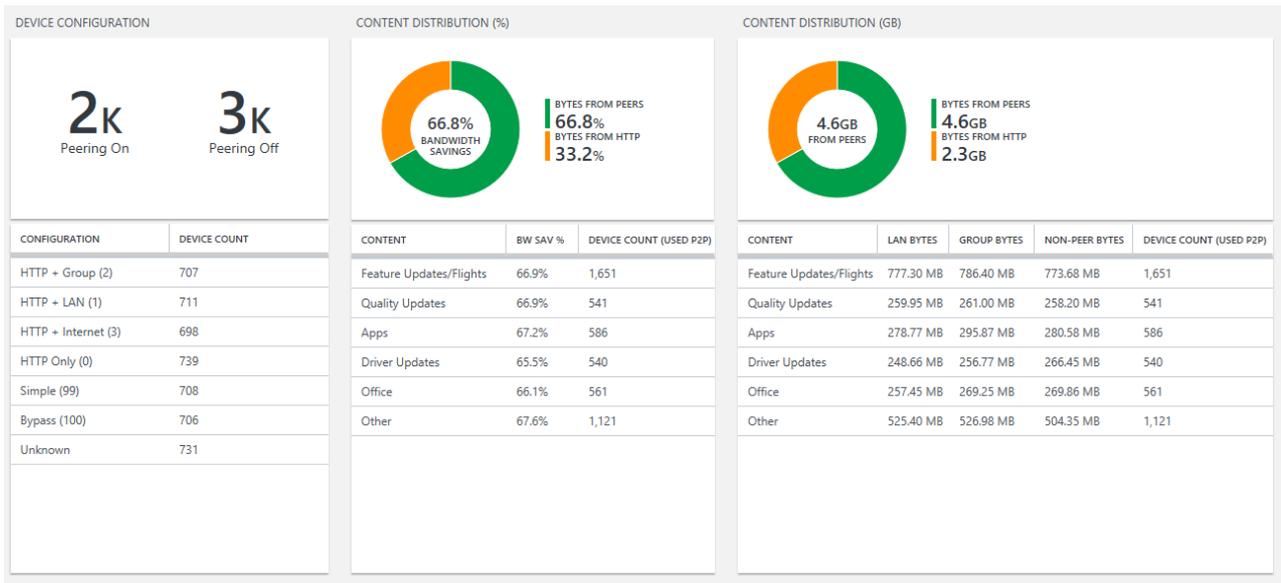
If `Path` is not specified, this cmdlet reads all logs from the dosvc log directory, which requires administrator permissions. If `Flush` is specified, the cmdlet stops dosvc before reading logs.

Log entries are written to the PowerShell pipeline as objects. To dump logs to a text file, run

`Get-DeliveryOptimizationLog | Set-Content <output file>` or something similar.

Monitor with Update Compliance

The Update Compliance solution of Windows Analytics provides you with information about your Delivery Optimization configuration, including the observed bandwidth savings across all devices that used peer-to-peer distribution over the past 28 days.



For details, see [Delivery Optimization in Update Compliance](#).

Delivery Optimization reference

6/6/2019 • 15 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

There are a great many details you can set in Delivery Optimization to customize it to do just what you need it to. This topic summarizes them for your reference.

Delivery Optimization options

You can use Group Policy or an MDM solution like Intune to configure Delivery Optimization.

You will find the Delivery Optimization settings in Group Policy under **Configuration\Policies\Administrative Templates\Windows Components\Delivery Optimization**. In MDM, the same settings are under **.Vendor/MSFT/Policy/Config/DeliveryOptimization/**.

Summary of Delivery Optimization settings :

GROUP POLICY SETTING	MDM SETTING	SUPPORTED FROM VERSION
Download mode	DODownloadMode	1511
Group ID	DOGroupID	1511
Minimum RAM (inclusive) allowed to use Peer Caching	DOMinRAMAllowedToPeer	1703
Minimum disk size allowed to use Peer Caching	DOMinDiskSizeAllowedToPeer	1703
Max Cache Age	DOMaxCacheAge	1511
Max Cache Size	DOMaxCacheSize	1511
Absolute Max Cache Size	DOAbsoluteMaxCacheSize	1607
Modify Cache Drive	DOModifyCacheDrive	1607
Minimum Peer Caching Content File Size	DOMinFileSizeToCache	1703
Maximum Download Bandwidth	DOMaxDownloadBandwidth	1607
Percentage of Maximum Download Bandwidth	DOPercentageMaxDownloadBandwidth	1607
Max Upload Bandwidth	DOMaxUploadBandwidth	1607

GROUP POLICY SETTING	MDM SETTING	SUPPORTED FROM VERSION
Monthly Upload Data Cap	DOMonthlyUploadDataCap	1607
Minimum Background QoS	DOMinBackgroundQoS	1607
Enable Peer Caching while the device connects via VPN	DOAllowVPNPeerCaching	1709
Allow uploads while the device is on battery while under set Battery level	DOMinBatteryPercentageAllowedToUpload	1709
MaxForegroundDownloadBandwidth	DOPercentageMaxForegroundBandwidth	1803
MaxBackgroundDownloadBandwidth	DOPercentageMaxBackgroundBandwidth	1803
SetHoursToLimitBackgroundDownloadBandwidth	DOSetHoursToLimitBackgroundDownloadBandwidth	1803
SetHoursToLimitForegroundDownloadBandwidth	DOSetHoursToLimitForegroundDownloadBandwidth	1803
Select a method to restrict Peer Selection	DORestrictPeerSelectionBy	1803
Select the source of Group IDs	DOGroupIDSource	1803
Delay background download from http (in secs)	DODelayBackgroundDownloadFromHttp	1803
Delay foreground download from http (in secs)	DODelayForegroundDownloadFromHttp	1803
Delay foreground download cache server fallback (in secs)	DelayCacheServerFallbackForeground	1903
Delay background download cache server fallback (in secs)	DelayCacheServerFallbackBackground	1903

More detail on Delivery Optimization settings:

Group ID, combined with Group [Download mode](#), enables administrators to create custom device groups that will share content between devices in the group.

Delivery Optimization uses locally cached updates. In cases where devices have ample local storage and you would like to cache more content, or if you have limited storage and would like to cache less, use the following settings to adjust the Delivery Optimization cache to suit your scenario:

- [Max Cache Size](#) and [Absolute Max Cache Size](#) control the amount of space the Delivery Optimization cache can use.
- [Max Cache Age](#) controls the retention period for each update in the cache.
- The system drive is the default location for the Delivery Optimization cache. [Modify Cache Drive](#) allows administrators to change that location.

NOTE

It is possible to configure preferred cache devices. For more information, see [Group ID](#).

All cached files have to be above a set minimum size. This size is automatically set by the Delivery Optimization cloud services, but when local storage is sufficient and the network isn't strained or congested, administrators might choose to change it to obtain increased performance. You can set the minimum size of files to cache by adjusting [Minimum Peer Caching Content File Size](#).

Additional options available that control the impact Delivery Optimization has on your network include the following:

- [Maximum Download Bandwidth](#) and [Percentage of Maximum Download Bandwidth](#) control the download bandwidth used by Delivery Optimization.
- [Max Upload Bandwidth](#) controls the Delivery Optimization upload bandwidth usage.
- [Monthly Upload Data Cap](#) controls the amount of data a client can upload to peers each month.
- [Minimum Background QoS](#) lets administrators guarantee a minimum download speed for Windows updates. This is achieved by adjusting the amount of data downloaded directly from Windows Update or WSUS servers, rather than other peers in the network.
- [Maximum Foreground Download Bandwidth](#) specifies the **maximum foreground download bandwidth** that Delivery Optimization uses, across all concurrent download activities, as a percentage of available download bandwidth.
- [Maximum Background Download Bandwidth](#) specifies the **maximum background download bandwidth** that Delivery Optimization uses, across all concurrent download activities, as a percentage of available download bandwidth.
- [Set Business Hours to Limit Background Download Bandwidth](#) specifies the maximum background download bandwidth that Delivery Optimization uses during and outside business hours across all concurrent download activities as a percentage of available download bandwidth.
- [Set Business Hours to Limit Foreground Download Bandwidth](#) specifies the maximum foreground download bandwidth that Delivery Optimization uses during and outside business hours across all concurrent download activities as a percentage of available download bandwidth.
- [Select a method to restrict Peer Selection](#) restricts peer selection by the options you select.
- [Select the source of Group IDs](#) restricts peer selection to a specific source.
- [Delay background download from http \(in secs\)](#) allows you to delay the use of an HTTP source in a background download that is allowed to use P2P.
- [Delay foreground download from http \(in secs\)](#) allows you to delay the use of an HTTP source in a foreground (interactive) download that is allowed to use P2P.

Administrators can further customize scenarios where Delivery Optimization will be used with the following settings:

- [Minimum RAM \(inclusive\) allowed to use Peer Caching](#) sets the minimum RAM required for peer caching to be enabled.
- [Minimum disk size allowed to use Peer Caching](#) sets the minimum disk size required for peer caching to be enabled.
- [Enable Peer Caching while the device connects via VPN](#) allows clients connected through VPN to use peer caching.
- [Allow uploads while the device is on battery while under set Battery level](#) controls the minimum battery level required for uploads to occur. You must enable this policy to allow upload while on battery.

Download mode

Download mode dictates which download sources clients are allowed to use when downloading Windows updates

in addition to Windows Update servers. The following table shows the available download mode options and what they do. Additional technical details for these policies are available in [Policy CSP - Delivery Optimization](#).

DOWNLOAD MODE OPTION	FUNCTIONALITY WHEN SET
HTTP Only (0)	This setting disables peer-to-peer caching but still allows Delivery Optimization to download content over HTTP from the download's original source. This mode uses additional metadata provided by the Delivery Optimization cloud services for a peerless reliable and efficient download experience.
LAN (1 – Default)	This default operating mode for Delivery Optimization enables peer sharing on the same network. The Delivery Optimization cloud service finds other clients that connect to the Internet using the same public IP as the target client. These clients then attempts to connect to other peers on the same network by using their private subnet IP.
Group (2)	When group mode is set, the group is automatically selected based on the device's Active Directory Domain Services (AD DS) site (Windows 10, version 1607) or the domain the device is authenticated to (Windows 10, version 1511). In group mode, peering occurs across internal subnets, between devices that belong to the same group, including devices in remote offices. You can use GroupID option to create your own custom group independently of domains and AD DS sites. Starting with Windows 10, version 1803, you can use the GroupIDSource parameter to take advantage of other method to create groups dynamically. Group download mode is the recommended option for most organizations looking to achieve the best bandwidth optimization with Delivery Optimization.
Internet (3)	Enable Internet peer sources for Delivery Optimization.
Simple (99)	Simple mode disables the use of Delivery Optimization cloud services completely (for offline environments). Delivery Optimization switches to this mode automatically when the Delivery Optimization cloud services are unavailable, unreachable or when the content file size is less than 10 MB. In this mode, Delivery Optimization provides a reliable download experience, with no peer-to-peer caching.
Bypass (100)	Bypass Delivery Optimization and use BITS, instead. You should only select this mode if you use WSUS and prefer to use BranchCache. You do not need to set this option if you are using SCCM. If you want to disable peer-to-peer functionality, it's best to set DownloadMode to 0 or 99 .

NOTE

Group mode is a best-effort optimization and should not be relied on for an authentication of identity of devices participating in the group.

Group ID

By default, peer sharing on clients using the group download mode is limited to the same domain in Windows 10, version 1511, and the same domain and AD DS site in Windows 10, version 1607. By using the Group ID setting,

you can optionally create a custom group that contains devices that should participate in Delivery Optimization but do not fall within those domain or AD DS site boundaries, including devices in another domain. Using Group ID, you can further restrict the default group (for example, you could create a sub-group representing an office building), or extend the group beyond the domain, allowing devices in multiple domains in your organization to be peers. This setting requires the custom group to be specified as a GUID on each device that participates in the custom group.

NOTE

To generate a GUID using Powershell, use `[guid]::NewGuid()`

This configuration is optional and not required for most implementations of Delivery Optimization.

Select the source of Group IDs

Starting in Windows 10, version 1803, set this policy to restrict peer selection to a specific source. The options are:

- 0 = not set
- 1 = AD Site
- 2 = Authenticated domain SID
- 3 = DHCP Option ID (with this option, the client will query DHCP Option ID 234 and use the returned GUID value as the Group ID)
- 4 = DNS Suffix

When set, the Group ID is assigned automatically from the selected source. If you set this policy, the GroupID policy will be ignored. The option set in this policy only applies to Group (2) download mode. If Group (2) isn't set as Download mode, this policy will be ignored. If you set the value to anything other than 0-4, the policy is ignored.

Minimum RAM (inclusive) allowed to use Peer Caching

This setting specifies the minimum RAM size in GB required to use Peer Caching. For example if the minimum set is 1 GB, then devices with 1 GB or higher available RAM will be allowed to use Peer caching. The recommended values are 1 to 4 GB, and the default value is 4 GB.

Minimum disk size allowed to use Peer Caching

This setting specifies the required minimum disk size (capacity in GB) for the device to use Peer Caching. The recommended values are 64 to 256 GB, and the default value is 32 GB.

NOTE

If the [Modify Cache Drive](#) policy is set, the disk size check will apply to the new working directory specified by this policy.

Max Cache Age

In environments configured for Delivery Optimization, you might want to set an expiration on cached updates and Windows application installation files. If so, this setting defines the maximum number of seconds each file can be held in the Delivery Optimization cache on each Windows 10 client device. The default Max Cache Age value is 259,200 seconds (3 days). Alternatively, organizations might choose to set this value to "0" which means "unlimited" to avoid peers re-downloading content. When "Unlimited" value is set, Delivery Optimization will hold the files in the cache longer and will clean up the cache as needed (for example when the cache size exceeded the maximum space allowed).

Max Cache Size

This setting limits the maximum amount of space the Delivery Optimization cache can use as a percentage of the available drive space, from 1 to 100. For example, if you set this value to 10 on a Windows 10 client device that has

100 GB of available drive space, then Delivery Optimization will use up to 10 GB of that space. Delivery Optimization will constantly assess the available drive space and automatically clear the cache to keep the maximum cache size under the set percentage. The default value for this setting is 20.

Absolute Max Cache Size

This setting specifies the maximum number of gigabytes the Delivery Optimization cache can use. This is different from the **Max Cache Size** setting, which is a percentage of available disk space. Also, if you configure this policy, it will override the **Max Cache Size** setting. The default value for this setting is 10 GB.

Minimum Peer Caching Content File Size

This setting specifies the minimum content file size in MB enabled to use Peer Caching. The recommended values are from 1 to 100000 MB.

Maximum Download Bandwidth

This setting specifies the maximum download bandwidth that can be used across all concurrent Delivery Optimization downloads in kilobytes per second (KB/s). A default value of 0 means that Delivery Optimization will dynamically adjust and optimize the maximum bandwidth used.

Maximum Foreground Download Bandwidth

Starting in Windows 10, version 1803, specifies the maximum foreground download bandwidth that Delivery Optimization uses across all concurrent download activities as a percentage of available download bandwidth. The default value of 0 means that Delivery Optimization dynamically adjusts to use the available bandwidth for foreground downloads. However, downloads from LAN peers are not throttled even when this policy is set.

Maximum Background Download Bandwidth

Starting in Windows 10, version 1803, specifies the maximum background download bandwidth that Delivery Optimization uses across all concurrent download activities as a percentage of available download bandwidth. The default value of 0 means that Delivery Optimization dynamically adjusts to use the available bandwidth for foreground downloads. However, downloads from LAN peers are not throttled even when this policy is set.

Percentage of Maximum Download Bandwidth

This setting specifies the maximum download bandwidth that Delivery Optimization can use across all concurrent download activities as a percentage of available download bandwidth. The default value 0 means that Delivery Optimization dynamically adjusts to use the available bandwidth for downloads.

Max Upload Bandwidth

This setting allows you to limit the amount of upload bandwidth individual clients can use for Delivery Optimization. Consider this setting when clients are providing content to requesting peers on the network. This option is set in kilobytes per second (KB/s). The default setting is 0, or "unlimited" which means Delivery Optimization dynamically optimizes for minimal usage of upload bandwidth; however it does not cap the upload bandwidth rate at a set rate.

Set Business Hours to Limit Background Download Bandwidth

Starting in Windows 10, version 1803, specifies the maximum background download bandwidth that Delivery Optimization uses during and outside business hours across all concurrent download activities as a percentage of available download bandwidth.

Set Business Hours to Limit Foreground Download Bandwidth

Starting in Windows 10, version 1803, specifies the maximum foreground download bandwidth that Delivery Optimization uses during and outside business hours across all concurrent download activities as a percentage of available download bandwidth.

Select a method to restrict peer selection

Starting in Windows 10, version 1803, set this policy to restrict peer selection via selected option.

Currently the only available option is **1 = Subnet mask**. This option (Subnet mask) applies to both Download Modes LAN (1) and Group (2).

Delay background download from http (in secs)

Starting in Windows 10, version 1803, this allows you to delay the use of an HTTP source in a background download that is allowed to use peer-to-peer.

Delay foreground download from http (in secs)

Starting in Windows 10, version 1803, allows you to delay the use of an HTTP source in a foreground (interactive) download that is allowed to use peer-to-peer.

Delay Foreground Download Cache Server Fallback (in secs)

Starting in Windows 10, version 1903, allows you to delay the fallback from cache server to the HTTP source for foreground content download by X seconds. If you set the policy to delay foreground download from http, it will apply first (to allow downloads from peers first).

Delay Background Download Cache Server Fallback (in secs)

Starting in Windows 10, version 1903, set this policy to delay the fallback from cache server to the HTTP source for a background content download by X seconds. If you set the policy to delay background download from http, it will apply first (to allow downloads from peers first).

Minimum Background QoS

This value specifies the minimum download speed guarantee that a client attempts to achieve and will fulfill by downloading more kilobytes from Windows Update servers or WSUS. Simply put, the lower this value is, the more content will be sourced using peers on the network rather than Windows Update. The higher this value, the more content is received from Windows Update servers or WSUS, versus peers on the local network.

Modify Cache Drive

This setting allows for an alternate Delivery Optimization cache location on the clients. By default, the cache is stored on the operating system drive through the %SYSTEMDRIVE% environment variable. You can set the value to an environment variable (e.g., %SYSTEMDRIVE%), a drive letter (e.g., D:), or a folder path (e.g., D:\DOCache).

Monthly Upload Data Cap

This setting specifies the total amount of data in gigabytes that a Delivery Optimization client can upload to Internet peers per month. A value of 0 means that an unlimited amount of data can be uploaded. The default value for this setting is 20 GB.

Enable Peer Caching while the device connects via VPN

This setting determines whether a device will be allowed to participate in Peer Caching while connected to VPN. Specify "true" to allow the device to participate in Peer Caching while connected via VPN to the domain network. This means the device can download from or upload to other domain network devices, either on VPN or on the corporate domain network.

Allow uploads while the device is on battery while under set Battery level

This setting specifies battery levels at which a device will be allowed to upload data. Specify any value between 1 and 100 (in percentage) to allow the device to upload data to LAN and Group peers while on DC power (Battery). Uploads will automatically pause when the battery level drops below the set minimum battery level. The recommended value to set if you allow uploads on battery is 40 (for 40%). The device can download from peers while on battery regardless of this policy.

IMPORTANT

By default, devices **will not upload while on battery**. To enable uploads while on battery, you need to enable this policy and set the battery value under which uploads pause.

Configure BranchCache for Windows 10 updates

5/31/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

BranchCache is a bandwidth-optimization feature that has been available since the Windows Server 2008 R2 and Windows 7 operating systems. Each client has a cache and acts as an alternate source for content that devices on its own network request. Windows Server Update Services (WSUS) and System Center Configuration Manager can use BranchCache to optimize network bandwidth during update deployment, and it's easy to configure for either of them. BranchCache has two operating modes: Distributed Cache mode and Hosted Cache mode.

- Distributed Cache mode operates like the [Delivery Optimization](#) feature in Windows 10: each client contains a cached version of the BranchCache-enabled files it requests and acts as a distributed cache for other clients requesting that same file.

TIP

Distributed Cache mode is preferred to Hosted Cache mode for Windows 10 updates to get the most benefit from peer-to-peer distribution.

- In Hosted Cache mode, designated servers at specific locations act as a cache for files requested by clients in its area. Then, rather than clients retrieving files from a latent source, the hosted cache server provides the content on its behalf.

For detailed information about how Distributed Cache mode and Hosted Cache mode work, see [BranchCache Overview](#).

Configure clients for BranchCache

Whether you use BranchCache with Configuration Manager or WSUS, each client that uses BranchCache must be configured to do so. You typically make your configurations through Group Policy. For step-by-step instructions on how to use Group Policy to configure BranchCache for Windows clients, see [Client Configuration](#) in the [BranchCache Early Adopter's Guide](#).

In Windows 10, version 1607, the Windows Update Agent uses Delivery Optimization by default, even when the updates are retrieved from WSUS. When using BranchCache with Windows 10, simply set the Delivery Optimization mode to Bypass to allow clients to use the Background Intelligent Transfer Service (BITS) protocol with BranchCache instead. For instructions on how to use BranchCache in Distributed Cache mode with WSUS, see the section [WSUS and Configuration Manager with BranchCache in Distributed Cache mode](#).

Configure servers for BranchCache

You can use WSUS and Configuration Manager with BranchCache in Distributed Cache mode. BranchCache in Distributed Cache mode is easy to configure for both WSUS and System Center Configuration Manager.

For a step-by-step guide to configuring BranchCache on Windows Server devices, see the [BranchCache](#)

[Deployment Guide \(Windows Server 2012\)](#) or [BranchCache Deployment Guide \(Windows Server 2016\)](#).

In addition to these steps, there is one requirement for WSUS to be able to use BranchCache in either operating mode: the WSUS server must be configured to download updates locally on the server to a shared folder. This way, you can select BranchCache publication for the share. For Configuration Manager, you can enable BranchCache on distribution points; no other server-side configuration is necessary for Distributed Cache mode.

NOTE

Configuration Manager only supports Distributed Cache mode.

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using Configuration Manager](#)
- [Manage device restarts after updates](#)

Windows Updates using forward and reverse differentials

5/31/2019 • 7 minutes to read • [Edit Online](#)

Windows 10 monthly quality updates are cumulative, containing all previously released fixes to ensure consistency and simplicity. For an operating system platform like Windows 10, which stays in support for multiple years, the size of monthly quality updates can quickly grow large, thus directly impacting network bandwidth consumption.

Today, this problem is addressed by using express downloads, where differential downloads for every changed file in the update are generated based on selected historical revisions plus the base version. In this paper, we introduce a new technique to build compact software update packages that are applicable to any revision of the base version, and then describe how Windows 10 quality updates uses this technique.

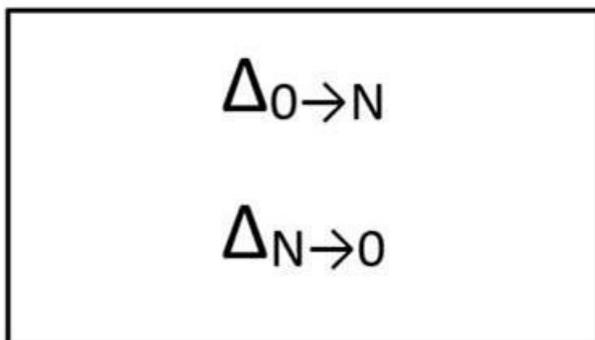
General Terms

The following general terms apply throughout this document:

- *Base version*: A major software release with significant changes, such as Windows 10, version 1809 (Windows 10 Build 17763.1)
- *Revision*: Minor releases in between the major version releases, such as KB4464330 (Windows 10 Build 17763.55)
- *Baseless Patch Storage Files (Baseless PSF)*: Patch storage files that contain full binaries or files

Introduction

In this paper, we introduce a new technique that can produce compact software updates optimized for any origin/destination revision pair. It does this by calculating forward the differential of a changed file from the base version and its reverse differential back to the base version. Both forward and reverse differentials are then packaged as an update and distributed to the endpoints running the software to be updated. The update package contents can be symbolized as follows:



The endpoints that have the base version of the file (V_0) hydrate the target revision (V_N) by applying a simple transformation:

$$V_0 + \Delta_{0 \rightarrow N} = V_N$$

The endpoints that have revision N of the file (V_N), hydrate the target revision (V_R) by applying the following set of transformations:

$$V_N + \Delta_{N \rightarrow 0} = V_0$$

$$V_0 + \Delta_{0 \rightarrow R} = V_R$$

The endpoints retain the reverse differentials for the software revision they are on, so that it can be used for hydrating and applying next revision update.

By using a common baseline, this technique produces a single update package with numerous advantages:

- Compact in size
- Applicable to all baselines
- Simple to build
- Efficient to install
- Redistributable

Historically, download sizes of Windows 10 quality updates (Windows 10, version 1803 and older supported versions of Windows 10) are optimized by using express download. Express download is optimized such that updating Windows 10 systems will download the minimum number of bytes. This is achieved by generating differentials for every updated file based on selected historical base revisions of the same file + its base or RTM version.

For example, if the October monthly quality update has updated Notepad.exe, differentials for Notepad.exe file changes from September to October, August to October, July to October, June to October, and from the original feature release to October are generated. All these differentials are stored in a Patch Storage File (PSF, also referred to as "express download files") and hosted or cached on Windows Update or other update management or distribution servers (for example, Windows Server Update Services (WSUS), System Center Configuration Manager, or a non-Microsoft update management or distribution server that supports express updates). A device leveraging express updates uses network protocol to determine optimal differentials, then downloads only what is needed from the update distribution endpoints.

The flipside of express download is that the size of PSF files can be very large depending on the number of historical baselines against which differentials were calculated. Downloading and caching large PSF files to on-premises or remote update distribution servers is problematic for most organizations, hence they are unable to leverage express updates to keep their fleet of devices running Windows 10 up to date. Secondly, due to the complexity of generating differentials and size of the express files that need to be cached on update distribution servers, it is only feasible to generate express download files for the most common baselines, thus express updates are only applicable to selected baselines. Finally, calculation of optimal differentials is expensive in terms of system memory utilization, especially for low-cost systems, impacting their ability to download and apply an update seamlessly.

In the following sections, we describe how Windows 10 quality updates will leverage this technique based on forward and reverse differentials for newer releases of Windows 10 and Windows Server to overcome the challenges with express downloads.

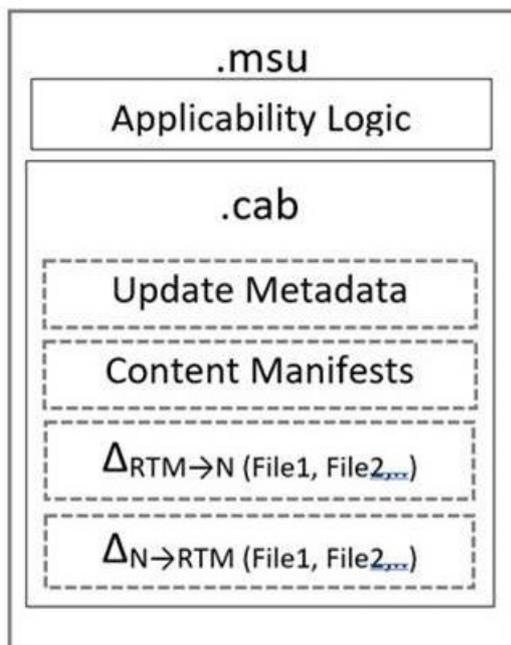
High-level Design

Update packaging

Windows 10 quality update packages will contain forward differentials from quality update RTM baselines

($\Delta_{RTM \rightarrow N}$) and reverse differentials back to RTM ($\Delta_{N \rightarrow RTM}$) for each file that has changed since RTM. By using the RTM version as the baseline, we ensure that all devices will have an identical payload. Update package metadata, content manifests, and forward and reverse differentials will be packaged into a cabinet file (.cab). This .cab file, and the applicability logic, will also be wrapped in Microsoft Standalone Update (.msu) format.

There can be cases where new files are added to the system during servicing. These files will not have RTM baselines, thus forward and reverse differentials cannot be used. In these scenarios, null differentials will be used to handle servicing. Null differentials are the slightly compressed and optimized version of the full binaries. Update packages can have either forward or reverse differentials, or null differential of any given binary in them. The following image symbolizes the content of a Windows 10 quality update installer:



Hydration and installation

Once the usual applicability checks are performed on the update package and are determined to be applicable, the Windows component servicing infrastructure will hydrate the full files during pre-installation and then proceed with the usual installation process.

Below is a high-level sequence of activities that the component servicing infrastructure will run in a transaction to complete installation of the update:

- Identify all files that are required to install the update.
- Hydrate each of necessary files using current version (V_N) of the file, reverse differential ($V_{N \rightarrow RTM}$) of the file back to quality update RTM/base version and forward differential ($V_{RTM \rightarrow N}$) from feature update RTM/base version to the target version. Also, use null differential hydration to hydrate null compressed files.
- Stage the hydrated files (full file), forward differentials (under 'f' folder) and reverse differentials (under 'r' folder) or null compressed files (under 'n' folder) in the component store (%windir%\WinSxS folder).
- Resolve any dependencies and install components.
- Clean up older state (V_{N-1}); the previous state V_N is retained for uninstallation and restoration or repair.

Resilient Hydration

To ensure resiliency against component store corruption or missing files that could occur due to susceptibility of certain types of hardware to file system corruption, a corruption repair service has been traditionally used to recover the component store automatically ("automatic corruption repair") or on demand ("manual corruption repair") using an online or local repair source. This service will continue to offer the ability to repair and recover

content for hydration and successfully install an update, if needed.

When corruption is detected during update operations, automatic corruption repair will start as usual and use the Baseless Patch Storage File published to Windows Update for each update to fix corrupted manifests, binary differentials, or hydrated or full files. Baseless patch storage files will contain reverse and forward differentials and full files for each updated component. Integrity of the repair files will be hash verified.

Corruption repair will use the component manifest to detect missing files and get hashes for corruption detection. During update installation, new registry flags for each differential staged on the machine will be set. When automatic corruption repair runs, it will scan hydrated files using the manifest and differential files using the flags. If the differential cannot be found or verified, it will be added to the list of corruptions to repair.

Lazy automatic corruption repair

"Lazy automatic corruption repair" runs during update operations to detect corrupted binaries and differentials. While applying an update, if hydration of any file fails, "lazy" automatic corruption repair automatically starts, identifies the corrupted binary or differential file, and then adds it to the corruption list. Later, the update operation continues as far as it can go, so that "lazy" automatic corruption repair can collect as many corrupted files to fix as possible. At the end of the hydration section, the update fails, and automatic corruption repair starts. Automatic corruption repair runs as usual and at the end of its operation, adds the corruption list generated by "lazy" automatic corruption repair on top of the new list to repair. Automatic corruption repair then repairs the files on the corruption list and installation of the update will succeed on the next attempt.

Best practices and recommendations for deploying Windows 10 Feature updates to mission critical devices

6/12/2019 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 10

Managing an environment with devices that provide mission critical services 24 hours a day, 7 days a week, can present challenges in keeping these devices current with Windows 10 feature updates. The processes that you use to keep regular devices current with Windows 10 feature updates, often aren't the most effective to service mission critical devices. This whitepaper will focus on the recommended approach of using the System Center Configuration Manager (current branch) software updates feature to deploy Windows 10 semi-annual feature updates.

For simplicity, we will outline the steps to deploy a feature update manually. If you prefer an automated approach, please see [Using Windows 10 servicing plans to deploy Windows 10 feature updates](#).

Devices and shared workstations that are online and available 24 hours a day, 7 days a week, can be serviced via one of two primary methods:

- **Service during maintenance windows** – Devices that have established maintenance windows will need to have feature updates scheduled to fit within these windows.
- **Service only when manually initiated** – Devices that need physical verification of the availability to update will need to have updates manually initiated by a technician.

You can use Configuration Manager to deploy feature updates to Windows 10 devices in two ways. The first option is to use the software updates feature. The second option is to use a task sequence to deploy feature updates. There are times when deploying a Windows 10 feature update requires the use of a task sequence—for example:

- **Upgrade to the next LTSC release.** With the LTSC servicing branch, feature updates are never provided to the Windows clients themselves. Instead, feature updates must be installed like a traditional in-place upgrade.
- **Additional required tasks.** When deploying a feature update requires additional steps (e.g., suspending disk encryption, updating applications), you can use task sequences to orchestrate the additional steps. Software updates do not have the ability to add steps to their deployments.
- **Language pack installs.** When deploying a feature update requires the installation of additional language packs, you can use task sequences to orchestrate the installation. Software updates do not have the ability to natively install language packs.

If you need to leverage a task sequence to deploy feature updates, please see [Using a task sequence to deploy Windows 10 updates](#) for more information. If you find that your requirement for a task sequence is based solely on the need to run additional tasks preformed pre-install or pre-commit, please see the new [run custom actions](#) functionality first introduced with Windows 10, version 1803. You may be able to leverage this functionality with the software updates deployment method.

Use the following information:

- [Deploy feature updates during maintenance windows](#)
- [Deploy feature updates for user-initiated installations](#)
- [Conclusion](#)

Deploy feature updates during maintenance windows

6/26/2019 • 18 minutes to read • [Edit Online](#)

Applies to: Windows 10

Use the following information to deploy feature updates during a maintenance window.

Get ready to deploy feature updates

Step 1: Configure maintenance windows

1. In the Configuration Manager console, choose **Assets and Compliance > Device Collections**.
2. In the **Device Collections** list, select the collection for which you intended to deploy the feature update(s).
3. On the **Home** tab, in the **Properties** group, choose **Properties**.
4. In the **Maintenance Windows** tab of the `<collection name>` Properties dialog box, choose the New icon.
5. Complete the `<new>` Schedule dialog.
6. Select from the Apply this schedule to drop-down list.
7. Choose **OK** and then close the `<collection name>` **Properties** dialog box.

Step 2: Review computer restart device settings

If you're not suppressing computer restarts and the feature update will be installed when no users are present, consider deploying a custom client settings policy to your feature update target collection to shorten the settings below or consider the total duration of these settings when defining your maintenance window duration.

For example, by default, 90 minutes will be honored before the system is rebooted after the feature update install. If users will not be impacted by the user logoff or restart, there is no need to wait a full 90 minutes before rebooting the computer. If a delay and notification is needed, ensure that the maintenance window takes this into account along with the total time needed to install the feature update.

NOTE

The following settings must be shorter in duration than the shortest maintenance window applied to the computer.

- **Display a temporary notification to the user that indicates the interval before the user is logged off or the computer restarts (minutes).**
- **Display a dialog box that the user cannot close, which displays the countdown interval before the user is logged off or the computer restarts (minutes).**

Step 3: Enable Peer Cache

Use **Peer Cache** to help manage deployment of content to clients in remote locations. Peer Cache is a built-in Configuration Manager solution that enables clients to share content with other clients directly from their local cache.

[Enable Configuration Manager client in full OS to share content](#) if you have clients in remote locations that would benefit from downloading feature update content from a peer instead of downloading it from a distribution point (or Microsoft Update).

Step 4: Override the default Windows setup priority (Windows 10, version 1709 and later)

If you're deploying **Feature update to Windows 10, version 1709** or later, by default, portions of setup are configured to run at a lower priority. This can result in a longer total install time for the feature update. When deploying within a maintenance window, we recommend that you override this default behavior to benefit from

faster total install times. To override the default priority, create a file called SetupConfig.ini on each machine to be upgraded in the below location containing the single section noted.

%systemdrive%\Users\Default\AppData\Local\Microsoft\Windows\WSUS\SetupConfig.ini

```
[SetupConfig]
Priority=Normal
```

You can use the new [Run Scripts](#) feature to run a PowerShell script like the sample below to create the SetupConfig.ini on target devices.

```
#Parameters
Param(
    [string] $PriorityValue = "Normal"
)

#Variable for ini file path
$iniFilePath = "$env:SystemDrive\Users\Default\AppData\Local\Microsoft\Windows\WSUS\SetupConfig.ini"

#Variables for SetupConfig
$iniSetupConfigSlogan = "[SetupConfig]"
$iniSetupConfigKeyValuePair = @{"Priority"=$PriorityValue;}

#Init SetupConfig content
$iniSetupConfigContent = @"
$iniSetupConfigSlogan
"@

#Build SetupConfig content with settings
foreach ($k in $iniSetupConfigKeyValuePair.Keys)
{
    $val = $iniSetupConfigKeyValuePair[$k]

    $iniSetupConfigContent = $iniSetupConfigContent.Insert($iniSetupConfigContent.Length, "`r`n$k=$val")
}

#Write content to file
New-Item $iniFilePath -ItemType File -Value $iniSetupConfigContent -Force

Disclaimer
Sample scripts are not supported under any Microsoft standard support program or service. The sample scripts
is
provided AS IS without warranty of any kind. Microsoft further disclaims all implied warranties including,
without
limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk
arising out of the use or performance of the sample script and documentation remains with you. In no event
shall
Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the scripts be
liable
for any damages whatsoever (including, without limitation, damages for loss of business profits, business
interruption,
loss of business information, or other pecuniary loss) arising out of the use of or inability to use the
sample script
or documentation, even if Microsoft has been advised of the possibility of such damages.
```

NOTE

If you elect not to override the default setup priority, you will need to increase the [maximum run time](#) value for Feature Update to Windows 10, version 1709 or higher from the default of 60 minutes. A value of 240 minutes may be required. Remember to ensure that your maintenance window duration is larger than your defined maximum run time value.

Manually deploy feature updates

The following sections provide the steps to manually deploy a feature update.

Step 1: Specify search criteria for feature updates

There are potentially a thousand or more feature updates displayed in the Configuration Manager console. The first step in the workflow for manually deploying feature updates is to identify the feature updates that you want to deploy.

1. In the Configuration Manager console, click **Software Library**.
2. In the Software Library workspace, expand **Windows 10 Servicing**, and click **All Windows 10 Updates**. The synchronized feature updates are displayed.
3. In the search pane, filter to identify the feature updates that you need by using one or both of the following steps:
 - In the search text box, type a search string that will filter the feature updates. For example, type the version number for a specific feature update, or enter a string that would appear in the title of the feature update.
 - Click **Add Criteria**, select the criteria that you want to use to filter software updates, click **Add**, and then provide the values for the criteria. For example, Title contains 1803, Required is greater than or equal to 1, and Language equals English.
4. Save the search for future use.

Step 2: Download the content for the feature update(s)

Before you deploy the feature updates, you can download the content as a separate step. Do this so you can verify that the content is available on the distribution points before you deploy the feature updates. This will help you to avoid any unexpected issues with the content delivery. Use the following procedure to download the content for feature updates before creating the deployment.

1. In the Configuration Manager console, navigate to **Software Library > Windows 10 Servicing**.
2. Choose the feature update(s) to download by using your saved search criteria. Select one or more of the feature updates returned, right click, and select Download.

The **Download Software Updates Wizard** opens.

3. On the **Deployment Package** page, configure the following settings: **Create a new deployment package**: Select this setting to create a new deployment package for the software updates that are in the deployment. Configure the following settings:
 - **Name**: Specifies the name of the deployment package. The package must have a unique name that briefly describes the package content. It is limited to 50 characters.
 - **Description**: Specifies the description of the deployment package. The package description provides information about the package contents and is limited to 127 characters.
 - **Package source**: Specifies the location of the feature update source files. Type a network path for the source location, for example, \server\sharename\path, or click **Browse** to find the network location. You must create the shared folder for the deployment package source files before you proceed to the next page.

NOTE

The deployment package source location that you specify cannot be used by another software deployment package.

IMPORTANT

The SMS Provider computer account and the user that is running the wizard to download the feature updates must both have Write NTFS permissions on the download location. You should carefully restrict access to the download location to reduce the risk of attackers tampering with the feature update source files.

IMPORTANT

You can change the package source location in the deployment package properties after Configuration Manager creates the deployment package. But if you do so, you must first copy the content from the original package source to the new package source location.

Click **Next**.

4. On the **Distribution Points** page, specify the distribution points or distribution point groups that will host the feature update files, and then click **Next**. For more information about distribution points, see [Distribution point configurations](#).

NOTE

The Distribution Points page is available only when you create a new software update deployment package.

5. On the **Distribution Settings** page, specify the following settings:
 - **Distribution priority:** Use this setting to specify the distribution priority for the deployment package. The distribution priority applies when the deployment package is sent to distribution points at child sites. Deployment packages are sent in priority order: High, Medium, or Low. Packages with identical priorities are sent in the order in which they were created. If there is no backlog, the package will process immediately regardless of its priority. By default, packages are sent using Medium priority.
 - **Enable for on-demand distribution:** Use this setting to enable on-demand content distribution to preferred distribution points. When this setting is enabled, the management point creates a trigger for the distribution manager to distribute the content to all preferred distribution points when a client requests the content for the package and the content is not available on any preferred distribution points. For more information about preferred distribution points and on-demand content, see [Content source location scenarios](#).
 - **Prestaged distribution point settings:** Use this setting to specify how you want to distribute content to prestaged distribution points. Choose one of the following options:
 - **Automatically download content when packages are assigned to distribution points:** Use this setting to ignore the prestage settings and distribute content to the distribution point.
 - **Download only content changes to the distribution point:** Use this setting to prestage the initial content to the distribution point, and then distribute content changes to the distribution point.
 - **Manually copy the content in this package to the distribution point:** Use this setting to always prestage content on the distribution point. This is the default setting.

For more information about prestaging content to distribution points, see [Use Prestaged content](#).

Click **Next**.

6. On the **Download Location** page, specify location that Configuration Manager will use to download the software update source files. As needed, use the following options:

- **Download software updates from the Internet:** Select this setting to download the software updates from the location on the Internet. This is the default setting.
- **Download software updates from a location on the local network:** Select this setting to download software updates from a local folder or shared network folder. Use this setting when the computer running the wizard does not have Internet access.

NOTE

When you use this setting, download the software updates from any computer with Internet access, and then copy the software updates to a location on the local network that is accessible from the computer running the wizard.

Click **Next**.

7. On the **Language Selection** page, specify the languages for which the selected feature updates are to be downloaded, and then click **Next**. Ensure that your language selection matches the language(s) of the feature updates selected for download. For example, if you selected English and German based feature updates for download, select those same languages on the language selection page.
8. On the **Summary** page, verify the settings that you selected in the wizard, and then click Next to download the software updates.
9. On the **Completion** page, verify that the software updates were successfully downloaded, and then click Close.

To monitor content status

1. To monitor the content status for the feature updates, click **Monitoring** in the Configuration Manager console.
2. In the Monitoring workspace, expand **Distribution Status**, and then click **Content Status**.
3. Select the feature update package that you previously identified to download the feature updates.
4. On the **Home** tab, in the Content group, click **View Status**.

Step 3: Deploy the feature update(s)

After you determine which feature updates you intend to deploy, you can manually deploy the feature update(s). Use the following procedure to manually deploy the feature update(s).

1. In the Configuration Manager console, click **Software Library**.
2. In the Software Library workspace, expand **Windows 10 Servicing**, and click **All Windows 10 Updates**.
3. Choose the feature update(s) to deploy by using your saved search criteria. Select one or more of the feature updates returned, right click, and select **Deploy**.

The **Deploy Software Updates Wizard** opens.

4. On the General page, configure the following settings:
 - **Name:** Specify the name for the deployment. The deployment must have a unique name that describes the purpose of the deployment and differentiates it from other deployments in the Configuration Manager site. By default, Configuration Manager automatically provides a name for the deployment in the following format: **Microsoft Software Updates - <date><time>**
 - **Description:** Specify a description for the deployment. The description provides an overview of the deployment and any other relevant information that helps to identify and differentiate the deployment among others in Configuration Manager site. The description field is optional, has a limit of 256 characters, and has a blank value by default.
 - **Software Update/Software Update Group:** Verify that the displayed software update group, or

software update, is correct.

- **Select Deployment Template:** Specify whether to apply a previously saved deployment template. You can configure a deployment template to contain multiple common software update deployment properties and then apply the template when you deploy subsequent software updates to ensure consistency across similar deployments and to save time.
- **Collection:** Specify the collection for the deployment, as applicable. Members of the collection receive the feature updates that are defined in the deployment.

5. On the Deployment Settings page, configure the following settings:

- **Type of deployment:** Specify the deployment type for the software update deployment. Select **Required** to create a mandatory software update deployment in which the feature updates are automatically installed on clients before a configured installation deadline.

IMPORTANT

After you create the software update deployment, you cannot later change the type of deployment.

NOTE

A software update group deployed as Required will be downloaded in background and honor BITS settings, if configured.

- **Use Wake-on-LAN to wake up clients for required deployments:** Specify whether to enable Wake On LAN at the deadline to send wake-up packets to computers that require one or more software updates in the deployment. Any computers that are in sleep mode at the installation deadline time will be awakened so the software update installation can initiate. Clients that are in sleep mode that do not require any software updates in the deployment are not started. By default, this setting is not enabled and is available only when Type of deployment is set to Required.

WARNING

Before you can use this option, computers and networks must be configured for Wake On LAN.

- **Detail level:** Specify the level of detail for the state messages that are reported by client computers.

6. On the Scheduling page, configure the following settings:

- **Schedule evaluation:** Specify whether the available time and installation deadline times are evaluated according to UTC or the local time of the computer running the Configuration Manager console.

NOTE

When you select local time, and then select **As soon as possible** for the **Software available time** or **Installation deadline**, the current time on the computer running the Configuration Manager console is used to evaluate when updates are available or when they are installed on a client. If the client is in a different time zone, these actions will occur when the client's time reaches the evaluation time.

- **Software available time:** Select **As soon as possible** to specify when the software updates will be available to clients:
 - **As soon as possible:** Select this setting to make the software updates in the deployment available to clients as soon as possible. When the deployment is created, the client policy is

updated, the clients are made aware of the deployment at their next client policy polling cycle, and then the software updates are available for installation.

- **Installation deadline:** Select **Specific time** to specify the installation deadline for the software updates in the deployment.

NOTE

You can configure the installation deadline setting only when **Type of deployment** is set to **Required** on the Deployment Settings page.

- **Specific time:** Select this setting to automatically install the software updates in the deployment at a specific date and time. Set the date and time value to correspond with your defined maintenance window for the target collection. Allow sufficient time for clients to download the content in advance of the deadline. Adjust accordingly if clients in your environment will need additional download time. E.g., slow or unreliable network links.

NOTE

The actual installation deadline time is the specific time that you configure plus a random amount of time up to 2 hours. This reduces the potential impact of all client computers in the destination collection installing the software updates in the deployment at the same time. Configure the Computer Agent client setting, **Disable deadline randomization** to disable the installation randomization delay for the required software updates to allow a greater chance for the installation to start and complete within your defined maintenance window. For more information, see [Computer Agent](#).

7. On the User Experience page, configure the following settings:

- **User notifications:** Specify whether to display notification of the software updates in Software Center on the client computer at the configured **Software available time** and whether to display user notifications on the client computers. When **Type of deployment** is set to **Available** on the Deployment Settings page, you cannot select **Hide in Software Center and all notifications**.
- **Deadline behavior:** Available only when **Type of deployment** is set to **Required** on the Deployment Settings page. Specify the behavior that is to occur when the deadline is reached for the software update deployment. Specify whether to install the software updates in the deployment. Also specify whether to perform a system restart after software update installation regardless of a configured maintenance window. For more information about maintenance windows, see [How to use maintenance windows](#).
- **Device restart behavior:** Available only when **Type of deployment** is set to **Required** on the Deployment Settings page. Specify whether to suppress a system restart on servers and workstations after software updates are installed and a system restart is required to complete the installation.

IMPORTANT

Suppressing system restarts can be useful in server environments or for cases in which you do not want the computers that are installing the software updates to restart by default. However, doing so can leave computers in an insecure state, whereas allowing a forced restart helps to ensure immediate completion of the software update installation.

- **Write filter handling for Windows Embedded devices:** When you deploy software updates to Windows Embedded devices that are write filter enabled, you can specify to install the software

update on the temporary overlay and either commit changes later or commit the changes at the installation deadline or during a maintenance window. When you commit changes at the installation deadline or during a maintenance window, a restart is required and the changes persist on the device.

NOTE

When you deploy a software update to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window.

- **Software updates deployment re-evaluation behavior upon restart:** Starting in Configuration Manager version 1606, select this setting to configure software updates deployments to have clients run a software updates compliance scan immediately after a client installs software updates and restarts. This enables the client to check for additional software updates that become applicable after the client restarts, and to then install them (and become compliant) during the same maintenance window.

8. On the Alerts page, configure how Configuration Manager and System Center Operations Manager will generate alerts for this deployment. You can configure alerts only when **Type of deployment** is set to **Required** on the Deployment Settings page.

NOTE

You can review recent software updates alerts from the Software Updates node in the Software Library workspace.

9. On the Download Settings page, configure the following settings:

- Specify whether the client will download and install the software updates when a client is connected to a slow network or is using a fallback content location.
- Specify whether to have the client download and install the software updates from a fallback distribution point when the content for the software updates is not available on a preferred distribution point.
- **Allow clients to share content with other clients on the same subnet:** Specify whether to enable the use of BranchCache for content downloads. For more information about BranchCache, see [Fundamental concepts for content management](#).
- **If software updates are not available on distribution point in current, neighbor or site groups, download content from Microsoft Updates:** Select this setting to have clients that are connected to the intranet download software updates from Microsoft Update if software updates are not available on distribution points. Internet-based clients can always go to Microsoft Update for software updates content.
- Specify whether to allow clients to download after an installation deadline when they use metered Internet connections. Internet providers sometimes charge by the amount of data that you send and receive when you are on a metered Internet connection.

NOTE

Clients request the content location from a management point for the software updates in a deployment. The download behavior depends upon how you have configured the distribution point, the deployment package, and the settings on this page. For more information, see [Content source location scenarios](#).

10. On the Summary page, review the settings. To save the settings to a deployment template, click **Save As**

Template, enter a name and select the settings that you want to include in the template, and then click **Save**. To change a configured setting, click the associated wizard page and change the setting.

11. Click **Next** to deploy the feature update(s).

Step 4: Monitor the deployment status

After you deploy the feature update(s), you can monitor the deployment status. Use the following procedure to monitor the deployment status:

1. In the Configuration Manager console, navigate to **Monitoring > Overview > Deployments**.
2. Click the software update group or software update for which you want to monitor the deployment status.
3. On the **Home** tab, in the **Deployment** group, click **View Status**.

Deploy feature updates for user-initiated installations (during a fixed service window)

6/10/2019 • 16 minutes to read • [Edit Online](#)

Applies to: Windows 10

Use the following steps to deploy a feature update for a user-initiated installation.

Get ready to deploy feature updates

Step 1: Enable Peer Cache

Use **Peer Cache** to help manage deployment of content to clients in remote locations. Peer Cache is a built-in Configuration Manager solution that enables clients to share content with other clients directly from their local cache.

[Enable Configuration Manager client in full OS to share content](#) if you have clients in remote locations that would benefit from downloading feature update content from a peer instead of downloading it from a distribution point (or Microsoft Update).

Step 2: Override the default Windows setup priority (Windows 10, version 1709 and later)

If you're deploying **Feature update to Windows 10, version 1709** or later, by default, portions of setup are configured to run at a lower priority. This can result in a longer total install time for the feature update. When deploying within a maintenance window, we recommend that you override this default behavior to benefit from faster total install times. To override the default priority, create a file called SetupConfig.ini on each machine to be upgraded in the below location containing the single section noted.

%systemdrive%\Users\Default\AppData\Local\Microsoft\Windows\WSUS\SetupConfig.ini

```
[SetupConfig]
Priority=Normal
```

You can use the new [Run Scripts](#) feature to run a PowerShell script like the sample below to create the SetupConfig.ini on target devices.

```

#Parameters
Param(
    [string] $PriorityValue = "Normal"
)

#Variable for ini file path
$iniFilePath = "$env:SystemDrive\Users\Default\AppData\Local\Microsoft\Windows\WSUS\SetupConfig.ini"

#Variables for SetupConfig
$iniSetupConfigSlogan = "[SetupConfig]"
$iniSetupConfigKeyValuePair = @{"Priority"=$PriorityValue;}

#Init SetupConfig content
$iniSetupConfigContent = @"
$iniSetupConfigSlogan
"@

#Build SetupConfig content with settings
foreach ($k in $iniSetupConfigKeyValuePair.Keys)
{
    $val = $iniSetupConfigKeyValuePair[$k]

    $iniSetupConfigContent = $iniSetupConfigContent.Insert($iniSetupConfigContent.Length, "`r`n$k=$val")
}

#Write content to file
New-Item $iniFilePath -ItemType File -Value $iniSetupConfigContent -Force

Disclaimer
Sample scripts are not supported under any Microsoft standard support program or service. The sample scripts
is
provided AS IS without warranty of any kind. Microsoft further disclaims all implied warranties including,
without
limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk
arising out of the use or performance of the sample script and documentation remains with you. In no event
shall
Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the scripts be
liable
for any damages whatsoever (including, without limitation, damages for loss of business profits, business
interruption,
loss of business information, or other pecuniary loss) arising out of the use of or inability to use the
sample script
or documentation, even if Microsoft has been advised of the possibility of such damages.

```

NOTE

If you elect not to override the default setup priority, you will need to increase the [maximum run time](#) value for Feature Update to Windows 10, version 1709 or higher from the default of 60 minutes. A value of 240 minutes may be required. Remember to ensure that your maintenance window duration is larger than your defined maximum run time value.

Manually deploy feature updates in a user-initiated installation

The following sections provide the steps to manually deploy a feature update.

Step 1: Specify search criteria for feature updates

There are potentially a thousand or more feature updates displayed in the Configuration Manager console. The first step in the workflow for manually deploying a feature update is to identify the feature updates that you want to deploy.

1. In the Configuration Manager console, click **Software Library**.

2. In the Software Library workspace, expand **Windows 10 Servicing**, and click **All Windows 10 Updates**. The synchronized feature updates are displayed.
3. In the search pane, filter to identify the feature updates that you need by using one or both of the following steps:
 - In the **search** text box, type a search string that will filter the feature updates. For example, type the version number for a specific feature update, or enter a string that would appear in the title of the feature update.
 - Click **Add Criteria**, select the criteria that you want to use to filter software updates, click **Add**, and then provide the values for the criteria. For example, Title contains 1803, **Required** is greater than or equal to 1, and **Language** equals English.
4. Save the search for future use.

Step 2: Download the content for the feature update(s)

Before you deploy the feature updates, you can download the content as a separate step. Do this so you can verify that the content is available on the distribution points before you deploy the feature updates. This will help you to avoid any unexpected issues with the content delivery. Use the following procedure to download the content for feature updates before creating the deployment.

1. In the Configuration Manager console, navigate to **Software Library > Windows 10 Servicing**.
2. Choose the feature update(s) to download by using your saved search criteria. Select one or more of the feature updates returned, right click, and select **Download**.

The **Download Software Updates Wizard** opens.

3. On the **Deployment Package** page, configure the following settings: **Create a new deployment package**: Select this setting to create a new deployment package for the software updates that are in the deployment. Configure the following settings:
 - **Name**: Specifies the name of the deployment package. The package must have a unique name that briefly describes the package content. It is limited to 50 characters.
 - **Description**: Specifies the description of the deployment package. The package description provides information about the package contents and is limited to 127 characters.
 - **Package source**: Specifies the location of the feature update source files. Type a network path for the source location, for example, \\server\sharename\path, or click **Browse** to find the network location. You must create the shared folder for the deployment package source files before you proceed to the next page.

NOTE

The deployment package source location that you specify cannot be used by another software deployment package.

IMPORTANT

The SMS Provider computer account and the user that is running the wizard to download the feature updates must both have Write NTFS permissions on the download location. You should carefully restrict access to the download location to reduce the risk of attackers tampering with the feature update source files.

IMPORTANT

You can change the package source location in the deployment package properties after Configuration Manager creates the deployment package. But if you do so, you must first copy the content from the original package source to the new package source location.

Click **Next**.

4. On the **Distribution Points** page, specify the distribution points or distribution point groups that will host the feature update files, and then click **Next**. For more information about distribution points, see [Distribution point configurations](#).

NOTE

The Distribution Points page is available only when you create a new software update deployment package.

5. On the **Distribution Settings** page, specify the following settings:
 - **Distribution priority:** Use this setting to specify the distribution priority for the deployment package. The distribution priority applies when the deployment package is sent to distribution points at child sites. Deployment packages are sent in priority order: **High**, **Medium**, or **Low**. Packages with identical priorities are sent in the order in which they were created. If there is no backlog, the package will process immediately regardless of its priority. By default, packages are sent using Medium priority.
 - **Enable for on-demand distribution:** Use this setting to enable on-demand content distribution to preferred distribution points. When this setting is enabled, the management point creates a trigger for the distribution manager to distribute the content to all preferred distribution points when a client requests the content for the package and the content is not available on any preferred distribution points. For more information about preferred distribution points and on-demand content, see [Content source location scenarios](#).
 - **Prestaged distribution point settings:** Use this setting to specify how you want to distribute content to prestaged distribution points. Choose one of the following options:
 - **Automatically download content when packages are assigned to distribution points:** Use this setting to ignore the prestige settings and distribute content to the distribution point.
 - **Download only content changes to the distribution point:** Use this setting to prestage the initial content to the distribution point, and then distribute content changes to the distribution point.
 - **Manually copy the content in this package to the distribution point:** Use this setting to always prestage content on the distribution point. This is the default setting.

For more information about prestaging content to distribution points, see [Use Prestaged content](#).

Click **Next**.

6. On the **Download Location** page, specify location that Configuration Manager will use to download the software update source files. As needed, use the following options:
 - **Download software updates from the Internet:** Select this setting to download the software updates from the location on the Internet. This is the default setting.
 - **Download software updates from a location on the local network:** Select this setting to download software updates from a local folder or shared network folder. Use this setting when the computer running the wizard does not have Internet access.

NOTE

When you use this setting, download the software updates from any computer with Internet access, and then copy the software updates to a location on the local network that is accessible from the computer running the wizard.

Click **Next**.

7. On the **Language Selection** page, specify the languages for which the selected feature updates are to be downloaded, and then click **Next**. Ensure that your language selection matches the language(s) of the feature updates selected for download. For example, if you selected English and German based feature updates for download, select those same languages on the language selection page.
8. On the **Summary** page, verify the settings that you selected in the wizard, and then click Next to download the software updates.
9. On the **Completion** page, verify that the software updates were successfully downloaded, and then click **Close**.

To monitor content status

1. To monitor the content status for the feature updates, click **Monitoring** in the Configuration Manager console.
2. In the Monitoring workspace, expand **Distribution Status**, and then click **Content Status**.
3. Select the feature update package that you previously identified to download the feature updates.
4. On the **Home** tab, in the Content group, click **View Status**.

Step 3: Deploy the feature update(s)

After you determine which feature updates you intend to deploy, you can manually deploy the feature update(s). Use the following procedure to manually deploy the feature update(s).

1. In the Configuration Manager console, click **Software Library**.
2. In the Software Library workspace, expand **Windows 10 Servicing**, and click **All Windows 10 Updates**.
3. Choose the feature update(s) to deploy by using your saved search criteria. Select one or more of the feature updates returned, right click, and select **Deploy**.

The **Deploy Software Updates Wizard** opens.

4. On the General page, configure the following settings:
 - **Name:** Specify the name for the deployment. The deployment must have a unique name that describes the purpose of the deployment and differentiates it from other deployments in the Configuration Manager site. By default, Configuration Manager automatically provides a name for the deployment in the following format: **Microsoft Software Updates - <date><time>**
 - **Description:** Specify a description for the deployment. The description provides an overview of the deployment and any other relevant information that helps to identify and differentiate the deployment among others in Configuration Manager site. The description field is optional, has a limit of 256 characters, and has a blank value by default.
 - **Software Update/Software Update Group:** Verify that the displayed software update group, or software update, is correct.
 - **Select Deployment Template:** Specify whether to apply a previously saved deployment template. You can configure a deployment template to contain multiple common software update deployment properties and then apply the template when you deploy subsequent software updates to ensure consistency across similar deployments and to save time.
 - **Collection:** Specify the collection for the deployment, as applicable. Members of the collection receive

the feature updates that are defined in the deployment.

5. On the Deployment Settings page, configure the following settings:

- **Type of deployment:** Specify the deployment type for the software update deployment. Select **Required** to create a mandatory software update deployment in which the feature updates are automatically installed on clients before a configured installation deadline.

IMPORTANT

After you create the software update deployment, you cannot later change the type of deployment.

NOTE

A software update group deployed as **Required** will be downloaded in background and honor BITS settings, if configured.

- **Use Wake-on-LAN to wake up clients for required deployments:** Specify whether to enable Wake On LAN at the deadline to send wake-up packets to computers that require one or more software updates in the deployment. Any computers that are in sleep mode at the installation deadline time will be awakened so the software update installation can initiate. Clients that are in sleep mode that do not require any software updates in the deployment are not started. By default, this setting is not enabled and is available only when **Type of deployment** is set to **Required**.

WARNING

Before you can use this option, computers and networks must be configured for Wake On LAN.

- **Detail level:** Specify the level of detail for the state messages that are reported by client computers.

6. On the Scheduling page, configure the following settings:

- **Schedule evaluation:** Specify whether the available time and installation deadline times are evaluated according to UTC or the local time of the computer running the Configuration Manager console.
- **Software available time:** Select **Specific time** to specify when the software updates will be available to clients:
 - **Specific time:** Select this setting to make the feature update in the deployment available to clients at a specific date and time. Specify a date and time that corresponds with the start of your fixed servicing window. When the deployment is created, the client policy is updated and clients are made aware of the deployment at their next client policy polling cycle. However, the feature update in the deployment is not available for installation until after the specified date and time are reached and the required content has been downloaded.
- **Installation deadline:** Select **Specific time** to specify the installation deadline for the software updates in the deployment.

NOTE

You can configure the installation deadline setting only when **Type of deployment** is set to **Required** on the Deployment Settings page.

- **Specific time:** Select this setting to automatically install the software updates in the deployment

at a specific date and time. However, for the purposes of the fixed servicing window, set the installation deadline date and time to a future value, well beyond the fixed servicing window.

Required deployments for software updates can benefit from functionality called advanced download. When the software available time is reached, clients will start downloading the content based on a randomized time. The feature update will not be displayed in Software Center for installation until the content is fully downloaded. This ensures that the feature update installation will start immediately when initiated.

7. On the User Experience page, configure the following settings:

- **User notifications:** Specify **Display in Software Center and show all notifications**.
- **Deadline behavior:** Available only when **Type of deployment** is set to **Required** on the Deployment Settings page. Specify the behavior that is to occur when the deadline is reached for the software update deployment. Specify whether to install the software updates in the deployment. Also specify whether to perform a system restart after software update installation regardless of a configured maintenance window.

NOTE

Remember that the installation deadline date and time will be well into the future to allow plenty of time for the user-initiated install during a fixed servicing window.

- **Device restart behavior:** Available only when **Type of deployment** is set to **Required** on the Deployment Settings page. Specify whether to suppress a system restart on servers and workstations after software updates are installed and a system restart is required to complete the installation.

IMPORTANT

Suppressing system restarts can be useful in server environments or for cases in which you do not want the computers that are installing the software updates to restart by default. However, doing so can leave computers in an insecure state, whereas allowing a forced restart helps to ensure immediate completion of the software update installation.

- **Write filter handling for Windows Embedded devices:** When you deploy software updates to Windows Embedded devices that are write filter enabled, you can specify to install the software update on the temporary overlay and either commit changes later or commit the changes at the installation deadline or during a maintenance window. When you commit changes at the installation deadline or during a maintenance window, a restart is required and the changes persist on the device.

NOTE

When you deploy a software update to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window.

- **Software updates deployment re-evaluation behavior upon restart:** Starting in Configuration Manager version 1606, select this setting to configure software updates deployments to have clients run a software updates compliance scan immediately after a client installs software updates and restarts. This enables the client to check for additional software updates that become applicable after the client restarts, and to then install them (and become compliant) during the same maintenance window.

8. On the Alerts page, configure how Configuration Manager and System Center Operations Manager will

generate alerts for this deployment. You can configure alerts only when **Type of deployment** is set to **Required** on the Deployment Settings page.

NOTE

You can review recent software updates alerts from the **Software Updates** node in the **Software Library** workspace.

9. On the Download Settings page, configure the following settings:

- Specify whether the client will download and install the software updates when a client is connected to a slow network or is using a fallback content location.
- Specify whether to have the client download and install the software updates from a fallback distribution point when the content for the software updates is not available on a preferred distribution point.
- **Allow clients to share content with other clients on the same subnet:** Specify whether to enable the use of BranchCache for content downloads. For more information about BranchCache, see [Fundamental concepts for content management](#).
- **If software updates are not available on distribution point in current, neighbor or site groups, download content from Microsoft Updates:** Select this setting to have clients that are connected to the intranet download software updates from Microsoft Update if software updates are not available on distribution points. Internet-based clients can always go to Microsoft Update for software updates content.
- Specify whether to allow clients to download after an installation deadline when they use metered Internet connections. Internet providers sometimes charge by the amount of data that you send and receive when you are on a metered Internet connection.

NOTE

Clients request the content location from a management point for the software updates in a deployment. The download behavior depends upon how you have configured the distribution point, the deployment package, and the settings on this page. For more information, see [Content source location scenarios](#).

10. On the Summary page, review the settings. To save the settings to a deployment template, click **Save As Template**, enter a name and select the settings that you want to include in the template, and then click **Save**. To change a configured setting, click the associated wizard page and change the setting.

11. Click **Next** to deploy the feature update(s).

Step 4: Monitor the deployment status

After you deploy the feature update(s), you can monitor the deployment status. Use the following procedure to monitor the deployment status:

1. In the Configuration Manager console, navigate to **Monitoring > Overview > Deployments**.
2. Click the software update group or software update for which you want to monitor the deployment status.
3. On the **Home** tab, in the **Deployment** group, click **View Status**.

Conclusion

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 10

Mission critical devices that need to be online 24x7 pose unique challenges for the IT Pro looking to stay current with the latest Windows 10 feature update. Because these devices are online continually, providing mission critical services, with only a small window of time available to apply feature updates, specific procedures are required to effectively keep these devices current, with as little downtime as possible.

Whether you have defined servicing windows at your disposal where feature updates can be installed automatically, or you require user initiated installs by a technician, this whitepaper provides guidelines for either approach. Improvements are continually being made to Windows 10 setup to reduce device offline time for feature updates. This whitepaper will be updated as enhancements become available to improve the overall servicing approach and experience.

Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to

- Windows 10 Mobile
- [Windows 10 IoT Mobile](#)

Looking for consumer information? See [Windows Update: FAQ](#)

TIP

If you're not familiar with the Windows 10 servicing or release channels, read [Servicing channels](#) first.

Devices running Windows 10 Mobile and Windows 10 IoT Mobile receive updates from the Semi-annual channel unless you [enroll the device in the Windows Insider Program](#) or assign the device to Current Branch for Business (CBB). Only devices running Windows 10 Mobile Enterprise or Windows 10 IoT Mobile can be assigned to CBB.

[Learn how to upgrade Windows 10 Mobile to Windows 10 Mobile Enterprise](#)

IMPORTANT

Due to [naming changes](#), older terms like CB,CBB and LTSB may still be displayed in some of our products.

In the following settings CB refers to Semi-Annual Channel (Targeted), while CBB refers to Semi-Annual Channel.

WINDOWS 10 EDITION	CB	CBB	INSIDER PROGRAM
Mobile	✓	✗	✓
Mobile Enterprise	✓	✓	✓
IoT Mobile	✓	✓	✓

Configuration of Windows 10 Mobile and Windows 10 IoT Mobile devices is limited to the feature set pertaining to Quality Updates only. That is, Windows Mobile Feature Updates are categorized the same as Quality Updates, and can only be deferred by setting the Quality Update deferral period, for a maximum period of 30 days. You can use mobile device management (MDM) to manage updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile. Updates cannot be managed for Windows 10 Mobile.

Windows 10, version 1511

Only the following Windows Update for Business policies are supported for Windows 10 Mobile and Windows 10 IoT Mobile:

- ../Vendor/MSFT/Policy/Config/Update/RequireDeferredUpgrade
- ../Vendor/MSFT/Policy/Config/Update/DeferUpdatePeriod
- ../Vendor/MSFT/Policy/Config/Update/PauseDeferrals

To defer the update period or pause deferrals, the device must be configured for CBB servicing branch by applying the **RequireDeferredUpgrade** policy.

Windows 10, version 1607

Only the following Windows Update for Business policies are supported for Windows 10 Mobile and Windows 10 IoT Mobile:

- ../Vendor/MSFT/Policy/Config/Update/BranchReadinessLevel
- ../Vendor/MSFT/Policy/Config/Update/DeferQualityUpdatesInDays
- ../Vendor/MSFT/Policy/Config/Update/PauseQualityUpdates

In version 1607, you can defer and pause updates for devices on both the CB and CBB servicing branches.

If a device running Windows 10 Mobile Enterprise or Windows 10 IoT Mobile, version 1511, has Windows Update for Business policies applied and is then updated to version 1607, version 1511 policies continue to apply until version 1607 policies are applied.

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)
- [Manage device restarts after updates](#)

Deploy updates using Windows Update for Business

6/7/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows Server 2016
- Windows Server 2019

Windows Update for Business is a free service that is available for Windows Pro, Enterprise, Pro for Workstation, and Education editions.

Looking for consumer information? See [Windows Update: FAQ](#)

Windows Update for Business enables IT administrators to keep the Windows 10 devices in their organization always up to date with the latest security defenses and Windows features by directly connecting these systems to Windows Update service. You can use Group Policy or MDM solutions such as Microsoft Intune to configure the Windows Update for Business settings that control how and when Windows 10 devices are updated.

Specifically, Windows Update for Business allows for control over update offering and experience to allow for reliability and performance testing on a subset of systems before rolling out updates across the organization as well as a positive update experience for those within your organization.

NOTE

To use Windows Update for Business, you must allow devices to access the Windows Update service.

Types of updates managed by Windows Update for Business

Windows Update for Business provides management policies for several types of updates to Windows 10 devices:

- **Feature updates:** previously referred to as upgrades, feature updates contain not only security and quality revisions, but also significant feature additions and changes; they are released semi-annually in the fall and in the spring.
- **Quality updates:** these are traditional operating system updates, typically released the second Tuesday of each month (though they can be released at any time). These include security, critical, and driver updates. Windows Update for Business also treats non-Windows updates (such as those for Microsoft Office or Visual Studio) as quality updates. These non-Windows Updates are known as "Microsoft updates" and can configure devices to receive or not receive such updates along with their Windows updates.
- **Driver updates:** these are non-Microsoft drivers that are applicable to your devices. Driver updates can be turned off by using Windows Update for Business policies.
- **Microsoft product updates:** these are updates for other Microsoft products, such as Office. These updates can be enabled or disabled by using Windows Update for Business policy.

Offering

You can control when updates are applied, for example by deferring when an update is installed on a device or

by pausing updates for a certain period of time.

Manage which updates are offered

Windows Update for Business offers you the ability to turn on or off both driver and Microsoft product updates.

- Drivers (on/off): When "on," this policy will not include drivers with Windows Update.
- Microsoft product updates (on/off): When "on" this policy will install updates for other Microsoft products.

Manage when updates are offered

You can defer or pause the installation of updates for a set period of time.

Defer or pause an update

A Windows Update for Business administrator can defer the installation of both feature and quality updates from deploying to devices within a bounded range of time from when those updates are first made available on the Windows Update service. You can use this deferral to allow time to validate deployments as they are pushed to devices. Deferrals work by allowing you to specify the number of days after an update is released before it is offered to a device (if you set a feature update deferral period of 365 days, the device will not install a feature update that has been released for less than 365 days). To defer feature updates use the **Select when Preview Builds and Feature Updates are Received** policy.

CATEGORY	MAXIMUM DEFERRAL
Feature updates	365 days
Quality updates	30 days
Non-deferrable	none

Pause an update

If you discover a problem while deploying a feature or quality update, the IT administrator can pause the update for 35 days to prevent other devices from installing it until the issue is mitigated.

If you pause a feature update, quality updates are still offered to devices to ensure they stay secure. The pause period for both feature and quality updates is calculated from a start date that you set.

To pause feature updates use the **Select when Preview Builds and Feature Updates are Received** policy and to pause quality updates use the **Select when Quality Updates are Received** policy. For more information, see [Pause feature updates](#) and [Pause quality updates](#).

Select branch readiness level for feature updates

The branch readiness level enables administrators to specify which channel of feature updates they want to receive. Today there are branch readiness level options for both pre-release and released updates:

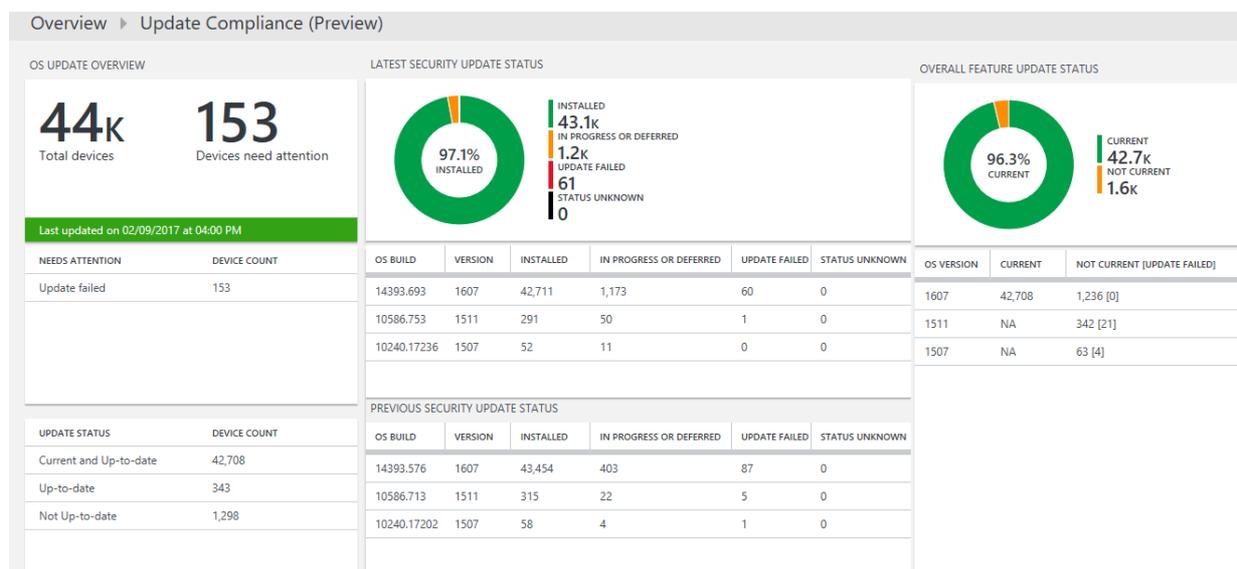
- Windows Insider Program for Business pre-release updates
 - Windows Insider Fast
 - Windows Insider Slow
 - Windows Insider Release Preview
- Semi-annual Channel for released updates

Prior to Windows 10, version 1903, there are two channels for released updates: Semi-annual Channel and Semi-annual Channel (Targeted). Deferral days are calculated against the release date of the chosen channel. Starting with Windows 10, version 1903 there is only the one release channel: Semi-annual Channel. All deferral days will be calculated against a release's Semi-annual Channel release date. To see release dates, visit [Windows Release Information](#). You can set the branch readiness level by using the **Select when Preview**

Builds and Feature Updates are Received policy. In order to use this to manage pre-release builds, first enable preview builds by using the **Manage preview Builds** policy.

Monitor Windows Updates by using Update Compliance

Update Compliance provides a holistic view of operating system update compliance, update deployment progress, and failure troubleshooting for Windows 10 devices. This service uses diagnostic data including installation progress, Windows Update configuration, and other information to provide such insights, at no extra cost and without additional infrastructure requirements. Whether used with Windows Update for Business or other management tools, you can be assured that your devices are properly updated.



For more information about Update Compliance, see [Monitor Windows Updates using Update Compliance](#).

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates
<input checked="" type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input checked="" type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input checked="" type="checkbox"/>	Deploy updates using Windows Update for Business (this topic) or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)

- Overview of Windows as a service
- Prepare servicing strategy for Windows 10 updates
- Build deployment rings for Windows 10 updates
- Assign devices to servicing channels for Windows 10 updates
- Optimize update delivery for Windows 10 updates
- Configure Delivery Optimization for Windows 10 updates
- Configure BranchCache for Windows 10 updates
- Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile
- Configure Windows Update for Business
- Integrate Windows Update for Business with management solutions
- Walkthrough: use Group Policy to configure Windows Update for Business
- Walkthrough: use Intune to configure Windows Update for Business
- Deploy Windows 10 updates using Windows Server Update Services
- Deploy Windows 10 updates using System Center Configuration Manager
- Manage device restarts after updates

Configure Windows Update for Business

5/31/2019 • 13 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile
- Windows Server 2016
- Windows Server 2019

Looking for consumer information? See [Windows Update: FAQ](#)

You can use Group Policy or your mobile device management (MDM) service to configure Windows Update for Business settings for your devices. The sections in this topic provide the Group Policy and MDM policies for Windows 10, version 1511 and above. The MDM policies use the OMA-URI setting from the [Policy CSP](#).

IMPORTANT

For Windows Update for Business policies to be honored, the diagnostic data level of the device must be set to **1 (Basic)** or higher. If it is set to **0 (Security)**, Windows Update for Business policies will have no effect. For instructions, see [Configure the operating system diagnostic data level](#).

Some Windows Update for Business policies are not applicable or behave differently for devices running Windows 10 Mobile Enterprise. Specifically, policies pertaining to Feature Updates will not be applied to Windows 10 Mobile Enterprise. All Windows 10 Mobile updates are recognized as Quality Updates, and can only be deferred or paused using the Quality Update policy settings. Additional information is provided in this topic and in [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#).

Start by grouping devices

By grouping devices with similar deferral periods, administrators are able to cluster devices into deployment or validation groups which can be as a quality control measure as updates are deployed in Windows 10. With deferral windows and the ability to pause updates, administrators can effectively control and measure update deployments, updating a small pool of devices first to verify quality, prior to a broader roll-out to their organization. For more information, see [Build deployment rings for Windows 10 updates](#).

TIP

In addition to setting up multiple rings for your update deployments, also incorporate devices enrolled in the Windows Insider Program as part of your deployment strategy. This will provide you the chance to not only evaluate new features before they are broadly available to the public, but it also increases the lead time to provide feedback and influence Microsoft's design on functional aspects of the product. For more information on Windows Insider program, see <https://insider.windows.com/>.

Configure devices for the appropriate service channel

With Windows Update for Business, you can set a device to be on either Windows Insider Preview or the Semi-Annual Channel servicing branch. For more information on this servicing model, see [Windows 10 servicing options](#).

Release branch policies

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 or later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > Select when Feature Updates are received	\Policies\Microsoft\Windows\WindowsUpdate\BranchReadinessLevel
GPO for Windows 10, version 1511: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Upgrades and Updates	\Policies\Microsoft\Windows\WindowsUpdate\DeferUpgrade
MDM for Windows 10, version 1607 or later: ../Vendor/MSFT/Policy/Config/Update/ BranchReadinessLevel	\Microsoft\PolicyManager\default\Update\BranchReadinessLevel
MDM for Windows 10, version 1511: ../Vendor/MSFT/Policy/Config/Update/ RequireDeferUpgrade	\Microsoft\PolicyManager\default\Update\RequireDeferUpgrade

Starting with Windows 10, version 1703, users can configure the branch readiness level for their device by using **Settings > Update & security > Windows Update > Advanced options**.

Choose when Feature Updates are installed

Choose the branch readiness level to determine when feature updates are installed:

Current Branch ▾

NOTE
Users will not be able to change this setting if it was configured by policy.

Configure when devices receive feature updates

After you configure the servicing branch (Windows Insider Preview or Semi-Annual Channel), you can then define if, and for how long, you would like to defer receiving Feature Updates following their availability from Microsoft on Windows Update. You can defer receiving these Feature Updates for a period of up to 365 days from their release by setting the `DeferFeatureUpdatesPeriodInDays` value.

For example, a device on the Semi-Annual Channel with `DeferFeatureUpdatesPeriodInDays=30` will not install a feature update that is first publicly available on Windows Update in September until 30 days later, in October.

Policy settings for deferring feature updates

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > Select when Feature Updates are received	\Policies\Microsoft\Windows\WindowsUpdate\DeferFeatureUpdates \Policies\Microsoft\Windows\WindowsUpdate\DeferFeatureUpdatesPeriodInDays

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1511: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Upgrades and Updates	\Policies\Microsoft\Windows\WindowsUpdate\DeferUpgradePeriod
MDM for Windows 10, version 1607 and later: ../Vendor/MSFT/Policy/Config/Update/ DeferFeatureUpdatesPeriodInDays	\Microsoft\PolicyManager\default\Update\DeferFeatureUpdatesPeriodInDays
MDM for Windows 10, version 1511: ../Vendor/MSFT/Policy/Config/Update/ DeferUpgrade	\Microsoft\PolicyManager\default\Update\RequireDeferUpgrade

NOTE

If not configured by policy, individual users can defer feature updates by using **Settings > Update & security > Windows Update > Advanced options**.

Pause feature updates

You can also pause a device from receiving Feature Updates by a period of up to 35 days from when the value is set. After 35 days has passed, the pause setting will automatically expire and the device will scan Windows Update for applicable Feature Updates. Following this scan, you can then pause Feature Updates for the device again.

Starting with Windows 10, version 1703, when you configure a pause by using policy, you must set a start date for the pause to begin. The pause period is calculated by adding 35 days to this start date.

In cases where the pause policy is first applied after the configured start date has passed, you can extend the pause period up to a total of 35 days by configuring a later start date.

IMPORTANT

In Windows 10, version 1703 and later versions, you can pause feature updates to 35 days, similar to the number of days for quality updates.

Policy settings for pausing feature updates

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 and later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > Select when Feature Updates are received	1607: \Policies\Microsoft\Windows\WindowsUpdate\PauseFeatureUpdates 1703 and later: \Policies\Microsoft\Windows\WindowsUpdate\PauseFeatureUpdatesStartDate
GPO for Windows 10, version 1511: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Upgrades and Updates	\Policies\Microsoft\Windows\WindowsUpdate\Pause

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
MDM for Windows 10, version 1607 and later: ../Vendor/MSFT/Policy/Config/Update/ PauseFeatureUpdates	1607: \Microsoft\PolicyManager\default\Update\PauseFeatureUpdates 1703 and later: \Microsoft\PolicyManager\default\Update\PauseFeatureUpdatesStartDate
MDM for Windows 10, version 1511: ../Vendor/MSFT/Policy/Config/Update/ DeferUpgrade	\Microsoft\PolicyManager\default\Update\Pause

You can check the date that Feature Updates were paused by checking the registry key **PausedFeatureDate** under **HKLM\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\Settings**.

The local group policy editor (GPEdit.msc) will not reflect whether the Feature Update pause period has expired. Although the device will resume Feature Updates after 35 days automatically, the pause checkbox will remain selected in the policy editor. To check whether a device has automatically resumed taking Feature Updates, check the status registry key **PausedFeatureStatus** under **HKLM\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\Settings** for the following values:

VALUE	STATUS
0	Feature Updates not paused
1	Feature Updates paused
2	Feature Updates have auto-resumed after being paused

NOTE

If not configured by policy, individual users can pause feature updates by using **Settings > Update & security > Windows Update > Advanced options**.

Starting with Windows 10, version 1703, using Settings to control the pause behavior provides a more consistent experience, specifically:

- Any active restart notification are cleared or closed.
- Any pending restarts are canceled.
- Any pending update installations are canceled.
- Any update installation running when pause is activated will attempt to roll back.

Configure when devices receive Quality Updates

Quality Updates are typically published on the first Tuesday of every month, although they can be released at any time. You can define if, and for how long, you would like to defer receiving Quality Updates following their availability. You can defer receiving these Quality Updates for a period of up to 35 days from their release by setting the **DeferQualityUpdatesPeriodinDays** value.

You can set your system to receive updates for other Microsoft products—known as Microsoft Updates (such as Microsoft Office, Visual Studio)—along with Windows Updates by setting the **AllowMUUpdateService** policy. When you do this, these Microsoft Updates will follow the same deferral and pause rules as all other Quality Updates.

IMPORTANT

This policy defers both Feature and Quality Updates on Windows 10 Mobile Enterprise.

Policy settings for deferring quality updates

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 and later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > Select when Quality Updates are received	\Policies\Microsoft\Windows\WindowsUpdate\DeferQuality Updates \Policies\Microsoft\Windows\WindowsUpdate\DeferQuality UpdatesPeriodInDays
GPO for Windows 10, version 1511: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Upgrades and Updates	\Policies\Microsoft\Windows\WindowsUpdate\DeferUpdate Period
MDM for Windows 10, version 1607 and later: ../Vendor/MSFT/Policy/Config/Update/ DeferQualityUpdatesPeriodInDays	\Microsoft\PolicyManager\default\Update\DeferQualityUpdatesPeriodInDays
MDM for Windows 10, version 1511: ../Vendor/MSFT/Policy/Config/Update/ DeferUpgrade	\Microsoft\PolicyManager\default\Update\RequireDeferUpgrade

NOTE

If not configured by policy, individual users can defer quality updates by using **Settings > Update & security > Windows Update > Advanced options**.

Pause quality updates

You can also pause a system from receiving Quality Updates for a period of up to 35 days from when the value is set. After 35 days has passed, the pause setting will automatically expire and the device will scan Windows Update for applicable quality Updates. Following this scan, you can then pause quality Updates for the device again.

Starting with Windows 10, version 1703, when you configure a pause by using policy, you must set a start date for the pause to begin. The pause period is calculated by adding 35 days to this start date.

In cases where the pause policy is first applied after the configured start date has passed, you can extend the pause period up to a total of 35 days by configuring a later start date.

NOTE

Starting with Windows 10, version 1809, IT administrators can prevent individual users from pausing updates.

Policy settings for pausing quality updates

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 and later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Windows Updates > Select when Quality Updates are received	1607: \Policies\Microsoft\Windows\WindowsUpdate\PauseQualityUpdates 1703: \Policies\Microsoft\Windows\WindowsUpdate\PauseQualityUpdatesStartTime
GPO for Windows 10, version 1511: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Defer Upgrades and Updates	\Policies\Microsoft\Windows\WindowsUpdate\Pause
MDM for Windows 10, version 1607 and later: ../Vendor/MSFT/Policy/Config/Update/ PauseQualityUpdates	1607: \Microsoft\PolicyManager\default\Update\PauseQualityUpdates 1703: \Microsoft\PolicyManager\default\Update\PauseQualityUpdatesStartTime
MDM for Windows 10, version 1511: ../Vendor/MSFT/Policy/Config/Update/ DeferUpgrade	\Microsoft\PolicyManager\default\Update\Pause

You can check the date that quality Updates were paused by checking the registry key **PausedQualityDate** under **HKLM\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\Settings**.

The local group policy editor (GPEdit.msc) will not reflect whether the quality Update pause period has expired. Although the device will resume quality Updates after 35 days automatically, the pause checkbox will remain selected in the policy editor. To check whether a device has automatically resumed taking quality Updates, check the status registry key **PausedQualityStatus** under **HKLM\SOFTWARE\Microsoft\WindowsUpdate\UpdatePolicy\Settings** for the following values:

VALUE	STATUS
0	Quality Updates not paused
1	Quality Updates paused
2	Quality Updates have auto-resumed after being paused

NOTE

If not configured by policy, individual users can pause quality updates by using **Settings > Update & security > Windows Update > Advanced options**.

Starting with Windows 10, version 1703, using Settings to control the pause behavior provides a more consistent experience, specifically:

- Any active restart notification are cleared or closed
- Any pending restarts are canceled
- Any pending update installations are canceled
- Any update installation running when pause is activated will attempt to rollback

Configure when devices receive Windows Insider Preview builds

Starting with Windows 10, version 1709, you can set policies to manage preview builds and their delivery:

The **Manage preview builds** setting gives administrators control over enabling or disabling preview build installation on a device. You can also decide to stop preview builds once the release is public.

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Windows Update/Windows Update for Business - Manage preview builds**
- MDM: **Update/ManagePreviewBuilds**
- System Center Configuration Manager: **Enable dual scan, manage through Windows Update for Business policy**

IMPORTANT

This policy replaces the "Toggle user control over Insider builds" policy under that is only supported up to Windows 10, version 1703. You can find the older policy here:

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Data Collection and Preview Builds/Toggle user control over Insider builds**
- MDM: **System/AllowBuildPreview**

The policy settings to **Select when Feature Updates are received** allows you to choose between preview flight rings, and allows you to defer or pause their delivery.

- Group Policy: **Computer Configuration/Administrative Templates/Windows Components/Windows Update/ Windows Update for Business - Select when Preview Builds and Feature Updates are received**
- MDM: **Update/BranchReadinessLevel**

Exclude drivers from Quality Updates

Starting with Windows 10, version 1607, you can selectively opt out of receiving driver update packages as part of your normal quality update cycle. This policy will not apply to updates to drivers provided with the operating system (which will be packaged within a security or critical update) or to Feature Updates, where drivers might be dynamically installed to ensure the Feature Update process can complete.

Policy settings to exclude drivers

POLICY	SETS REGISTRY KEY UNDER HKLM\SOFTWARE
GPO for Windows 10, version 1607 and later: Computer Configuration > Administrative Templates > Windows Components > Windows Update > Do not include drivers with Windows Updates	\Policies\Microsoft\Windows\WindowsUpdate\ExcludeWUDriversInQualityUpdate
MDM for Windows 10, version 1607 and later: ../Vendor/MSFT/Policy/Config/Update/ ExcludeWUDriversInQualityUpdate	\Microsoft\PolicyManager\default\Update\ExcludeWUDriversInQualityUpdate

Summary: MDM and Group Policy settings for Windows 10, version 1703 and later

The following are quick-reference tables of the supported policy values for Windows Update for Business in Windows 10, version 1607 and later.

GPO: HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate

GPO KEY	KEY TYPE	VALUE
BranchReadinessLevel	REG_DWORD	16: systems take Feature Updates for the Current Branch (CB) 32: systems take Feature Updates for the Current Branch for Business (CBB) Note: Other value or absent: receive all applicable updates (CB)
DeferQualityUpdates	REG_DWORD	1: defer quality updates Other value or absent: don't defer quality updates
DeferQualityUpdatesPeriodinDays	REG_DWORD	0-35: defer quality updates by given days
PauseQualityUpdatesStartDate	REG_DWORD	1: pause quality updates Other value or absent: don't pause quality updates
DeferFeatureUpdates	REG_DWORD	1: defer feature updates Other value or absent: don't defer feature updates
DeferFeatureUpdatesPeriodinDays	REG_DWORD	0-365: defer feature updates by given days
PauseFeatureUpdatesStartDate	REG_DWORD	1: pause feature updates Other value or absent: don't pause feature updates
ExcludeWUDriversInQualityUpdate	REG_DWORD	1: exclude Windows Update drivers Other value or absent: offer Windows Update drivers

MDM: HKEY_LOCAL_MACHINE\Software\Microsoft\Policies\default\Update

MDM KEY	KEY TYPE	VALUE
BranchReadinessLevel	REG_DWORD	16: systems take Feature Updates for the Current Branch (CB) 32: systems take Feature Updates for the Current Branch for Business (CBB) Note: Other value or absent: receive all applicable updates (CB)
DeferQualityUpdatesPeriodinDays	REG_DWORD	0-35: defer quality updates by given days
PauseQualityUpdatesStartDate	REG_DWORD	1: pause quality updates Other value or absent: don't pause quality updates
DeferFeatureUpdatesPeriodinDays	REG_DWORD	0-365: defer feature updates by given days

MDM KEY	KEY TYPE	VALUE
PauseFeatureUpdatesStartDate	REG_DWORD	1: pause feature updates Other value or absent: don't pause feature updates
ExcludeWUDriversinQualityUpdate	REG_DWORD	1: exclude Windows Update drivers Other value or absent: offer Windows Update drivers

Update devices to newer versions

Due to the changes in Windows Update for Business, Windows 10, version 1607 uses different GPO and MDM keys than those available in version 1511. Windows 10, version 1703 also uses a few GPO and MDM keys that are different from those available in version 1607. However, Windows Update for Business devices running older versions will still see their policies honored after they update to a newer version; the old policy keys will continue to exist with their values ported forward during the update. Following the update to a newer version, only the old keys will be populated and not the new version keys, until the newer keys are explicitly defined on the device by the administrator.

How older version policies are respected on newer versions

When a device running a newer version sees an update available on Windows Update, the device first evaluates and executes the Windows Updates for Business policy keys for its current (newer) version. If these are not present, it then checks whether any of the older version keys are set and defer accordingly. Update keys for newer versions will always supersede the older equivalent.

Comparing keys in Windows 10, version 1607 to Windows 10, version 1703

VERSION 1607 KEY	VERSION 1703 KEY
PauseFeatureUpdates	PauseFeatureUpdatesStartTime
PauseQualityUpdates	PauseQualityUpdatesStartTime

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)

- [Manage device restarts after updates](#)

Integrate Windows Update for Business with management solutions

6/6/2019 • 3 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

You can integrate Windows Update for Business deployments with existing management tools such as Windows Server Update Services (WSUS) and System Center Configuration Manager.

Integrate Windows Update for Business with Windows Server Update Services

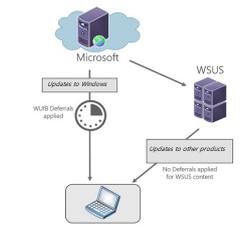
For Windows 10, version 1607, devices can now be configured to receive updates from both Windows Update (or Microsoft Update) and Windows Server Update Services (WSUS). In a joint WSUS and Windows Update for Business setup:

- Devices will receive their Windows content from Microsoft and defer these updates according to Windows Update for Business policy
- All other content synced from WSUS will be directly applied to the device; that is, updates to products other than Windows will not follow your Windows Update for Business deferral policies

Configuration example #1: Deferring Windows Update updates with other update content hosted on WSUS

Configuration:

- Device is configured to defer Windows Quality Updates using Windows Update for Business
- Device is also configured to be managed by WSUS
- Device is not configured to enable Microsoft Update (**Update/AllowMUUpdateService** = not enabled)
- Admin has opted to put updates to Office and other products on WSUS
- Admin has also put 3rd party drivers on WSUS

CONTENT	METADATA SOURCE	PAYLOAD SOURCE	DEFERRED?	
Updates to Windows	Windows Update	Windows Update	Yes	
Updates to Office and other products	WSUS	WSUS	No	
Third-party drivers	WSUS	WSUS	No	

Configuration example #2: Excluding drivers from Windows Quality Updates using Windows Update for Business

Configuration:

- Device is configured to defer Windows Quality Updates and to exclude drivers from Windows Update

Quality Updates (**ExcludeWUDriversInQualityUpdate** = enabled)

- Device is also configured to be managed by WSUS
- Admin has opted to put Windows Update drivers on WSUS

CONTENT	METADATA SOURCE	PAYLOAD SOURCE	DEFERRED?	
Updates to Windows (excluding drivers)	Windows Update	Windows Update	Yes	
Updates to Office and other products	WSUS	WSUS	No	
Drivers	WSUS	WSUS	No	

Configuration example #3: Device configured to receive Microsoft updates

Configuration:

- Device is configured to defer Quality Updates using Windows Update for Business and to be managed by WSUS
- Device is configured to “receive updates for other Microsoft products” along with updates to Windows (**Update/AllowMUUpdateService** = enabled)
- Admin has also placed Microsoft Update, third-party, and locally-published update content on the WSUS server

In this example, the deferral behavior for updates to Office and other non-Windows products is slightly different than if WSUS were not enabled.

- In a non-WSUS case, these updates would be deferred just as any update to Windows would be.
- However, with WSUS also configured, these updates are sourced from Microsoft but deferral policies are not applied.

CONTENT	METADATA SOURCE	PAYLOAD SOURCE	DEFERRED?	
Updates to Windows (excluding drivers)	Microsoft Update	Microsoft Update	Yes	
Updates to Office and other products	Microsoft Update	Microsoft Update	No	
Drivers, third-party applications	WSUS	WSUS	No	

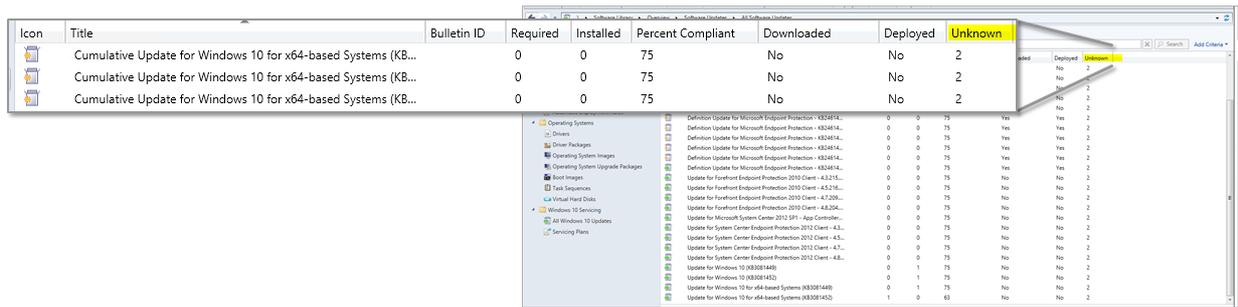
NOTE

Because the admin enabled **Update/AllowMUUpdateService**, placing the content on WSUS was not needed for the particular device, as the device will always receive Microsoft Update content from Microsoft when configured in this manner.

Integrate Windows Update for Business with System Center Configuration Manager

For Windows 10, version 1607, organizations already managing their systems with a Configuration Manager solution can also have their devices configured for Windows Update for Business (i.e. setting deferral policies

on those devices). Such devices will be visible in the Configuration Manager console, however they will appear with a detection state of **Unknown**.



Icon	Title	Bulletin ID	Required	Installed	Percent Compliant	Downloaded	Deployed	Unknown
	Cumulative Update for Windows 10 for x64-based Systems (KB...		0	0	75	No	No	2
	Cumulative Update for Windows 10 for x64-based Systems (KB...		0	0	75	No	No	2
	Cumulative Update for Windows 10 for x64-based Systems (KB...		0	0	75	No	No	2

Category	Title	Required	Installed	Percent Compliant	Downloaded	Deployed	Unknown
Operating Systems	Definition Update for Microsoft Endpoint Protection - KB24814...	0	0	75	Yes	Yes	2
Operating Systems	Definition Update for Microsoft Endpoint Protection - KB24814...	0	0	75	Yes	Yes	2
Operating Systems	Definition Update for Microsoft Endpoint Protection - KB24814...	0	0	75	Yes	Yes	2
Operating System Images	Definition Update for Microsoft Endpoint Protection - KB24814...	0	0	75	Yes	Yes	2
Operating System Upgrade Packages	Definition Update for Microsoft Endpoint Protection - KB24814...	0	0	75	Yes	Yes	2
Task Sequences	Update for Forefront Endpoint Protection 2010 Client - 4.5.216...	0	0	75	No	No	2
Windows 10 Servicing	Update for Forefront Endpoint Protection 2010 Client - 4.7.206...	0	0	75	No	No	2
All Windows 10 Updates	Update for Microsoft System Center 2012 SP1 - App Controller...	0	0	75	No	No	2
Service Plans	Update for System Center Endpoint Protection 2012 Client - 4.5...	0	0	75	No	No	2
	Update for System Center Endpoint Protection 2012 Client - 4.7...	0	0	75	No	No	2
	Update for System Center Endpoint Protection 2012 Client - 4.8...	0	0	75	No	No	2
	Update for Windows 10 (KB3001449)	0	1	75	No	No	2
	Update for Windows 10 (KB3001452)	0	1	75	No	No	2
	Update for Windows 10 for x64-based Systems (KB3001448)	0	1	75	No	No	2
	Update for Windows 10 for x64-based Systems (KB3001452)	1	0	63	No	No	2

For more information, see [Integration with Windows Update for Business in Windows 10](#).

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)
- [Manage device restarts after updates](#)

Walkthrough: use Group Policy to configure Windows Update for Business

6/18/2019 • 5 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

Overview

You can use Group Policy through the Group Policy Management Console (GPMC) to control how Windows Update for Business works. You should consider and devise a deployment strategy for updates before you make changes to the Windows Update for Business settings. See

An IT administrator can set policies for Windows Update for Business by using Group Policy, or they can be set locally (per device). All of the relevant policies are under the path **Computer configuration > Administrative Templates > Windows Components > Windows Update**.

To manage updates with Windows Update for Business as described in this topic, you should prepare with these steps, if you haven't already:

- Create Active Directory security groups that align with the deployment rings you use to phase deployment of updates. See [Build deployment rings for Windows 10 updates](#) to learn more about deployment rings in Windows 10.
- Allow access to the Windows Update service.
- Download and install ADMX templates appropriate to your Windows 10 version. For more information, see [How to create and manage the Central Store for Group Policy Administrative Templates in Windows](#) and [Step-By-Step: Managing Windows 10 with Administrative templates](#).

Set up Windows Update for Business

In this example, one security group is used to manage updates. Typically we would recommend having at least three rings (early testers for pre-release builds, broad deployment for releases, critical devices for mature releases) to deploy. See [Build deployment rings for Windows 10 updates](#) for more information.

Follow these steps on a device running the Remote Server Administration Tools or on a domain controller:

Set up a ring

1. Start Group Policy Management Console (gpmc.msc).
2. Expand ****Forest > Domains > <your domain>**.
3. Right-click **<your domain>** and select **Create a GPO in this domain and link it here**.
4. In the **New GPO** dialog box, enter *Windows Update for Business - Group 1* as the name of the new Group Policy Object.
5. Right-click the ****Windows Update for Business - Group 1"** object, and then select **Edit**.
6. In the Group Policy Management Editor, go to **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Update**. You are now ready to start assigning policies to this ring (group) of devices.

Offering

You can control when updates are applied, for example by deferring when an update is installed on a device or by pausing updates for a certain period of time.

Manage which updates are offered

Windows Update for Business offers you the ability to turn on or off both driver and Microsoft product updates.

- Drivers (on/off): **Computer configuration > Administrative Templates > Windows Components > Windows Update > Do not include drivers with Windows Updates**
- Microsoft product updates (on/off): **Computer configuration > Administrative Templates > Windows Components > Windows Update > Get updates for other Microsoft Products**

We recommend that you allow the driver policy to allow drivers to updated on devices (the default), but you can turn this setting off if you prefer to manage drivers manually. We also recommend that you leave the "Microsoft product updates" setting on.

Manage when updates are offered

You can defer or pause the installation of updates for a set period of time.

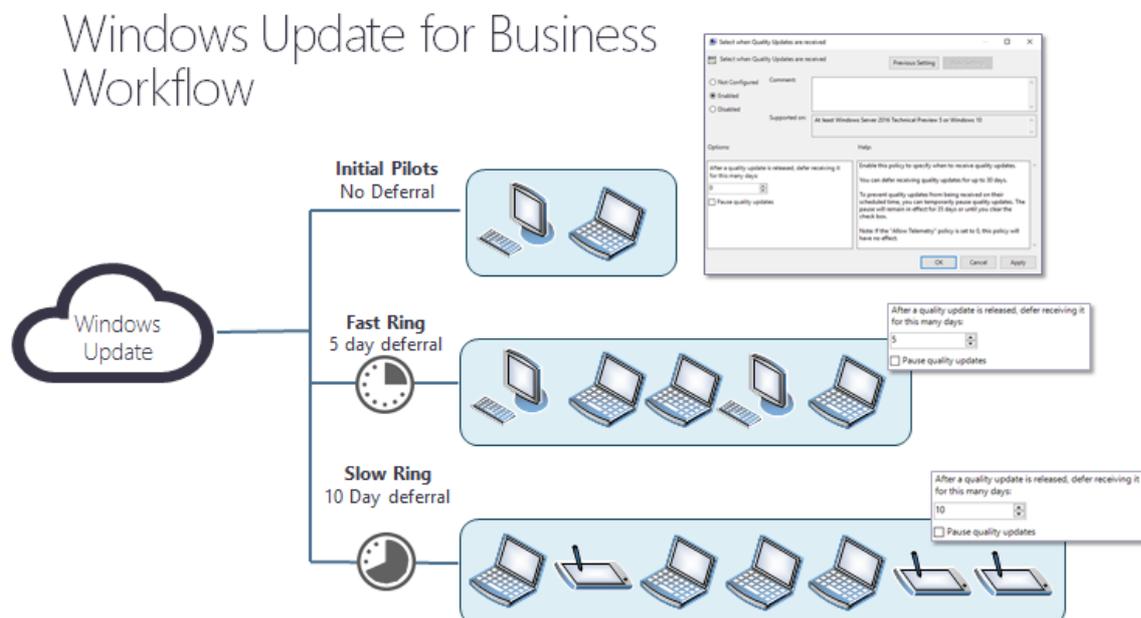
Defer or pause an update

A Windows Update for Business administrator can defer or pause updates and preview builds. You can defer features updates for up to 365 days. You can pause feature or quality updates for up to 35 days from a given start date that you specify.

- Defer or pause a feature update: **Computer configuration > Administrative Templates > Windows Components > Windows Update > Windows Update for Business > Select when Preview Builds and Feature Updates are Received**
- Defer or pause a quality update: **Computer configuration > Administrative Templates > Windows Components > Windows Update > Windows Update for Business > Select when Quality Updates are Received**

Example

In this example, there are three rings for quality updates. The first ring ("pilot") has a deferral period of 0 days. The second ring ("fast") has a deferral of five days. The third ring ("slow") has a deferral of ten days.

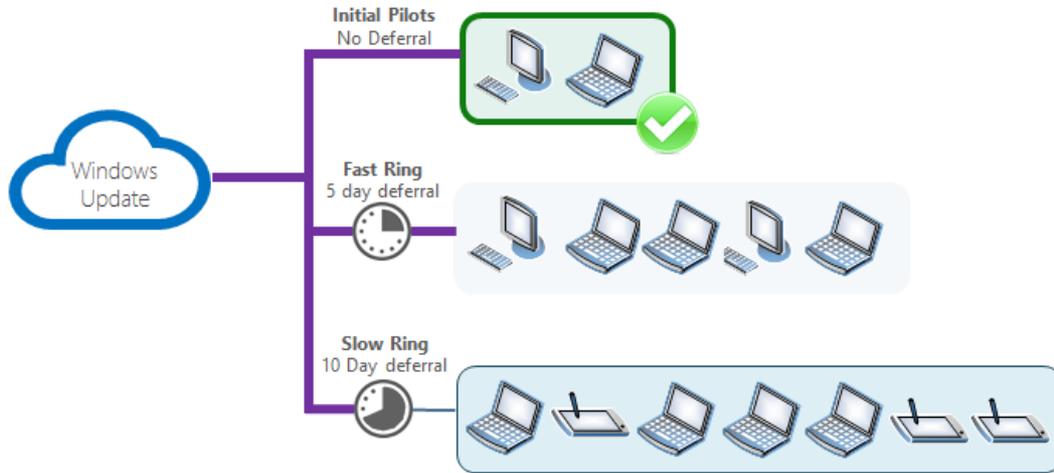


When the quality update is released, it is offered to devices in the pilot ring the next time they scan for updates.

Five days later

The devices in the fast ring are offered the quality update the next time they scan for updates.

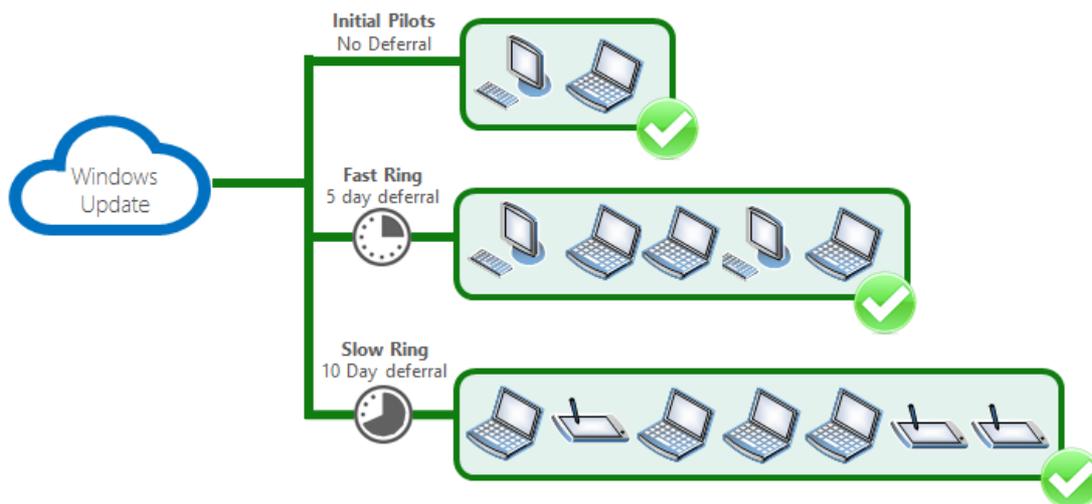
Windows Update for Business Workflow



Ten days later

Ten days after the quality update is released, it is offered to the devices in the slow ring the next time they scan for updates.

Windows Update for Business Workflow



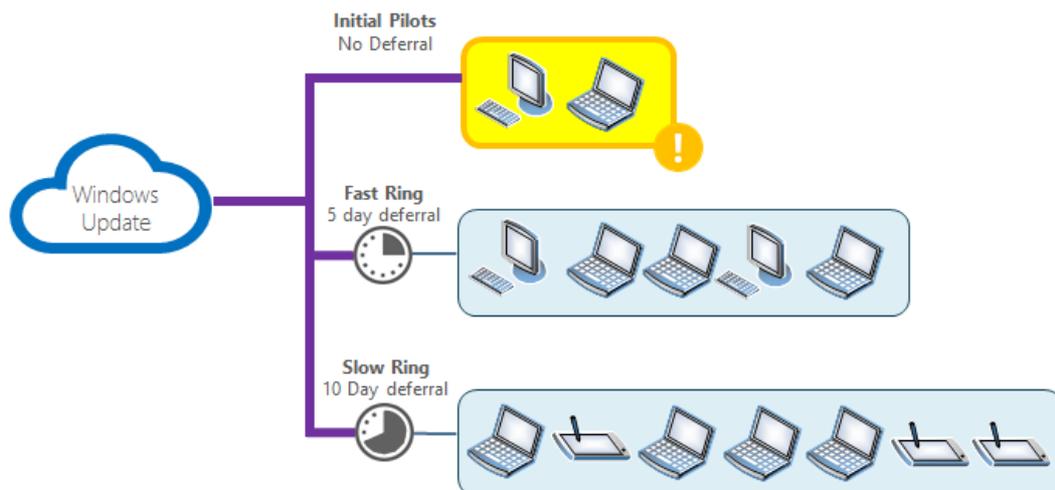
If no problems occur, all of the devices that scan for updates will be offered the quality update within ten days of its release, in three waves.

What if a problem occurs with the update?

In this example, some problem is discovered during the deployment of the update to the "pilot" ring.

Addressing deployment issues

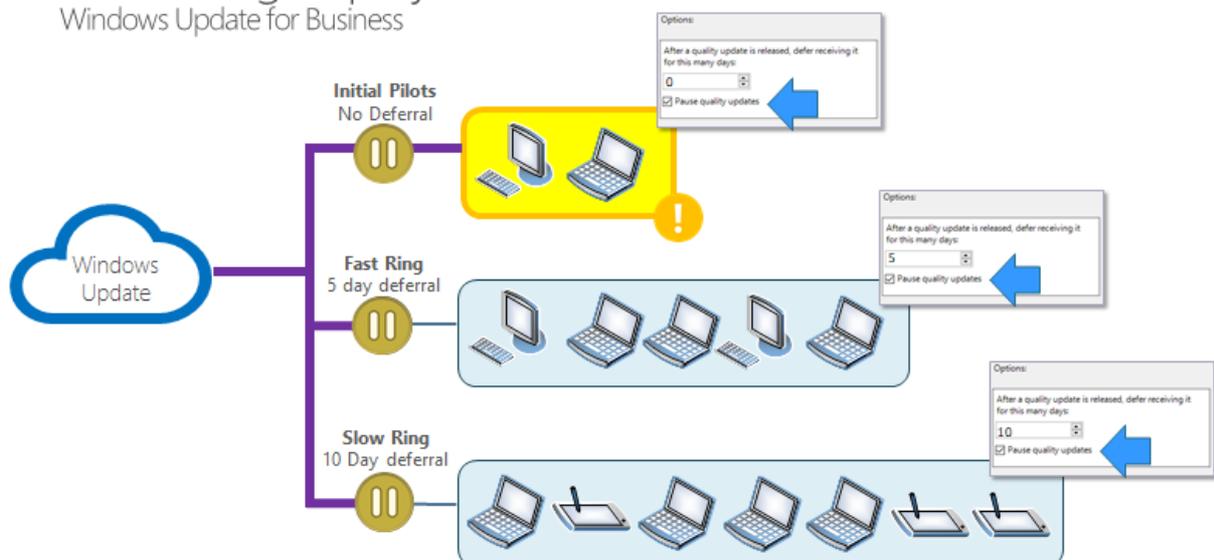
Windows Update for Business



At this point, the IT administrator can set a policy to pause the update. In this example, the admin selects the **Pause quality updates** check box.

Addressing deployment issues

Windows Update for Business



Now all devices are paused from updating for 35 days. When the pause is removed, they will be offered the *next* quality update, which ideally will not have the same issue. If there is still an issue, the IT admin can pause updates again.

Set branch readiness level for feature updates

This policy only applies to feature updates. To enable preview builds for devices in your organization, set the "Enable preview builds" policy and then use the "Select when preview builds and feature updates are received" policy.

We recommend that you set up a ring to receive preview builds by joining the Windows Insider Program for Business. By having a ring of devices receiving "pre-release slow" builds and learning about commercial pre-release features, you can ensure that any issues you have with the release are fixed before it is ever released and far before you broadly deploy.

- Enable preview builds: **Computer configuration > Administrative Templates > Windows Components > Windows Update > Windows Update for Business > Manage Preview Builds**
- Set branch readiness level: **Computer configuration > Administrative Templates > Windows Components > Windows Update > Windows Update for Business > Select when Preview Builds and Feature Updates are Received**

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using Windows Server Update Services](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)
- [Manage device restarts after updates](#)

Deploy Windows 10 updates using Windows Server Update Services (WSUS)

6/12/2019 • 14 minutes to read • [Edit Online](#)

Applies to

- Windows 10

Looking for consumer information? See [Windows Update: FAQ](#)

IMPORTANT

Due to [naming changes](#), older terms like CB,CBB and LTSB may still be displayed in some of our products.

In the following settings CB refers to Semi-Annual Channel (Targeted), while CBB refers to Semi-Annual Channel.

WSUS is a Windows Server role available in the Windows Server operating systems. It provides a single hub for Windows updates within an organization. WSUS allows companies not only to defer updates but also to selectively approve them, choose when they're delivered, and determine which individual devices or groups of devices receive them. WSUS provides additional control over Windows Update for Business but does not provide all the scheduling options and deployment flexibility that System Center Configuration Manager provides.

When you choose WSUS as your source for Windows updates, you use Group Policy to point Windows 10 client devices to the WSUS server for their updates. From there, updates are periodically downloaded to the WSUS server and managed, approved, and deployed through the WSUS administration console or Group Policy, streamlining enterprise update management. If you're currently using WSUS to manage Windows updates in your environment, you can continue to do so in Windows 10.

Requirements for Windows 10 servicing with WSUS

To be able to use WSUS to manage and deploy Windows 10 feature updates, you must have WSUS 4.0, which is available in the Windows Server 2012 R2 and Windows Server 2012 operating systems. In addition to WSUS 4.0, you must install the [KB3095113](#) and [KB3159706](#) patches on the WSUS server.

WSUS scalability

To use WSUS to manage all Windows updates, some organizations may need access to WSUS from a perimeter network, or they might have some other complex scenario. WSUS is highly scalable and configurable for organizations of any size or site layout. For specific information about scaling WSUS, including upstream and downstream server configuration, branch offices, WSUS load balancing, and other complex scenarios, see [Choose a Type of WSUS Deployment](#).

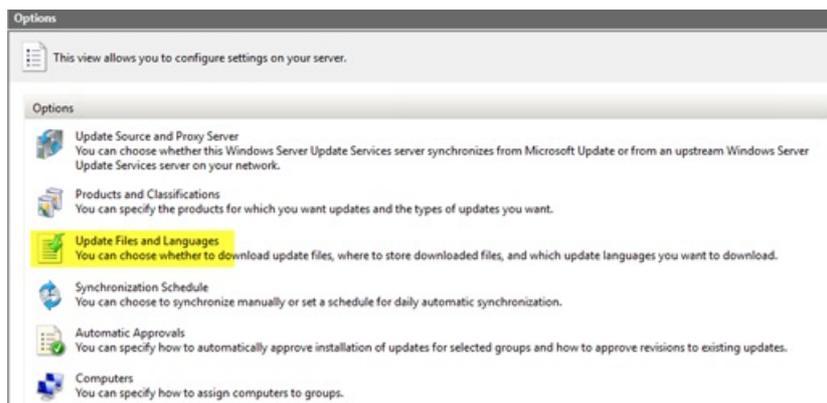
Express Installation Files

With Windows 10, quality updates will be larger than traditional Windows Updates because they're cumulative. To manage the bandwidth clients downloading large updates like these will need, WSUS has a feature called *Express Installation Files*.

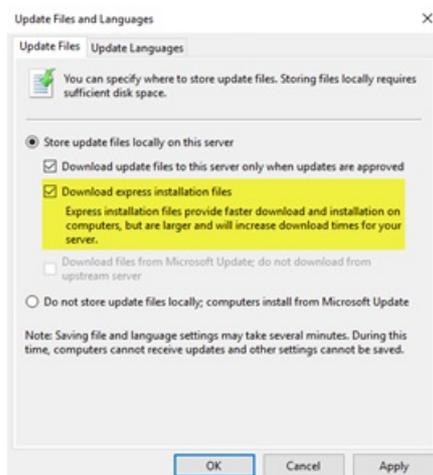
At a binary level, files associated with updates may not change a lot. In fact, with cumulative quality updates, most of the content will be from previous updates. Rather than downloading the entire update when only a small percentage of the payload is actually different, Express Installation Files analyze the differences between the new files associated with an update and the existing files on the client. This approach significantly reduces the amount of bandwidth used because only a fraction of the update content is actually delivered.

To configure WSUS to download Express Update Files

1. Open the WSUS Administration Console.
2. In the navigation pane, go to *Your_Server*\Options.
3. In the **Options** section, click **Update Files and Languages**.



4. In the **Update Files and Languages** dialog box, select **Download express installation files**.



NOTE

Because Windows 10 updates are cumulative, enabling Express Installation Files when WSUS is configured to download Windows 10 updates will significantly increase the amount of disk space that WSUS requires. Alternatively, when using Express Installation Files for previous versions of Windows, the feature's positive effects aren't noticeable because the updates aren't cumulative.

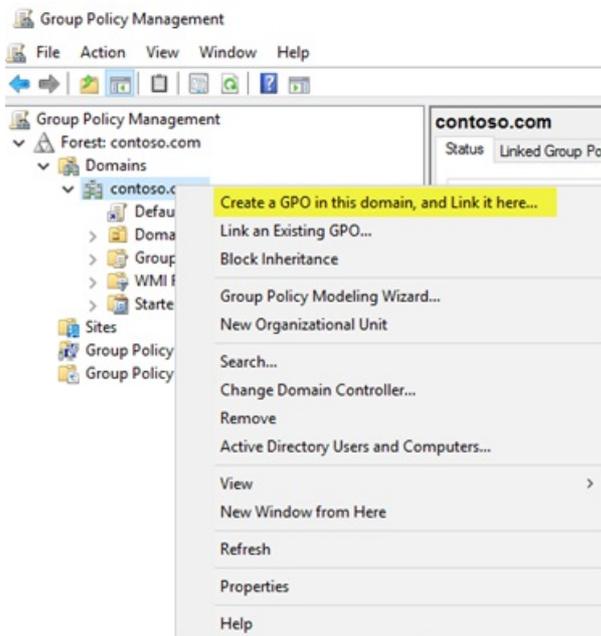
Configure automatic updates and update service location

When using WSUS to manage updates on Windows client devices, start by configuring the **Configure Automatic Updates** and **Intranet Microsoft Update Service Location** Group Policy settings for your environment. Doing so forces the affected clients to contact the WSUS server so that it can manage them. The following process describes how to specify these settings and deploy them to all devices in the domain.

To configure the Configure Automatic Updates and Intranet Microsoft Update Service Location

Group Policy settings for your environment

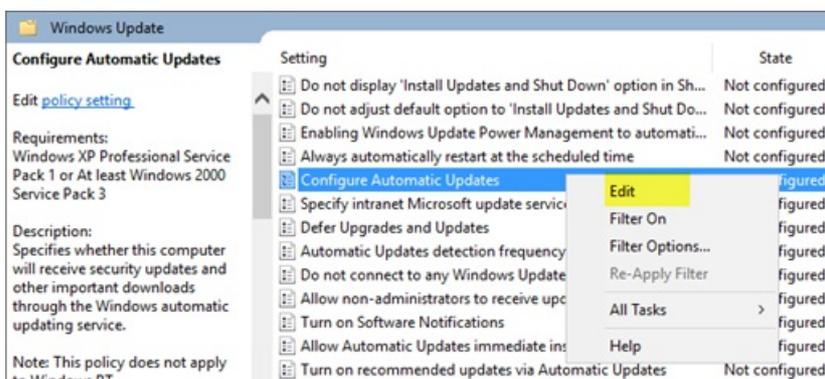
1. Open GPMC.
2. Expand Forest\Domains*Your_Domain*.
3. Right-click *Your_Domain*, and then click **Create a GPO in this domain, and Link it here**.



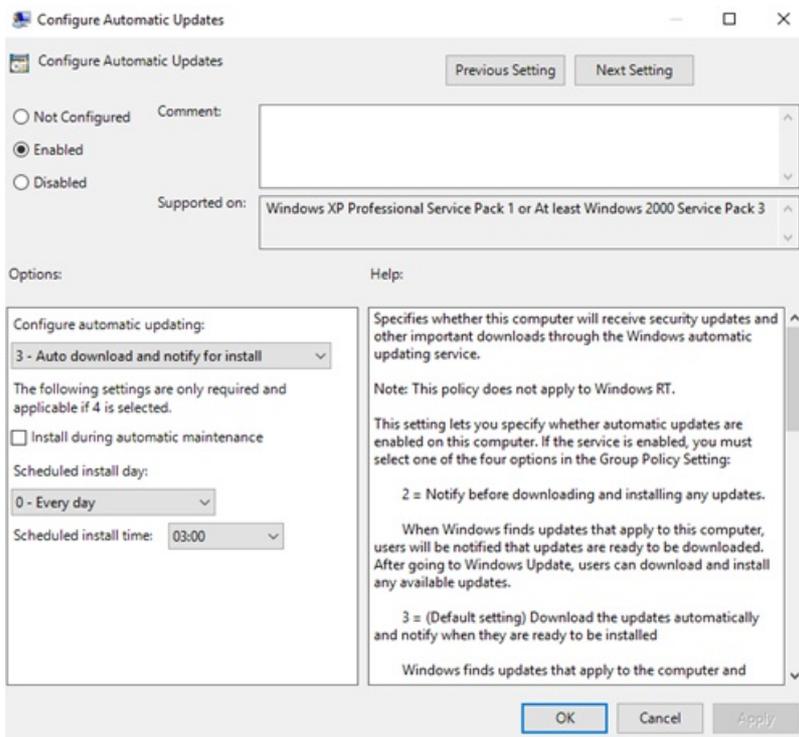
NOTE

In this example, the **Configure Automatic Updates** and **Intranet Microsoft Update Service Location** Group Policy settings are specified for the entire domain. This is not a requirement; you can target these settings to any security group by using Security Filtering or a specific OU.

4. In the **New GPO** dialog box, name the new GPO **WSUS – Auto Updates and Intranet Update Service Location**.
5. Right-click the **WSUS – Auto Updates and Intranet Update Service Location** GPO, and then click **Edit**.
6. In the Group Policy Management Editor, go to Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Update.
7. Right-click the **Configure Automatic Updates** setting, and then click **Edit**.



8. In the **Configure Automatic Updates** dialog box, select **Enable**.
9. Under **Options**, from the **Configure automatic updating** list, select **3 - Auto download and notify for install**, and then click **OK**.



NOTE

?There are three other settings for automatic update download and installation dates and times. This is simply the option this example uses. For more examples of how to control automatic updates and other related policies, see [Configure Automatic Updates by Using Group Policy](#).

10. Right-click the **Specify intranet Microsoft update service location** setting, and then click **Edit**.
11. In the **Specify intranet Microsoft update service location** dialog box, select **Enable**.
12. Under **Options**, in the **Set the intranet update service for detecting updates** and **Set the intranet statistics server** options, type **http://Your_WSUS_Server_FQDN:PortNumber**, and then click **OK**.

NOTE

The URL `http://CONTOSO-WSUS1.contoso.com:8530` in the following image is just an example. In your environment, be sure to use the server name and port number for your WSUS instance.

Specify intranet Microsoft update service location

Previous Setting Next Setting

Not Configured Comment:
 Enabled
 Disabled

Supported on: At least Windows XP Professional Service Pack 1 or Windows 2000 Service Pack 3, excluding Windows RT

Options:

Set the intranet update service for detecting updates:

Set the intranet statistics server:

 (example: http://IntranetUpd01)

Help:

Specifies an intranet server to host updates from Microsoft Update. You can then use this update service to automatically update computers on your network.

This setting lets you specify a server on your network to function as an internal update service. The Automatic Updates client will search this service for updates that apply to the computers on your network.

To use this setting, you must set two servername values: the server from which the Automatic Updates client detects and downloads updates, and the server to which updated workstations upload statistics. You can set both values to be the same server.

If the status is set to Enabled, the Automatic Updates client connects to the specified intranet Microsoft update service, instead of Windows Update, to search for and download updates. Enabling this setting means that end users in your organization don't have to go through a firewall to get updates, and it gives you the opportunity to test updates before deploying

OK Cancel Apply

NOTE

The default HTTP port for WSUS is 8530, and the default HTTP over Secure Sockets Layer (HTTPS) port is 8531. If you're unsure which port WSUS is using for client communication, right-click the WSUS Administration site in IIS Manager, and then click **Edit Bindings**.

As Windows clients refresh their computer policies (the default Group Policy refresh setting is 90 minutes and when a computer restarts), computers start to appear in WSUS. Now that clients are communicating with the WSUS server, create the computer groups that align with your deployment rings.

Create computer groups in the WSUS Administration Console

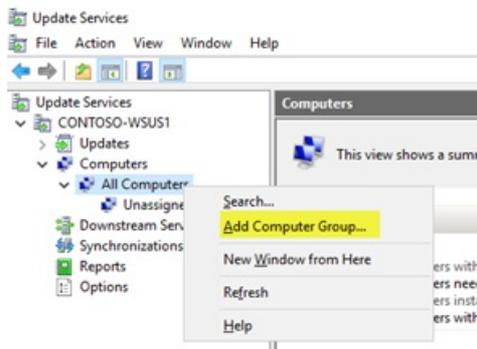
NOTE

The following procedures use the groups from Table 1 in [Build deployment rings for Windows 10 updates](#) as examples.

You can use computer groups to target a subset of devices that have specific quality and feature updates. These groups represent your deployment rings, as controlled by WSUS. You can populate the groups either manually by using the WSUS Administration Console or automatically through Group Policy. Regardless of the method you choose, you must first create the groups in the WSUS Administration Console.

To create computer groups in the WSUS Administration Console

1. Open the WSUS Administration Console.
2. Go to *Server_Name*\Computers\All Computers, and then click **Add Computer Group**.



3. Type **Ring 2 Pilot Business Users** for the name, and then click **Add**.
4. Repeat these steps for the **Ring 3 Broad IT** and **Ring 4 Broad Business Users** groups. When you're finished, there should be three deployment ring groups.

Now that the groups have been created, add the computers to the computer groups that align with the desired deployment rings. You can do this through [Group Policy](#) or manually by using the [WSUS Administration Console](#).

Use the WSUS Administration Console to populate deployment rings

Adding computers to computer groups in the WSUS Administration Console is simple, but it could take much longer than managing membership through Group Policy, especially if you have many computers to add. Adding computers to computer groups in the WSUS Administration Console is called *server-side targeting*.

In this example, you add computers to computer groups in two different ways: by manually assigning unassigned computers and by searching for multiple computers.

Manually assign unassigned computers to groups

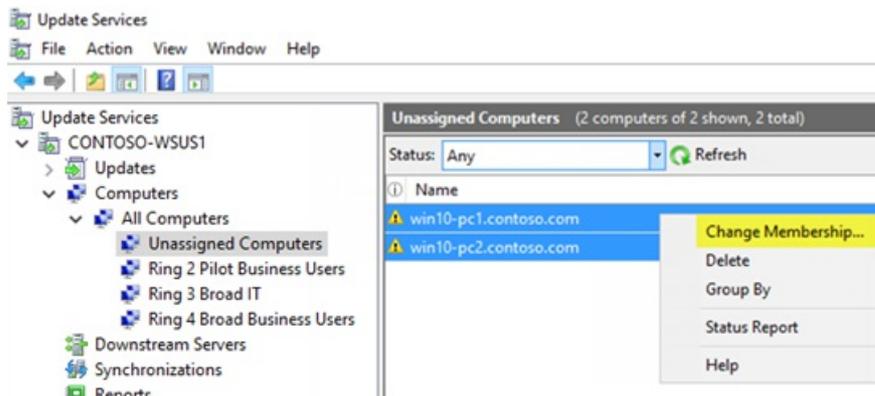
When new computers communicate with WSUS, they appear in the **Unassigned Computers** group. From there, you can use the following procedure to add computers to their correct groups. For these examples, you use two Windows 10 PCs (WIN10-PC1 and WIN10-PC2) to add to the computer groups.

To assign computers manually

1. In the WSUS Administration Console, go to *Server_Name*\Computers\All Computers\Unassigned Computers.

Here, you see the new computers that have received the GPO you created in the previous section and started communicating with WSUS. This example has only two computers; depending on how broadly you deployed your policy, you will likely have many computers here.

2. Select both computers, right-click the selection, and then click **Change Membership**.



3. In the **Set Computer Group Membership** dialog box, select the **Ring 2 Pilot Business Users** deployment ring, and then click **OK**.

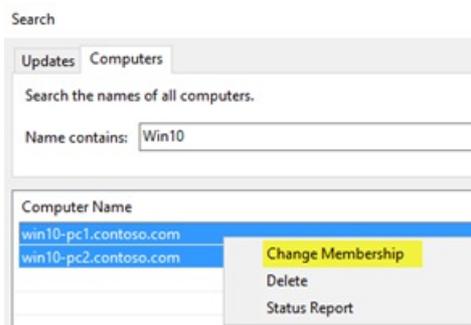
Because they were assigned to a group, the computers are no longer in the **Unassigned Computers** group. If you select the **Ring 2 Pilot Business Users** computer group, you will see both computers there.

Search for multiple computers to add to groups

Another way to add multiple computers to a deployment ring in the WSUS Administration Console is to use the search feature.

To search for multiple computers

1. In the WSUS Administration Console, go to *Server_Name*\Computers\All Computers, right-click **All Computers**, and then click **Search**.
2. In the search box, type **WIN10**.
3. In the search results, select the computers, right-click the selection, and then click **Change Membership**.



4. Select the **Ring 3 Broad IT** deployment ring, and then click **OK**.

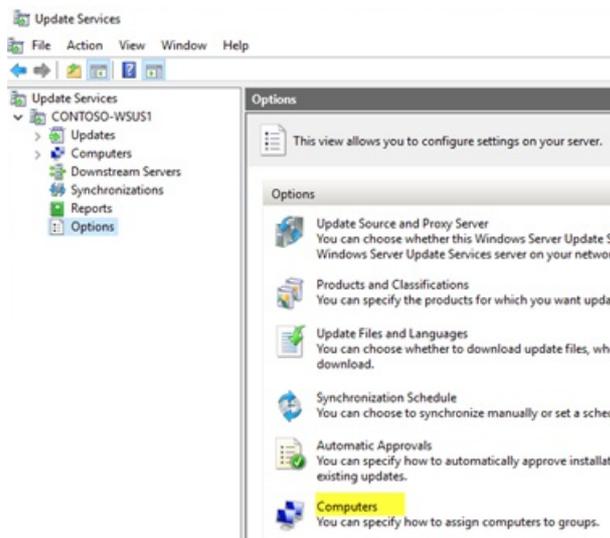
You can now see these computers in the **Ring 3 Broad IT** computer group.

Use Group Policy to populate deployment rings

The WSUS Administration Console provides a friendly interface from which you can manage Windows 10 quality and feature updates. When you need to add many computers to their correct WSUS deployment ring, however, it can be time-consuming to do so manually in the WSUS Administration Console. For these cases, consider using Group Policy to target the correct computers, automatically adding them to the correct WSUS deployment ring based on an Active Directory security group. This process is called *client-side targeting*. Before enabling client-side targeting in Group Policy, you must configure WSUS to accept Group Policy computer assignment.

To configure WSUS to allow client-side targeting from Group Policy

1. Open the WSUS Administration Console, and go to *Server_Name*\Options, and then click **Computers**.



2. In the **Computers** dialog box, select **Use Group Policy or registry settings on computers**, and then click **OK**.

NOTE

This option is exclusively either-or. When you enable WSUS to use Group Policy for group assignment, you can no longer manually add computers through the WSUS Administration Console until you change the option back.

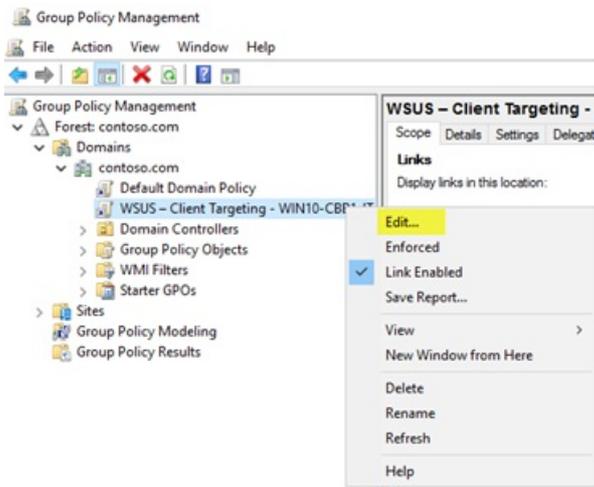
Now that WSUS is ready for client-side targeting, complete the following steps to use Group Policy to configure client-side targeting:

To configure client-side targeting

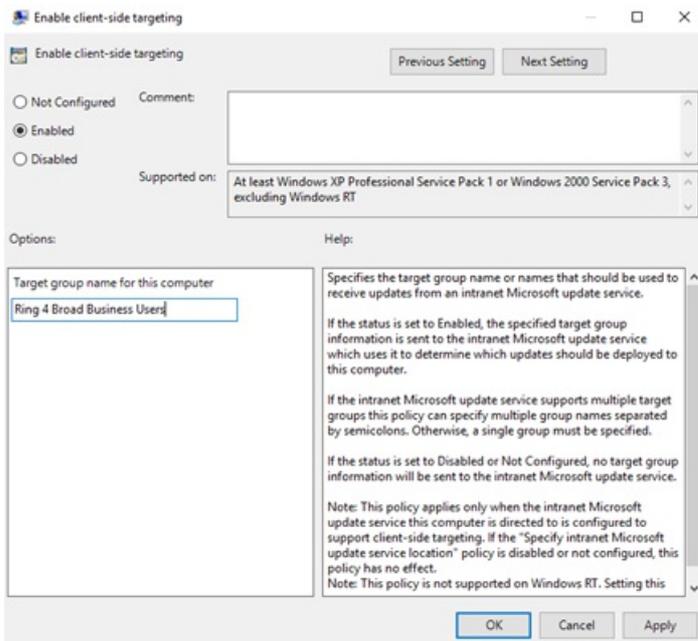
TIP

When using client-side targeting, consider giving security groups the same names as your deployment rings. Doing so simplifies the policy-creation process and helps ensure that you don't add computers to the incorrect rings.

1. Open GPMC.
2. Expand Forest\Domains*Your_Domain*.
3. Right-click *Your_Domain*, and then click **Create a GPO in this domain, and Link it here**.
4. In the **New GPO** dialog box, type **WSUS – Client Targeting – Ring 4 Broad Business Users** for the name of the new GPO.
5. Right-click the **WSUS – Client Targeting – Ring 4 Broad Business Users** GPO, and then click **Edit**.



6. In the Group Policy Management Editor, go to Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Update.
7. Right-click **Enable client-side targeting**, and then click **Edit**.
8. In the **Enable client-side targeting** dialog box, select **Enable**.
9. In the **Target group name for this computer** box, type **Ring 4 Broad Business Users**. This is the name of the deployment ring in WSUS to which these computers will be added.

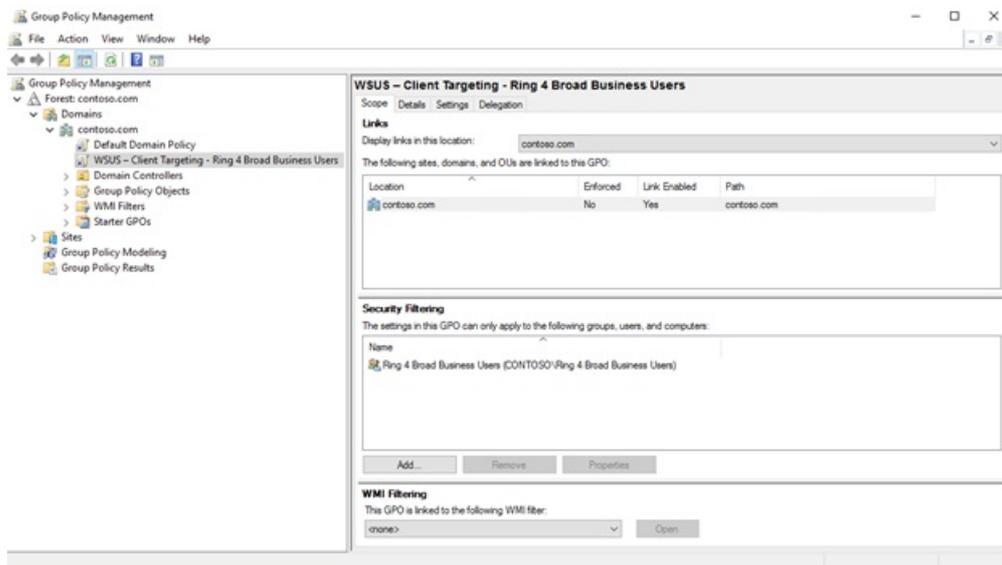


10. Close the Group Policy Management Editor.

Now you're ready to deploy this GPO to the correct computer security group for the **Ring 4 Broad Business Users** deployment ring.

To scope the GPO to a group

1. In GPMC, select the **WSUS - Client Targeting - Ring 4 Broad Business Users** policy.
2. Click the **Scope** tab.
3. Under **Security Filtering**, remove the default **AUTHENTICATED USERS** security group, and then add the **Ring 4 Broad Business Users** group.



The next time the clients in the **Ring 4 Broad Business Users** security group receive their computer policy and contact WSUS, they will be added to the **Ring 4 Broad Business Users** deployment ring.

Automatically approve and deploy feature updates

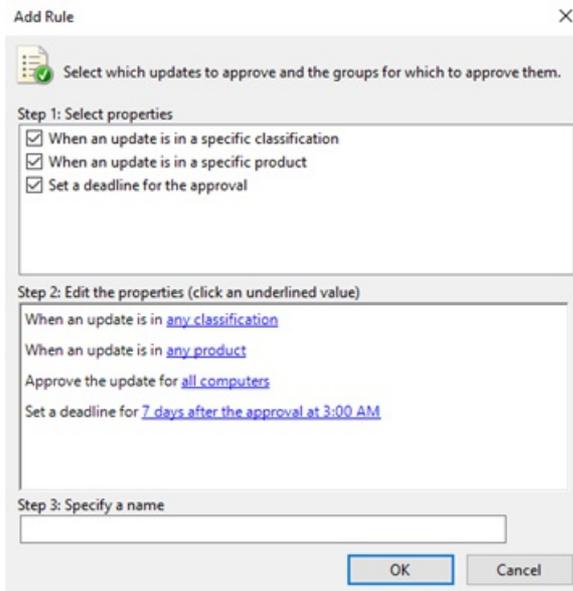
For clients that should have their feature updates approved as soon as they're available, you can configure Automatic Approval rules in WSUS.

NOTE

WSUS respects the client's servicing branch. If you approve a feature update while it is still Current Branch (CB), WSUS will install the update only on PCs that are in the CB servicing branch. When Microsoft releases the build for Current Branch for Business (CBB), the PCs in the CBB servicing branch will install it. Windows Update for Business branch settings do not apply to feature updates through WSUS.

To configure an Automatic Approval rule for Windows 10 feature updates and approve them for the Ring 3 Broad IT deployment ring

1. In the WSUS Administration Console, go to Update Services*Server_Name*\Options, and then select **Automatic Approvals**.
2. On the **Update Rules** tab, click **New Rule**.
3. In the **Add Rule** dialog box, select the **When an update is in a specific classification**, **When an update is in a specific product**, and **Set a deadline for the approval** check boxes.



4. In the **Edit the properties** area, select **any classification**. Clear everything except **Upgrades**, and then click **OK**.

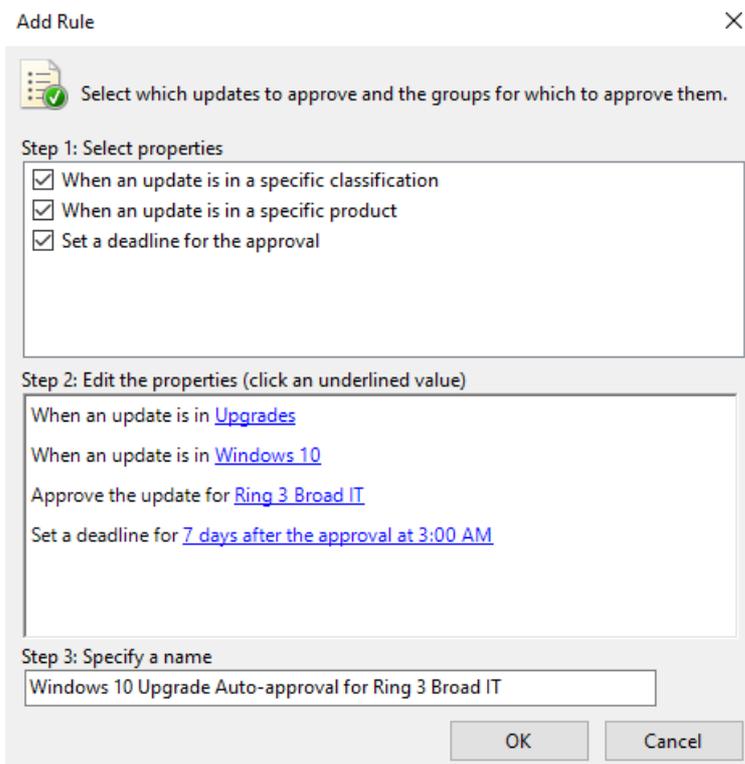
5. In the **Edit the properties area**, click the **any product** link. Clear all check boxes except **Windows 10**, and then click **OK**.

Windows 10 is under All Products\Microsoft\Windows.

6. In the **Edit the properties** area, click the **all computers** link. Clear all the computer group check boxes except **Ring 3 Broad IT**, and then click **OK**.

7. Leave the deadline set for **7 days after the approval at 3:00 AM**.

8. In the **Step 3: Specify a name** box, type **Windows 10 Upgrade Auto-approval for Ring 3 Broad IT**, and then click **OK**.



9. In the **Automatic Approvals** dialog box, click **OK**.

NOTE

WSUS does not honor any existing month/week/day deferral settings for CB or CBB. That said, if you're using Windows Update for Business for a computer for which WSUS is also managing updates, when WSUS approves the update, it will be installed on the computer regardless of whether you configured Group Policy to wait.

Now, whenever Windows 10 feature updates are published to WSUS, they will automatically be approved for the **Ring 3 Broad IT** deployment ring with an installation deadline of 1 week.

Manually approve and deploy feature updates

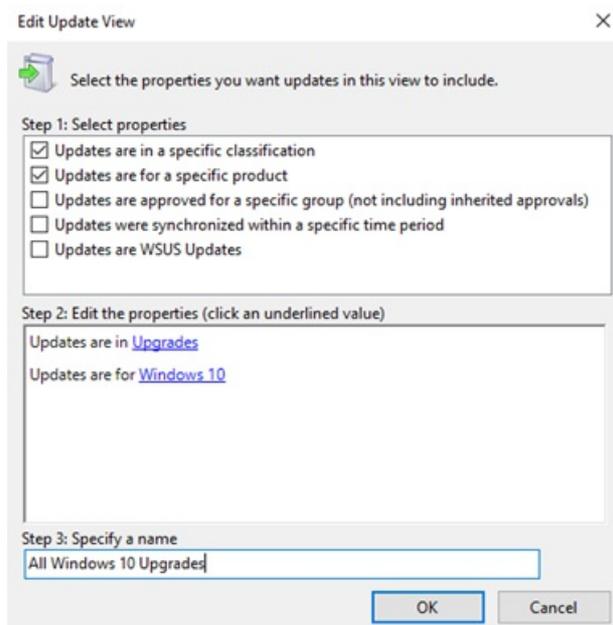
You can manually approve updates and set deadlines for installation within the WSUS Administration Console, as well. To simplify the manual approval process, start by creating a software update view that contains only Windows 10 updates.

To approve and deploy feature updates manually

1. In the WSUS Administration Console, go to Update Services*Server_Name*\Updates. In the **Action** pane, click **New Update View**.
2. In the **Add Update View** dialog box, select **Updates are in a specific classification** and **Updates are for a specific product**.
3. Under **Step 2: Edit the properties**, click **any classification**. Clear all check boxes except **Upgrades**, and then click **OK**.
4. Under **Step 2: Edit the properties**, click **any product**. Clear all check boxes except **Windows 10**, and then click **OK**.

Windows 10 is under All Products\Microsoft\Windows.

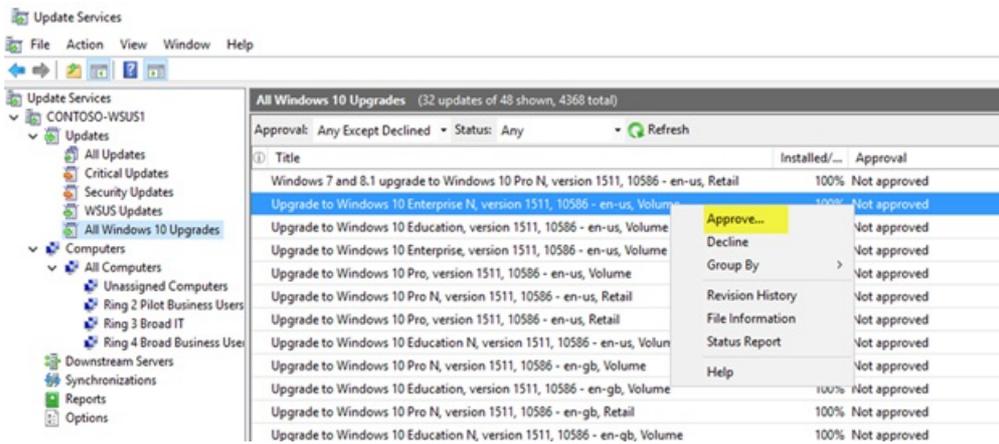
5. In the **Step 3: Specify a name** box, type **All Windows 10 Upgrades**, and then click **OK**.



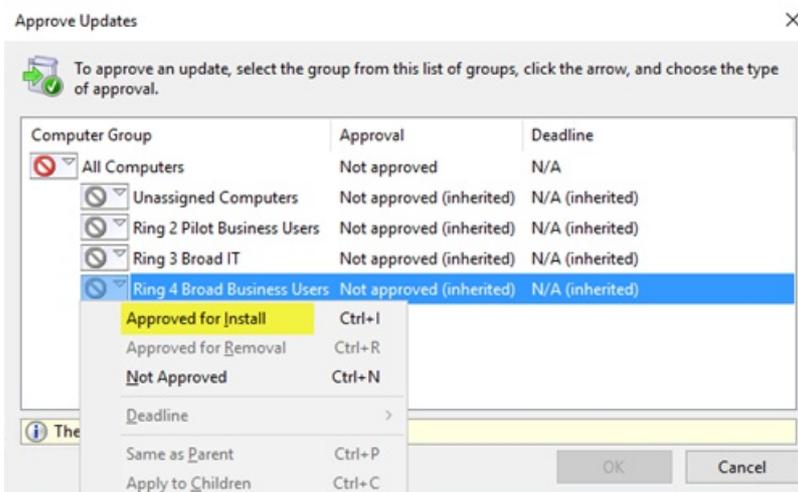
Now that you have the All Windows 10 Upgrades view, complete the following steps to manually approve an update for the **Ring 4 Broad Business Users** deployment ring:

1. In the WSUS Administration Console, go to Update Services*Server_Name*\Updates\All Windows 10 Upgrades.

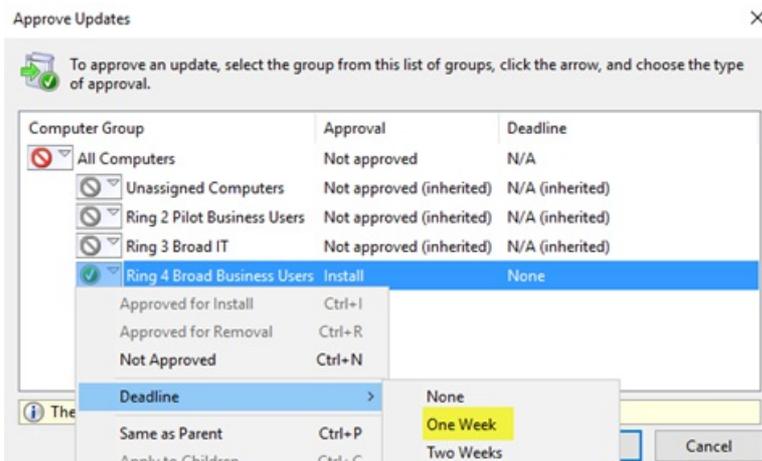
- Right-click the feature update you want to deploy, and then click **Approve**.



- In the **Approve Updates** dialog box, from the **Ring 4 Broad Business Users** list, select **Approved for Install**.

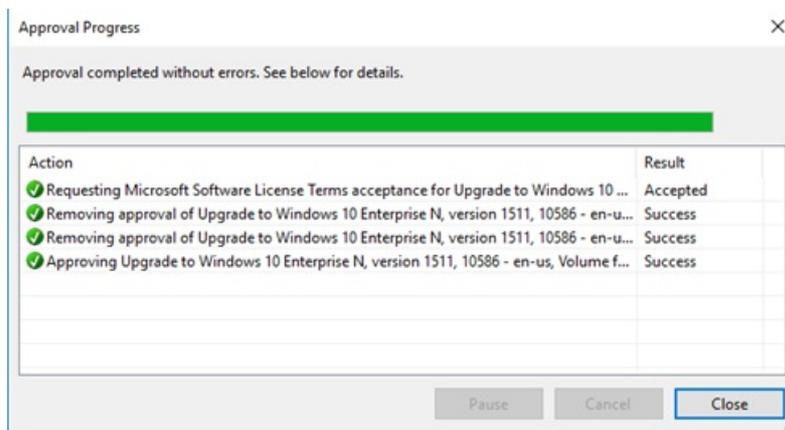


- In the **Approve Updates** dialog box, from the **Ring 4 Broad Business Users** list, click **Deadline**, click **One Week**, and then click **OK**.



- If the **Microsoft Software License Terms** dialog box opens, click **Accept**.

If the deployment is successful, you should receive a successful progress report.



6. In the **Approval Progress** dialog box, click **Close**.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates
<input checked="" type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input checked="" type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input checked="" type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services (this topic) or Deploy Windows 10 updates using System Center Configuration Manager

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Prepare servicing strategy for Windows 10 updates](#)
- [Build deployment rings for Windows 10 updates](#)
- [Assign devices to servicing channels for Windows 10 updates](#)
- [Optimize update delivery for Windows 10 updates](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Deploy updates using Windows Update for Business](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)

- [Walkthrough: use Intune to configure Windows Update for Business](#)
- [Deploy Windows 10 updates using System Center Configuration Manager](#)
- [Manage device restarts after updates](#)

How to make Features on Demand and language packs available when you're using WSUS/SCCM

5/31/2019 • 2 minutes to read • [Edit Online](#)

Applies to: Windows 10

As of Windows 10 version 1709, you cannot use Windows Server Update Services (WSUS) to host [Features on Demand](#) (FOD) and language packs for Windows 10 clients locally. Instead, you can enforce a Group Policy setting that tells the clients to pull them directly from Windows Update. You can also host FOD and language packs on a network share, but starting with Windows 10 version 1809, FOD and language packs can only be installed from Windows Update.

For Windows domain environments running WSUS or SCCM, change the **Specify settings for optional component installation and component repair** policy to enable downloading FOD and language packs from Windows Update. This setting is located in `Computer Configuration\Administrative Templates\System` in the Group Policy Editor.

Changing this policy does not affect how other updates are distributed. They continue to come from WSUS or SCCM as you have scheduled them.

Learn about other client management options, including using Group Policy and administrative templates, in [Manage clients in Windows 10](#).

Deploy Windows 10 updates using System Center Configuration Manager

5/31/2019 • 14 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

IMPORTANT

Due to [naming changes](#), older terms like CB,CBB and LTSB may still be displayed in some of our products.

In the following settings CB refers to Semi-Annual Channel (Targeted), while CBB refers to Semi-Annual Channel.

System Center Configuration Manager provides maximum control over quality and feature updates for Windows 10. Unlike other servicing tools, Configuration Manager has capabilities that extend beyond servicing, such as application deployment, antivirus management, software metering, and reporting, and provides a secondary deployment method for LTSB clients. Configuration Manager can effectively control bandwidth usage and content distribution through a combination of BranchCache and distribution points. Microsoft encourages organizations currently using Configuration Manager for Windows update management to continue doing so for Windows 10 client computers.

You can use Configuration Manager to service Windows 10 devices in two ways. The first option is to use Windows 10 Servicing Plans to deploy Windows 10 feature updates automatically based on specific criteria, similar to an Automatic Deployment Rule for software updates. The second option is to use a task sequence to deploy feature updates, along with anything else in the installation.

NOTE

This topic focuses on updating and upgrading Windows 10 after it has already been deployed. To use Configuration Manager to upgrade your systems from the Windows 8.1, Windows 8, or Windows 7 operating system, see [Upgrade to Windows 10 with System Center Configuration Manager](#).

Windows 10 servicing dashboard

The Windows 10 servicing dashboard gives you a quick-reference view of your active servicing plans, compliance for servicing plan deployment, and other key information about Windows 10 servicing. For details about what each tile on the servicing dashboard represents, see [Manage Windows as a service using System Center Configuration Manager](#).

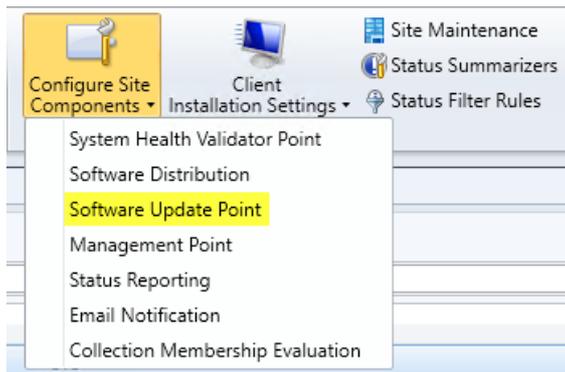
For the Windows 10 servicing dashboard to display information, you must adhere to the following requirements:

- **Heartbeat discovery.** Enable heartbeat discovery for the site receiving Windows 10 servicing information. Configuration for heartbeat discovery can be found in Administration\Overview\Hierarchy Configuration\Discovery Methods.

- **Windows Server Update Service (WSUS).** System Center Configuration Manager must have the Software update point site system role added and configured to receive updates from a WSUS 4.0 server with the hotfix KB3095113 installed.
- **Service connection point.** Add the Service connection point site system role in Online, persistent connection mode.
- **Upgrade classification.** Select **Upgrade** from the list of synchronized software update classifications.

To configure Upgrade classification

1. Go to Administration\Overview\Site Configuration\Sites, and then select your site from the list.
2. On the Ribbon, in the **Settings** section, click **Configure Site Components**, and then click **Software Update Point**.



3. In the **Software Update Point Component Properties** dialog box, on the **Classifications** tab, click **Upgrades**.

When you have met all these requirements and deployed a servicing plan to a collection, you'll receive information on the Windows 10 servicing dashboard.

Create collections for deployment rings

Regardless of the method by which you deploy Windows 10 feature updates to your environment, you must start the Windows 10 servicing process by creating collections of computers that represent your deployment rings. In this example, you create two collections: **Windows 10 – All Current Branch for Business** and **Ring 4 Broad business users**. You'll use the **Windows 10 – All Current Branch for Business** collection for reporting and deployments that should go to all CBB clients. You'll use the **Ring 4 Broad business users** collection as a deployment ring for the first CBB users.

NOTE

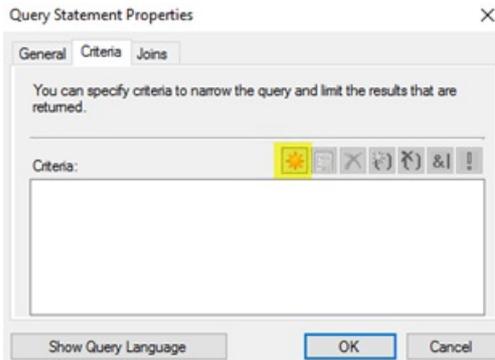
The following procedures use the groups from Table 1 in [Build deployment rings for Windows 10 updates](#) as examples.

To create collections for deployment rings

1. In the Configuration Manager console, go to Assets and Compliance\Overview\Device Collections.
2. On the Ribbon, in the **Create** group, click **Create Device Collection**.
3. In the Create Device Collection Wizard, in the **name** box, type **Windows 10 – All Current Branch for Business**.
4. Click **Browse** to select the limiting collection, and then click **All Systems**.
5. In **Membership rules**, click **Add Rule**, and then click **Query Rule**.

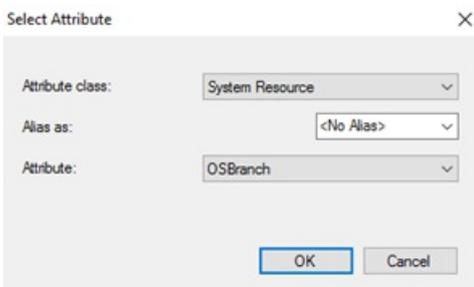
6. Name the rule **CBB Detection**, and then click **Edit Query Statement**.

7. On the **Criteria** tab, click the **New** icon.



8. In the **Criterion Properties** dialog box, leave the type as **Simple Value**, and then click **Select**.

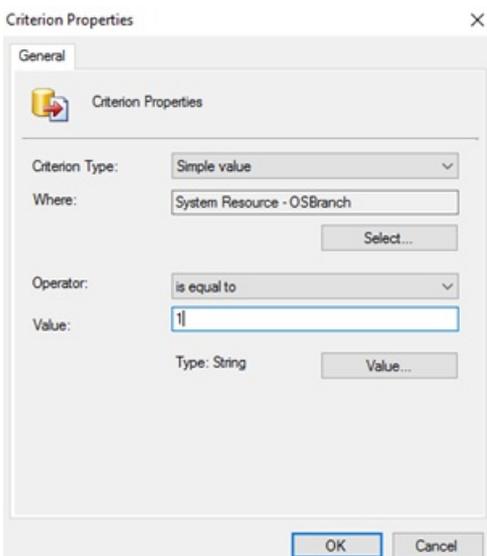
9. In the **Select Attribute** dialog box, from the **Attribute class** list, select **System Resource**. From the **Attribute** list, select **OSBranch**, and then click **OK**.



NOTE

Configuration Manager discovers clients' servicing branch and stores that value in the **OSBranch** attribute, which you will use to create collections based on servicing branch. The values in this attribute can be **0 (Current Branch)**, **1 (Current Branch for Business)**, or **2 (Long-Term Servicing Branch)**.

10. Leave **Operator** set to **is equal to**; in the **Value** box, type **1**. Click **OK**.

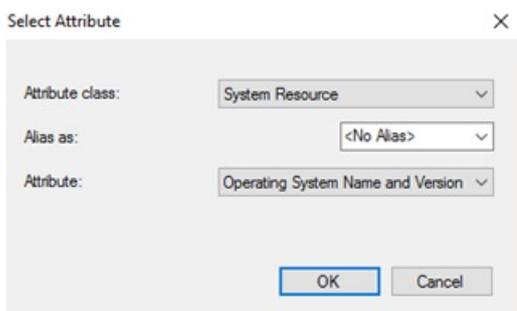


11. Now that the **OSBranch** attribute is correct, verify the operating system version.

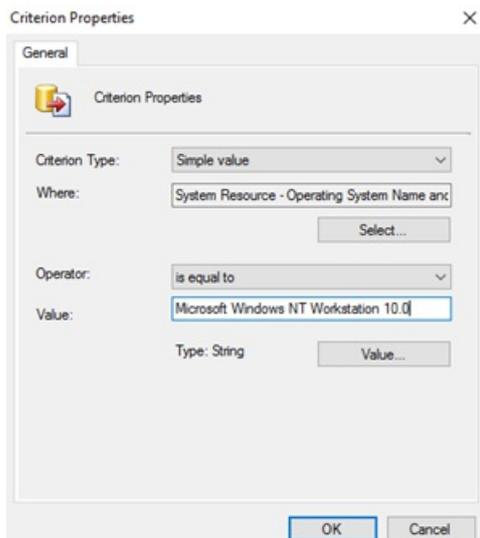
12. On the **Criteria** tab, click the **New** icon again to add criteria.

13. In the **Criterion Properties** dialog box, click **Select**.

- From the **Attribute class** list, select **System Resource**. From the **Attribute** list, select **Operating System Name and Version**, and then click **OK**.



- In the **Value** box, type **Microsoft Windows NT Workstation 10.0**, and then click **OK**.



- In the **Query Statement Properties** dialog box, you see two values. Click **OK**, and then click **OK** again to continue to the Create Device Collection Wizard.
- Click **Summary**, and then click **Next**.
- Close the wizard.

IMPORTANT

Windows Insider PCs are discovered the same way as CB or CBB devices. If you have Windows Insider PCs that you use Configuration Manager to manage, then you should create a collection of those PCs and exclude them from this collection. You can create the membership for the Windows Insider collection either manually or by using a query where the operating system build doesn't equal any of the current CB or CBB build numbers. You would have to update each periodically to include new devices or new operating system builds.

After you have updated the membership, this new collection will contain all managed clients on the CBB servicing branch. You will use this collection as a limiting collection for future CBB-based collections and the **Ring 4 Broad business users** collection. Complete the following steps to create the **Ring 4 Broad business users** device collection, which you'll use as a CBB deployment ring for servicing plans or task sequences.

- In the Configuration Manager console, go to Assets and Compliance\Overview\Device Collections.
- On the Ribbon, in the **Create** group, click **Create Device Collection**.
- In the Create Device Collection Wizard, in the **name** box, type **Ring 4 Broad business users**.
- Click **Browse** to select the limiting collection, and then click **Windows 10 – All Current Branch for**

Business.

5. In **Membership rules**, click **Add Rule**, and then click **Direct Rule**.
6. In the **Create Direct Membership Rule Wizard** dialog box, click **Next**.
7. In the **Value** field, type all or part of the name of a device to add, and then click **Next**.
8. Select the computer that will be part of the **Ring 4 Broad business users** deployment ring, and then click **Next**.
9. Click **Next**, and then click **Close**.
10. In the **Create Device Collection Wizard** dialog box, click **Summary**.
11. Click **Next**, and then click **Close**.

Use Windows 10 servicing plans to deploy Windows 10 feature updates

There are two ways to deploy Windows 10 feature updates with System Center Configuration Manager. The first is to use servicing plans, which provide an automated method to update devices consistently in their respective deployment rings, similar to Automatic Deployment Rules for software updates.

To configure Windows feature updates for CBB clients in the Ring 4 Broad business users deployment ring using a servicing plan

1. In the Configuration Manager console, go to Software Library\Overview\Windows 10 Servicing, and then click **Servicing Plans**.
2. On the Ribbon, in the **Create** group, click **Create Servicing Plan**.
3. Name the plan **Ring 4 Broad business users Servicing Plan**, and then click **Next**.
4. On the **Servicing Plan** page, click **Browse**. Select the **Ring 4 Broad business users** collection, which you created in the [Create collections for deployment rings](#) section, click **OK**, and then click **Next**.

IMPORTANT

Microsoft added a new protection feature to Configuration Manager that prevents accidental installation of high-risk deployments such as operating system upgrades on site systems. If you select a collection (All Systems in this example) that has a site system in it, you may receive the following message.



For details about how to manage the settings for high-risk deployments in Configuration Manager, see [Settings to manage high-risk deployments for System Center Configuration Manager](#).

5. On the **Deployment Ring** page, select the **Business Ready (Current Branch for Business)** readiness state, leave the delay at **0 days**, and then click **Next**.

Doing so deploys CBB feature updates to the broad business users deployment ring immediately after

they are released to CBB.

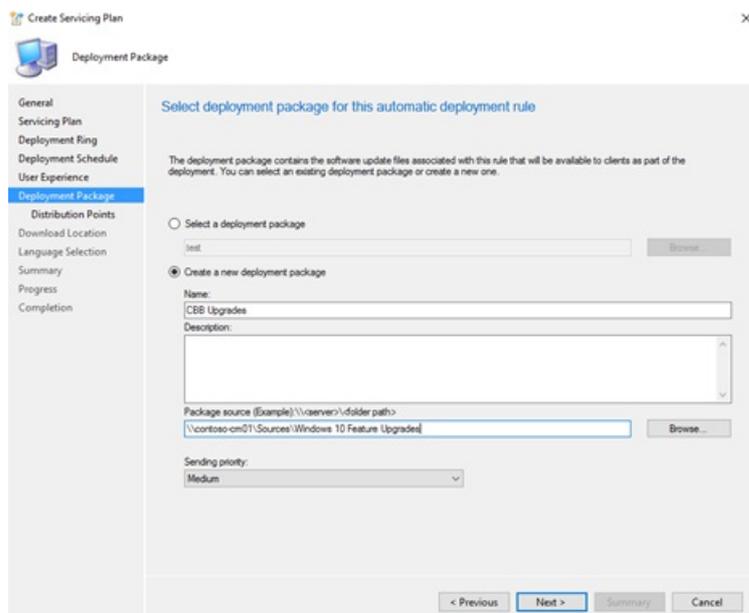
On the **Upgrades** page, you specify filters for the feature updates to which this servicing plan is applicable. For example, if you wanted this plan to be only for Windows 10 Enterprise, you could select **Title**, and then type **Enterprise**.

6. For this example, on the **Upgrades** page, click **Next** to leave the criterion blank.
7. On the **Deployment Schedule** page, click **Next** to keep the default values of making the content available immediately and requiring installation by the 7-day deadline.
8. On the **User Experience** page, from the **Deadline behavior** list, select **Software Installation and System restart (if necessary)**. From the **Device restart behavior** list, select **Workstations**, and then click **Next**.

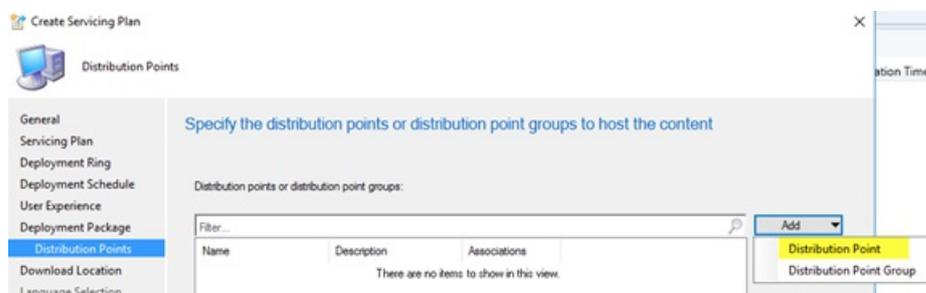
Doing so allows installation and restarts after the 7-day deadline on workstations only.

9. On the **Deployment Package** page, select **Create a new deployment package**. In **Name**, type **CBB Upgrades**, select a share for your package source location, and then click **Next**.

In this example, `\contoso-cm01\Sources\Windows 10 Feature Upgrades` is a share on the Configuration Manager server that contains all the Windows 10 feature updates.



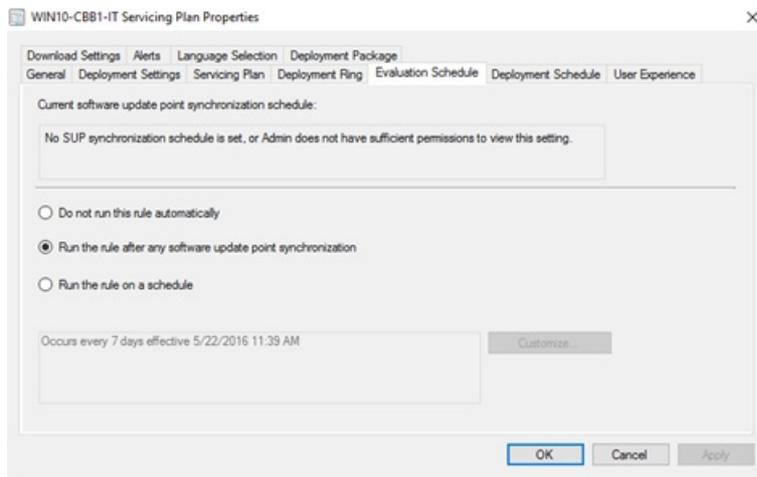
10. On the **Distribution Points** page, from the **Add** list, select **Distribution Point**.



Select the distribution points that serve the clients to which you're deploying this servicing plan, and then click **OK**.

11. Click **Summary**, click **Next** to complete the servicing plan, and then click **Close**.

You have now created a servicing plan for the **Ring 4 Broad business users** deployment ring. By default, this rule is evaluated each time the software update point is synchronized, but you can modify this schedule by viewing the service plan's properties on the **Evaluation Schedule** tab.



Use a task sequence to deploy Windows 10 updates

There are times when deploying a Windows 10 feature update requires the use of a task sequence—for example:

- **LTSB feature updates.** With the LTSB servicing branch, feature updates are never provided to the Windows clients themselves. Instead, feature updates must be installed like a traditional in-place upgrade.
- **Additional required tasks.** When deploying a feature update requires additional steps (e.g., suspending disk encryption, updating applications), you must use task sequences to orchestrate the additional steps. Servicing plans do not have the ability to add steps to their deployments.

Each time Microsoft releases a new Windows 10 build, it releases a new .iso file containing the latest build, as well. Regardless of the scenario that requires a task sequence to deploy the Windows 10 upgrade, the base process is the same. Start by creating an Operating System Upgrade Package in the Configuration Manager console:

1. In the Configuration Manager console, go to Software Library\Overview\Operating Systems\Operating System Upgrade Packages.
2. On the Ribbon, in the **Create** group, click **Add Operating System Upgrade Package**.
3. On the **Data Source** page, type the path of the extracted .iso file of the new version of Windows 10 you're deploying, and then click **Next**.

In this example, the Windows 10 Enterprise 1607 installation media is deployed to \contoso-cm01\Sources\Operating Systems\Windows 10 Enterprise\Windows 10 Enterprise - Version 1607.

NOTE

System Center Configuration Manager version 1606 is required to manage machines running Windows 10, version 1607.

4. On the **General** page, in the **Name** field, type the name of the folder (**Windows 10 Enterprise - Version 1607** in this example). Set the **Version** to **1607**, and then click **Next**.
5. On the **Summary** page, click **Next** to create the package.
6. On the **Completion** page, click **Close**.

Now that the operating system upgrade package has been created, the content in that package must be distributed to the correct distribution points so that the clients can access the content. Complete the following steps to distribute the package content to distribution points:

1. In the Configuration Manager console, go to Software Library\Overview\Operating Systems\Operating System Upgrade Packages, and then select the **Windows 10 Enterprise – Version 1607** software upgrade package.
2. On the Ribbon, in the **Deployment** group, click **Distribute Content**.
3. In the Distribute Content Wizard, on the **General** page, click **Next**.
4. On the **Content Destination** page, click **Add**, and then click **Distribution Point**.
5. In the **Add Distribution Points** dialog box, select the distribution point that will serve the clients receiving this package, and then click **OK**.
6. On the **Content Destination** page, click **Next**.
7. On the **Summary** page, click **Next** to distribute the content to the selected distribution point.
8. On the **Completion** page, click **Close**.

Now that the upgrade package has been created and its contents distributed, create the task sequence that will use it. Complete the following steps to create the task sequence, using the previously created deployment package:

1. In the Configuration Manager console, go to Software Library\Overview\Operating Systems\Task Sequences.
2. On the Ribbon, in the **Create** group, click **Create Task Sequence**.
3. In the Create Task Sequence Wizard, on the **Create a new task sequence** page, select **Upgrade an operating system from upgrade package**, and then click **Next**.
4. On the **Task Sequence Information** page, in **Task sequence name**, type **Upgrade Windows 10 Enterprise – Version 1607**, and then click **Next**.
5. On the **Upgrade the Windows Operating system** page, click **Browse**, select the deployment package you created in the previous steps, and then click **OK**.
6. Click **Next**.
7. On the **Include Updates** page, select **Available for installation – All software updates**, and then click **Next**.
8. On the **Install Applications** page, click **Next**.
9. On the **Summary** page, click **Next** to create the task sequence.
10. On the **Completion** page, click **Close**.

With the task sequence created, you're ready to deploy it. If you're using this method to deploy most of your Windows 10 feature updates, you may want to create deployment rings to stage the deployment of this task sequence, with delays appropriate for the respective deployment ring. In this example, you deploy the task sequence to the **Ring 4 Broad business users collection**.

IMPORTANT

This process deploys a Windows 10 operating system feature update to the affected devices. If you're testing, be sure to select the collection to which you deploy this task sequence carefully.

To deploy your task sequence

1. In the Configuration Manager console, go to Software Library\Overview\Operating Systems\Task Sequences, and then select the **Upgrade Windows 10 Enterprise – Version 1607** task sequence.
2. On the Ribbon, in the **Deployment** group, click **Deploy**.
3. In the Deploy Software Wizard, on the **General** page, click **Browse**. Select the target collection, click **OK**, and then click **Next**.
4. On the **Deployment Settings** page, for **purpose**, select **Required**, and then click **Next**.
5. On the **Scheduling** page, select the **Schedule when this deployment will become available** check box (it sets the current time by default). For **Assignment schedule**, click **New**.
6. In the **Assignment Schedule** dialog box, click **Schedule**.
7. In the **Custom Schedule** dialog box, select the desired deadline, and then click **OK**.
8. In the **Assignment Schedule** dialog box, click **OK**, and then click **Next**.
9. On the **User Experience** page, in the **When the scheduled assignment time is reached, allow the following activities to be performed outside of the maintenance window** section, select **Software Installation** and **System restart** (if required to complete the installation), and then click **Next**.
10. Use the defaults for the remaining settings.
11. Click **Summary**, and then click **Next** to deploy the task sequence.
12. Click **Close**.

Steps to manage updates for Windows 10

<input checked="" type="checkbox"/>	Learn about updates and servicing channels
<input checked="" type="checkbox"/>	Prepare servicing strategy for Windows 10 updates
<input checked="" type="checkbox"/>	Build deployment rings for Windows 10 updates
<input checked="" type="checkbox"/>	Assign devices to servicing channels for Windows 10 updates
<input checked="" type="checkbox"/>	Optimize update delivery for Windows 10 updates
<input checked="" type="checkbox"/>	Deploy updates using Windows Update for Business or Deploy Windows 10 updates using Windows Server Update Services or Deploy Windows 10 updates using System Center Configuration Manager (this topic)

See also

[Manage Windows as a service using System Center Configuration Manager](#)

Related topics

- Update Windows 10 in the enterprise
- Overview of Windows as a service
- Prepare servicing strategy for Windows 10 updates
- Build deployment rings for Windows 10 updates
- Assign devices to servicing channels for Windows 10 updates
- Optimize update delivery for Windows 10 updates
- Configure Delivery Optimization for Windows 10 updates
- Configure BranchCache for Windows 10 updates
- Deploy updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile
- Deploy updates using Windows Update for Business
- Configure Windows Update for Business
- Integrate Windows Update for Business with management solutions
- Walkthrough: use Group Policy to configure Windows Update for Business
- Walkthrough: use Intune to configure Windows Update for Business
- Deploy Windows 10 updates using Windows Server Update Services
- Manage device restarts after updates

Manage device restarts after updates

6/14/2019 • 10 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

You can use Group Policy settings, mobile device management (MDM) or Registry (not recommended) to configure when devices will restart after a Windows 10 update is installed. You can schedule update installation and set policies for restart, configure active hours for when restarts will not occur, or you can do both.

Schedule update installation

In Group Policy, within **Configure Automatic Updates**, you can configure a forced restart after a specified installation time.

To set the time, you need to go to **Configure Automatic Updates**, select option **4 - Auto download and schedule the install**, and then enter a time in the **Scheduled install time** dropdown. Alternatively, you can specify that installation will occur during the automatic maintenance time (configured using **Computer Configuration\Administrative Templates\Windows Components\Maintenance Scheduler**).

Always automatically restart at the scheduled time forces a restart after the specified installation time and lets you configure a timer to warn a signed-in user that a restart is going to occur.

While not recommended, the same result can be achieved through Registry. Under **HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU**, set **AuOptions** to **4**, set the install time with **ScheduledInstallTime**, enable **AlwaysAutoRebootAtScheduledTime** and specify the delay in minutes through **AlwaysAutoRebootAtScheduledTimeMinutes**. Similar to Group Policy, **AlwaysAutoRebootAtScheduledTimeMinutes** sets the timer to warn a signed-in user that a restart is going to occur.

For a detailed description of these registry keys, see [Registry keys used to manage restart](#).

Delay automatic reboot

When **Configure Automatic Updates** is enabled in Group Policy, you can enable one of the following additional policies to delay an automatic reboot after update installation:

- **Turn off auto-restart for updates during active hours** prevents automatic restart during active hours.
- **No auto-restart with logged on users for scheduled automatic updates installations** prevents automatic restart when a user is signed in. If a user schedules the restart in the update notification, the device will restart at the time the user specifies even if a user is signed in at the time. This policy only applies when **Configure Automatic Updates** is set to option **4-Auto download and schedule the install**.

NOTE

When using Remote Desktop Protocol connections, only active RDP sessions are considered as logged on users. Devices that do not have locally logged on users, or active RDP sessions, will be restarted.

You can also use Registry, to prevent automatic restarts when a user is signed in. Under **HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU**, set **AuOptions** to **4** and enable **NoAutoRebootWithLoggedOnUsers**. As with Group Policy, if a user schedules the restart in the update notification, it will override this setting.

For a detailed description of these registry keys, see [Registry keys used to manage restart](#).

Configure active hours

Active hours identify the period of time when you expect the device to be in use. Automatic restarts after an update will occur outside of the active hours.

By default, active hours are from 8 AM to 5 PM on PCs and from 5 AM to 11 PM on phones. Users can change the active hours manually.

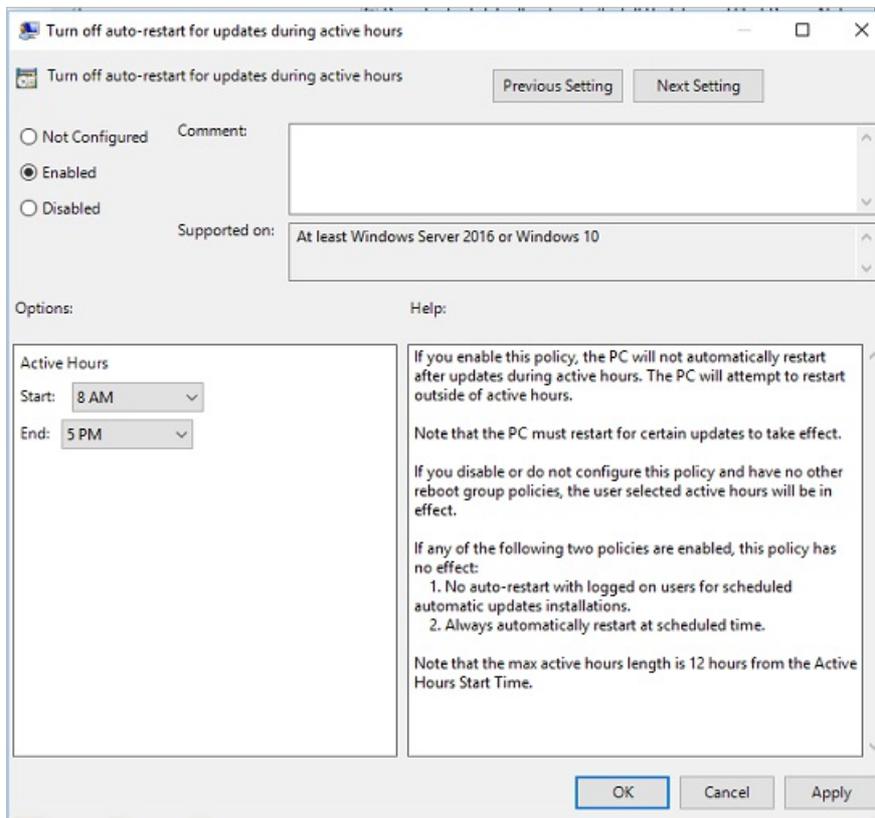
Starting with Windows 10, version 1703, you can also specify the max active hours range. The specified range will be counted from the active hours start time.

Administrators can use multiple ways to set active hours for managed devices:

- You can use Group Policy, as described in the procedure that follows.
- You can use MDM, as described in [Configuring active hours with MDM](#).
- While not recommended, you can also configure active hours, as described in [Configuring active hours through Registry](#).

Configuring active hours with Group Policy

To configure active hours using Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and open the **Turn off auto-restart for updates during active hours** policy setting. When the policy is enabled, you can set the start and end times for active hours.



Configuring active hours with MDM

MDM uses the [Update/ActiveHoursStart](#) and [Update/ActiveHoursEnd](#) and [Update/ActiveHoursMaxRange](#) settings in the [Policy CSP](#) to configure active hours.

Configuring active hours through Registry

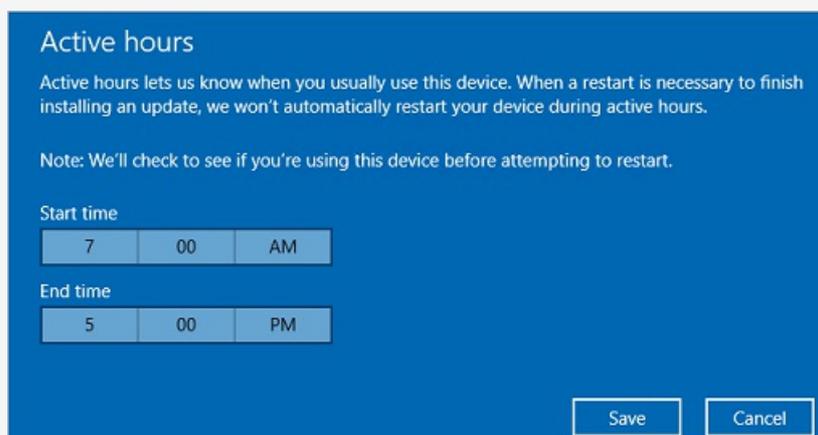
This method is not recommended, and should only be used when neither Group Policy or MDM are available. Any settings configured through Registry may conflict with any existing configuration that uses any of the methods mentioned above.

You should set a combination of the following registry values, in order to configure active hours. Under **HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate** use **SetActiveHours** to enable or disable active hours and **ActiveHoursStart,ActiveHoursEnd** to specify the range of active hours.

For a detailed description of these registry keys, see [Registry keys used to manage restart](#).

NOTE

To configure active hours manually on a single device, go to **Settings > Update & security > Windows Update** and select **Change active hours**.



Configuring active hours max range

With Windows 10, version 1703, administrators can specify the max active hours range users can set. This option gives you additional flexibility to leave some of the decision for active hours on the user's side, while making sure you allow enough time for updating. The max range is calculated from active hours start time.

To configure active hours max range through Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and open the **Specify active hours range for auto-restarts**.

To configure active hours max range through MDM, use [Update/ActiveHoursMaxRange](#).

Limit restart delays

After an update is installed, Windows 10 attempts automatic restart outside of active hours. If the restart does not succeed after 7 days (by default), the user will see a notification that restart is required. You can use the **Specify deadline before auto-restart for update installation** policy to change the delay from 7 days to a number of days between 2 and 14.

Control restart notifications

In Windows 10, version 1703, we have added settings to control restart notifications for users.

Auto-restart notifications

Administrators can override the default behavior for the auto-restart required notification. By default, this notification will dismiss automatically.

To configure this behavior through Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and select **Configure auto-restart required notification for updates**. When configured to **2 - User Action**, a user that gets this notification must manually dismiss it.

To configure this behavior through MDM, use [Update/AutoRestartRequiredNotificationDismissal](#)

You can also configure the period prior to an update that this notification will show up on. The default value is 15 minutes.

To change it through Group Policy, select **Configure auto-restart-reminder notifications for updates** under **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and select the period in minutes.

To change it through MDM, use [Update/AutoRestartNotificationSchedule](#).

In some cases, you don't need a notification to show up.

To do so through Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and select **Turn off auto-restart notifications for update installations**.

To do so through MDM, use [Update/SetAutoRestartNotificationDisable](#).

Scheduled auto-restart warnings

Since users are not able to postpone a scheduled restart once the deadline has been reached, you can configure a warning reminder prior to the scheduled restart. You can also configure a warning prior to the restart, to notify users once the restart is imminent and allow them to save their work.

To configure both through Group Policy, find **Configure auto-restart warning notifications schedule for updates** under **Computer Configuration\Administrative Templates\Windows Components\Windows Update**. The warning reminder can be configured by **Reminder (hours)** and the warning prior to an imminent auto-restart can be configured by **Warning (mins)**.

In MDM, the warning reminder is configured using [Update/ScheduleRestartWarning](#) and the auto-restart imminent warning is configured using [Update/ScheduleImminentRestartWarning](#).

Engaged restart

Engaged restart is the period of time when users are required to schedule a restart. Initially, Windows will auto-restart outside of working hours. Once the set period ends (7 days by default), Windows transitions to user scheduled restarts.

The following settings can be adjusted for engaged restart:

- Period of time before auto-restart transitions to engaged restart.
- The number of days that users can snooze engaged restart reminder notifications.
- The number of days before a pending restart automatically executes outside of working hours.

In Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update** and pick **Specify Engaged restart transition and notification schedule for updates**.

In MDM, use [Update/EngagedRestartTransitionSchedule](#), [Update/EngagedRestartSnoozeSchedule](#) and [Update/EngagedRestartDeadline](#) respectively.

Group Policy settings for restart

In the Group Policy editor, you will see a number of policy settings that pertain to restart behavior in **Computer Configuration\Administrative Templates\Windows Components\Windows Update**. The following table shows which policies apply to Windows 10.

POLICY	APPLIES TO WINDOWS 10	NOTES
Turn off auto-restart for updates during active hours		Use this policy to configure active hours, during which the device will not be restarted. This policy has no effect if the No auto-restart with logged on users for scheduled automatic updates installations or Always automatically restart at the scheduled time policies are enabled.
Always automatically restart at the scheduled time		Use this policy to configure a restart timer (between 15 and 180 minutes) that will start immediately after Windows Update installs important updates. This policy has no effect if the No auto-restart with logged on users for scheduled automatic updates installations policy is enabled.
Specify deadline before auto-restart for update installation		Use this policy to specify how many days (between 2 and 14) an automatic restart can be delayed. This policy has no effect if the No auto-restart with logged on users for scheduled automatic updates installations or Always automatically restart at the scheduled time policies are enabled.

POLICY	APPLIES TO WINDOWS 10	NOTES
No auto-restart with logged on users for scheduled automatic updates installations	✓	Use this policy to prevent automatic restart when a user is logged on. This policy applies only when the Configure Automatic Updates policy is configured to perform scheduled installations of updates. There is no equivalent MDM policy setting for Windows 10 Mobile.
Re-prompt for restart with scheduled installations	✗	
Delay Restart for scheduled installations	✗	
Reschedule Automatic Updates scheduled installations	✗	

NOTE

You can only choose one path for restart behavior. If you set conflicting restart policies, the actual restart behavior may not be what you expected. When using RDP, only active RDP sessions are considered as logged on users.

Registry keys used to manage restart

The following tables list registry values that correspond to the Group Policy settings for controlling restarts after updates in Windows 10.

HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate

REGISTRY KEY	KEY TYPE	VALUE
ActiveHoursEnd	REG_DWORD	0-23: set active hours to end at a specific hour starts with 12 AM (0) and ends with 11 PM (23)
ActiveHoursStart	REG_DWORD	0-23: set active hours to start at a specific hour starts with 12 AM (0) and ends with 11 PM (23)
SetActiveHours	REG_DWORD	0: disable automatic restart after updates outside of active hours 1: enable automatic restart after updates outside of active hours

HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU

REGISTRY KEY	KEY TYPE	VALUE
--------------	----------	-------

REGISTRY KEY	KEY TYPE	VALUE
AlwaysAutoRebootAtScheduledTime	REG_DWORD	0: disable automatic reboot after update installation at scheduled time 1: enable automatic reboot after update installation at ascheduled time
AlwaysAutoRebootAtScheduledTimeM inutes	REG_DWORD	15-180: set automatic reboot to occur after given minutes
AUOptions	REG_DWORD	2: notify for download and notify for installation of updates 3: automatically download and notify for installation of updates 4: Automatically download and schedule installation of updates 5: allow the local admin to configure these settings Note: To configure restart behavior, set this value to 4
NoAutoRebootWithLoggedOnUsers	REG_DWORD	0: disable do not reboot if users are logged on 1: do not reboot after an update installation if a user is logged on Note: If disabled : Automatic Updates will notify the user that the computer will automatically restart in 5 minutes to complete the installation
ScheduledInstallTime	REG_DWORD	0-23: schedule update installation time to a specific hour starts with 12 AM (0) and ends with 11 PM (23)

There are 3 different registry combinations for controlling restart behavior:

- To set active hours, **SetActiveHours** should be **1**, while **ActiveHoursStart** and **ActiveHoursEnd** should define the time range.
- To schedule a specific installation and reboot time, **AUOptions** should be **4**, **ScheduledInstallTime** should specify the installation time, **AlwaysAutoRebootAtScheduledTime** set to **1** and **AlwaysAutoRebootAtScheduledTimeMinutes** should specify number of minutes to wait before rebooting.
- To delay rebooting if a user is logged on, **AUOptions** should be **4**, while **NoAutoRebootWithLoggedOnUsers** is set to **1**.

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Manage updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)
- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Integrate Windows Update for Business with management solutions](#)
- [Walkthrough: use Group Policy to configure Windows Update for Business](#)

- [Walkthrough: use Intune to configure Windows Update for Business](#)

Manage additional Windows Update settings

6/14/2019 • 11 minutes to read • [Edit Online](#)

Applies to

- Windows 10
- Windows 10 Mobile

Looking for consumer information? See [Windows Update: FAQ](#)

You can use Group Policy settings or mobile device management (MDM) to configure the behavior of Windows Update (WU) on your Windows 10 devices. You can configure the update detection frequency, select when updates are received, specify the update service location and more.

IMPORTANT

In Windows 10, any Group Policy user configuration settings for Windows Update were deprecated and are no longer supported on this platform.

Summary of Windows Update settings

GROUP POLICY SETTING	MDM SETTING	SUPPORTED FROM VERSION
Specify Intranet Microsoft update service location	UpdateServiceUrl and UpdateServiceUrlAlternate	All
Automatic Updates Detection Frequency	DetectionFrequency	1703
Remove access to use all Windows Update features		All
Do not connect to any Windows Update Internet locations		All
Enable client-side targeting		All
Allow signed updates from an intranet Microsoft update service location	AllowNonMicrosoftSignedUpdate	All
Do not include drivers with Windows Updates	ExcludeWUDriversInQualityUpdate	1607
Configure Automatic Updates	AllowAutoUpdate	All

IMPORTANT

Additional information about settings to manage device restarts and restart notifications for updates is available on [Manage device restarts after updates](#).

Additional settings that configure when Feature and Quality updates are received are detailed on [Configure Windows Update for Business](#).

Scanning for updates

With Windows 10, admins have a lot of flexibility in configuring how their devices scan and receive updates.

[Specify Intranet Microsoft update service location](#) allows admins to point devices to an internal Microsoft update service location, while [Do not connect to any Windows Update Internet locations](#) gives them the option to restrict devices to just that internal update service. [Automatic Updates Detection Frequency](#) controls how frequently devices scan for updates.

You can make custom device groups that'll work with your internal Microsoft update service by using [Enable client-side targeting](#). You can also make sure your devices receive updates that were not signed by Microsoft from your internal Microsoft update service, through [Allow signed updates from an intranet Microsoft update service location](#).

Finally, to make sure the updating experience is fully controlled by the admins, you can [Remove access to use all Windows Update features](#) for users.

For additional settings that configure when Feature and Quality updates are received, see [Configure Windows Update for Business](#).

Specify Intranet Microsoft update service location

Specifies an intranet server to host updates from Microsoft Update. You can then use this update service to automatically update computers on your network. This setting lets you specify a server on your network to function as an internal update service. The Automatic Updates client will search this service for updates that apply to the computers on your network.

To use this setting in Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows Update\Specify Intranet Microsoft update service location**. You must set two server name values: the server from which the Automatic Updates client detects and downloads updates, and the server to which updated workstations upload statistics. You can set both values to be the same server. An optional server name value can be specified to configure Windows Update Agent to download updates from an alternate download server instead of the intranet update service.

If the setting is set to **Enabled**, the Automatic Updates client connects to the specified intranet Microsoft update service (or alternate download server), instead of Windows Update, to search for and download updates. Enabling this setting means that end users in your organization don't have to go through a firewall to get updates, and it gives you the opportunity to test updates after deploying them. If the setting is set to **Disabled** or **Not Configured**, and if Automatic Updates is not disabled by policy or user preference, the Automatic Updates client connects directly to the Windows Update site on the Internet.

The alternate download server configures the Windows Update Agent to download files from an alternative download server instead of the intranet update service. The option to download files with missing URLs allows content to be downloaded from the Alternate Download Server when there are no download URLs for files in the update metadata. This option should only be used when the intranet update service does not provide download URLs in the update metadata for files which are present on the alternate download server.

NOTE

If the "Configure Automatic Updates" policy is disabled, then this policy has no effect.

If the "Alternate Download Server" is not set, it will use the intranet update service by default to download updates.

The option to "Download files with no Url..." is only used if the "Alternate Download Server" is set.

To configure this policy with MDM, use [UpdateServiceUrl](#) and [UpdateServiceUrlAlternate](#).

Automatic Updates detection frequency

Specifies the hours that Windows will use to determine how long to wait before checking for available updates. The exact wait time is determined by using the hours specified here minus zero to twenty percent of the hours specified. For example, if this policy is used to specify a 20-hour detection frequency, then all clients to which this policy is applied will check for updates anywhere between 16 to 20 hours.

To set this setting with Group Policy, navigate to **Computer Configuration\Administrative Templates\Windows Components\Windows Update\Automatic Updates detection frequency**.

If the setting is set to **Enabled**, Windows will check for available updates at the specified interval. If the setting is set to **Disabled** or **Not Configured**, Windows will check for available updates at the default interval of 22 hours.

NOTE

The "Specify intranet Microsoft update service location" setting must be enabled for this policy to have effect.

If the "Configure Automatic Updates" policy is disabled, this policy has no effect.

To configure this policy with MDM, use [DetectionFrequency](#).

Remove access to use all Windows Update features

By enabling the Group Policy setting under **Computer Configuration\Administrative Templates\Windows Components\Windows update\Remove access to use all Windows update features**, administrators can disable the "Check for updates" option for users. Any background update scans, downloads and installations will continue to work as configured.

Do not connect to any Windows Update Internet locations

Even when Windows Update is configured to receive updates from an intranet update service, it will periodically retrieve information from the public Windows Update service to enable future connections to Windows Update, and other services like Microsoft Update or the Microsoft Store.

Use **Computer Configuration\Administrative Templates\Windows Components\Windows update\Do not connect to any Windows Update Internet locations** to enable this policy. When enabled, this policy will disable the functionality described above, and may cause connection to public services such as the Microsoft Store, Windows Update for Business and Delivery Optimization to stop working.

NOTE

This policy applies only when the device is configured to connect to an intranet update service using the "Specify intranet Microsoft update service location" policy.

Enable client-side targeting

Specifies the target group name or names that should be used to receive updates from an intranet Microsoft update service. This allows admins to configure device groups that will receive different updates from sources like WSUS or SCCM.

This Group Policy setting can be found under **Computer Configuration\Administrative Templates\Windows Components\Windows update\Enable client-side targeting**. If the setting is set to **Enabled**, the specified target group information is sent to the intranet Microsoft update service which uses it to determine which updates should be deployed to this computer. If the setting is set to **Disabled** or **Not Configured**, no target group information will be sent to the intranet Microsoft update service.

If the intranet Microsoft update service supports multiple target groups, this policy can specify multiple group names separated by semicolons. Otherwise, a single group must be specified.

NOTE

This policy applies only when the intranet Microsoft update service the device is directed to is configured to support client-side targeting. If the "Specify intranet Microsoft update service location" policy is disabled or not configured, this policy has no effect.

Allow signed updates from an intranet Microsoft update service location

This policy setting allows you to manage whether Automatic Updates accepts updates signed by entities other than Microsoft when the update is found on an intranet Microsoft update service location.

To configure this setting in Group Policy, go to **Computer Configuration\Administrative Templates\Windows Components\Windows update\Allow signed updates from an intranet Microsoft update service location**.

If you enable this policy setting, Automatic Updates accepts updates received through an intranet Microsoft update service location, as specified by [Specify Intranet Microsoft update service location](#), if they are signed by a certificate found in the "Trusted Publishers" certificate store of the local computer. If you disable or do not configure this policy setting, updates from an intranet Microsoft update service location must be signed by Microsoft.

NOTE

Updates from a service other than an intranet Microsoft update service must always be signed by Microsoft and are not affected by this policy setting.

To configure this policy with MDM, use [AllowNonMicrosoftSignedUpdate](#).

Installing updates

To add more flexibility to the update process, settings are available to control update installation.

[Configure Automatic Updates](#) offers 4 different options for automatic update installation, while [Do not include drivers with Windows Updates](#) makes sure drivers are not installed with the rest of the received updates.

Do not include drivers with Windows Updates

Allows admins to exclude Windows Update (WU) drivers during updates.

To configure this setting in Group Policy, use **Computer Configuration\Administrative Templates\Windows Components\Windows update\Do not include drivers with Windows Updates**. Enable this policy to not include drivers with Windows quality updates. If you disable or do not configure this policy, Windows Update will include updates that have a Driver classification.

Configure Automatic Updates

Enables the IT admin to manage automatic update behavior to scan, download, and install updates.

Configuring Automatic Updates by using Group Policy

Under **Computer Configuration\Administrative Templates\Windows Components\Windows update\Configure Automatic Updates**, you must select one of the four options:

2 - Notify for download and auto install - When Windows finds updates that apply to this device, users will be notified that updates are ready to be downloaded. After going to **Settings > Update & security > Windows Update**, users can download and install any available updates.

3 - Auto download and notify for Install - Windows finds updates that apply to the device and downloads them in the background (the user is not notified or interrupted during this process). When the downloads are complete, users will be notified that they are ready to install. After going to **Settings > Update & security > Windows Update**, users can install them.

4 - Auto download and schedule the install - Specify the schedule using the options in the Group Policy Setting. For more information about this setting, see [Schedule update installation](#).

5 - Allow local admin to choose setting - With this option, local administrators will be allowed to use the settings app to select a configuration option of their choice. Local administrators will not be allowed to disable the configuration for Automatic Updates.

If this setting is set to *Disabled*, any updates that are available on Windows Update must be downloaded and installed manually. To do this, users must go to **Settings > Update & security > Windows Update**.

If this setting is set to *Not Configured*, an administrator can still configure Automatic Updates through the settings app, under **Settings > Update & security > Windows Update > Advanced options**.

Configuring Automatic Updates by editing the registry

![Note] Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require you to reinstall the operating system. Microsoft cannot guarantee that these problems can be resolved. Modify the registry at your own risk.

In an environment that does not have Active Directory deployed, you can edit registry settings to configure group policies for Automatic Update.

To do this, follow these steps:

1. Select **Start**, search for "regedit", and then open Registry Editor.
2. Open the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU
```

3. Add one of the following registry values to configure Automatic Update.

- NoAutoUpdate (REG_DWORD):
 - **0**: Automatic Updates is enabled (default).
 - **1**: Automatic Updates is disabled.
- AUOptions (REG_DWORD):
 - **1**: Keep my computer up to date is disabled in Automatic Updates.
 - **2**: Notify of download and installation.
 - **3**: Automatically download and notify of installation.
 - **4**: Automatically download and scheduled installation.

- ScheduledInstallDay (REG_DWORD):
 - **0**: Every day.
 - **1** through **7**: The days of the week from Sunday (1) to Saturday (7).

- ScheduledInstallTime (REG_DWORD):

n, where **n** equals the time of day in a 24-hour format (0-23).

- UseWU Server (REG_DWORD)

Set this value to **1** to configure Automatic Updates to use a server that is running Software Update Services instead of Windows Update.

- RescheduleWaitTime (REG_DWORD)

m, where **m** equals the time period to wait between the time Automatic Updates starts and the time that it begins installations where the scheduled times have passed. The time is set in minutes from 1 to 60, representing 1 minute to 60 minutes)

![Note] This setting only affects client behavior after the clients have updated to the SUS SP1 client version or later versions.

- NoAutoRebootWithLoggedOnUsers (REG_DWORD):

0 (false) or **1** (true). If set to **1**, Automatic Updates does not automatically restart a computer while users are logged on.

![Note] This setting affects client behavior after the clients have updated to the SUS SP1 client version or later versions.

To use Automatic Updates with a server that is running Software Update Services, see the Deploying Microsoft Windows Server Update Services 2.0 guidance.

When you configure Automatic Updates directly by using the policy registry keys, the policy overrides the preferences that are set by the local administrative user to configure the client. If an administrator removes the registry keys at a later date, the preferences that were set by the local administrative user are used again.

To determine the WSUS server that the client computers and servers connect to for updates, add the following registry values to the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\
```

- WU Server (REG_SZ)

This value sets the WSUS server by HTTP name (for example, http://IntranetSUS).
- WU Status Server (REG_SZ)

This value sets the SUS statistics server by HTTP name (for example, http://IntranetSUS).

Related topics

- [Update Windows 10 in the enterprise](#)
- [Overview of Windows as a service](#)
- [Manage updates for Windows 10 Mobile Enterprise and Windows 10 IoT Mobile](#)

- [Configure Delivery Optimization for Windows 10 updates](#)
- [Configure BranchCache for Windows 10 updates](#)
- [Configure Windows Update for Business](#)
- [Manage device restarts after updates](#)

2 minutes to read

Windows Analytics overview

6/14/2019 • 2 minutes to read • [Edit Online](#)

Windows Analytics is a set of solutions for Azure Portal that provide you with extensive data about the state of devices in your deployment. There are currently three solutions which you can use singly or in any combination:

Device Health

[Device Health](#) provides the following:

- Identification of devices that crash frequently, and therefore might need to be rebuilt or replaced
- Identification of device drivers that are causing device crashes, with suggestions of alternative versions of those drivers that might reduce the number of crashes
- Notification of Windows Information Protection misconfigurations that send prompts to end users

Update Compliance

[Update Compliance](#) shows you the state of your devices with respect to the Windows updates so that you can ensure that they are on the most current updates as appropriate. In addition, Update Compliance provides the following:

- Dedicated drill-downs for devices that might need attention
- An inventory of devices, including the version of Windows they are running and their update status
- The ability to track protection and threat status for Windows Defender Antivirus-enabled devices
- An overview of Windows Update for Business deferral configurations (Windows 10, version 1607 and later)
- Powerful built-in log analytics to create useful custom queries
- Cloud-connected access utilizing Windows 10 diagnostic data means no need for new complex, customized infrastructure

Upgrade Readiness

[Upgrade Readiness](#) offers a set of tools to plan and manage the upgrade process end to end, allowing you to adopt new Windows releases more quickly. With new Windows versions being released multiple times a year, ensuring application and driver compatibility on an ongoing basis is key to adopting new Windows versions as they are released. Upgrade Readiness not only supports upgrade management from Windows 7 and Windows 8.1 to Windows 10, but also Windows 10 upgrades in the Windows as a service model.

Use Upgrade Readiness to get:

- A visual workflow that guides you from pilot to production
- Detailed computer and application inventory
- Powerful computer-level search and drill-downs
- Guidance and insights into application and driver compatibility issues, with suggested fixes
- Data-driven application rationalization tools
- Application usage information, allowing targeted validation; workflow to track validation progress and decisions
- Data export to commonly used software deployment tools, including System Center Configuration Manager

To get started with any of these solutions, visit the links for instructions to add it to Azure Portal.

NOTE

For details about licensing requirements and costs associated with using Windows Analytics solutions, see [What are the requirements and costs for Windows Analytics solutions?](#)

Windows Analytics in the Azure Portal

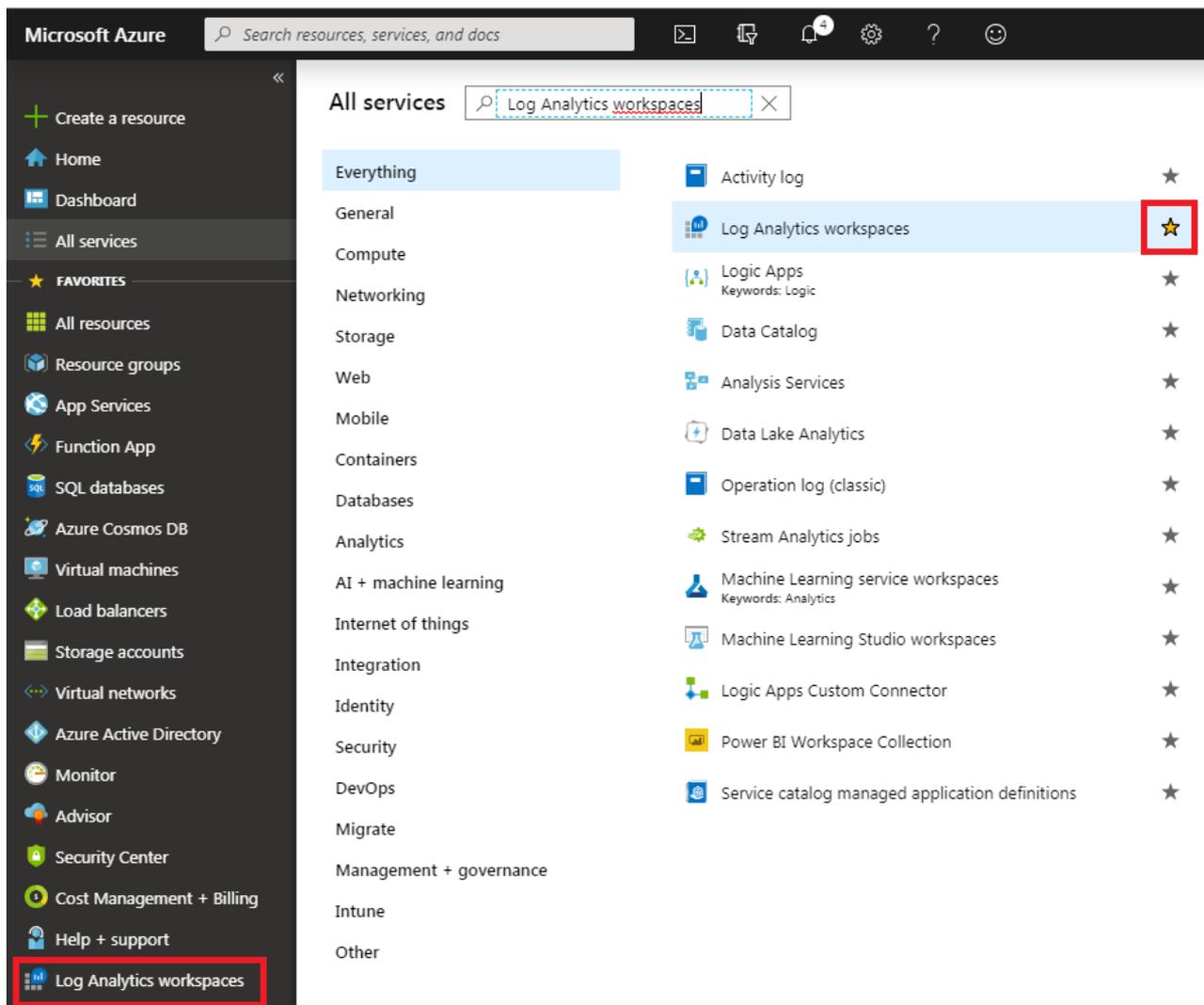
6/14/2019 • 2 minutes to read • [Edit Online](#)

Windows Analytics uses Azure Log Analytics workspaces (formerly known as Operations Management Suite or OMS), a collection of cloud-based services for monitoring and automating your on-premises and cloud environments.

The OMS portal has been deprecated; you should start using the Azure portal instead as soon as possible. Many experiences are the same in the two portals, but there are some key differences, which this topic will explain. For much more information about the transition from OMS to Azure, see [OMS portal moving to Azure](#).

Navigation and permissions in the Azure portal

Go to the [Azure portal](#), select **All services**, and search for *Log Analytics workspaces*. Once it appears, you can select the star to add it to your favorites for easy access in the future.



The screenshot shows the Microsoft Azure portal interface. On the left is a navigation sidebar with a search bar at the top. The 'All services' section is active, and a search box contains the text 'Log Analytics workspaces'. Below the search box, a list of services is displayed, including 'Activity log', 'Log Analytics workspaces', 'Logic Apps', 'Data Catalog', 'Analysis Services', 'Data Lake Analytics', 'Operation log (classic)', 'Stream Analytics jobs', 'Machine Learning service workspaces', 'Machine Learning Studio workspaces', 'Logic Apps Custom Connector', 'Power BI Workspace Collection', and 'Service catalog managed application definitions'. The 'Log Analytics workspaces' entry is highlighted in blue, and its star icon is circled in red. In the bottom-left corner of the sidebar, the 'Log Analytics workspaces' link is also highlighted with a red box.

Permissions

It's important to understand the difference between Azure Active Directory and an Azure subscription:

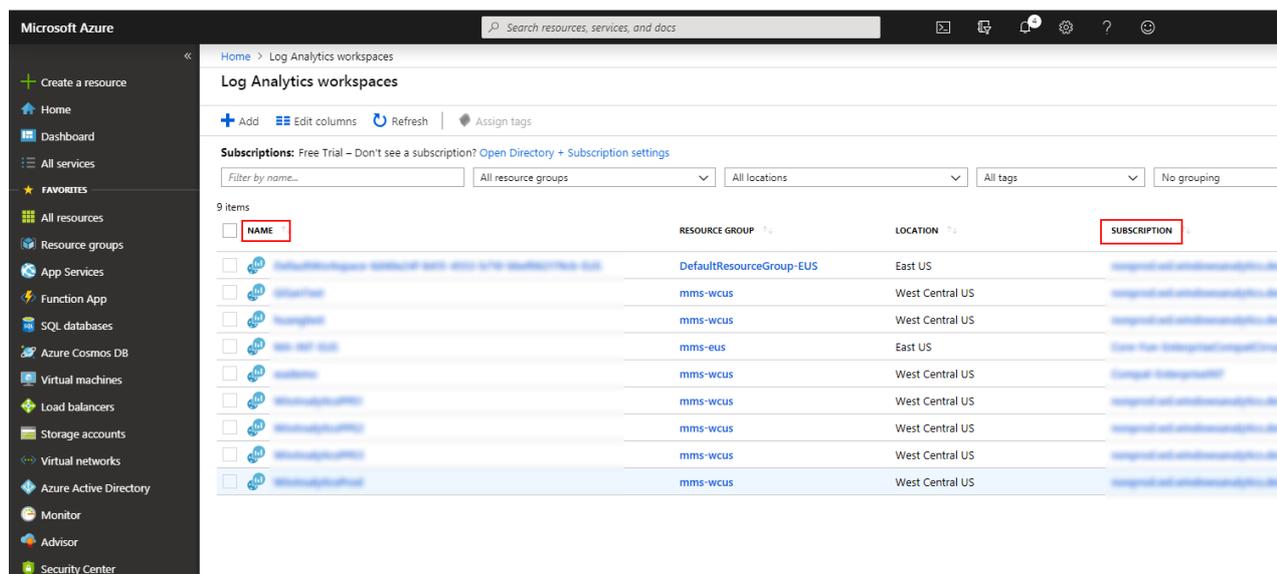
Azure Active Directory is the directory that Azure uses. Azure Active Directory (Azure AD) is a separate service which sits by itself and is used by all of Azure and also Office 365.

An **Azure subscription** is a container for billing, but also acts as a security boundary. Every Azure subscription has a trust relationship with at least one Azure AD instance. This means that a subscription trusts that directory to authenticate users, services, and devices.

IMPORTANT

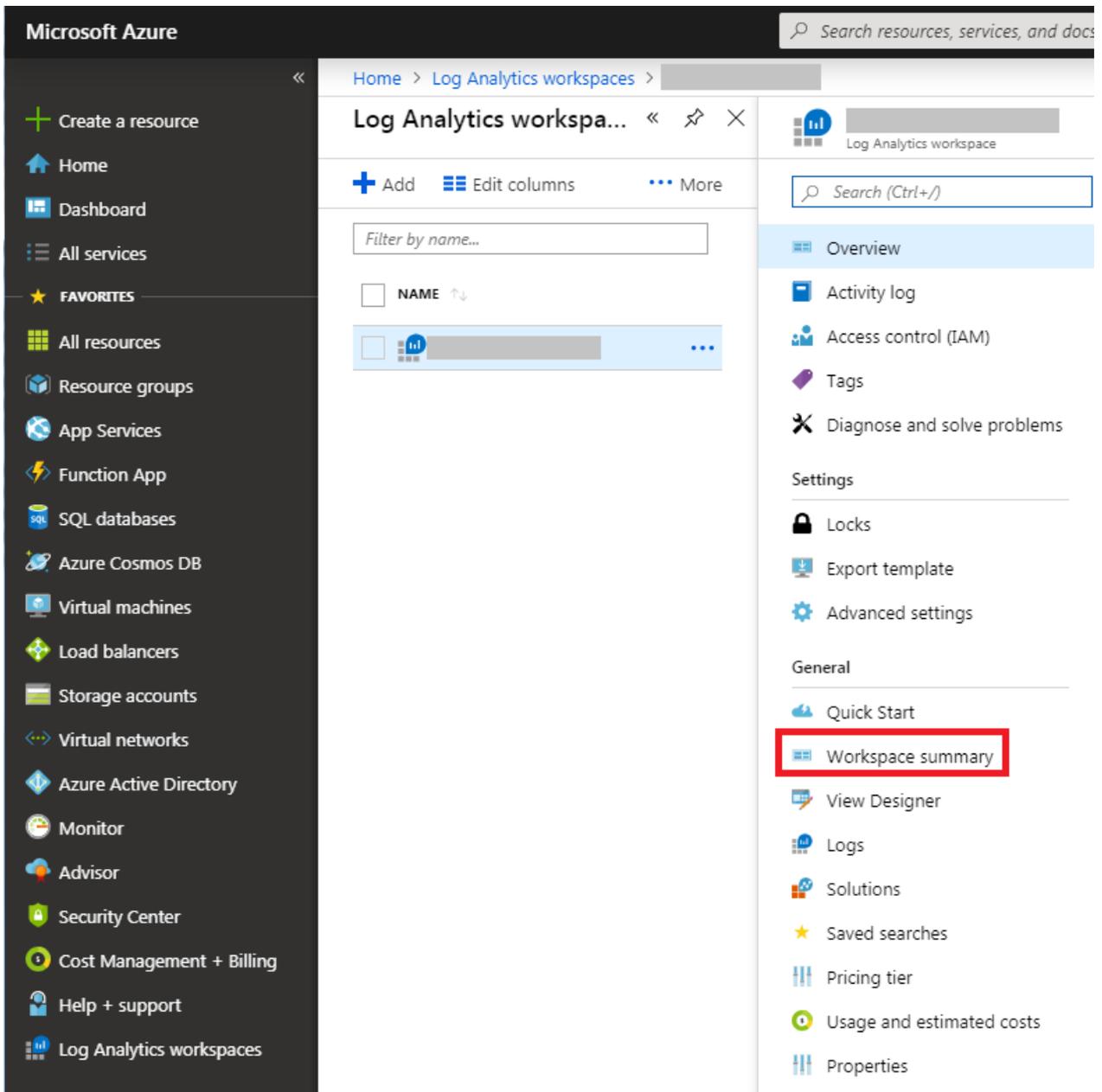
Unlike the OMS portal (which only requires permission to access the Azure Log Analytics workspace), the Azure portal also requires access to be configured to either the linked *Azure subscription* or *Azure resource group*.

To check the Log Analytics workspaces you can access, select **Log Analytics workspaces**. You should see a grid control listing all workspaces, along with the Azure subscription each is linked to:



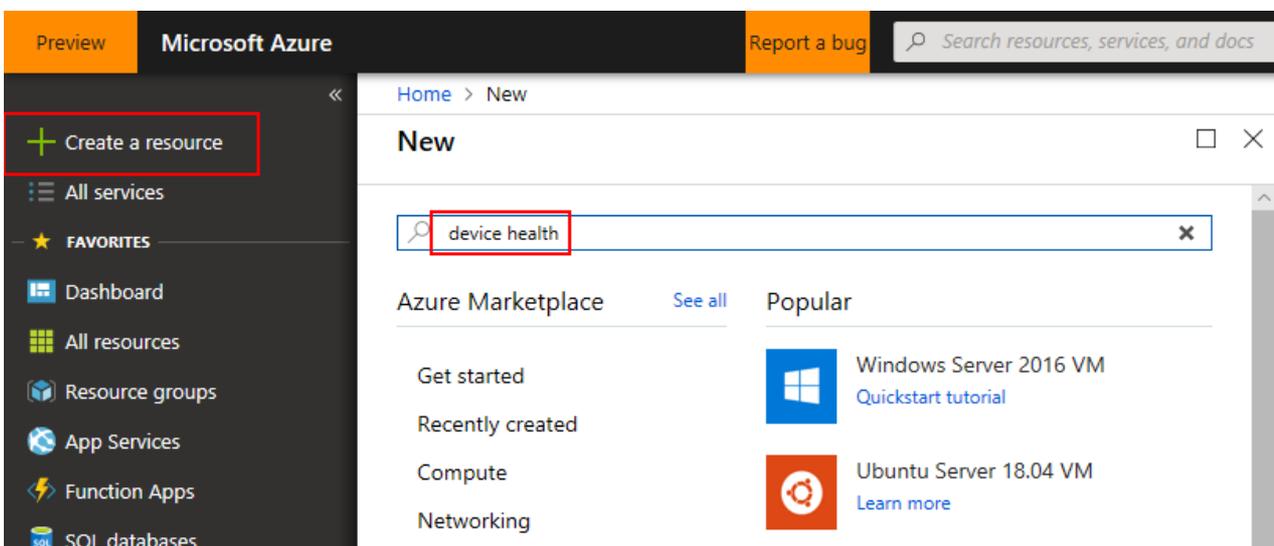
If you do not see your workspace in this view, but you are able to access the workspace from the classic portal, that means you do not have access to the workspace's Azure subscription or resource group. To remedy this, you will need to find someone with admin rights to grant you access, which they can do by selecting the subscription name and selecting **Access control (IAM)** (alternatively they can configure your access at the resource group level). They should either grant you "Log Analytics Reader" access (for read-only access) or "Log Analytics Contributor" access (which enables making changes such as creating deployment plans and changing application readiness states).

When permissions are configured, you can select the workspace and then select **Workspace summary** to see information similar to what was shown in the OMS overview page.



Adding Windows Analytics solutions

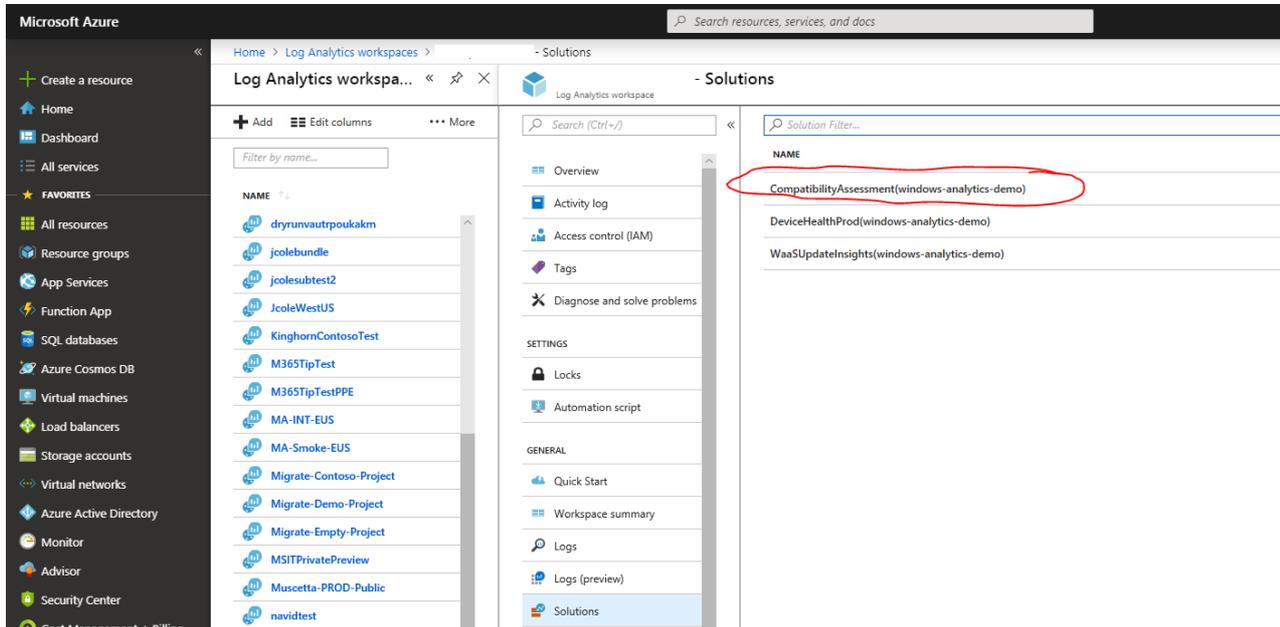
In the Azure portal, the simplest way to add Windows Analytics solutions (Upgrade Readiness, Update Compliance, and Device Health) is to select **+ Create a resource** and then type the solution name in the search box. In this example, the search is for "Device Health":



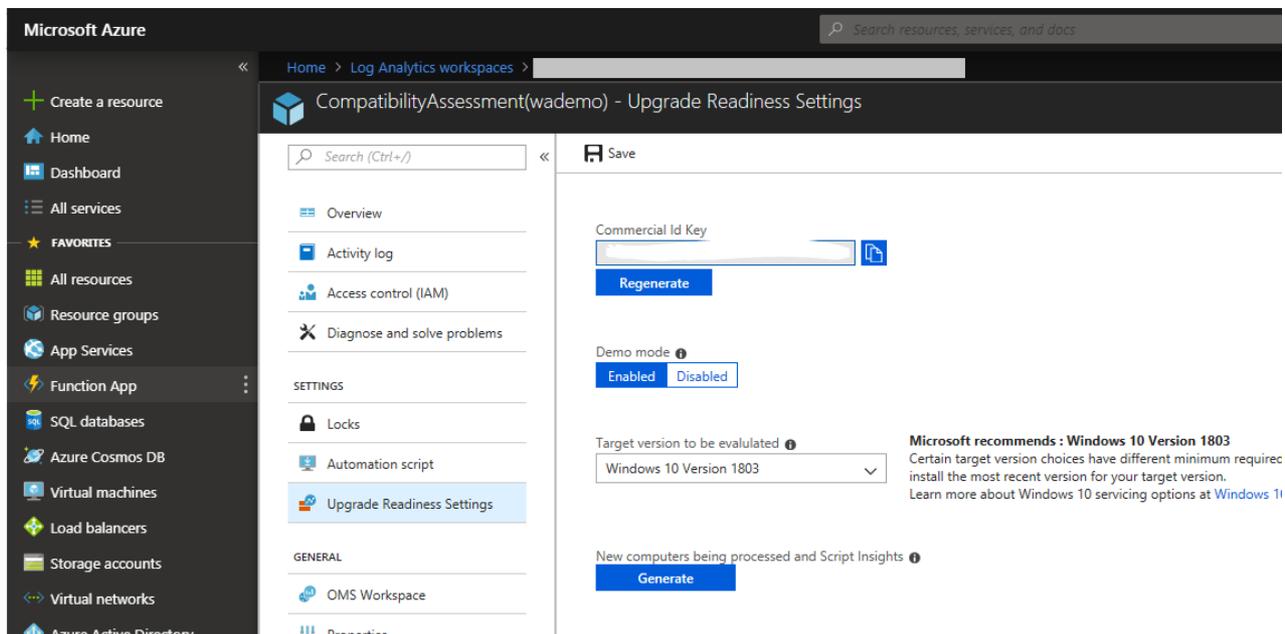
Select the solution from the list that is returned by the search, and then select **Create** to add the solution.

Navigating to Windows Analytics solutions settings

To adjust settings for a Windows Analytics solution, first navigate to the **Solutions** tab for your workspace, and then select the solution to configure. In this example, Upgrade Readiness is being adjusted by selecting **CompatibilityAssessment**:



From there, select the settings page to adjust specific settings:



NOTE

To access these settings, both the subscription and workspace require "contributor" permissions. You can view your current role and make changes in other roles by using the **Access control (IAM)** tab in Azure.

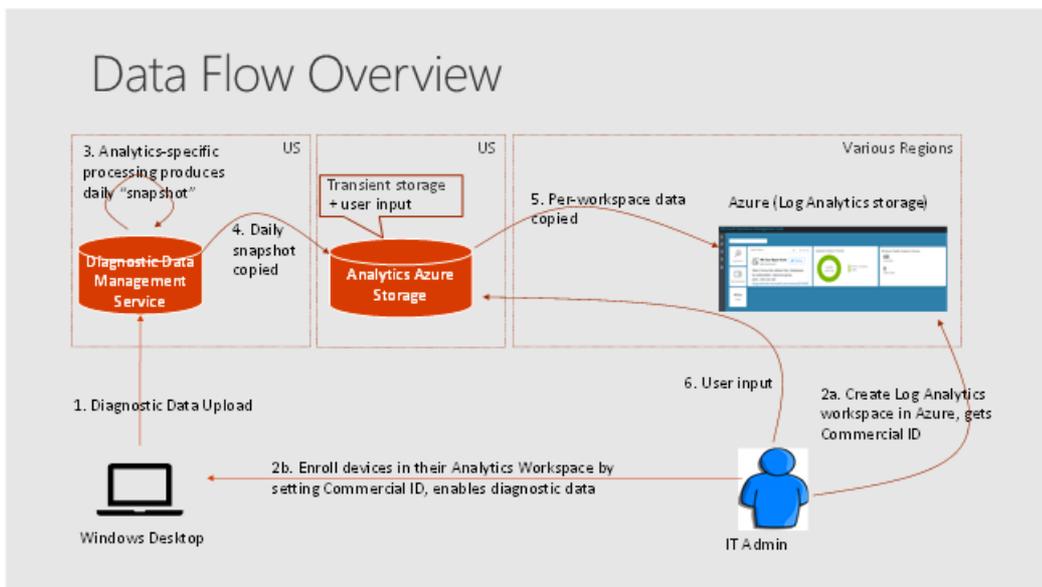
Windows Analytics and privacy

6/14/2019 • 2 minutes to read • [Edit Online](#)

Windows Analytics is fully committed to privacy, centering on these tenets:

- **Transparency:** We fully document the Windows Analytics diagnostic events (see the links for additional information) so you can review them with your company's security and compliance teams. The Diagnostic Data Viewer lets you see diagnostic data sent from a given device (see [Diagnostic Data Viewer Overview](#) for details).
- **Control:** You ultimately control the level of diagnostic data you wish to share. In Windows 10, version 1709 we added a new policy to Limit enhanced diagnostic data to the minimum required by Windows Analytics
- **Security:** Your data is protected with strong security and encryption
- **Trust:** Windows Analytics supports the Microsoft Online Service Terms

The following illustration shows how diagnostic data flows from individual devices through the Diagnostic Data Service, Azure Log Analytics storage, and to your Log Analytics workspace:



The data flow sequence is as follows:

1. Diagnostic data is sent from devices to the Microsoft Diagnostic Data Management service, which is hosted in the US.
2. An IT administrator creates an Azure Log Analytics workspace. The administrator chooses the location, copies the Commercial ID (which identifies that workspace), and then pushes Commercial ID to devices they want to monitor. This is the mechanism that specifies which devices appear in which workspaces.
3. Each day Microsoft produces a "snapshot" of IT-focused insights for each workspace in the Diagnostic Data Management service.
4. These snapshots are copied to transient storage which is used only by Windows Analytics (also hosted in US data centers) where they are segregated by Commercial ID.
5. The snapshots are then copied to the appropriate Azure Log Analytics workspace.
6. If the IT administrator is using the Upgrade Readiness solution, user input from the IT administrator (specifically, the target operating system release and the importance and upgrade readiness per app) is stored in the Windows Analytics Azure Storage. (Upgrade Readiness is the only Windows Analytics solution that takes such user input.)

See these topics for additional background information about related privacy issues:

- [Windows 10 and the GDPR for IT Decision Makers](#)
- [Configure Windows diagnostic data in your organization](#)
- [Windows 7, Windows 8, and Windows 8.1 Appraiser Telemetry Events, and Fields](#)
- [Windows 10, version 1809 basic level Windows diagnostic events and fields](#)
- [Windows 10, version 1803 basic level Windows diagnostic events and fields](#)
- [Windows 10, version 1709 basic level Windows diagnostic events and fields](#)
- [Windows 10, version 1703 basic level Windows diagnostic events and fields](#)
- [Windows 10, version 1709 enhanced diagnostic data events and fields used by Windows Analytics](#)
- [Diagnostic Data Viewer Overview](#)
- [Licensing Terms and Documentation](#)
- [Learn about security and privacy at Microsoft datacenters](#)
- [Confidence in the trusted cloud](#)
- [Trust Center](#)

Can Windows Analytics be used without a direct client connection to the Microsoft Data Management Service?

No, the entire service is powered by Windows diagnostic data, which requires that devices have this direct connectivity.

Can I choose the data center location?

Yes for Azure Log Analytics, but no for the Microsoft Data Management Service (which is hosted in the US).

Manage Windows upgrades with Upgrade Readiness

6/19/2019 • 2 minutes to read • [Edit Online](#)

Upgrading to new operating systems has traditionally been a challenging, complex, and slow process for many enterprises. Discovering applications and drivers and then testing them for potential compatibility issues have been among the biggest pain points.

With the release of Upgrade Readiness, enterprises now have the tools to plan and manage the upgrade process end to end, allowing them to adopt new Windows releases more quickly. With new Windows versions being released multiple times a year, ensuring application and driver compatibility on an ongoing basis is key to adopting new Windows versions as they are released. Windows Upgrade Readiness not only supports upgrade management from Windows 7, Windows 8.1 to Windows 10, but also Windows 10 upgrades in the [Windows as a service](#) model.

Microsoft developed Upgrade Readiness in response to demand from enterprise customers looking for additional direction and details about upgrading to Windows 10. Upgrade Readiness was built taking into account multiple channels of customer feedback, testing, and Microsoft's experience upgrading millions of devices to Windows 10.

With Windows diagnostic data enabled, Upgrade Readiness collects system, application, and driver data for analysis. We then identify compatibility issues that can block an upgrade and suggest fixes when they are known to Microsoft.

Use Upgrade Readiness to get:

- A visual workflow that guides you from pilot to production
- Detailed computer and application inventory
- Powerful computer level search and drill-downs
- Guidance and insights into application and driver compatibility issues, with suggested fixes
- Data driven application rationalization tools
- Application usage information, allowing targeted validation; workflow to track validation progress and decisions
- Data export to commonly used software deployment tools, including System Center Configuration Manager

The Upgrade Readiness workflow steps you through the discovery and rationalization process until you have a list of computers that are ready to be upgraded.

Important For system, application, and driver data to be shared with Microsoft, you must configure user computers to send data. For information about what diagnostic data Microsoft collects and how that data is used and protected by Microsoft, see:

- [Configure Windows diagnostic data in your organization](#)
- [Manage connections from Windows operating system components to Microsoft services](#)
- [Windows 7, Windows 8, and Windows 8.1 appraiser diagnostic data events and fields](#)

Related topics

[Upgrade Readiness architecture](#)

[Upgrade Readiness requirements](#)

[Upgrade Readiness release notes](#)

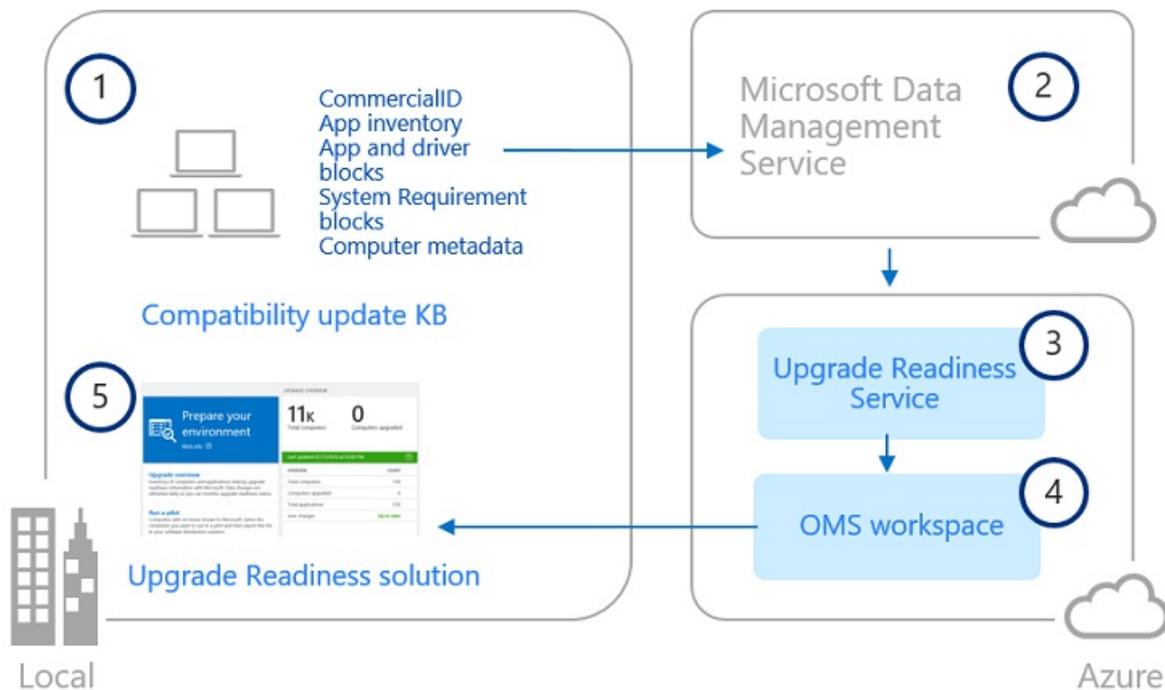
[Get started with Upgrade Readiness](#)

[Use Upgrade Readiness to manage Windows upgrades](#)

Upgrade Readiness architecture

6/14/2019 • 2 minutes to read • [Edit Online](#)

Microsoft analyzes system, application, and driver diagnostic data to help you determine when computers are upgrade-ready, allowing you to simplify and accelerate Windows upgrades in your organization. The diagram below illustrates how Upgrade Readiness components work together in a typical installation.



After you enable Windows diagnostic data on user computers and install the compatibility update KB (1), user computers send computer, application and driver diagnostic data to a secure Microsoft data center through the Microsoft Data Management Service (2). After you configure Upgrade Readiness, diagnostic data is analyzed by the Upgrade Readiness Service (3) and pushed to your workspace (4). You can then use the Upgrade Readiness solution (5) to plan and manage Windows upgrades.

For more information about what diagnostic data Microsoft collects and how that data is used and protected by Microsoft, see:

[Configure Windows diagnostic data in your organization](#)

[Manage connections from Windows operating system components to Microsoft services](#)

[Windows 7, Windows 8, and Windows 8.1 appraiser diagnostic data events and fields](#)

Related topics

[Upgrade Readiness requirements](#)

[Upgrade Readiness release notes](#)

[Get started with Upgrade Readiness](#)

Upgrade Readiness requirements

6/14/2019 • 4 minutes to read • [Edit Online](#)

This article introduces concepts and steps needed to get up and running with Upgrade Readiness. We recommend that you review this list of requirements before getting started as you may need to collect information, such as account credentials, and get approval from internal IT groups, such as your network security group, before you can start using Upgrade Readiness.

Supported upgrade paths

Windows 7 and Windows 8.1

To perform an in-place upgrade, user computers must be running the latest version of either Windows 7 SP1 or Windows 8.1. After you enable Windows diagnostic data, Upgrade Readiness performs a full inventory of computers so that you can see which version of Windows is installed on each computer.

The compatibility update that sends diagnostic data from user computers to Microsoft data centers works with Windows 7 SP1 and Windows 8.1 only. Upgrade Readiness cannot evaluate Windows XP or Windows Vista for upgrade eligibility.

If you need to update user computers to Windows 7 SP1 or Windows 8.1, use Windows Update or download and deploy the applicable package from the Microsoft Download Center.

NOTE

Upgrade Readiness is designed to best support in-place upgrades. In-place upgrades do not support migrations from BIOS to UEFI or from 32-bit to 64-bit architecture. If you need to migrate computers in these scenarios, use the wipe-and-reload method. Upgrade Readiness insights are still valuable in this scenario, however, you can ignore in-place upgrade specific guidance.

See [Windows 10 Specifications](#) for additional information about computer system requirements.

Windows 10

Keeping Windows 10 up to date involves deploying a feature update, and Upgrade Readiness tools help you prepare and plan for these Windows updates. The latest cumulative updates must be installed on Windows 10 computers to make sure that the required compatibility updates are installed. You can find the latest cumulative update on the [Microsoft Update Catalog](#).

While Upgrade Readiness can be used to assist with updating devices from Windows 10 Long-Term Servicing Channel (LTSC) to Windows 10 Semi-Annual Channel, Upgrade Readiness does not support updates to Windows 10 LTSC. The Long-Term Servicing Channel of Windows 10 is not intended for general deployment, and does not receive feature updates, therefore it is not a supported target with Upgrade Readiness. See [Windows as a service overview](#) to understand more about LTSC.

Operations Management Suite or Azure Log Analytics

Upgrade Readiness is offered as a solution in Azure Portal and Azure Log Analytics, a collection of cloud-based services for managing on premises and cloud computing environments. For more information about Azure Portal, see [Windows Analytics in the Azure Portal](#) or the Azure [Log Analytics overview](#).

If you're already using Azure Portal or Azure Log Analytics, you'll find Upgrade Readiness in the Solutions Gallery. Click the **Upgrade Readiness** tile in the gallery and then click **Add** on the solution's details page. Upgrade

Readiness is now visible in your workspace.

If you are not using Azure Portal or Azure Log Analytics, go to [Log Analytics](#) on Microsoft.com and select **Start free** to start the setup process. During the process, you'll create a workspace and add the Upgrade Readiness solution to it.

IMPORTANT

You can use either a Microsoft Account or a Work or School account to create a workspace. If your company is already using Azure Active Directory, use a Work or School account when you sign in to Azure Portal. Using a Work or School account allows you to use identities from your Azure AD to manage permissions in Azure Portal. You also need an Azure subscription to link to your Azure Portal workspace. The account you used to create the workspace must have administrator permissions on the Azure subscription in order to link the workspace to the Azure account. Once the link has been established, you can revoke the administrator permissions.

System Center Configuration Manager integration

Upgrade Readiness can be integrated with your installation of Configuration Manager. For more information, see [Integrate Upgrade Readiness with System Center Configuration Manager](#).

Important information about this release

Before you get started configuring Upgrade Analytics, review the following tips and limitations about this release.

Upgrade Readiness does not support on-premises Windows deployments. Upgrade Readiness is built as a cloud service, which allows Upgrade Readiness to provide you with insights based on the data from user computers and other Microsoft compatibility services. Cloud services are easy to get up and running and are cost-effective because there is no requirement to physically implement and maintain services on-premises.

In-region data storage requirements. Windows diagnostic data from user computers is encrypted, sent to, and processed at Microsoft-managed secure data centers located in the US. Our analysis of the upgrade readiness-related data is then provided to you through the Upgrade Readiness solution in Azure Portal. Upgrade Readiness is supported in all Azure regions; however, selecting an international Azure region does not prevent diagnostic data from being sent to and processed in Microsoft's secure data centers in the US.

Tips

- When viewing inventory items in table view, the maximum number of rows that can be viewed and exported is limited to 5,000. If you need to view or export more than 5,000 items, reduce the scope of the query so you can export a list with fewer items.
- Sorting data by clicking a column heading may not sort your complete list of items. For information about how to sort data in Azure Portal, see [Sorting DocumentDB data using Order By](#).

Get started

See [Get started with Upgrade Readiness](#) for detailed, step-by-step instructions for configuring Upgrade Readiness and getting started on your Windows upgrade project.

Get started with Upgrade Readiness

6/14/2019 • 3 minutes to read • [Edit Online](#)

IMPORTANT

The OMS portal has been deprecated; you should start using the [Azure portal](#) instead as soon as possible. Many experiences are the same in the two portals, but there are some key differences. See [Windows Analytics in the Azure Portal](#) for steps to use Windows Analytics in the Azure portal. For much more information about the transition from OMS to Azure, see [OMS portal moving to Azure](#).

This topic explains how to obtain and configure Upgrade Readiness for your organization.

You can use Upgrade Readiness to plan and manage your upgrade project end-to-end. Upgrade Readiness works by establishing communications between computers in your organization and Microsoft. Upgrade Readiness collects computer, application, and driver data for analysis. This data is used to identify compatibility issues that can block your upgrade and to suggest fixes that are known to Microsoft.

Before you begin, consider reviewing the following helpful information:

- [Upgrade Readiness requirements](#): Provides detailed requirements to use Upgrade Readiness.
- [Upgrade Readiness blog](#): Contains announcements of new features and provides helpful tips for using Upgrade Readiness.

If you are using System Center Configuration Manager, also check out information about how to integrate Upgrade Readiness with Configuration Manager: [Integrate Upgrade Readiness with System Center Configuration Manager](#).

When you are ready to begin using Upgrade Readiness, perform the following steps:

1. Review [data collection and privacy](#) information.
2. [Add the Upgrade Readiness solution to your Azure subscription](#).
3. [Enroll devices in Windows Analytics](#).
4. [Use Upgrade Readiness to manage Windows Upgrades](#) once your devices are enrolled.

Data collection and privacy

To enable system, application, and driver data to be shared with Microsoft, you must configure user computers to send data. For information about what diagnostic data Microsoft collects and how that data is used and protected by Microsoft, see the following topics, refer to [Frequently asked questions and troubleshooting Windows Analytics](#), which discusses the issues and provides links to still more detailed information.

Add the Upgrade Readiness solution to your Azure subscription

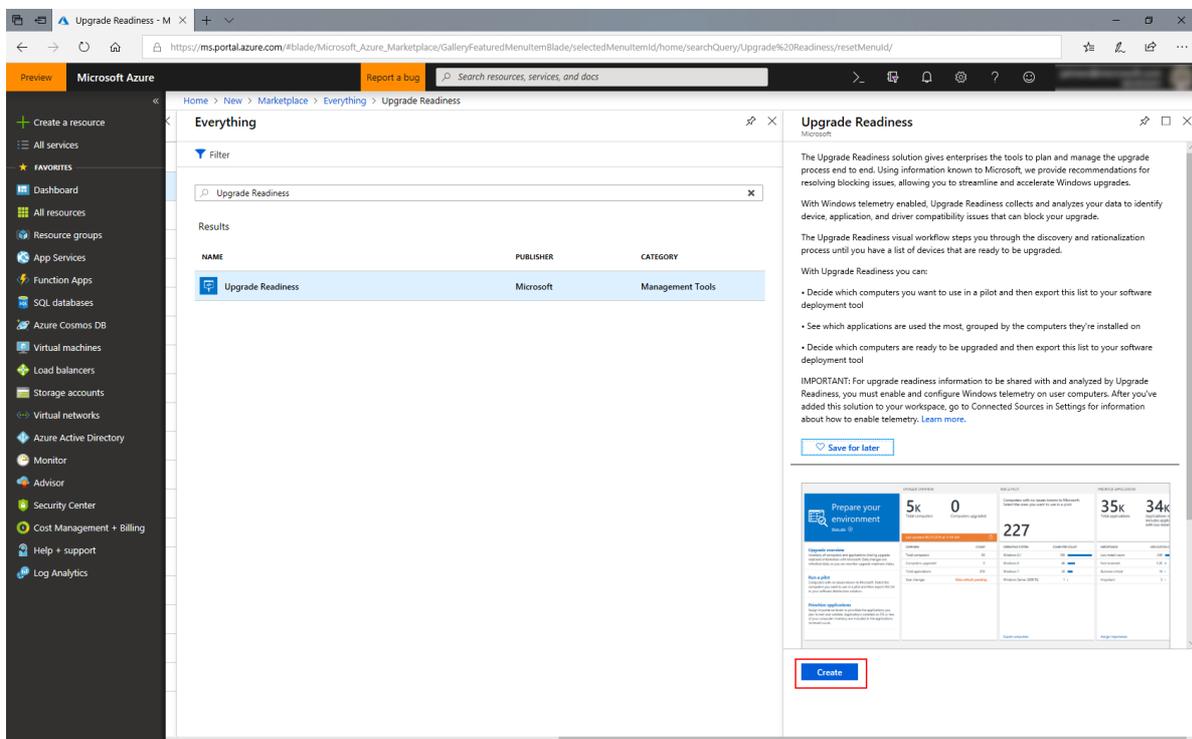
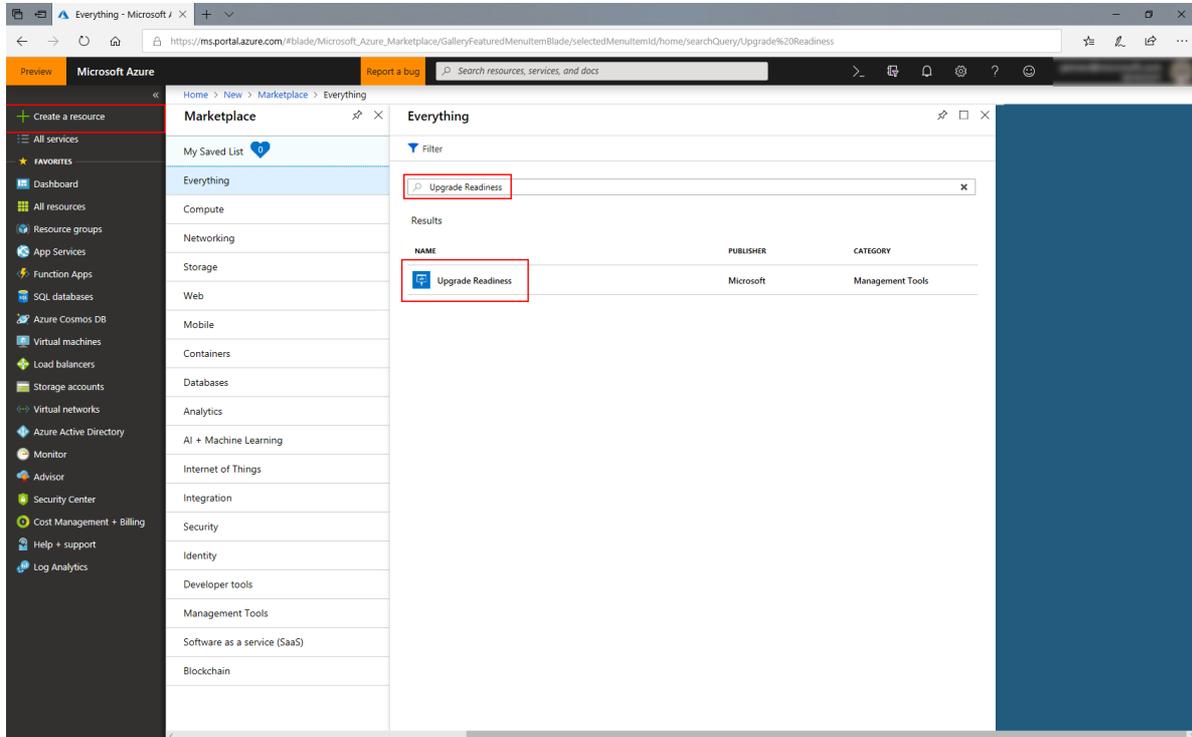
Upgrade Readiness is offered as a *solution* which you link to a new or existing [Azure Log Analytics workspace](#) within your Azure *subscription*. To configure this, follows these steps:

1. Sign in to the [Azure Portal](#) with your work or school account or a Microsoft account. If you don't already have an Azure subscription you can create one (including free trial options) through the portal.

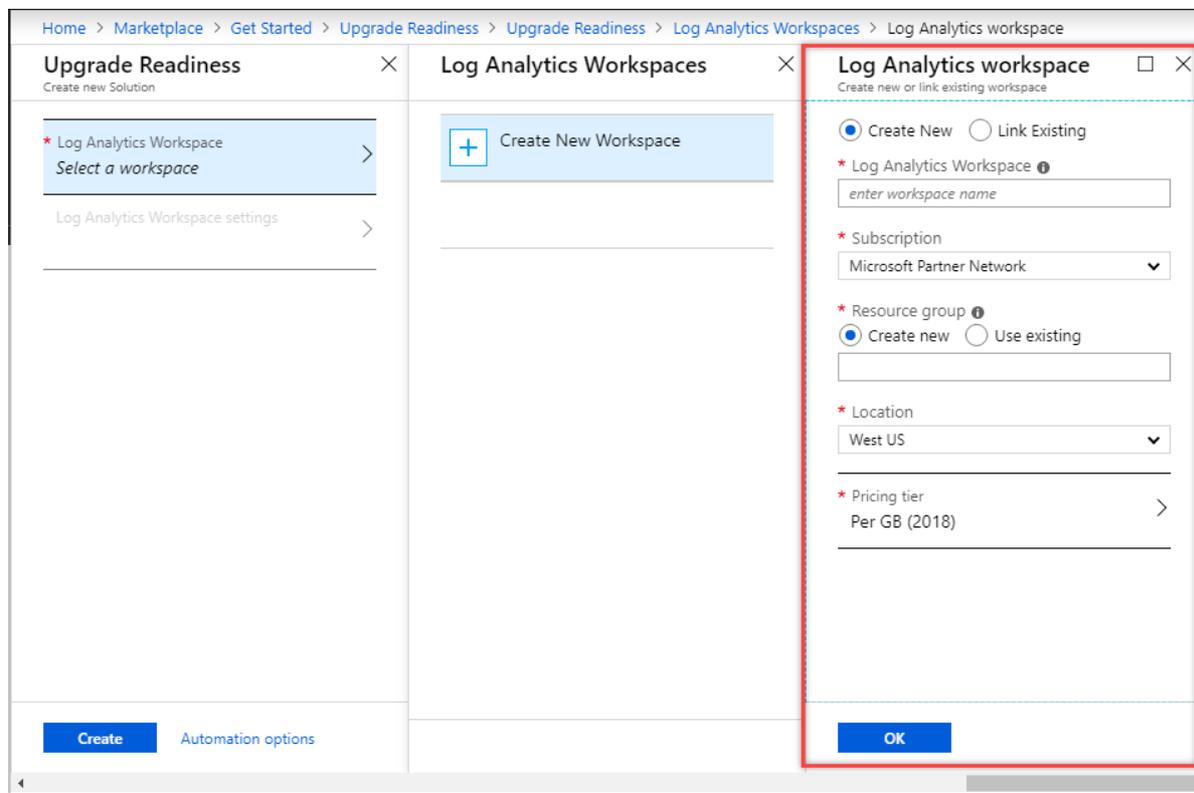
NOTE

Upgrade Readiness is included at no additional cost with Windows 10 Professional, Education, and Enterprise editions. An Azure subscription is required for managing and using Upgrade Readiness, but no Azure charges are expected to accrue to the subscription as a result of using Upgrade Readiness.

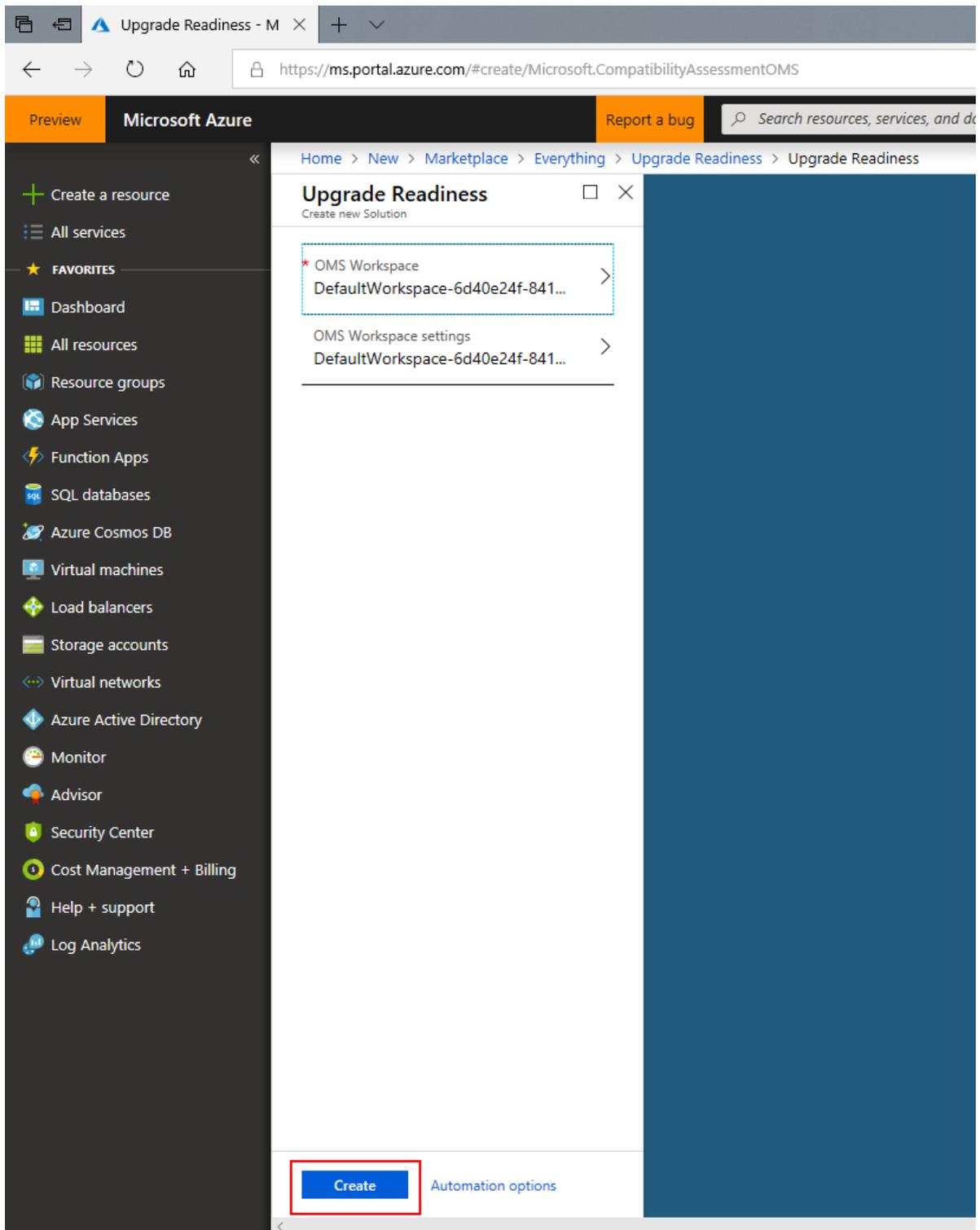
2. In the Azure portal select **Create a resource**, search for "Upgrade Readiness", and then select **Create** on the **Upgrade Readiness** solution.



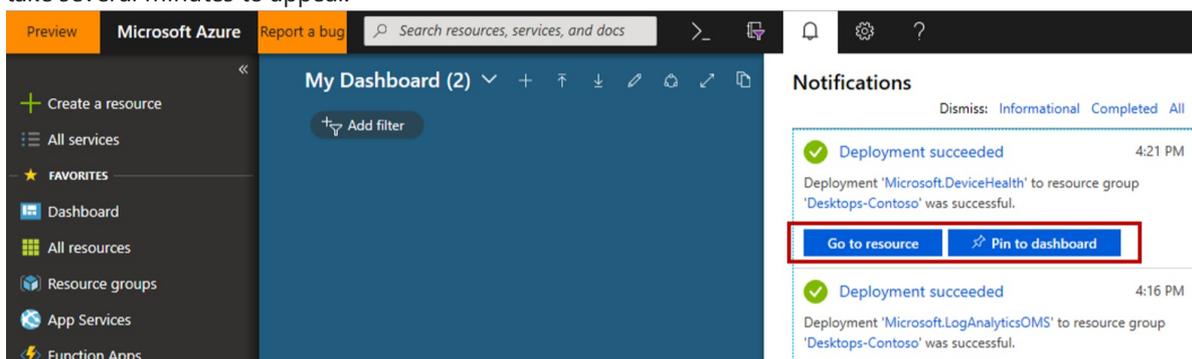
3. Choose an existing workspace or create a new workspace to host the Upgrade Readiness solution.



- If you are using other Windows Analytics solutions (Device Health or Update Compliance) you should add Upgrade Readiness to the same workspace.
 - If you are creating a new workspace, and your organization does not have policies governing naming conventions and structure, consider the following workspace settings to get started:
 - Choose a workspace name which reflects the scope of planned usage in your organization, for example *PC-Analytics*.
 - For the resource group setting select **Create new** and use the same name you chose for your new workspace.
 - For the location setting, choose the Azure region where you would prefer the data to be stored.
 - For the pricing tier select **per GB**.
4. Now that you have selected a workspace, you can go back to the Upgrade Readiness blade and select **Create**.



5. Watch for a Notification (in the Azure portal) that "Deployment 'Microsoft.CompatibilityAssessmentOMS' to resource group 'YourResourceGroupName' was successful." and then select **Go to resource**. This might take several minutes to appear.



- Suggestion: Choose the **Pin to Dashboard** option to make it easy to navigate to your newly added

Upgrade Readiness solution.

- Suggestion: If a "resource unavailable" error occurs when navigating to the solution, try again after one hour.

Enroll devices in Windows Analytics

Once you've added Upgrade Readiness to a workspace in your Azure subscription, you can start enrolling the devices in your organization. For full instructions, see [Enrolling devices in Windows Analytics](#).

Use Upgrade Readiness to manage Windows Upgrades

Now that your devices are enrolled, you can move on to [Use Upgrade Readiness to manage Windows Upgrades](#).

Upgrade Readiness deployment script

6/26/2019 • 16 minutes to read • [Edit Online](#)

To automate the steps provided in [Get started with Upgrade Readiness](#), and to troubleshoot data sharing issues, you can run the [Upgrade Readiness deployment script](#), developed by Microsoft.

IMPORTANT

Upgrade Readiness was previously called Upgrade Analytics. References to Upgrade Analytics in any scripts or online content pertain to the Upgrade Readiness solution.

For detailed information about using the Upgrade Readiness (also known as upgrade analytics) deployment script, see the [Upgrade Analytics blog](#).

The following guidance applies to version 11.11.16 or later of the Upgrade Readiness deployment script. If you are using an older version, download the latest from the [Download Center](#).

The Upgrade Readiness deployment script does the following:

1. Sets commercial ID key + CommercialDataOptIn + RequestAllAppraiserVersions keys.
2. Verifies that user computers can send data to Microsoft.
3. Checks whether the computer has a pending restart.
4. Verifies that the latest version of KB package 10.0.x is installed (version 10.0.14348 or later is required, but version 10.0.14913 or later is recommended).
5. If enabled, turns on verbose mode for troubleshooting.
6. Initiates the collection of the diagnostic data that Microsoft needs to assess your organization's upgrade readiness.
7. If enabled, displays the script's progress in a cmd window, providing you immediate visibility into issues (success or fail for each step) and/or writes to log file.

Running the script

There should be no performance impact caused by the script. The script is a light wrapper of Windows in-box components that undergo performance testing and optimization to avoid any performance impact. However, typically the script is scheduled to be run outside of working hours.

Do not run the script at each sign-on. It is recommended to run the script once every 30 days.

The length of time the script takes to run on each system depends on the number of apps and drivers, and the type of hardware. Anti-virus software scanning simultaneously can increase the script run time, but the script should require no longer than 10 minutes to run, and typically the time is much shorter. If the script is observed running for an extended period of time, please run the Pilot script, and collect logs to share with Microsoft. Log files are created in the drive that is specified in the RunConfig.bat file. By default this is set to: **%SystemDrive%\UADiagnostics**.

To run the Upgrade Readiness deployment script:

1. Download the [Upgrade Readiness deployment script](#) and extract the .zip file. Inside, there are two folders: **Pilot** and **Deployment**. The **Pilot** folder contains advanced logging that can help troubleshoot issues and

is intended to be run from an elevated command prompt. The **Deployment** folder offers a lightweight script intended for broad deployment through ConfigMgr or other software deployment system. We recommend manually running the Pilot version of the script on 5-10 machines to verify that everything is configured correctly. Once you have confirmed that data is flowing successfully, proceed to run the Deployment version throughout your organization.

2. Edit the following parameters in RunConfig.bat:

- a. Provide a storage location for log information. You can store log information on a remote file share or a local directory. If the script is blocked from creating the log file for the given path, it creates the log files in the drive with the Windows directory. Example: %SystemDrive%\UADiagnostics
- b. Input your commercial ID key. To find your commercial ID, first navigate to the **Solutions** tab for your workspace, and then select the solution. From there, select the **Settings** page, where you can find and copy your commercial ID:
- c. By default, the script sends log information to both the console and the log file. To change the default behavior, use one of the following options:

```
logMode = 0 log to console only
```

```
logMode = 1 log to file and console
```

```
logMode = 2 log to file only
```

3. To enable Internet Explorer data collection, set AllowIEData to IEDataOptIn. By default, AllowIEData is set to Disable. Then use one of the following options to determine what Internet Explorer data can be collected:

```
IEOptInLevel = 0 Internet Explorer data collection is disabled
```

```
IEOptInLevel = 1 Data collection is enabled for sites in the Local intranet + Trusted sites + Machine local zones
```

```
IEOptInLevel = 2 Data collection is enabled for sites in the Internet + Restricted sites zones
```

```
IEOptInLevel = 3 Data collection is enabled for all sites
```

4. A recent version (03.02.17) of the deployment script is configured to collect and send diagnostic and debugging data to Microsoft. If you wish to disable sending diagnostic and debugging data to Microsoft, set **AppInsightsOptIn = false**. By default, **AppInsightsOptIn** is set to **true**.

The data that is sent is the same data that is collected in the text log file that captures the events and error codes while running the script. This file is named in the following format:

UA_yyyy_mm_dd_hh_mm_ss_machineID.txt. Log files are created in the drive that is specified in the RunConfig.bat file. By default this is set to: **%SystemDrive%\UADiagnostics**.

This data gives us the ability to determine the status of your machines and to help troubleshoot issues. If you choose to opt-in to and send this data to Microsoft, you must also allow https traffic to be sent to the following wildcard endpoints:

```
*vortex*.data.microsoft.com
```

```
*settings*.data.microsoft.com
```

5. The latest version (03.28.2018) of the deployment script configures insider builds to continue to send the device name to the diagnostic data management service and the analytics portal. If you do not want to have insider builds send the device name sent to analytics and be available in the analytics portal, set **DeviceNameOptIn = false**. By default it is true, which preserves the behavior on previous versions of Windows. This setting only applies to insider builds. Note that the device name is also sent to AppInsights,

so to ensure the device name is not sent to either place you would need to also set **AppInsightsOptIn = false**.

- After you finish editing the parameters in RunConfig.bat, you are ready to run the script. If you are using the Pilot version, run RunConfig.bat from an elevated command prompt. If you are using the Deployment version, use ConfigMgr or other software deployment service to run RunConfig.bat as system.

Exit codes

The deployment script displays the following exit codes to let you know if it was successful, or if an error was encountered.

EXIT CODE	SUGGESTED FIX
0 - Success	N/A
1 - Unexpected error occurred while executing the script.	The files in the deployment script are likely corrupted. Download the latest script from the download center and try again.
2 - Error when logging to console. \$logMode = 0. (console only)	Try changing the \$logMode value to 1 and try again. \$logMode value 1 logs to both console and file.
3 - Error when logging to console and file. \$logMode = 1.	Verify that you have set the logPath parameter in RunConfig.bat, and that the configuration script has access to connect and write to this location.
4 - Error when logging to file. \$logMode = 2.	Verify that you have set the logPath parameter in RunConfig.bat, and that the configuration script has access to connect and write to this location.
5 - Error when logging to console and file. \$logMode = unknown.	Verify that you have set the logPath parameter in RunConfig.bat, and that the configuration script has access to connect and write to this location.
6 - The commercialID parameter is set to unknown.	Modify the runConfig.bat file to set the CommercialID value. The value for parameter in the runconfig.bat file should match the Commercial ID key for your workspace. See Generate your Commercial ID key for instructions on generating a Commercial ID key for your workspace.
8 - Failure to create registry key path: HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection . The Commercial Id property is set at the following registry key path: HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection	Verify that the context under which the script is running has access to the registry key.
9 - The script failed to write Commercial Id to registry.	
Error creating or updating registry key: CommercialId at HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection	Verify that the context under which the script is running has access to the registry key.
10 - Error when writing CommercialDataOptIn to the registry at HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection	Verify that the deployment script is running in a context that has access to the registry key.

EXIT CODE	SUGGESTED FIX
<p>11 - Function SetupCommercialId failed with an unexpected exception. The SetupCommercialId function updates the Commercial Id at the registry key path: HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection</p>	<p>Verify that the configuration script has access to this location.</p>
<p>12 - Can't connect to Microsoft - Vortex. Check your network/proxy settings.</p>	<p>Http Get on the end points did not return a success exit code. For Windows 10, connectivity is verified by connecting to https://v10.vortex-win.data.microsoft.com/health/keepalive. For previous operating systems, connectivity is verified by connecting to https://vortex-win.data.microsoft.com/health/keepalive. If there is an error verifying connectivity, this will prevent the collected data from being sent to Upgrade Readiness. To resolve this issue, verify that the required endpoints are correctly whitelisted. For more information, see Enrolling devices in Windows Analytics</p>
<p>13 - Can't connect to Microsoft - setting.</p>	<p>An error occurred connecting to https://settings.data.microsoft.com/qos. This error will prevent the collected data from being sent to Upgrade Readiness. To resolve this issue, verify that the required endpoints are correctly whitelisted. For more information, see Enrolling devices in Windows Analytics. Verify that the required endpoints are whitelisted correctly. See Whitelist select endpoints for more details.</p>
<p>14 - Can't connect to Microsoft - compatexchange. An error occurred connecting to CompatibilityExchangeService.svc.</p>	<p>This error will prevent the collected data from being sent to Upgrade Readiness. To resolve this issue, verify that the required endpoints are correctly whitelisted. For more information, see Enrolling devices in Windows Analytics.</p>
<p>15 - Function CheckVortexConnectivity failed with an unexpected exception.</p>	<p>This error will prevent the collected data from being sent to Upgrade Readiness. To resolve this issue, verify that the required endpoints are correctly whitelisted. For more information, see Enrolling devices in Windows Analytics. Check the logs for the exception message and the HRESULT.</p>
<p>16 - The computer requires a reboot before running the script.</p>	<p>Restart the device to complete the installation of the compatibility update and related updates. Reboot the computer before running the Upgrade Readiness deployment script.</p>
<p>17 - Function CheckRebootRequired failed with an unexpected exception.</p>	<p>Restart the device to complete installation of the compatibility update and related updates. Check the logs for the exception message and the HRESULT.</p>
<p>18 - Appraiser KBs not installed or appraiser.dll not found.</p>	<p>Either the Appraiser-related updates are not installed, or the appraiser.dll file was not found. For more information, see appraiser diagnostic data events and fields information in the Data collection and privacy topic.</p>
<p>19 - Function CheckAppraiserKB, which checks the compatibility update KBs, failed with unexpected exception.</p>	<p>Check the logs for the Exception message and HRESULT. The script will not run further if this error is not fixed.</p>

EXIT CODE	SUGGESTED FIX
20 - An error occurred when creating or updating the registry key RequestAllAppraiserVersions at HKLM:\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\AppCompatFlags\Appraiser	The registry key is required for data collection to work correctly. Verify that the script is running in a context that has access to the registry key.
21 - Function SetRequestAllAppraiserVersions failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
22 - RunAppraiser failed with unexpected exception.	Check the logs for the exception message and HRESULT. Check the %windir%\System32 directory for the file CompatTelRunner.exe . If the file does not exist, reinstall the required compatibility updates which include this file, and check your organization's Group Policy to verify it does not remove this file.
23 - Error finding system variable %WINDIR% .	Verify that this environment variable is configured on the computer.
24 - The script failed when writing IEDataOptIn to the registry. An error occurred when creating registry key IEOptInLevel at HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection	This is a required registry key for IE data collection to work correctly. Verify that the deployment script is running in a context that has access to the registry key. Check the logs for the exception message and HRESULT.
25 - The function SetIEDataOptIn failed with unexpected exception.	Check the logs for the exception message and HRESULT.
27 - The script is not running under System account.	The Upgrade Readiness configuration script must be run as System .
28 - Could not create log file at the specified logPath .	Make sure the deployment script has access to the location specified in the logPath parameter.
29 - Connectivity check failed for proxy authentication.	Install cumulative updates on the device and enable the DisableEnterpriseAuthProxy authentication proxy setting. The DisableEnterpriseAuthProxy setting is enabled by default for Windows 7. For Windows 8.1 computers, set the DisableEnterpriseAuthProxy setting to 0 (not disabled). For more information on authentication proxy support, see Authentication proxy support added in new version (12.28.16) of the Upgrade Readiness deployment script .
30 - Connectivity check failed. Registry key property DisableEnterpriseAuthProxy is not enabled.	The DisableEnterpriseAuthProxy setting is enabled by default for Windows 7. For Windows 8.1 computers, set the DisableEnterpriseAuthProxy setting to 0 (not disabled). For more information on authentication proxy support, see this blog post .
31 - There is more than one instance of the Upgrade Readiness data collector running at the same time on this computer. Use Task Manager to check if CompatTelRunner.exe is running, and wait until it has completed to rerun the script. The Upgrade Readiness task is scheduled by default to run daily at 0300.	

EXIT CODE	SUGGESTED FIX
32 - Appraiser version on the machine is outdated.	The configuration script detected a version of the compatibility update module that is older than the minimum required to correctly collect the data required by Upgrade Readiness solution. Use the latest version of the compatibility update for Windows 7 SP1/Windows 8.1.
33 - CompatTelRunner.exe exited with an exit code	CompatTelRunner.exe runs the appraise task on the device. If it fails, it will provide a specific exit code. The script will return exit code 33 when CompatTelRunner.exe itself exits with an exit code. Check the logs for more details. Also see the Note following this table for additional steps to follow.
34 - Function CheckProxySettings failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
35 - Function CheckAuthProxy failed with an unexpected exception. Check the logs for the exception message and HRESULT.	
36 - Function CheckAppraiserEndPointsConnectivity failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
37 - Diagnose_internal.cmd failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
38 - Function Get-SqmID failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
39 - For Windows 10: AllowTelemetry property is not set to 1 or higher at registry key path HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection or HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection	For Windows 10 devices, the AllowTelemetry property should be set to 1 or greater to enable data collection. The script will return an error if this is not true. For more information, see Configure Windows diagnostic data in your organization .
40 - Function CheckTelemetryOptIn failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
41 - The script failed to impersonate the currently logged on user.	The script mimics the UTC client to collect upgrade readiness data. When auth proxy is set, the UTC client impersonates the user that is logged on. The script also tries to mimic this, but the process failed.
42 - Function StartImpersonatingLoggedOnUser failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
43 - Function EndImpersonatingLoggedOnUser failed with an unexpected exception.	Check the logs for the exception message and HRESULT.
44 - Diagtrack.dll version is old, so Auth Proxy will not work.	Update the device using Windows Update or Windows Server Update Services.
45 - Diagtrack.dll was not found.	Update the device using Windows Update or Windows Server Update Services.

EXIT CODE	SUGGESTED FIX
48 - CommercialID mentioned in RunConfig.bat should be a GUID.	Copy the commercial ID from your workspace. To find your commercial ID, first navigate to the Solutions tab for your workspace in Azure Portal, and then select the solution. From there, select the Settings page, where you can find and copy your commercial ID.
50 - Diagtrack Service is not running.	The Diagtrack service is required to send data to Microsoft. Enable and run the "Connected User Experiences and Telemetry" service.
51 - RunCensus failed with an unexpected exception.	RunCensus explicitly runs the process used to collect device information. The method failed with an unexpected exception. The most common cause is incorrect setup of diagnostic data. Check the ExceptionHResult and ExceptionMessage for more details.
52 - DeviceCensus.exe not found on a Windows 10 machine.	On computers running Windows 10, the process devicecensus.exe should be present in the \system32 directory. Error code 52 is returned if the process was not found. Ensure that it exists at the specified location.
53 - There is a different CommercialID present at the GPO path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection . This will take precedence over the CommercialID provided in the script.	Provide the correct CommercialID at the GPO location.
54 - Microsoft Account Sign In Assistant Service is Disabled.	This service is required for devices running Windows 10. The diagnostic data client relies on the Microsoft Account Sign In Assistant (MSA) to get the Global Device ID for the device. Without the MSA service running, the global device ID will not be generated and sent by the client and Windows Update will no longer offer feature updates to devices running Windows 10 1709 or higher. See Feature updates are not being offered while other updates are .
55 - SetDeviceNameOptIn function failed to create registry key path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection	The function SetDeviceNameOptIn sets the registry key value which determines whether to send the device name in diagnostic data. The function tries to create the registry key path if it does not already exist. Verify that the account has the correct permissions to change or add registry keys.
56 - SetDeviceNameOptIn function failed to create property AllowDeviceNameInTelemetry at registry key path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection	Verify that the account has the correct permissions to change or add registry keys.
57 - SetDeviceNameOptIn function failed to update AllowDeviceNameInTelemetry property to value 1 at registry key path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection	Verify that the account has the correct permissions to change or add registry keys.
58 - SetDeviceNameOptIn function failed with unexpected exception	The function SetDeviceNameOptIn failed with an unexpected exception.

EXIT CODE	SUGGESTED FIX
59 - CleanupOneSettings failed to delete LastPersistedEventTimeOrFirstBoot property at registry key path: HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\Diagtrack	The CleanupOneSettings function clears some of the cached values needed by the Appraiser which is the data collector on the monitored device. This helps in the download of the most recent for accurate running of the data collector. Verify that the account has the correct permissions to change or add registry keys.
60 - CleanupOneSettings failed to delete registry key: HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\Diagtrack\SettingsRequests	Verify that the account has the correct permissions to change or add registry keys.
61 - CleanupOneSettings failed with an exception	CleanupOneSettings failed with an unexpected exception.
63 - Diagnostic data is disabled for the device	If AllowTelemetry == 0, devices cannot send diagnostic data. To resolve this, set the AllowTelemetry value at HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection .

NOTE

Additional steps to follow if you receive exit code 33

Check the exit code for any of these messages:

- CompatTelRunner.exe exited with last error code: 0x800703F1
- CompatTelRunner.exe exited with last error code: 0x80070005
- CompatTelRunner.exe exited with last error code: 0x80080005

If the exit code includes any of those messages, then run these commands from an elevated command prompt:

1. Net stop diagtrack
2. Net stop pcasvc
3. Net stop dps
4. Del %windir%\appcompat\programs\amcache.hve
5. reg delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags" /v AmiHivePermissionsCorrect /f
6. reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags" /v LogFlags /t REG_DWORD /d 4 /f
7. Net start diagtrack
8. Net start pcasvc
9. Net start dps

Then run the Enterprise Config script (RunConfig.bat) again.

If the script still fails, then send mail to uasupport@microsoft.com including log files from the RunConfig.bat script. These log files are stored on the drive that is specified in the RunConfig.bat file. By default this is set to

%SystemDrive%\UADiagnostics. The log file is named with the format **UA_yyyy_mm_dd_hh_mm_ss_machineID.txt**.

There will be some additional logs generated under your **<system drive>\Windows\Temp** directory with the names similar to **AsLog_....txt**. You should send those logs as well.

Use Upgrade Readiness to manage Windows upgrades

6/14/2019 • 3 minutes to read • [Edit Online](#)

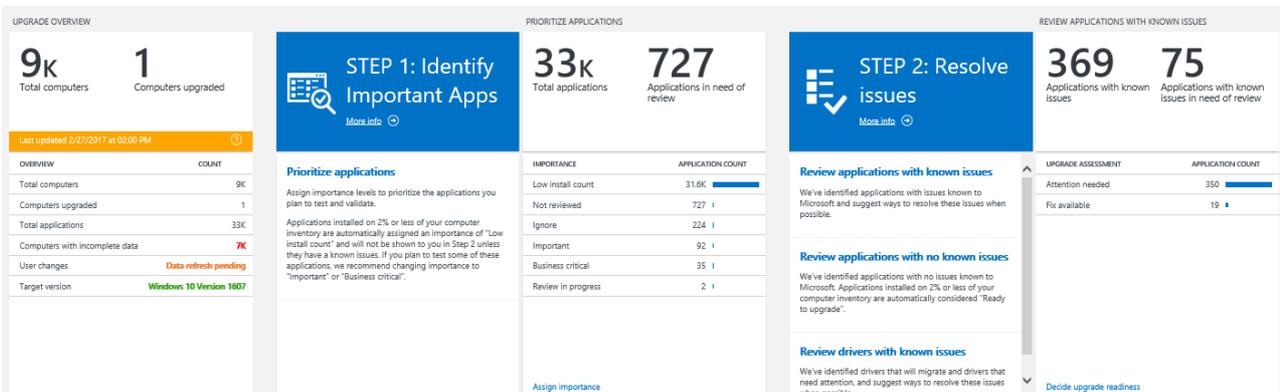
IMPORTANT

The OMS portal has been deprecated, so you need to switch to the [Azure portal](#) now. The two portals offer the same experience, with some key differences. Learn how to use [Windows Analytics in the Azure Portal](#). Find out more about the [OMS portal moving to Azure](#), or jump right in and [Get started with Upgrade Readiness](#).

You can use Upgrade Readiness to prioritize and work through application and driver issues, assign and track issue resolution status, and identify computers that are ready to upgrade. Upgrade Readiness enables you to deploy Windows with confidence, knowing that you've addressed potential blocking issues.

- Based on diagnostic data from user computers, Upgrade Readiness identifies application and driver compatibility issues that may block Windows upgrades, allowing you to make data-driven decisions about your organization's upgrade readiness.
- Information is refreshed daily so you can monitor upgrade progress. Any changes your team makes, such as assigning application importance and marking applications as ready to upgrade, are reflected 24 hours after you make them.

When you are ready to begin the upgrade process, a workflow is provided to guide you through critical high-level tasks.



Each step in the workflow is enumerated using blue tiles. Helpful data is provided on white tiles to help you get started, to monitor your progress, and to complete each step.

Important: You can use the [Target version](#) setting to evaluate computers that are running a specified version of Windows before starting the Upgrade Readiness workflow. By default, the Target version is configured to the released version of Windows 10 for the Current Branch for Business (CBB).

The following information and workflow is provided:

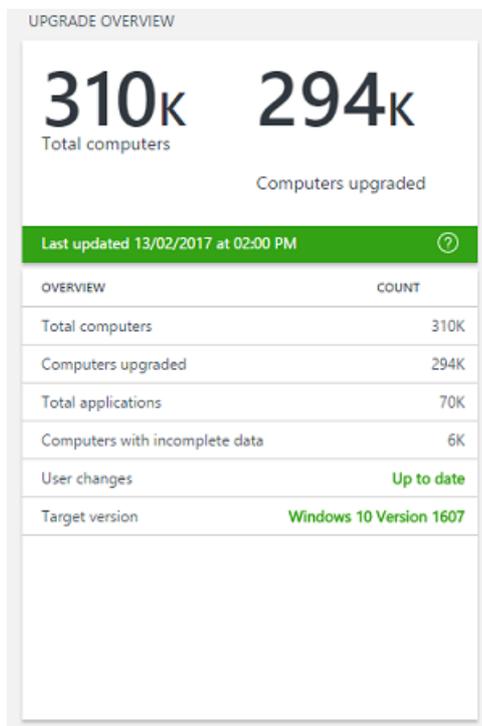
- [Upgrade overview](#): Review compatibility and usage information about computers, applications, and drivers.
- [Step 1: Identify important apps](#): Assign importance levels to prioritize your applications.
- [Step 2: Resolve issues](#): Identify and resolve problems with applications.
- [Step 3: Deploy](#): Start the upgrade process.

Also see the following topic for information about additional items that can be affected by the upgrade process:

- [Additional insights](#): Find out which MS Office add-ins are installed, and review web site activity.

Target version

The target version setting is used to evaluate the number of computers that are already running the default version of Windows 10, or a later version. The target version of Windows 10 is displayed on the upgrade overview tile. See the following example:

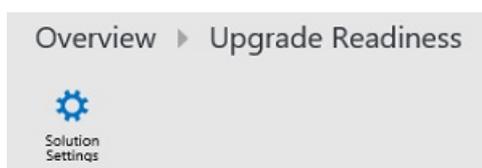


The default target version in Upgrade Readiness is set to the released version of the Current Branch for Business (CBB). CBB can be determined by reviewing [Windows 10 release information](#). The target version setting is used to evaluate the number of computers that are already running this version of Windows, or a later version.

The number displayed under **Computers upgraded** in the Upgrade Overview blade is the total number of computers that are already running the same or a later version of Windows compared to the target version. It also is used in the evaluation of apps and drivers: Known issues and guidance for the apps and drivers in Upgrade Readiness is based on the target operating system version.

You now have the ability to change the Windows 10 version you wish to target. The available options currently are: Windows 10 version 1507, Windows 10 version 1511, Windows 10 version 1607, Windows 10 version 1703, Windows 10 version 1709 and Windows 10 version 1803.

To change the target version setting, click on **Solutions Settings**, which appears at the top when you open your Upgrade Readiness solution:



You must be signed in to Upgrade Readiness as an administrator to view settings.

On the **Upgrade Readiness Settings** page, choose one of the options in the drop down box and click **Save**. The changes in the target version setting are reflected in evaluations when a new snapshot is uploaded to your

workspace.

Overview ▶ Upgrade Readiness Settings

 Save  Cancel

Commercial ID Key

361d3921-78e9-43ea-8aff-d02a83edcb7d 

Target version to be evaluated

Use the dropdown below to select the operating system version that you are planning to upgrade to. (Don't forget to click save in the top left corner of the screen!)
Note that changes to your target operating system will take approximately 24 hours to be reflected in the tool.

Windows 10 Version 1607 ▼

Microsoft recommends: Windows 10 Version 1607
Learn more about Windows 10 servicing options at [Windows 10 release information](#).
Certain target version choices have different minimum required [compatibility update KBs](#). Make sure you install the most recent version for your target version.

Upgrade Readiness - Upgrade overview

6/14/2019 • 3 minutes to read • [Edit Online](#)

The first blade in the Upgrade Readiness solution is the upgrade overview blade. This blade displays the total count of computers sharing data with Microsoft, and the count of computers upgraded. As you successfully upgrade computers, the count of computers upgraded increases.

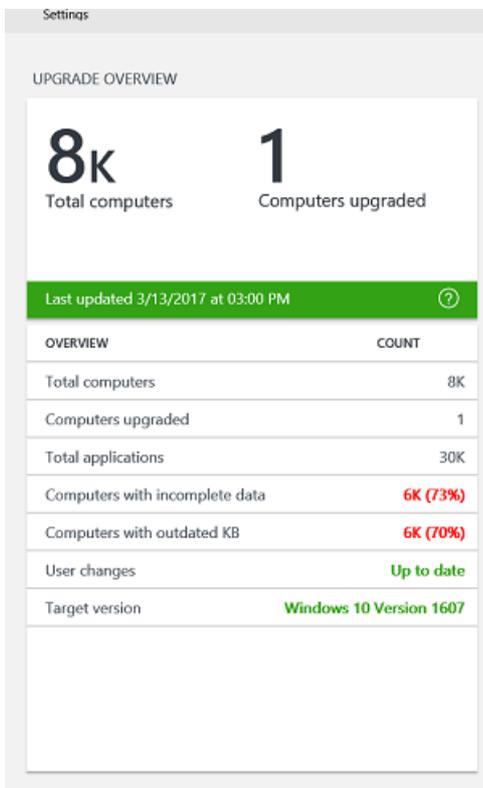
The upgrade overview blade displays data refresh status, including the date and time of the most recent data update and whether user changes are reflected. The upgrade overview blade also displays the current target OS version. For more information about the target OS version, see [target version](#).

The following color-coded status changes are reflected on the upgrade overview blade:

- The "Last updated" banner:
 - No delay in processing device inventory data = "Last updated" banner is displayed in green.
 - Delay processing device inventory data = "Last updated" banner is displayed in amber.
- Computers with incomplete data:
 - Less than 4% = Count is displayed in green.
 - 4% - 10% = Count is displayed in amber.
 - Greater than 10% = Count is displayed in red.
- Computers with outdated KB:
 - Less than 10% = Count is displayed in green.
 - 10% - 30% = Count is displayed in amber.
 - Greater than 30% = Count is displayed in red.
- User changes:
 - Pending user changes = User changes count displays "Data refresh pending" in amber.
 - No pending user changes = User changes count displays "Up to date" in green.
- Target version:
 - If the current value matches the recommended value, the version is displayed in green.
 - If the current value is an older OS version than the recommended value, but not deprecated, the version is displayed in amber.
 - If the current value is a deprecated OS version, the version is displayed in red.

Click a row to drill down and see details about individual computers. If updates are missing, see [Enrolling devices in Windows Analytics](#) for information on required updates.

In the following example, there is no delay in data processing, more than 10% of computers (6k\8k) have incomplete data, more than 30% of computers (6k/8k) require an update, there are no pending user changes, and the currently selected target OS version is the same as the recommended version:



If data processing is delayed, the "Last updated" banner will indicate the date on which data was last updated. You can continue using your workspace as normal. However, any changes or additional information that is added might not be displayed until data is refreshed. When your workspace is in this state, there is no action required; data is typically refreshed and the display will return to normal again within 24 hours.

If there are computers with incomplete data, verify that you have installed the latest compatibility updates. Install the updates if necessary and then run the most recent [Update Readiness deployment script](#) from the Microsoft download center. The updated data payload should appear in Upgrade Readiness within 48 hours of a successful run on the deployment script.

Select **Total computers** for a list of computers and details about them, including:

- Computer ID and computer name
- Computer manufacturer
- Computer model
- Operating system version and build
- Count of system requirement, application, and driver issues per computer
- Upgrade assessment based on analysis of computer diagnostic data
- Upgrade decision status

Select **Total applications** for a list of applications discovered on user computers and details about them, including:

- Application vendor
- Application version
- Count of computers the application is installed on
- Count of computers that opened the application at least once in the past 30 days
- Percentage of computers in your total computer inventory that opened the application in the past 30 days
- Issues detected, if any
- Upgrade assessment based on analysis of application data
- Rollup level

Upgrade Readiness - Step 1: Identify important apps

6/14/2019 • 3 minutes to read • [Edit Online](#)

This is the first step of the Upgrade Readiness workflow. In this step, applications are listed and grouped by importance level. Setting the importance level enables you to prioritize applications for upgrade.



Select **Assign importance** to change an application's importance level. By default, applications are marked **Not reviewed** or **Low install count** until you assign a different importance level to them.

To change an application's importance level:

1. Select **Not reviewed** or **Low install count** on the **Prioritize applications** blade to view the list of applications with that importance level.
2. Select the applications you want to change to a specific importance level and then select the appropriate option from the **Select importance level** list.
3. Click **Save** when finished.

Importance levels include:

IMPORTANCE LEVEL	WHEN TO USE IT	RECOMMENDATION
------------------	----------------	----------------

IMPORTANCE LEVEL	WHEN TO USE IT	RECOMMENDATION
Low install count	<p>We give you a head start by identifying applications that are installed on 2% or less of your total computer inventory. [Number of computers application is installed on/total number of computers in your inventory.]</p> <p>Low install count applications are automatically marked as Ready to upgrade in the UpgradeDecision column unless they have issues that need attention.</p>	<p>Be sure to review low install count applications for any business critical or important applications that are not yet upgrade-ready, despite their low installation rates. For example, payroll apps or tax accounting apps tend to be installed on a relatively small number of machines but are still considered business critical applications.</p>
Not reviewed	<p>Applications that are installed on more than 2% of your total computer inventory are marked not reviewed until you set their importance level.</p>	<p>Once you've started to investigate an application to determine its importance level and upgrade readiness, change its status to Review in progress in both the Importance and UpgradeDecision columns.</p>
Business critical	<p>By default, no applications are marked as business critical because only you can make that determination. If you know that an application is critical to your organization's functioning, mark it Business critical.</p>	<p>You may also want to change the application's status to Review in progress in the UpgradeDecision column to let other team members know that you're working on getting this business critical application upgrade-ready. Once you've fixed any issues and validated that the application will migrate successfully, change the upgrade decision to Ready to upgrade.</p>
Important	<p>By default, no applications are marked as important because only you can make that determination. If the application is important but not critical to your organization's functioning, mark it Important.</p>	<p>You may also want to change the application's status to Review in progress in the UpgradeDecision column to let other team members know that you're working on getting this important application upgrade-ready. Once you've fixed any issues and validated that the application will migrate successfully, change the upgrade decision to Ready to upgrade.</p>
Ignore	<p>By default, no applications are marked as ignore because only you can make that determination. If the application is not important to your organization's functioning, such as user-installed applications and games, you may not want to spend time and money validating that these applications will migrate successfully. Mark these applications Ignore.</p>	<p>Set the application's importance level to Ignore to let other team members know that it can be left as-is with no further investigation or testing. If you set the importance level to ignore, and this is an app that you are not planning on testing or validating, consider changing the upgrade decision to Ready to upgrade. By marking these apps ready to upgrade, you are indicating that you are comfortable upgrading with the app remaining in its current state.</p>

IMPORTANCE LEVEL	WHEN TO USE IT	RECOMMENDATION
Review in progress	<p>Once you've started to investigate an application to determine its importance level and upgrade readiness, change its status to Review in progress in both the Importance and UpgradeDecision columns.</p>	<p>As you learn more about the application's importance to your organization's functioning, change the importance level to Business critical, Important, or Ignore.</p> <p>Until you've determined that priority applications will migrate successfully, leave the upgrade decision status as Review in progress.</p>

Upgrade Readiness - Step 2: Resolve app and driver issues

6/14/2019 • 17 minutes to read • [Edit Online](#)

This section of the Upgrade Readiness workflow reports application and driver inventory and shows you which applications have known issues, which applications have no known issues, and which drivers have issues. We identify applications and drivers that need attention and suggest fixes when we know about them.

In this section

The blades in the **Step 2: Resolve issues** section are:

- [Review applications with known issues](#)
- [Review known driver issues](#)
- [Review low-risk apps and drivers](#)
- [Prioritize app and driver testing](#)

You can change an application's upgrade decision and a driver's upgrade decision from the blades in this section. To change an application's or a driver's importance level, select **User changes**. Select the item you want to change and then select the appropriate option from the **Select upgrade decision** list.

Upgrade decisions include:

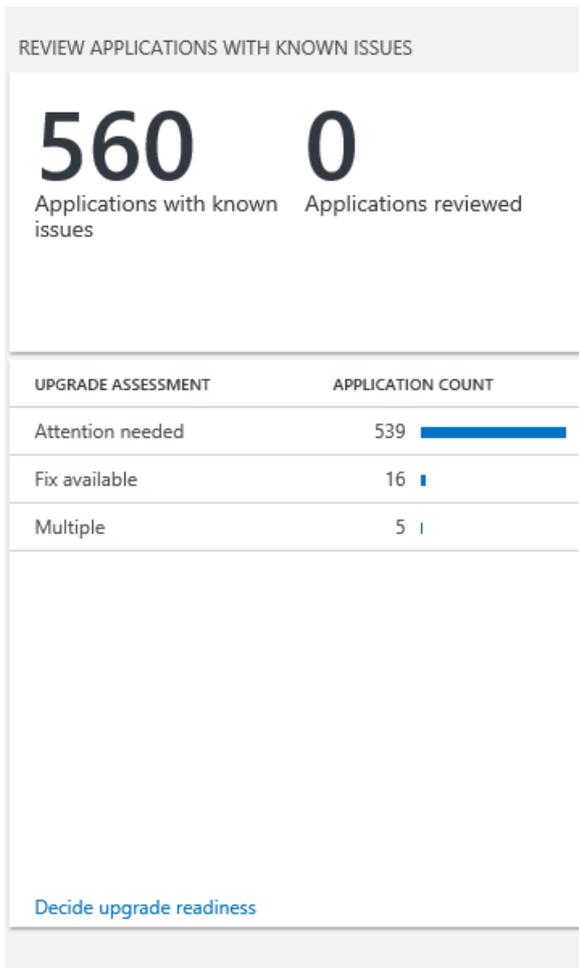
UPGRADE DECISION	WHEN TO USE IT	GUIDANCE
Not reviewed	<p>All drivers are marked as Not reviewed by default.</p> <p>Any app that has not been marked Low install count will also have an upgrade decision of Not reviewed by default.</p>	<p>Apps you have not yet reviewed or are waiting to review later should be marked as Not reviewed. When you start to investigate an application or a driver to determine upgrade readiness, change their upgrade decision to Review in progress.</p>
Review in progress	<p>When you start to investigate an application or a driver to determine upgrade readiness, change its upgrade decision to Review in progress.</p> <p>Until you've determined that applications and drivers will migrate successfully or you've resolved blocking issues, leave the upgrade decision status as Review in progress.</p>	<p>Once you've fixed any issues and validated that the application or driver will migrate successfully, change the upgrade decision to Ready to upgrade.</p>

UPGRADE DECISION	WHEN TO USE IT	GUIDANCE
Ready to upgrade	<p>Mark applications and drivers Ready to upgrade once you've resolved all blocking issues and you're confident that they will upgrade successfully, or if you've decided to upgrade them as-is.</p>	<p>Applications with no known issues and with low installation rates are marked Ready to upgrade by default.</p> <p>In Step 1, you might have marked some of your apps as Ignore. These should be marked as Ready to upgrade. Apps with low installation rates are marked as Ready to upgrade by default. Be sure to review any low install count applications for any business critical or important applications that are not yet upgrade-ready, despite their low installation rates.</p>
Won't upgrade	<p>By default, no applications or drivers are marked Won't upgrade because only you can make that determination.</p> <p>Use Won't upgrade for applications and drivers that you do not work on your target operating system, or that you are unable to upgrade.</p>	<p>If, during your investigation into an application or driver, you determine that they should not or cannot be upgraded, mark them Won't upgrade.</p>

As you review applications with known issues, you can also see ISV support statements or applications using [Ready for Windows](#).

Review applications with known issues

Applications with issues known to Microsoft are listed, grouped by upgrade assessment into **Attention needed** or **Fix available**.



To change an application's upgrade decision:

1. Select **Decide upgrade readiness** to view applications with issues.
2. In the table view, select an **UpgradeDecision** value.
3. Select **Decide upgrade readiness** to change the upgrade decision for each application.
4. Select the applications you want to change to a specific upgrade decision and then select the appropriate option from the **Select upgrade decision** list.
5. Click **Save** when finished.

IMPORTANT: Ensure that you have the most recent versions of the compatibility update and related KBs installed to get the most up-to-date compatibility information.

For applications assessed as **Attention needed**, review the table below for details about known issues and for guidance about how to resolve them, when possible.

UPGRADE ASSESSMENT	ACTION REQUIRED PRIOR TO UPGRADE?	ISSUE	WHAT IT MEANS	GUIDANCE
Attention needed	No	Application is removed during upgrade	Compatibility issues were detected and the application will not migrate to the new operating system.	No action is required for the upgrade to proceed.

UPGRADE ASSESSMENT	ACTION REQUIRED PRIOR TO UPGRADE?	ISSUE	WHAT IT MEANS	GUIDANCE
Attention needed	Yes	Blocking upgrade	Blocking issues were detected and Upgrade Readiness is not able to remove the application during upgrade. The application may work on the new operating system.	Remove the application before upgrading, and reinstall and test on new operating system.
Attention needed	No	Evaluate application on new OS	The application will migrate, but issues were detected that may impact its performance on the new operating system.	No action is required for the upgrade to proceed, but be sure to test the application on the new operating system.
Attention needed	No	Does not work with new OS, but won't block upgrade	The application is not compatible with the new operating system, but won't block the upgrade.	No action is required for the upgrade to proceed, however, you'll have to install a compatible version of the application on the new operating system.
Attention needed	Yes	Does not work with new OS, and will block upgrade	The application is not compatible with the new operating system and will block the upgrade.	Remove the application before upgrading. A compatible version of the application may be available.
Attention needed	Yes	May block upgrade, test application	Issues were detected that may interfere with the upgrade, but need to be investigated further.	Test the application's behavior during upgrade. If it blocks the upgrade, remove it before upgrading and reinstall and test it on the new operating system.
Attention needed	Maybe	Multiple	Multiple issues are affecting the application. See detailed view for more information.	When you see Multiple in the query detailed view, click Query to see details about what issues were detected with the different versions of the application.

For applications assessed as **Fix available**, review the table below for details about known issues and ways to fix them that are known to Microsoft.

UPGRADE ASSESSMENT	ACTION REQUIRED PRIOR TO UPGRADE?	ISSUE	WHAT IT MEANS	GUIDANCE
Fix available	Yes	Blocking upgrade, update application to newest version	The existing version of the application is not compatible with the new operating system and won't migrate. A compatible version of the application is available.	Update the application before upgrading.
Fix available	No	Reinstall application after upgrading	The application is compatible with the new operating system, but must be reinstalled after upgrading. The application is removed during the upgrade process.	No action is required for the upgrade to proceed. Reinstall application on the new operating system.
Fix available	Yes	Blocking upgrade, but can be reinstalled after upgrading	The application is compatible with the new operating system, but won't migrate.	Remove the application before upgrading and reinstall on the new operating system.
Fix available	Yes	Disk encryption blocking upgrade	The application's encryption features are blocking the upgrade.	Disable the encryption feature before upgrading and enable it again after upgrading.

ISV support for applications with Ready for Windows

[Ready for Windows](#) lists software solutions that are supported and in use for Windows 10. This site leverages data about application adoption from commercial Windows 10 installations and helps IT managers upgrade to Windows 10 with confidence. For more information, see [Ready for Windows Frequently Asked Questions](#).

Click **Review Applications With Known Issues** to see the status of applications for Ready for Windows and corresponding guidance. For example:

Guidance	: This application has been installed on at least 100,000 commercial Windows 10 devices.
Importance	: Not reviewed
UpgradeDecision	: Ready to upgrade
ReadyForWindows	: Highly adopted [Link]

If there are known issues with an application, the specific guidance for that known issue takes precedence over the Ready for Windows guidance.

Guidance	: Application may have issues on new OS. No action is required for upgrade to proceed.
Importance	: Not reviewed
UpgradeDecision	: Ready to upgrade
ReadyForWindows	: Highly adopted [Link]

If you query with RollupLevel="NamePublisher", each version of the application can have a different status for Ready for Windows. In this case, different values appear for Ready for Windows.

Type=UApp (RollupLevel=NamePublisher)

TIP

Within the Upgrade Readiness data model, an object of Type **UApp** refers to a particular application installed on a specific computer.

To support dynamic aggregation and summation of data the Upgrade Readiness solution "rolls up" (aggregates) data in preprocessing. Rolling up to the **Granular** level enables display of the **App** level. In Upgrade Readiness terminology, an **App** is a unique combination of: app name, app vendor, app version, and app language. Thus, at the Granular level, you can see attributes such as **total install count**, which is the number of machines with a specific **App** installed.

Upgrade Readiness also has a roll up level of **NamePublisher**. This level enables you to ignore different app versions within your organization for a particular app. In other words, **NamePublisher** displays statistics about a given app, aggregated across all versions.

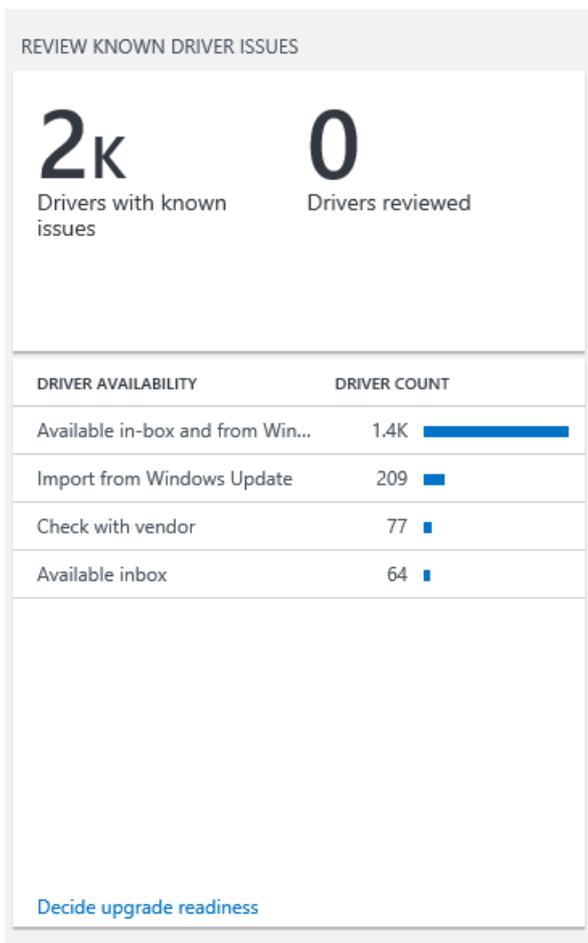
The following table lists possible values for **ReadyForWindows** and what they mean. For more information, see [What does the Adoption Status mean?](#)

READY FOR WINDOWS STATUS	QUERY ROLLUP LEVEL	WHAT THIS MEANS	GUIDANCE
Supported version available	Granular	The software provider has declared support for one or more versions of this application on Windows 10.	The ISV has declared support for a version of this application on Windows 10.
Highly adopted	Granular	This version of this application has been highly adopted within the Windows 10 Enterprise ecosystem.	This application has been installed on at least 100,000 commercial Windows 10 devices.
Adopted	Granular	This version of this application has been adopted within the Windows 10 Enterprise ecosystem.	This application has been installed on at least 10,000 commercial Windows 10 devices.
Insufficient Data	Granular	Too few commercial Windows 10 devices are sharing information about this version of this application for Microsoft to categorize its adoption.	N/A
Contact developer	Granular	There may be compatibility issues with this version of the application, so Microsoft recommends contacting the software provider to learn more.	Check Ready for Windows for additional information.
Supported version available	NamePublisher	The software provider has declared support for this application on Windows 10.	The ISV has declared support for a version of this application on Windows 10.

READY FOR WINDOWS STATUS	QUERY ROLLUP LEVEL	WHAT THIS MEANS	GUIDANCE
Adoption status available	NamePublisher	A Ready for Windows adoption status is available for one or more versions of this application. Please check Ready for Windows to learn more.	Check Ready for Windows for adoption information for this application.
Unknown	Any	There is no Ready for Windows information available for this version of this application. Information may be available for other versions of the application at Ready for Windows .	N/A

Review drivers with known issues

Drivers that won't migrate to the new operating system are listed, grouped by availability.



Availability categories are explained in the table below.

DRIVER AVAILABILITY	ACTION REQUIRED BEFORE OR AFTER UPGRADE?	WHAT IT MEANS	GUIDANCE
---------------------	--	---------------	----------

DRIVER AVAILABILITY	ACTION REQUIRED BEFORE OR AFTER UPGRADE?	WHAT IT MEANS	GUIDANCE
Available in-box	No, for awareness only	The currently installed version of an application or driver won't migrate to the new operating system; however, a compatible version is installed with the new operating system.	No action is required for the upgrade to proceed.
Import from Windows Update	Yes	The currently installed version of a driver won't migrate to the new operating system; however, a compatible version is available from Windows Update.	If the computer automatically receives updates from Windows Update, no action is required. Otherwise, import a new driver from Windows Update after upgrading.
Available in-box and from Windows Update	Yes	The currently installed version of a driver won't migrate to the new operating system. Although a new driver is installed during upgrade, a newer version is available from Windows Update.	If the computer automatically receives updates from Windows Update, no action is required. Otherwise, import a new driver from Windows Update after upgrading.
Check with vendor	Yes	The driver won't migrate to the new operating system and we are unable to locate a compatible version.	Check with the independent hardware vendor (IHV) who manufactures the driver for a solution.

To change a driver's upgrade decision:

1. Select **Decide upgrade readiness** and then select the group of drivers you want to review. Select **Table** to view the list in a table.
2. Select **User changes** to enable user input.
3. Select the drivers you want to change to a specific upgrade decision and then select the appropriate option from the **Select upgrade decision** list.
4. Click **Save** when finished.

Review low-risk apps and drivers

Applications and drivers that meet certain criteria to be considered low risk are displayed on this blade.

REVIEW LOW-RISK APPS AND DRIVERS	
4	171
Low-risk applications in need of review	Low-risk drivers in need of review
Learn how to use this blade effectively on our blog .	
CRITERIA	ITEM COUNT
Apps with an ISV support statement	0
Apps that are "Highly adopted"	4
Apps that are "Adopted"	0
Apps you have marked "Ignore"	0
Drivers available on Windows Update	169
Drivers available in-box	2
OTHER APPS AND DRIVERS IN NEED OF REVIEW	
CRITERIA	ITEM COUNT
Mission and Business critical apps	2
All apps that are not yet reviewed	76
All drivers that are not yet reviewed	258

The first row reports the number of your apps that have an official statement of support on Windows 10 from the software vendor, so you can be confident that they will work on your target operating system.

The second row (**Apps that are "Highly adopted"**) shows apps that have a ReadyForWindows status of "Highly adopted". This means that they have been installed on at least 100,000 commercial Windows 10 devices, and that Microsoft has not detected significant issues with the app in diagnostic data. Since these apps are prevalent in the ecosystem at large, you can be confident that they will work in your environment as well.

Each row of the blade uses a different criterion to filter your apps or drivers. You can view a list of applications that meet the criterion by clicking into a row of the blade. For example, if you click the row that says "Apps that are 'Highly adopted'", the result is a list of apps that have a ReadyForWindows status of "Highly adopted". From here, you can bulk-select the results, select **Ready to upgrade**, and then click **Save**. This will mark all apps meeting the "Highly adopted" criterion as "Ready to upgrade"--no further validation is required. Any applications that you have marked as *Mission critical* or *Business critical* are filtered out, as well as any app that has an issue known to Microsoft. This allows you to work with apps in bulk without having to worry about missing a critical app.

You can customize the criteria further by using the Log Search query language. For example, if a ReadyForWindows status of "Adopted" is not sufficient by itself for you to be confident in an app's compatibility, you can add additional filters. To do this, click the row labeled **Apps that are 'Adopted'**. Then, modify the resulting query to fit your company's risk tolerance. If, for example, you prefer that an app must be "Adopted" and have fewer than 1,000 installations, then add *TotalInstalls < 1000* to the end of the Log Search query. Similarly, you can append additional criteria by using other attributes such as monthly active users or app importance.

NOTE

Apps that you have designated as *Mission critical* or *Business critical* are automatically **excluded** from the counts on this blade. If an app is critical, you should always validate it manually prior to upgrading.

At the bottom of the blade, the **OTHER APPS AND DRIVERS IN NEED OF REVIEW** section allows you to quickly access apps you have designated as **Mission critical** or **Business critical**, your remaining apps that still need to be reviewed, and your remaining drivers that need to be reviewed.

Prioritize app and driver testing

Planning and executing an OS upgrade project can be overwhelming. When you are tasked with evaluating thousands of applications and drivers to ensure a successful upgrade, it can be difficult to decide where to start. The Upgrade Readiness solution provides valuable assistance for you, helping to determine the most important apps and drivers to unblock and enabling you to create a proposed action plan.

Proposed action plan

The Upgrade Readiness proposed action plan is an optimally ordered list of apps and drivers that are in need of review. By testing apps and drivers in the order suggested by the proposed action plan, you are able to increase your number of "Ready to upgrade" computers in an efficient manner. The action plan can be a very powerful tool during upgrade planning – but it's most helpful when it's used correctly. This topic explains the proposed action plan, describes how to use it, and calls out a few misconceptions and invalid use cases that you should avoid.

The proposed action plan represents the order that Microsoft recommends you rationalize the upgrade-readiness of your apps and drivers. By validating apps and drivers in the order proposed, you can ensure that you are testing efficiently.

Each item in the proposed action plan represents either an application or a driver that you have not yet marked "Ready to upgrade."

Since "Low install count" apps are automatically marked "Ready to upgrade", you will not see any of these apps in the proposed action plan.

Each item in the plan has the following attributes:

ATTRIBUTE	DESCRIPTION	EXAMPLE VALUE
ItemRank	The location of this item in the context of the proposed action plan. For example, the item with ItemRank 7 is the 7th item in the Plan. It is crucial that the Plan is viewed in order by increasing ItemRank. Sorting the Plan in any other way invalidates the insights that the Plan provides.	7
ItemType	Whether this item is an app or driver -- possible values are: "App" and "Driver."	App
ItemName	The name of the app or driver that is in need of review.	Microsoft Visual C++ 2005 Redistributable (x64)
ItemVendor	The vendor of the app or driver.	Microsoft Corporation
ItemVersion	The version of the app or driver.	12.1.0.1
ItemLanguage	If this item is an application, then this field will be the language of the app. If the item is a driver, then this will say "N/A."	English
ItemHardwareId	If this item is a driver, then this field will be the hardware id of the driver. If the item is an app, then this will say "N/A."	N/A
Upgrade Decision	The upgrade decision you have provided for this app or driver. If you have not defined an upgrade decision, then you will see the default value of "Not reviewed."	Review in progress

ATTRIBUTE	DESCRIPTION	EXAMPLE VALUE
ComputersUnblocked	Assuming you have already marked all previous items in the proposed action plan "Ready to upgrade", this represents the number of additional computers that will become "Ready to upgrade" by testing this app or driver and giving it an upgrade decision of "Ready to upgrade". For example, if ComputersUnblocked is 200, then resolving any issues associated with the app/driver in question will make 200 new computers "Ready to upgrade."	200
CumulativeUnblocked	The total number of computers that will become "Ready to upgrade" if you validate and mark this and all prior items in the proposed action plan "Ready to upgrade". For example, if ItemRank is 7, and CumulativeUnblocked is 950, then fixing items 1 thru 7 in the proposed action plan will cause 950 of your computers to become "Ready to upgrade."	950
CumulativeUnblockedPct	The percentage of your machines that will become "Ready to upgrade" if you make this and all prior items in the proposed action plan "Ready to upgrade."	0.24

See the following example action plan items (click the image for a full-size view):

The screenshot displays the Microsoft Operations Management Suite interface. On the left, there is a sidebar with navigation icons and filters. The main area shows a search bar and a table of 18 items. The table columns are: ITEM RANK, ITEM NAME, ITEM VERSION, ITEM VENDOR, ITEM TYPE, ITEM LANGUAGE, ITEM HARDWARE ID, COMPUTERS UNBLOCKED, CUMULATIVE UNBLOCKED, CUMULATIVE UNBLOCKED PCT, and UPGRADE DECISION. The 3rd item is Microsoft Bing Sports, version 4.20.951.0, published by Microsoft. The table shows that 1014 computers are unblocked for this item, and 14779 computers are cumulative unblocked for the first three items.

ITEM RANK	ITEM NAME	ITEM VERSION	ITEM VENDOR	ITEM TYPE	ITEM LANGUAGE	ITEM HARDWARE ID	COMPUTERS UNBLOCKED	CUMULATIVE UNBLOCKED	CUMULATIVE UNBLOCKED PCT	UPGRADE DECISION
1	Microsoft.BingNe...	4.20.951.0	CN=Microsoft Co...	App	English (United S...		1026	1026	0.76	Not reviewed
2	Microsoft.BingWe...	4.20.951.0	CN=Microsoft Co...	App	English (United S...		12739	13765	10.21	Not reviewed
3	Microsoft.BingSp...	4.20.951.0	CN=Microsoft Co...	App	English (United S...		1014	14779	10.96	Not reviewed
4	Microsoft.BingFin...	4.20.951.0	CN=Microsoft Co...	App	English (United S...		20991	35770	26.53	Not reviewed
5	46928bounde.Ecl...	2.2.2.38	CN=4EAD3E48-B...	App	English (United S...		3871	39641	29.4	Not reviewed
6	Drawboard Draw...	5.1.6.0.0	CN=3CCF3A32-E...	App	English (United S...		3661	43302	32.12	Not reviewed
7	Microsoft.Micros...	1.1701.5019.0	CN=Microsoft Co...	App	English (United S...		2978	46280	34.33	Not reviewed
8	Microsoft.Minecr...	1.0.20.0	CN=Microsoft Co...	App	English (United S...		4518	50798	37.68	Not reviewed
9	Microsoft Silverf...	5.1.50906.0	Microsoft Corpor...	App	English (United S...		2596	53494	39.68	Not reviewed
10	Microsoft Azure ...	2.0.50130.0	Microsoft Corpor...	App	English (United S...		2483	55977	41.52	Not reviewed
11	NVIDIA Update 2...	2.11.4.0	NVIDIA Corporati...	App	Unknown		2480	58457	43.36	Not reviewed
12	Microsoft Project...	16.0.7369.2127	Microsoft Corpor...	App	Unknown		6	58463	43.36	Not reviewed
13	Microsoft Visio P...	16.0.7369.2127	Microsoft Corpor...	App	Unknown		3026	61489	45.61	Not reviewed
14	Intel(R) Processo...	10.18.10.4358	Intel Corporation	App	Unknown		1359	62848	46.62	Not reviewed
15	Vulkan Run Time ...	1.0.8.1	LunarG Inc.	App	Unknown		1306	64154	47.59	Not reviewed
16	Microsoft Visual ...	14.0.23107.156	Microsoft Corpor...	App	English (United S...		1190	65344	48.47	Not reviewed
17	Synaptics Pointin...	5.00.8498.1000	Microsoft Corpor...	App	English (United S...		1200	66544	49.36	Not reviewed
18	NVIDIA Graphics ...	354.45	NVIDIA Corporati...	App	Unknown		1176	67720	50.23	Not reviewed

In this example, the 3rd item is an application: **Microsoft Bing Sports**, a modern app, version **4.20.951.0**, published by Microsoft. By validating this app and making its UpgradeDecision "Ready to upgrade", you can potentially make **1014** computers "Ready to upgrade" – but only after you have already validated items 1 and 2 in the list. By marking items 1, 2, and 3 "Ready to upgrade", 14779 of your computers will become upgrade-ready. This represents 10.96% of

the machines in this workspace.

Using the proposed action plan

There are several valid use cases for the proposed action plan. But it's always important to remember that the information presented in the Plan is only accurate when sorted by increasing Item Rank! Here are three potential cases in which you could use the proposed action plan:

1. Quickly determine how many apps and drivers you'll need to validate in order to make x% of your computers upgrade-ready. To determine this, simply find the first item in the Plan with a CumulativeUnblockedPct greater than or equal to your desired percentage of upgrade-ready computers. The corresponding ItemRank represents the smallest number of apps and drivers that you can validate in order to reach your upgrade readiness goal. The prior items in the proposed action plan itself represent the most efficient route to reaching your goal.
2. Use the proposed action plan to prepare a small portion of your machines for a pilot of your target Operating System. Let's say you want to test a new Operating System by upgrading a few hundred computers. You can use the proposed action plan to determine how many apps and drivers you will need to validate before you can be confident that your pilot will be successful.
3. If your project deadline is approaching and you only have time to validate a few more apps and drivers, you can use the proposed action plan to determine which apps and drivers you should focus on to maximize the number of computers that you can confidently upgrade.

Misconceptions and things to avoid

The most common misconceptions about the proposed action plan involve the assumption that each item in the plan is independent of those around it. The apps and drivers in the plan must be considered in the correct order to draw valid conclusions. For example, if you choose to validate items 1, 3, 4, and 5 and mark each of them "Ready to upgrade," the proposed action plan cannot tell you how many computers will become upgrade-ready as a result of your testing. Even the non-cumulative "ComputersUnblocked" count is dependent upon all prior issues having already been resolved.

If an item with ItemRank = 7 has a ComputersUnblocked value of 50, do not assume that 50 of your computers will become upgrade-ready if you test this item. However, if you validate items 1 through 6 in the plan, you can make an additional 50 computers upgrade-ready by validating the 7th item in the plan.

Upgrade Readiness - Step 3: Deploy Windows

6/14/2019 • 4 minutes to read • [Edit Online](#)

All of your work up to now involved reviewing and resolving application and driver issues. Along the way, as you've resolved issues and decided which applications and drivers are ready to upgrade, you've been building a list of computers that are upgrade ready. The blades in the **Deploy** section are:

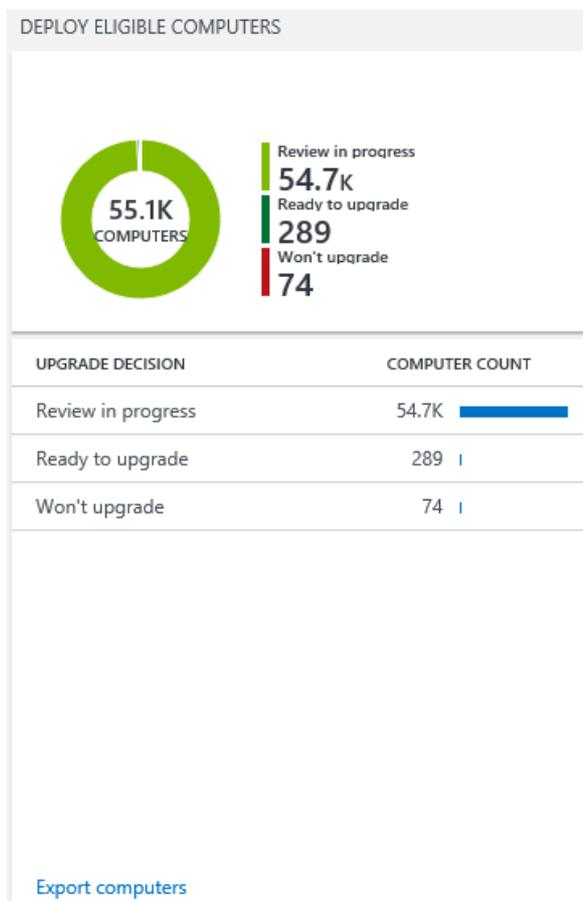
- [Deploy eligible computers](#)
- [Deploy computers by group](#)

Computers that are listed in this step are assigned an **UpgradeDecision** value, and the total count of computers in each upgrade decision category is displayed. Additionally, computers are assigned an **UpgradeAssessment** value. This value is displayed by drilling down into a specific upgrade decision category. For information about upgrade assessment values, see [Upgrade assessment](#).

Deploy eligible computers

In this blade, computers grouped by upgrade decision are listed. The upgrade decision on the machines is a calculated value based on the upgrade decision status for the apps and drivers installed on the computer. This value cannot be modified directly. The upgrade decision is calculated in the following ways:

- **Review in progress:** At least one app or driver installed on the computer is marked **Review in progress**.
- **Ready to upgrade:** All apps and drivers installed on the computer are marked as **Ready to Upgrade**.
- **Won't upgrade:** At least one app or driver installed on the computer is marked as **Won't upgrade**, or a system requirement is not met.



Select **Export computers** for more details, including computer name, manufacturer and model, and Windows edition currently running on the computer. Sort or further query the data and then select **Export** to generate and save a comma-separated value (csv) list of upgrade-ready computers.

Important

When viewing inventory items in table view, the maximum number of rows that can be viewed and exported is limited to 5,000. If you need to view or export more than 5,000 items, reduce the scope of the query so you can export fewer items at a time.

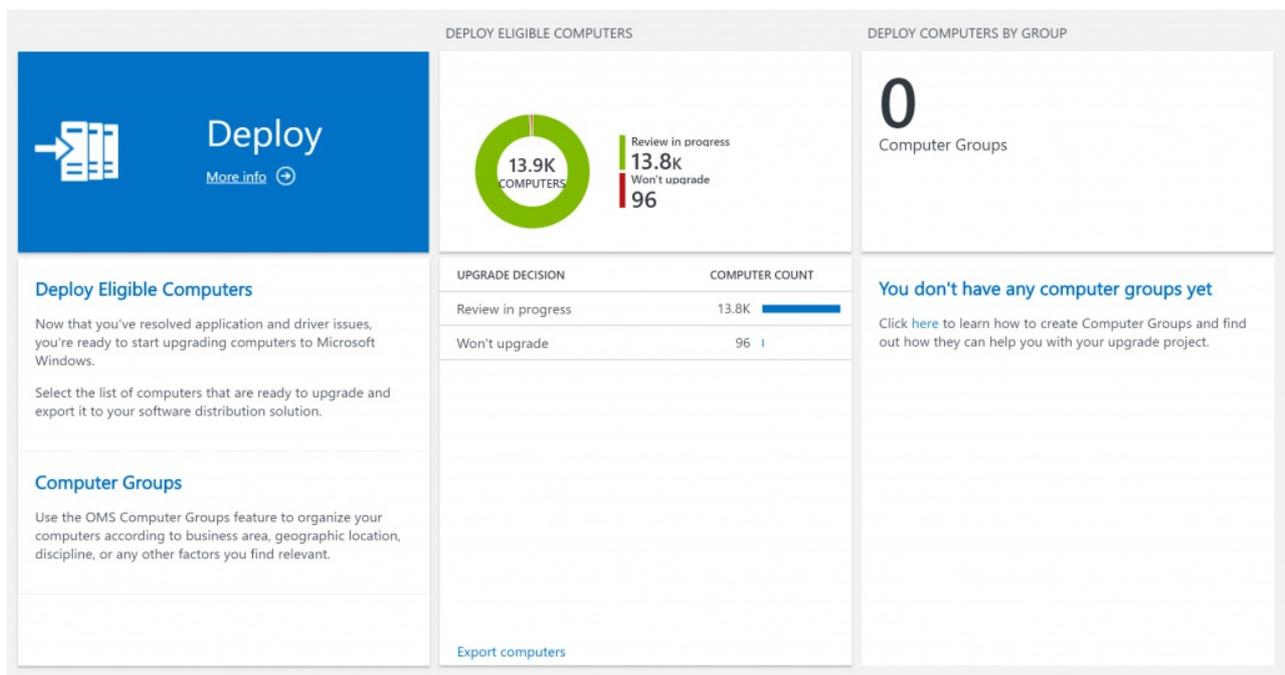
Computer groups

Computer groups allow you to segment your environment by creating device groups based on log search results, or by importing groups from Active Directory, WSUS or System Center Configuration Manager. Computer groups are an OMS feature. For more information, see [Computer groups in OMS](#).

Query based computer groups are recommended in the initial release of this feature. A feature known as **Configuration Manager Upgrade Readiness Connector** is anticipated in a future release that will enable synchronization of **ConfigMgr Collections** with computer groups in OMS.

Getting started with Computer Groups

When you sign in to OMS, you will see a new blade entitled **Computer Groups**. See the following example:



To create a computer group, open **Log Search** and create a query based on **Type=UAComputer**, for example:

```
Type=UAComputer Manufacturer=DELL
```

When you are satisfied that the query is returning the intended results, add the following text to your search:

| measure count() by Computer

This will ensure every computer only shows up once. Then, save your group by clicking **Save** and **Yes**. See the following example:

Your new computer group will now be available in Upgrade Readiness. See the following example:

DEPLOY ELIGIBLE COMPUTERS
DEPLOY COMPUTERS BY GROUP

Deploy

[More info](#) ➔

13.9K COMPUTERS

Review in progress: 13.8k
Won't upgrade: 96

1

Computer Groups

Deploy Eligible Computers

Now that you've resolved application and driver issues, you're ready to start upgrading computers to Microsoft Windows.

Select the list of computers that are ready to upgrade and export it to your software distribution solution.

UPGRADE DECISION	COMPUTER COUNT
Review in progress	13.8K <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
Won't upgrade	96 <div style="width: 100%; height: 10px; background-color: #C00000;"></div>

[Export computers](#)

GROUP NAME	COMPUTER COUNT
Dell Computers	4K <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>

Using Computer Groups

When you drill into a computer group, you will see that computers are categorized by **UpgradeDecision**. For computers with the status **Review in progress** or **Won't upgrade** you can drill down to view issues that cause a computer to be in each category, or you can simply display a list of the computers in the category. For computers that are designated **Ready to upgrade**, you can go directly to the list of computers that are ready.

Type=UAComputer Computer in \$ComputerGroups[Dell Computers] | measure count() AS ComputerCount by UpgradeDecision
⏸

3 Results
[Chart](#)
[Table](#)

UPGRADEDECISION	COMPUTERCOUNT
Review in progress [Details] [Computers]	16,399 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
Ready to upgrade	100 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
Won't upgrade [Details] [Computers]	52 <div style="width: 100%; height: 10px; background-color: #C00000;"></div>

Viewing a list of computers in a certain status is self-explanatory, Let's look at what happens when you click the details link on **Review in progress**:

(Type=UAApp or Type=UADriver) (UpgradeDecision="Not reviewed" OR UpgradeDecision="Review in progress") Computer in \$ComputerGroups[Dell Computers]
✕ 🔍

2 Results
[Chart](#)
[Table](#)

TYPE	TYPECOUNT
UAApp [Issues]	667,347 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
UADriver [Issues]	8,491 <div style="width: 100%; height: 10px; background-color: #C00000;"></div>

Next, select if you want to see application issues (**UAApp**) or driver issues (**UADriver**). See the following example of selecting **UAApp**:

22 Results | Chart | Table

APPNAME	APPENDOR	UPGRADEDECISION	APPLICATIONCOUNT
Windows Firewall Configuration Provider	Microsoft Corporation	Review in progress	2,938
Microsoft Visual C++ 2010 x86 Redistribut...	Microsoft Corporation	Review in progress	2,751
Microsoft.Reader	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,266
Microsoft.XboxLIVEGames	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,203
Microsoft.BingMaps	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,157
Microsoft.BingNews	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,130
Microsoft.BingFinance	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,124
Microsoft.WindowsScan	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,101
microsoft.windowscommunicationsapps	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,100
Microsoft.BingSports	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,096
fs.vpn.client	CN=Microsoft Windows O=Microsoft Corp...	Not reviewed	2,076
SonicWALL.MobileConnect	CN=Microsoft Windows O=Microsoft Corp...	Not reviewed	2,076
CheckPoint.VPN	CN=Microsoft Windows O=Microsoft Corp...	Not reviewed	2,076
JuniperNetworks.JunosPulseVpn	CN=Microsoft Windows O=Microsoft Corp...	Not reviewed	2,076
Microsoft.MoCamera	CN=Microsoft Windows O=Microsoft Corp...	Not reviewed	2,072
Microsoft.WindowsCalculator	CN=Microsoft Corporation O=Microsoft C...	Not reviewed	2,068

A list of apps that require review so that Dell Computers are ready for upgrade to Windows 10 is displayed.

Upgrade assessment

Upgrade assessment and guidance details are explained in the following table.

UPGRADE ASSESSMENT	ACTION REQUIRED BEFORE OR AFTER UPGRADE PILOT?	ISSUE	WHAT IT MEANS	GUIDANCE
No known issues	No	None	Computers will upgrade seamlessly.	OK to use as-is in pilot.
OK to pilot, fixed during upgrade	No, for awareness only	Application or driver will not migrate to new OS	The currently installed version of an application or driver won't migrate to the new operating system; however, a compatible version is installed with the new operating system.	OK to use as-is in pilot.
OK to pilot with new driver from Windows Update	Yes	Driver will not migrate to new OS	The currently installed version of a driver won't migrate to the new operating system; however, a newer, compatible version is available from Windows Update.	Although a compatible version of the driver is installed during upgrade, a newer version is available from Windows Update. If the computer automatically receives updates from Windows Update, no action is required. Otherwise, replace the new in-box driver with the Windows Update version after upgrading.

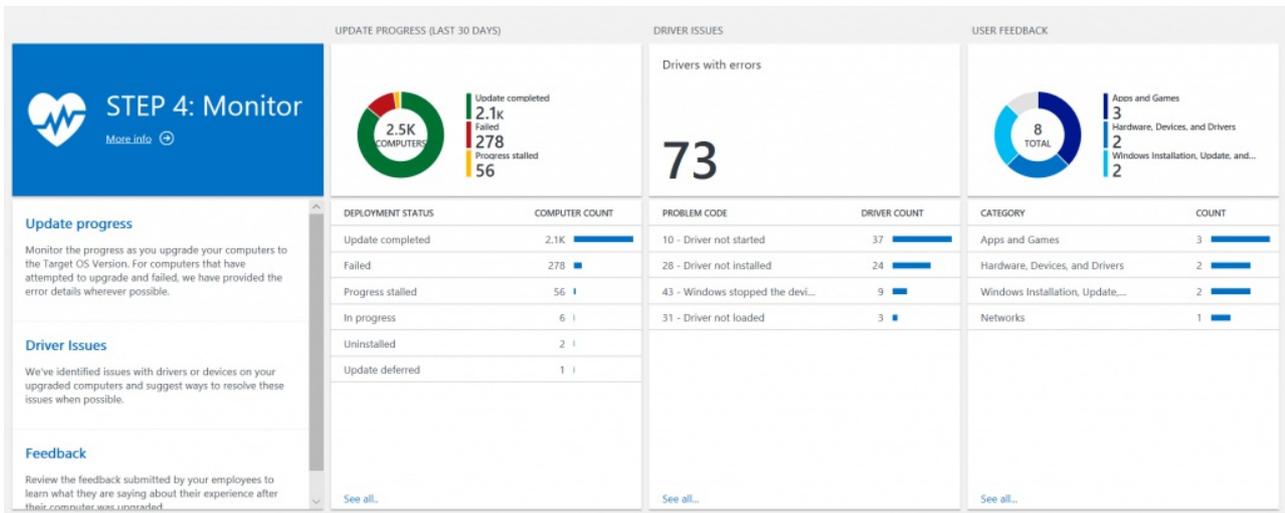
Select **Export computers** to view pilot-ready computers organized by operating system. After you select the computers you want to use in a pilot, click Export to generate and save a comma-separated value (csv) file.

Important> When viewing inventory items in table view, the maximum number of rows that can be viewed and exported is limited to 5,000. If you need to view or export more than 5,000 items, reduce the scope of the query so you can export fewer items at a time.

Upgrade Readiness - Step 4: Monitor

6/14/2019 • 2 minutes to read • [Edit Online](#)

Now that you have started deploying an update with Upgrade Readiness, you can use it to monitor important elements.



Update progress

The **Update progress** blade allows you to monitor the progress and status of your deployment. Any device that has attempted to upgrade in the last 30 days displays the **DeploymentStatus** attribute. You'll be able to see the number of computers that have successfully upgraded, failed to upgrade, are stalled, etc.

Selecting this blade allows you to view device-level details about the deployment. For example, select **Failed** to view the original operating system version, the target operating system version, and the reason the update failed for each of the devices that failed to upgrade. In the case of the device illustrated in the following image, an attempt was made to upgrade from Windows 10, version 1703 to 1709, but the operation timed out.

12/17/2017 12:00:00.000 AM | UAComputer

```
... TimeGenerated      : 12/17/2017 12:00:00.000 AM
... Computer          : Sales157.Contoso
... Manufacturer      : MICROSOFT_CORPORATION
... Model             : Surface Book
... OSVersion         : Windows 10
... OSBuild           : 10.0.15063.674.amd64fre.rs2_release.170317-1834
... DeploymentStatus  : Failed
... DeploymentErrorDetails : The installation process did not complete within the required time limit.
... OriginBuild       : 15063
... OriginOSVersion   : 1703
... TargetBuild       : 16299
... TargetOSVersion   : 1709
... HoursToUninstall  : -1
... TotalIssues       : 0
```

Driver issues

The **Driver issues** blade allows you to see Device Manager errors for your upgraded devices. We include data for all compatibility-related device errors, such as "driver not found" and "driver not started." The blade summarizes errors by error type, but you can select a particular error type to see device-level details about which device(s) are failing and where to obtain a driver.

For example, by selecting error code **28 - driver not installed**, you would see that the device in the following image is missing the driver for a network controller. Upgrade Readiness also notifies that a suitable driver is available online through Windows Update. If this device is configured to automatically receive updates from Windows Update, this issue would likely resolve itself following the device's next Windows Update scan. If this device does not automatically receive updates from Windows Update, you would need to deliver the driver manually.

12/17/2017 12:00:00.000 AM | UADriverProblemCodes

```
... TimeGenerated      : 12/17/2017 12:00:00.000 AM
... Computer          : MaryRe-Dsktp.Contoso
... HardwareName     : Network Controller
... HardwareID       : pci\ven_14e4&dev_43a0&subsys_86591043&rev_03
... DriverAvailability : AvailableOnlineWindowsUpdate
... ProblemCode      : 28 - Driver not installed
... Guidance         : Reinstall the device driver manually. \[More Info\]
... SourceSystem     : AzureStorage
... ComputerID       : 515ad2aa-d44f-46ca-b2d8-b09ea26d90f8
```

[\[-\] show less](#)

User feedback

The **User Feedback** blade focuses on gathering subjective feedback from your end users. If a user submits feedback through the Feedback Hub app on a device in your workspace, we will make that feedback visible to you in this blade. The Feedback Hub app is built into Windows 10 and can be accessed by typing "Feedback Hub" in the Cortana search bar.

We recommend that you encourage your end users to submit any feedback they have through Feedback Hub. Not only will this feedback be sent directly to Microsoft for review, but you'll also be able to see it by using Upgrade Readiness. You should be aware that **feedback submitted through Feedback Hub will be publicly visible**, so it's best to avoid submitting feedback about internal line-of-business applications.

When viewing user feedback in Upgrade Readiness, you'll be able to see the raw "Title" and "Feedback" text from the user's submission in Feedback Hub, as well as the number of upvotes the submission has received. (Since feedback is publicly visible, the number of upvotes is a global value and not specific to your company.) If a Microsoft engineer has responded to the submission in Feedback Hub, we'll pull in the Microsoft response for you to see as well.

12/17/2017 12:00:00.000 AM | UAFeedback

```
... TimeGenerated      : 12/17/2017 12:00:00.000 AM
... Category          : Apps and Games
... Computer          : AaronSi-Booth.Contoso
... Title             : Joining WebEx meetings is taking much longer than usual
... Feedback         :
```

Lately, it's been taking a very long time for me to join WebEx meetings from my desktop. When I click Join Meeting, the loading bar spins for at least 30-45 seconds before dropping me into the meeting room. In the past, this process never took more than a few seconds.

```
... TotalUpvotes      : 5
... FeedbackSubmittedDate : 8/22/2017 5:00:00.000 PM
... MicrosoftResponse : We've received your feedback! We're looking into this.
... AppName           : Cisco WebEx Meetings
... SourceSystem      : AzureStorage
... ComputerID        : fb4bdb89-2dbe-4d68-80dd-63b61c1392c4
```

[\[-\] show less](#)

Upgrade Readiness - Additional insights

6/19/2019 • 4 minutes to read • [Edit Online](#)

This topic provides information on additional features that are available in Upgrade Readiness to provide insights into your environment. These include:

- **Spectre and Meltdown protections:** Status of devices with respect to their anti-virus, security update, and firmware updates related to protection from the "Spectre" and "Meltdown" vulnerabilities.
- **Site discovery:** An inventory of web sites that are accessed by client computers running Windows 7, Windows 8.1, or Windows 10 using Internet Explorer.
- **Office add-ins:** A list of the Microsoft Office add-ins that are installed on client computers.

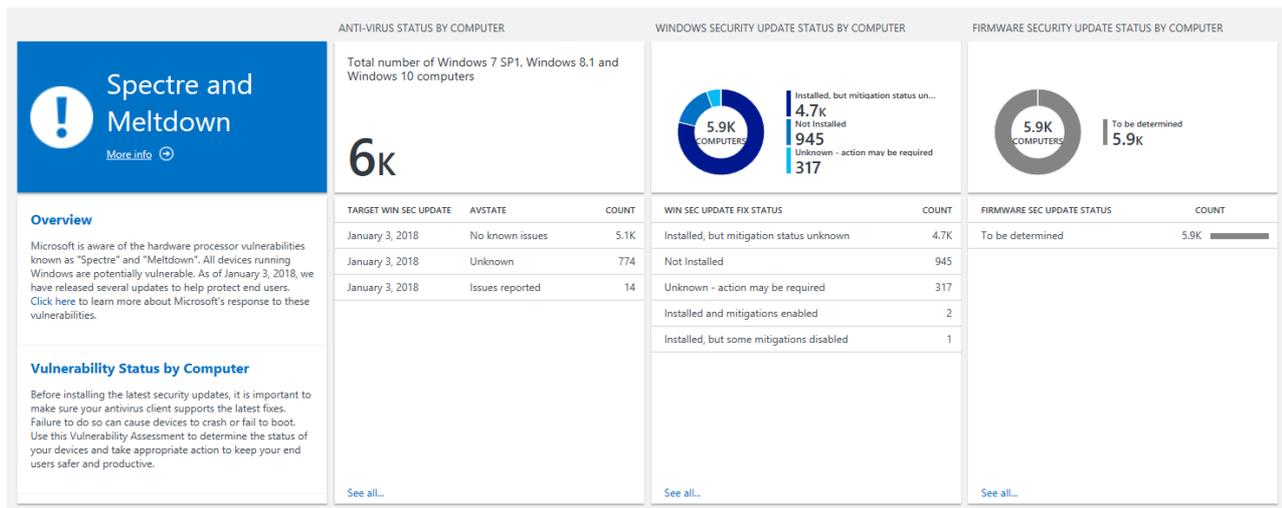
Spectre and Meltdown protection status

Microsoft has published guidance for IT Pros that outlines the steps you can take to improve protection against the hardware vulnerabilities known as "Spectre" and "Meltdown." See [Windows Client Guidance for IT Pros to protect against speculative execution side-channel vulnerabilities](#) for details about the vulnerabilities and steps you can take.

Microsoft recommends three steps to help protect against the Spectre and Meltdown vulnerabilities:

- Verify that you are running a supported antivirus application.
- Apply all available Windows operating system updates, including the January 2018 and later Windows security updates.
- Apply any applicable processor firmware (microcode) updates provided by your device manufacturer(s).

Upgrade Readiness reports on status of your devices in these three areas.



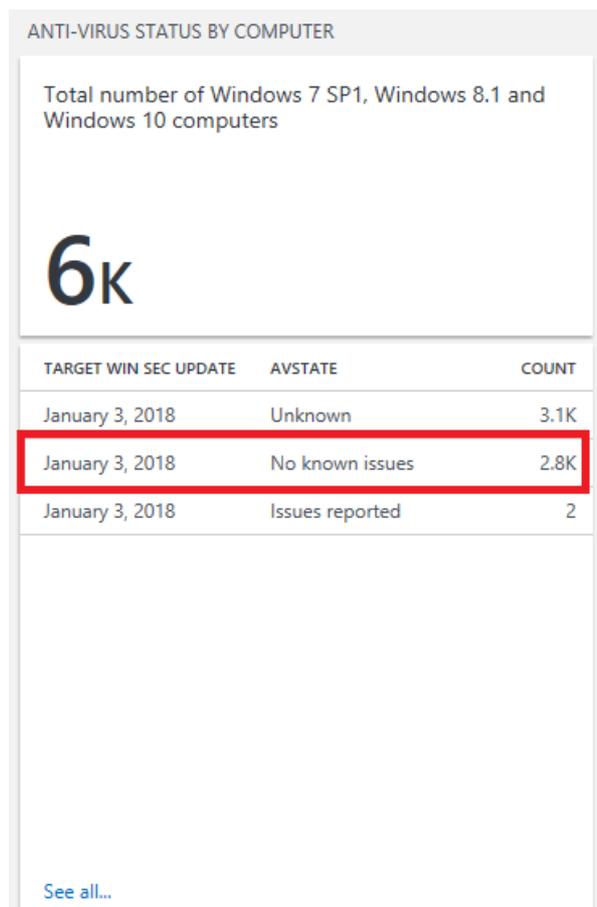
IMPORTANT

To provide these blades with data, ensure that your devices can reach the endpoint <http://adl.windows.com>. (See [Enrolling devices in Windows Analytics](#) for more about necessary endpoints and how to whitelist them.)

Anti-virus status blade

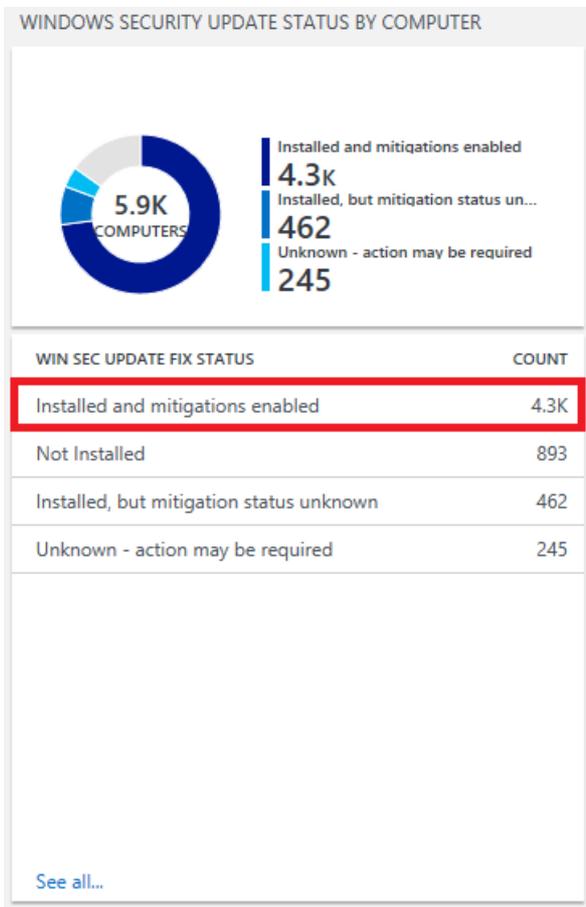
This blade helps you determine if your devices' anti-virus solution is compatible with the latest Windows operating system updates. It shows the number of devices that have an anti-virus solution with no known issues, issues

reported, or an unknown status for a particular Windows security update. In the following example, an anti-virus solution that has no known issues with the January 3, 2018 Windows update is installed on about 2,800 devices.



Security update status blade

This blade indicates whether a Windows security update that includes Spectre- or Meltdown-related fixes (January 3, 2018 or later) has been installed, as well as whether specific fixes have been disabled. Though protections are enabled by default on devices running Windows (but not Windows Server) operating systems, some IT administrators might choose to disable specific protections. In the following example, about 4,300 devices have a Windows security update that includes Spectre or Meltdown protections installed, and those protections are enabled.



IMPORTANT

If you are seeing computers with statuses of either “Unknown – action may be required” or “Installed, but mitigation status unknown,” it is likely that you need to whitelist the <http://adl.windows.com> endpoint.

Firmware update status blade

This blade reports the number of devices that have installed a firmware update that includes Spectre or Meltdown protections. The blade might report a large number of blank, “unknown”, or “to be determined” statuses at first. As CPU information is provided by partners, the blade will automatically update with no further action required on your part.

Site discovery

The IE site discovery feature in Upgrade Readiness provides an inventory of web sites that are accessed by client computers using Internet Explorer on Windows 7, Windows 8.1, and Windows 10. Site discovery does not include sites that are accessed using other Web browsers, such as Microsoft Edge. Site inventory information is provided as optional data related to upgrading to Windows 10 and Internet Explorer 11, and is meant to help prioritize compatibility testing for web applications. You can make more informed decisions about testing based on usage data.

NOTE

Site discovery data is disabled by default; you can find documentation on what is collected in the [Windows 7, Windows 8, and Windows 8.1 appraiser diagnostic data events and fields](#). After you turn on this feature, data is collected on all sites visited by Internet Explorer, except during InPrivate sessions. The data collection process is silent, without notification to the employee. You are responsible for ensuring that your use of this feature complies with all applicable local laws and regulatory requirements, including any requirements to provide notice to employees.

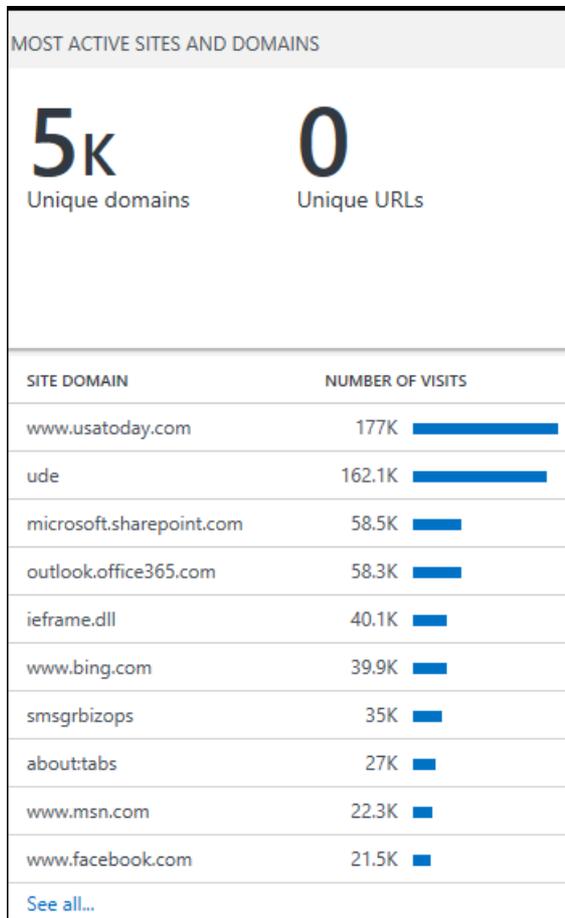
IE site discovery is disabled on devices running Windows 7 and Windows 8.1 that are in Switzerland and EU countries.

In order to use site discovery, a separate opt-in is required; see [Enrolling devices in Windows Analytics](#).

Review most active sites

This blade indicates the most visited sites by computers in your environment. Review this list to determine which web applications and sites are used most frequently. The number of visits is based on the total number of views, and not by the number of unique devices accessing a page.

For each site, the fully qualified domain name will be listed. You can sort the data by domain name or by URL.



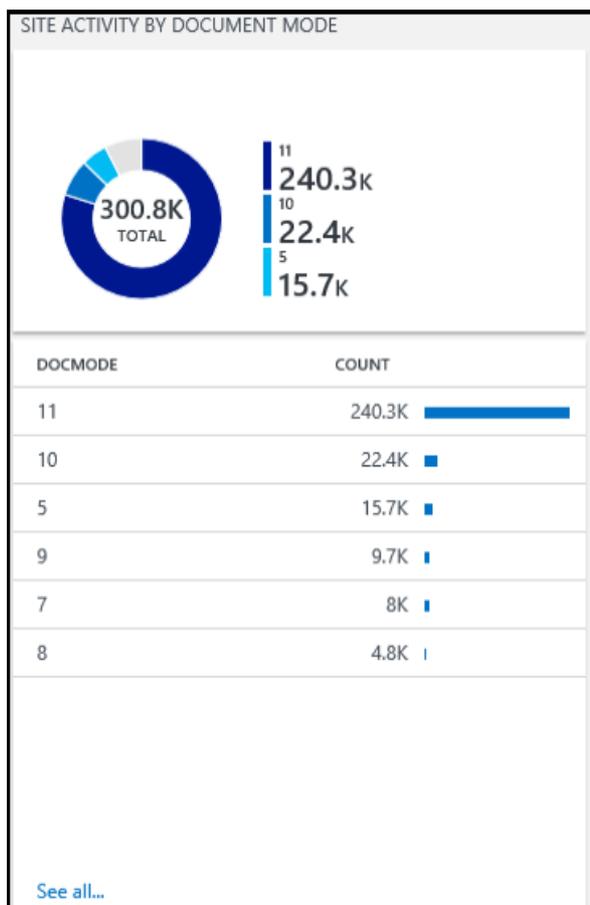
Click the name of any site in the list to drill down into more details about the visits, including the time of each visit and the computer name.

1K Results [List](#) [Table](#)

TIMEGENERATED	COMPUTER	SITENAME	NUMBEROFVISITS	URL	DOCMODE
9/25/2016 3:00...	TestDevice1	www.microsoft.com	1	http://www.microsoft.com/...	IE11 Document M...
9/25/2016 3:00...	TestDevice1	www.microsoft.com	2	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice1	www.microsoft.com	6	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice2	www.microsoft.com	4	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice2	www.microsoft.com	1	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice2	www.microsoft.com	2	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice2	www.microsoft.com	3	https://www.microsoft.com...	IE11 Document M...
9/25/2016 3:00...	TestDevice3	www.microsoft.com	2	https://www.microsoft.com...	IE11 Document M...

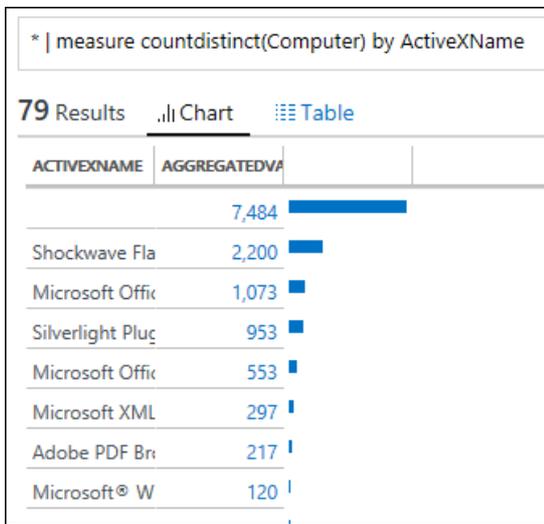
Review document modes in use

This blade provides information about which document modes are used in the sites that are visited in your environment. Document modes are used to provide compatibility with older versions of Internet Explorer. Sites that use older technologies may require additional testing and are less likely to be compatible with Microsoft Edge. Counts are based on total page views and not the number of unique devices. For more information about document modes, see [Deprecated document modes](#).



Run browser-related queries

You can run predefined queries to capture more info, such as sites that have Enterprise Mode enabled, or the number of unique computers that have visited a site. For example, this query returns the most used ActiveX controls. You can modify and save the predefined queries.



Office add-ins

Office add-ins provides a list of the Microsoft Office add-ins in your environment, and enumerates the computers that have these add-ins installed. This information should not affect the upgrade decision workflow, but can be helpful to an administrator.

Related topics

[Manage Windows upgrades with Upgrade Readiness](#)

Targeting a new operating system version

6/14/2019 • 2 minutes to read • [Edit Online](#)

After you've used Upgrade Readiness to help deploy a given version of Windows 10, you might want to use it again to help deploy a newer version of Windows 10. When you change the target operating system version (as described in [Use Upgrade Readiness to manage Windows upgrades](#)), the app states (Importance, AppOwner, UpgradeDecision, TestPlan, and TestResult) are not reset. Follow this guidance to preserve or reset these states as needed:

TestResults

If you want to preserve the TestResults from the previous operating system version testing, there is nothing you need to do.

If you want to reset them, click any of the rows in the **Prioritize Application** blade (described in [Upgrade Readiness - Step 1: Identify important apps](#)). This will take you to the **Log Search** user experience. Replace the query in that window with the following query:

```
search in (UApp) IsRollup == true and RollupLevel == "Granular" and TestResult <> "Not started"
```

After a short period of time, you will see the "user input" perspective render, which will let you bulk-edit the results. Select the check box in the table header, click the **bulk edit** button, and then set the **TestResult** to *Not started*. Leave all other fields as they are.

UpgradeDecision

If you want to preserve the UpgradeDecision from the previous operating system version testing, there is nothing you need to do.

If you want to reset them, keep these important points in mind:

- Make sure to *not* reset the **Ready to upgrade** decision for the "long tail" of apps that have importance of **Ignore** or **Low install count**. Doing this will make it extremely difficult to complete the Upgrade Readiness workflow.
- Decide which decisions to reset. For example, one option is just to reset the decisions marked **Ready to upgrade** (in order to retest those), while preserving states of apps marked **Won't upgrade**. Doing this means you won't lose track of this previous marking. Or you can reset everything.

To do this, type the following query in **Log Search**:

```
search in (UApp) IsRollup == true and RollupLevel == "Granular" and Importance <> "Ignore" and Importance <> "Low install count" and UpgradeDecision == "Ready to upgrade"
```

NOTE

If you just want to reset all **UpgradeDecision** values, you can simply remove

```
'and UpgradeDecision == "Ready to upgrade"' from the query.
```

After a short period of time, you will see the "user input" perspective render, which will let you bulk-edit the results. Select the check box in the table header, click the **bulk edit** button, and then set the **UpgradeDecision** to *Not reviewed*. Leave all other fields as they are.

Bulk-approving apps from a given vendor

You can bulk-approve all apps from a given vendor (for example, Microsoft) if there are no known compatibility issues. To do this, type the following query in **Log Search**:

```
search in (UApp) IsRollup == true and RollupLevel == "Granular" and AppVendor has "Microsoft" and UpgradeAssessment=="No known issues" and UpgradeDecision<>"Ready to upgrade"
```

After a short period of time, you will see the "user input" perspective render, which will let you bulk-edit the results. Select the check box in the table header, click the **bulk edit** button, and then set the **UpgradeDecision** to *Ready to upgrade*. Leave all other fields as they are.

Related topics

[Windows Analytics overview](#)

[Manage Windows upgrades with Upgrade Readiness](#)

[Get started with Upgrade Readiness](#)

Monitor Windows Updates with Update Compliance

5/31/2019 • 2 minutes to read • [Edit Online](#)

Introduction

Update Compliance is a [Windows Analytics solution](#) that enables organizations to:

- Monitor Windows 10 Professional, Education, and Enterprise security, quality, and feature updates.
- View a report of device and update issues related to compliance that need attention.
- See the status of Windows Defender Antivirus signatures and threats.
- Check bandwidth savings incurred across multiple content types by using [Delivery Optimization](#).

Update Compliance is offered through the Azure portal, and is available free for devices that meet the [prerequisites](#).

Update Compliance uses Windows 10 and Windows Defender Antivirus diagnostic data for all of its reporting. It collects system data including update deployment progress, [Windows Update for Business](#) configuration data, Windows Defender Antivirus data, and Delivery Optimization usage data, and then sends this data to a secure cloud to be stored for analysis and usage in [Azure Log Analytics](#).

See the following topics in this guide for detailed information about configuring and using the Update Compliance solution:

- [Get started with Update Compliance](#): How to add Update Compliance to your environment.
- [Using Update Compliance](#): How to begin using Update Compliance.

Update Compliance architecture

The Update Compliance architecture and data flow is summarized by the following four-step process:

1. User computers send diagnostic data to a secure Microsoft data center using the Microsoft Data Management Service.
2. Diagnostic data is analyzed by the Update Compliance Data Service.
3. Diagnostic data is pushed from the Update Compliance Data Service to your Azure Monitor workspace.
4. Diagnostic data is available in the Update Compliance solution.

NOTE

This process assumes that Windows diagnostic data is enabled and data sharing is enabled as described in [Enrolling devices in Windows Analytics](#).

Related topics

[Get started with Update Compliance](#)

[Use Update Compliance to monitor Windows Updates](#)

Get started with Update Compliance

6/6/2019 • 3 minutes to read • [Edit Online](#)

This topic explains the steps necessary to configure your environment for Windows Analytics: Update Compliance.

Steps are provided in sections that follow the recommended setup process:

1. Ensure you meet the [Update Compliance prerequisites](#).
2. [Add Update Compliance to your Azure subscription](#).
3. [Enroll devices in Windows Analytics](#).
4. [Use Update Compliance](#) to monitor Windows Updates, Windows Defender Antivirus status, and Delivery Optimization.

Update Compliance prerequisites

Before you begin the process to add Update Compliance to your Azure subscription, first ensure you can meet the prerequisites:

1. Update Compliance works only with Windows 10 Professional, Education, and Enterprise editions. Update Compliance only provides data for the standard Desktop Windows 10 version and is not currently compatible with Windows Server, Surface Hub, IoT, etc.
2. Update Compliance provides detailed deployment data for devices on the Semi-Annual Channel and the Long-term Servicing Channel. Update Compliance will show Windows Insider Preview devices, but currently will not provide detailed deployment information for them.
3. Update Compliance requires at least the Basic level of diagnostic data and a Commercial ID to be enabled on the device.
4. To show device names for versions of Windows 10 starting with 1803 in Windows Analytics you must opt in. For details about this, see the "AllowDeviceNameInTelemetry (in Windows 10)" entry in the table in the [Distributing policies at scale](#) section of [Enrolling devices in Windows Analytics](#).
5. To use the Windows Defender Status, devices must be E3-licensed and have Cloud Protection enabled. E5-licensed devices will not appear here. For E5 devices, you should use [Windows Defender ATP](#) instead. For more information on Windows 10 Enterprise licensing, see [Windows 10 Enterprise: FAQ for IT Professionals](#).

Add Update Compliance to your Azure subscription

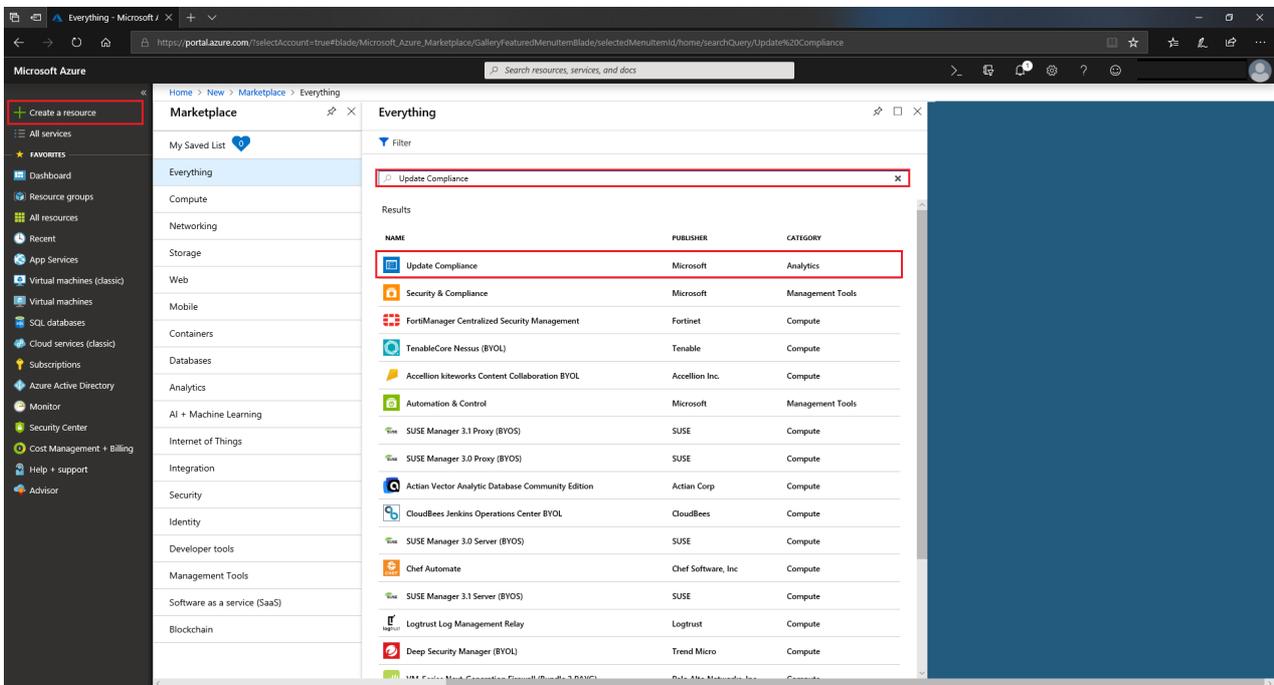
Update Compliance is offered as a solution which is linked to a new or existing [Azure Log Analytics](#) workspace within your Azure subscription. To configure this, follow these steps:

1. Sign in to the [Azure Portal](#) with your work or school account or a Microsoft account. If you don't already have an Azure subscription you can create one (including free trial options) through the portal.

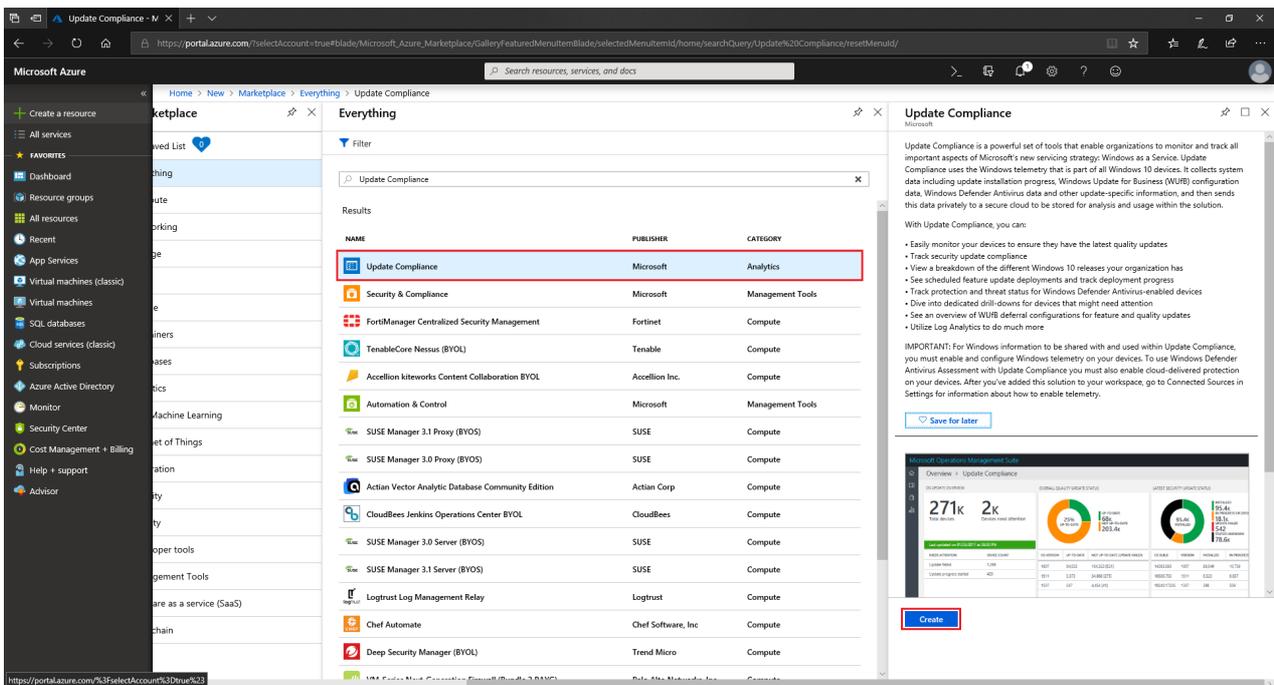
NOTE

Update Compliance is included at no additional cost with Windows 10 Professional, Education, and Enterprise editions. An Azure subscription is required for managing and using Update Compliance, but no Azure charges are expected to accrue to the subscription as a result of using Update Compliance.

2. In the Azure portal select + **Create a resource**, and search for "Update Compliance". You should see it in the results below.

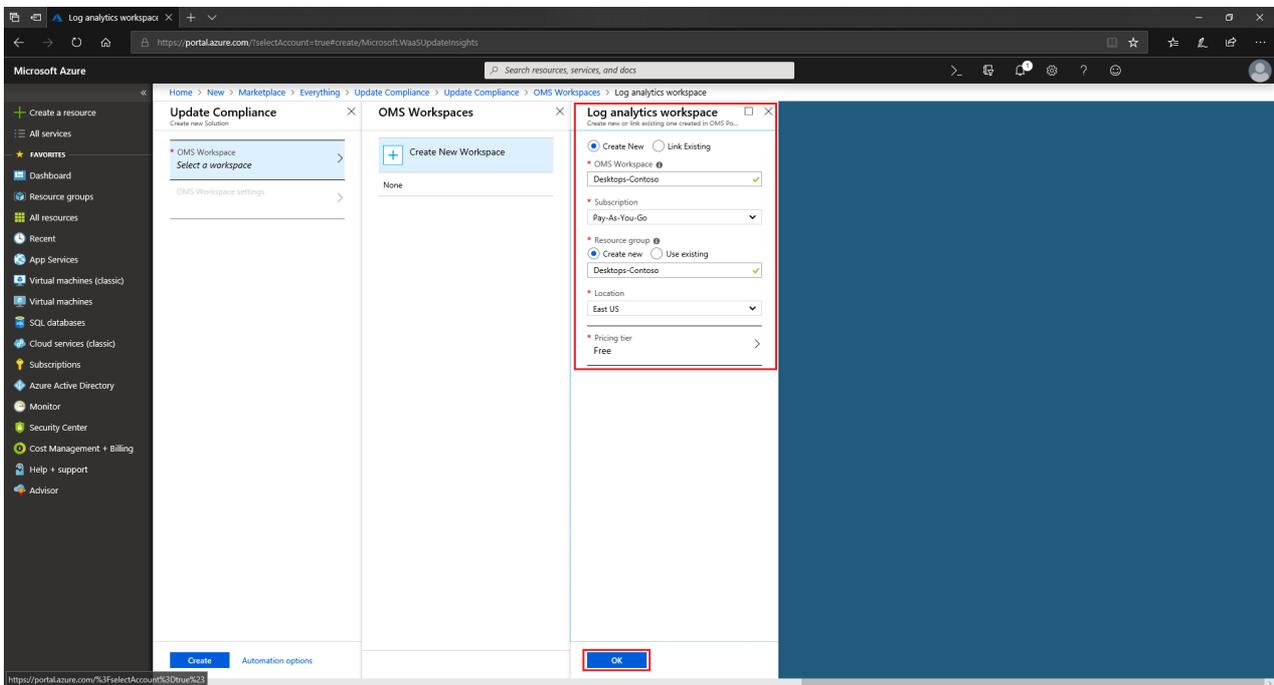


3. Select **Update Compliance** and a blade will appear summarizing the solution's offerings. At the bottom, select **Create** to begin adding the solution to Azure.

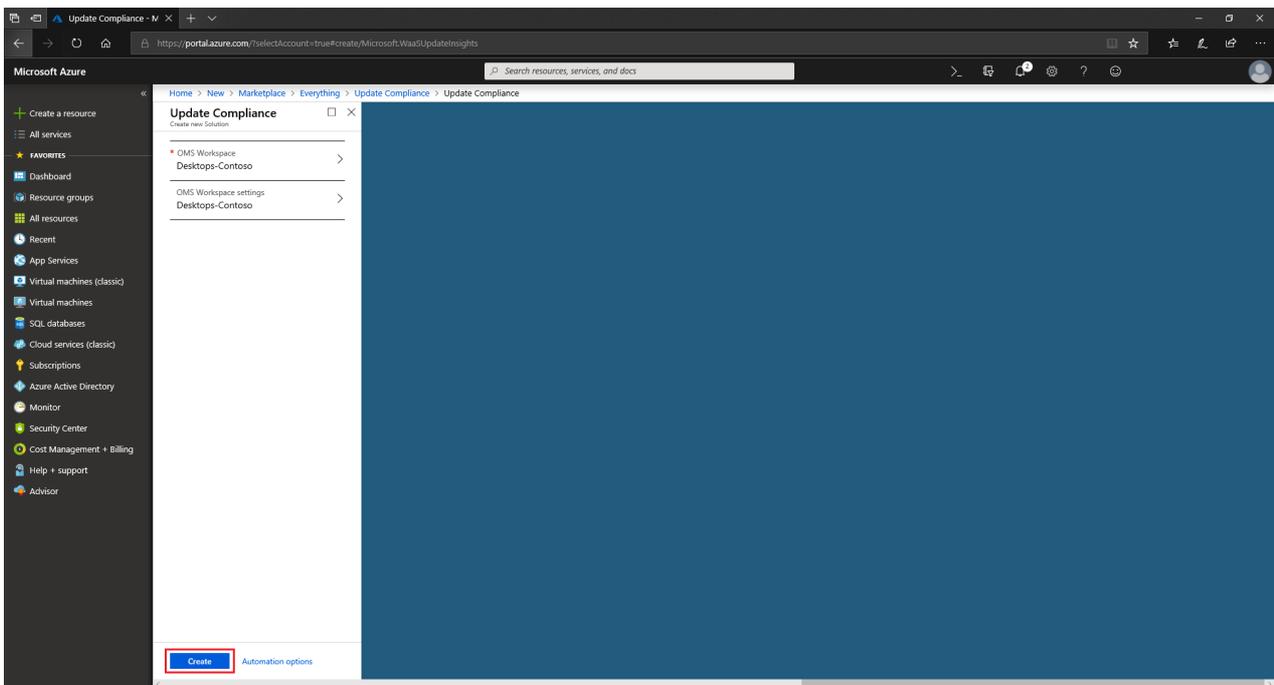


4. Choose an existing workspace or create a new workspace that will be assigned to the Update Compliance solution.

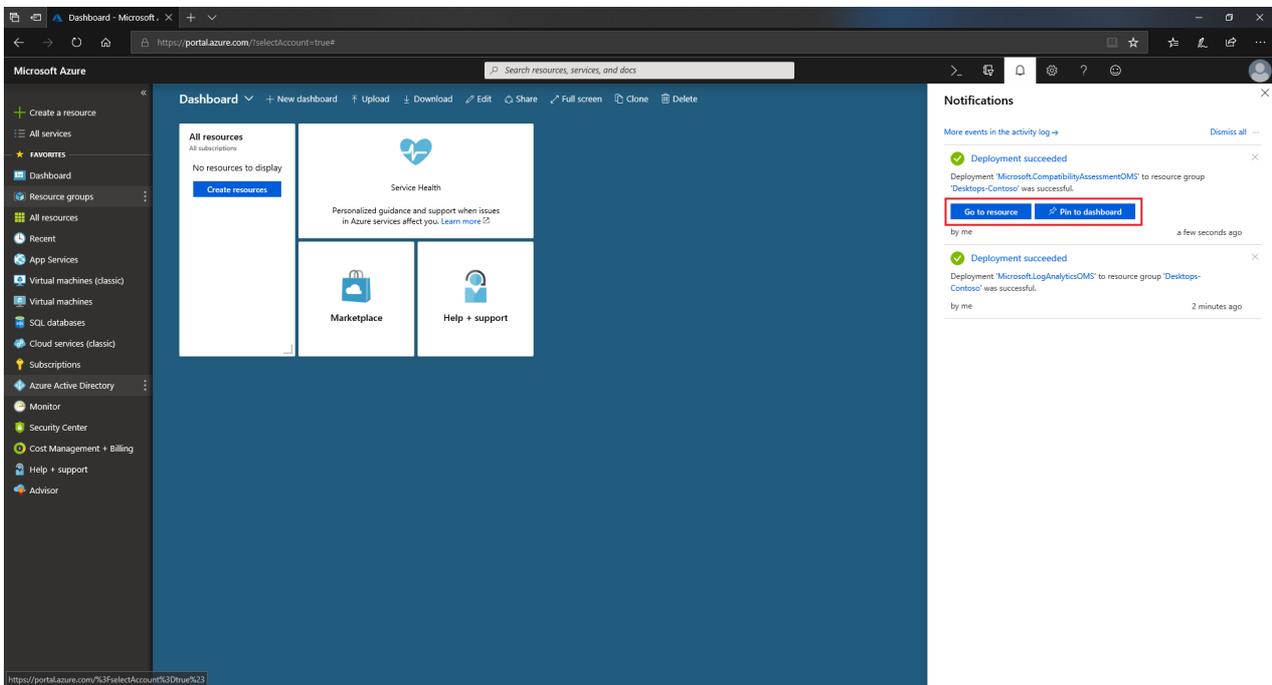
- If you already have another Windows Analytics solution, you should use the same workspace.
- If you are creating a new workspace, and your organization does not have policies governing naming conventions and structure, consider the following workspace settings to get started:
 - Choose a workspace name which reflects the scope of planned usage in your organization, for example *PC-Analytics*.
 - For the resource group setting select **Create new** and use the same name you chose for your new workspace.
 - For the location setting, choose the Azure region where you would prefer the data to be stored.
 - For the pricing tier select **per GB**.



5. The resource group and workspace creation process could take a few minutes. After this, you are able to use that workspace for Update Compliance. Select **Create**.



6. Watch for a notification in the Azure portal that your deployment has been successful. This might take a few minutes. Then, select **Go to resource**.



Enroll devices in Windows Analytics

Once you've added Update Compliance to a workspace in your Azure subscription, you can start enrolling the devices in your organization. For Update Compliance there are two key steps for enrollment:

1. Deploy your Commercial ID (from the Update Compliance Settings page) to your Windows 10 devices (typically by using Group Policy, [Mobile Device Management](#), [System Center Configuration Manager](#) or similar).
2. Ensure the Windows Diagnostic Data setting on devices is set to at least Basic (typically using Group Policy or similar). For full enrollment instructions and troubleshooting, see [Enrolling devices in Windows Analytics](#).

After enrolling your devices (by deploying your CommercialID and Windows Diagnostic Data settings), it might take 48-72 hours for the first data to appear in the solution. Until then, Update Compliance will indicate it is still assessing devices.

Use Update Compliance

6/6/2019 • 5 minutes to read • [Edit Online](#)

In this section you'll learn how to use Update Compliance to monitor your device's Windows updates and Windows Defender Antivirus status. To configure your environment for use with Update Compliance, refer to [Get started with Update Compliance](#).

Update Compliance:

- Provides detailed deployment data for Windows 10 security, quality, and feature updates.
- Reports when devices have issues related to updates that need attention.
- Shows Windows Defender AV status information for devices that use it and meet the [prerequisites](#).
- Shows bandwidth usage and savings for devices that are configured to use [Delivery Optimization](#).
- Provides all of the above data in [Log Analytics](#), which affords additional querying and export capabilities.

The Update Compliance tile

After Update Compliance has successfully been [added to your Azure subscription](#), you'll see this tile:

Update Compliance (Preview)

Performing Assessment

No devices have been detected. Note that it can take up to 24 hours for configured devices to appear and may take longer if they are not currently connected to internet. Refer to <http://aka.ms/UpdateCompliance> for more info on how to configure devices.

When the solution is added, data is not immediately available. Data will begin to be collected after data is sent up that belongs to the Commercial ID associated with the device. This process assumes that Windows diagnostic data is enabled and data sharing is enabled as described in [Enrolling devices in Windows Analytics](#). After Microsoft has collected and processed any device data associated with your Commercial ID, the tile will be replaced with the following summary:

Update Compliance

5k

Total devices

563

Devices need attention

Last updated on 09/27/2018 at 10:00 PM

The summary details the total number of devices that Microsoft has received data from with your Commercial ID. It also provides the number of devices that need attention if any. Finally, it details the last point at which your Update Compliance workspace was refreshed.

The Update Compliance workspace

NEED ATTENTION!

Device issues

173

DEVICE ISSUES	COUNT
Out of support OS Version	0
Missing multiple security updates	173

Update issues

503

UPDATE ISSUES	COUNT
Failed	264
Progress stalled	239
Canceled	0
Rollback	0
Uninstalled	0

Setup Diagnostic Tool

LIST OF QUERIES

- Update deployment failures
WaaSDeploymentStatus | where DeploymentStatus == "Failed" |...
- Devices pending reboot to complete update
WaaSDeploymentStatus | where DetailedStatus == "Reboot pend...
- OS Servicing branch distribution for the devices
WaaSUpdateStatus | summarize DeviceCount = count() by OSSer...
- OS Edition distribution for the devices
WaaSUpdateStatus | summarize DeviceCount = count() by OSEdit...
- Deferral configurations for Feature Update
WaaSUpdateStatus | summarize DeviceCount = count() by Featur...
- Pause configurations for Feature Update
WaaSUpdateStatus | summarize DeviceCount = count() by Featur...
- Deferral configurations for Quality Update
WaaSUpdateStatus | summarize DeviceCount = count() by Qualit...
- Pause configurations for Quality Update
WaaSUpdateStatus | summarize DeviceCount = count() by Qualit...
- Devices not assessed for Defender AV
WDAVStatus | where UpdateStatus == "Not assessed" | render ta...
- Inventory of devices on Insider builds
WaaSInsiderStatus

When you select this tile, you will be redirected to the Update Compliance workspace. The workspace is organized with the Overview blade providing a hub from which to navigate to different reports of your devices' data.

Overview blade

OVERVIEW: 5,000 DEVICES (0 ON INSIDER)

■ UP-TO-DATE
■ NOT UP-TO-DATE

Item	Up-to-date	Not up-to-date
Security Update	4.3k	1.1k
Feature Update	4.3k	0.5k
AV Signature	3.3k	1.1k

Last updated on 09/27/2018 at 10:00 PM

Need Attention! 563

of devices that need attention

Security Update Status 73.6%

% of devices on latest security update

Feature Update Status 86.9%

% of devices on latest feature update

Windows Defender AV Status 70.6%

% of assessed devices with insufficient protection

Delivery Optimization Status 66.5%

% bandwidth savings in Feature and Quality Updates over the last 28 days

Update Compliance's overview blade summarizes all the data Update Compliance provides. It functions as a hub from which you can navigate to different sections. The total number of devices detected by Update Compliance is reported in the title of this blade. What follows is a distribution for all devices as to whether they are up to date on the following items:

- Security updates: A device is up to date on quality updates whenever it has the latest applicable quality update installed. Quality updates are monthly cumulative updates that are specific to a version of Windows 10.
- Feature updates: A device is up to date on feature updates whenever it has the latest applicable feature update

installed. Update Compliance considers [Servicing Channel](#) when determining update applicability.

- **AV Signature:** A device is up to date on Antivirus Signature when the latest Windows Defender Signatures have been downloaded. This distribution only considers devices that are running Windows Defender Antivirus.

The blade also provides the time at which your Update Compliance workspace was [refreshed](#).

The following is a breakdown of the different sections available in Update Compliance:

- **Need Attention!** - This section is the default section when arriving to your Update Compliance workspace. It provides a summary of the different issues devices are facing relative to Windows 10 updates.
- **Security Update Status** - This section lists the percentage of devices that are on the latest security update released for the version of Windows 10 it is running. Selecting this section provides blades that summarize the overall status of security updates across all devices and a summary of their deployment progress towards the latest two security updates.
- **Feature Update Status** - This section lists the percentage of devices that are on the latest feature update that is applicable to a given device. Selecting this section provides blades that summarize the overall feature update status across all devices and a summary of deployment status for different versions of Windows 10 in your environment.
- **Windows Defender AV Status** - This section lists the percentage of devices running Windows Defender Antivirus that are not sufficiently protected. Selecting this section provides a summary of signature and threat status across all devices that are running Windows Defender Antivirus. This section is not applicable to devices not running Windows Defender Antivirus or devices that do not meet the [prerequisites](#) to be assessed.
- **Delivery Optimization Status** - This section summarizes bandwidth savings incurred by utilizing Delivery Optimization in your environment. It provides a breakdown of Delivery Optimization configuration across devices, and summarizes bandwidth savings and utilization across multiple content types.

Update Compliance data latency

Update Compliance uses Windows 10 diagnostic data as its data source. After you add Update Compliance and appropriately configure your devices, it could take 48-72 hours before they first appear. The process that follows is as follows:

Update Compliance is refreshed every 12 hours. This means that every 12 hours all data that has been gathered over the last 12-hour interval is pushed to Log Analytics. However, the rate that each data type is sent and how long it takes to be ready for Update Compliance varies, roughly outlined below.

DATA TYPE	REFRESH RATE	DATA LATENCY
WaaSUpdateStatus	Once per day	4 hours
WaaSInsiderStatus	Once per day	4 hours
WaaSDeploymentStatus	Every update event (Download, install, etc.)	24-36 hours
WDAVStatus	On signature update	24 hours
WDAVThreat	On threat detection	24 hours
WUDOAggregatedStatus	On update event, aggregated over time	24-36 hours
WUDOStatus	Once per day	12 hours

This means you should generally expect to see new data every 24-36 hours, except for WaaSDeploymentStatus

and WUDOAgregatedStatus, which may take 36-48 hours (if it misses the 36th hour refresh, it would be in the 48th, so the data will be present in the 48th hour refresh).

Using Log Analytics

Update Compliance is built on the Log Analytics platform that is integrated into Operations Management Suite. All data in the workspace is the direct result of a query. Understanding the tools and features at your disposal, all integrated within Azure Portal, can deeply enhance your experience and complement Update Compliance.

See below for a few topics related to Log Analytics:

- Learn how to effectively execute custom Log Searches by referring to Microsoft Azure's excellent documentation on [querying data in Log Analytics](#).
- To develop your own custom data views in Operations Management Suite or [Power BI](#); check out documentation on [analyzing data for use in Log Analytics](#).
- [Gain an overview of Log Analytics' alerts](#) and learn how to use it to always stay informed about the most critical issues you care about.

Related topics

[Get started with Update Compliance](#)

Needs attention!

5/31/2019 • 2 minutes to read • [Edit Online](#)

NEED ATTENTION!

Device issues

173

DEVICE ISSUES	COUNT
Out of support OS Version	0
Missing multiple security updates	173

Update issues

503

UPDATE ISSUES	COUNT
Failed	264
Progress stalled	239
Canceled	0
Rollback	0
Uninstalled	0

[Setup Diagnostic Tool](#)

LIST OF QUERIES

- Update deployment failures
[WaaSDeploymentStatus | where DeploymentStatus == "Failed" |...](#)
- Devices pending reboot to complete update
[WaaSDeploymentStatus | where DetailedStatus == "Reboot pend...](#)
- OS Servicing branch distribution for the devices
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by OSSer...](#)
- OS Edition distribution for the devices
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by OSEdit...](#)
- Deferral configurations for Feature Update
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by Featur...](#)
- Pause configurations for Feature Update
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by Featur...](#)
- Deferral configurations for Quality Update
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by Qualit...](#)
- Pause configurations for Quality Update
[WaaSUpdateStatus | summarize DeviceCount = count\(\) by Qualit...](#)
- Devices not assessed for Defender AV
[WDAVStatus | where UpdateStatus == "Not assessed" | render ta...](#)
- Inventory of devices on Insider builds
[WaaSInsiderStatus](#)

The **Needs attention!** section provides a breakdown of all Windows 10 device and update issues detected by Update Compliance. The summary tile for this section counts the number of devices that have issues, while the blades within break down the issues encountered. Finally, a [list of queries](#) blade in this section contains queries that provide values but do not fit within any other main section.

NOTE

The summary tile counts the number of devices that have issues, while the blades within the section break down the issues encountered. A single device can have more than one issue, so these numbers might not add up.

The different issues are broken down by Device Issues and Update Issues:

Device Issues

- **Missing multiple security updates:** This issue occurs when a device is behind by two or more security updates. These devices might be more vulnerable and should be investigated and updated.
- **Out of support OS Version:** This issue occurs when a device has fallen out of support due to the version of Windows 10 it is running. When a device has fallen out of support, it will no longer receive important security updates, and might be vulnerable. These devices should be updated to a supported version of Windows 10.

Update Issues

- **Failed:** This issue occurs when an error halts the process of downloading and applying an update on a device. Some of these errors might be transient, but should be investigated further to be sure.
- **Cancelled:** This issue occurs when a user cancels the update process.
- **Rollback:** This issue occurs when a fatal error occurs during a feature update, and the device is rolled back to the previous version.
- **Uninstalled:** This issue occurs when a feature update is uninstalled from a device by a user or an administrator.

Note that this might not be a problem if the uninstallation was intentional, but is highlighted as it might need attention.

- **Progress stalled:** This issue occurs when an update is in progress, but has not completed over a period of 10 days.

Selecting any of the issues will take you to a [Log Analytics](#) view with all devices that have the given issue.

NOTE

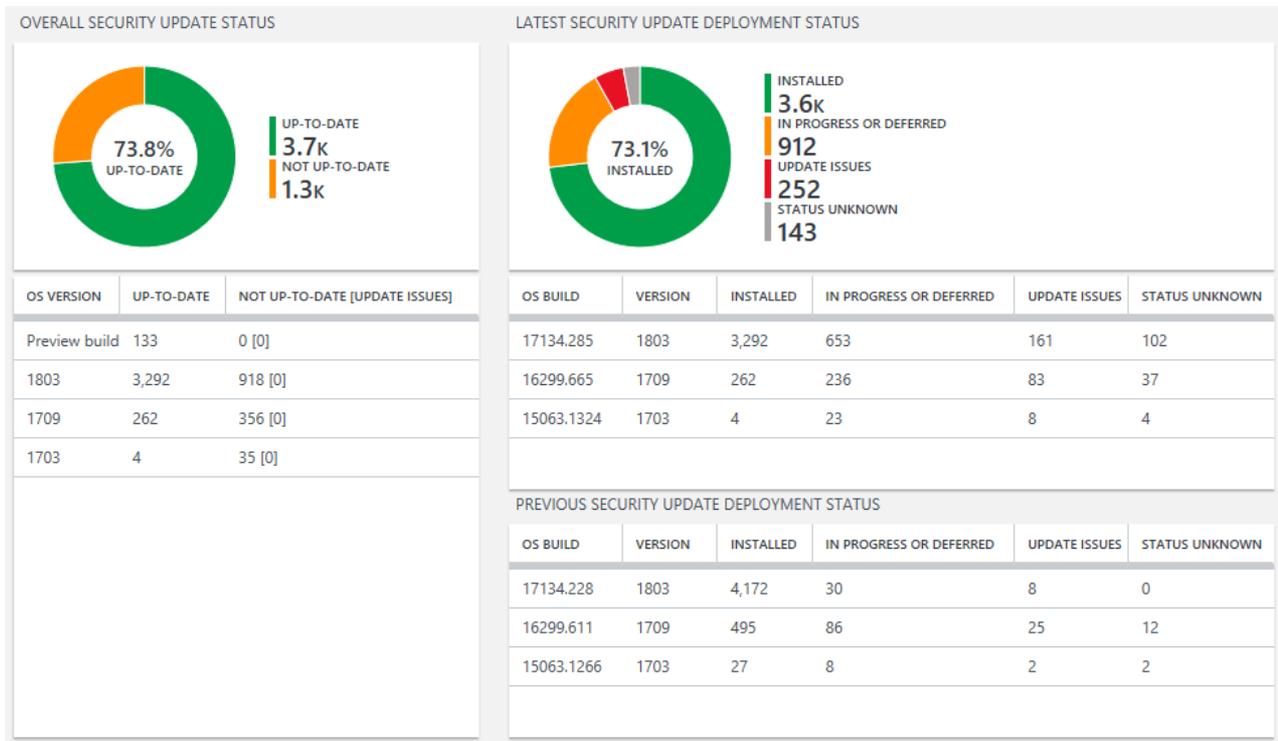
This blade also has a link to the [Setup Diagnostic Tool](#), a standalone tool you can use to obtain details about why a Windows 10 feature update was unsuccessful.

List of Queries

The **List of Queries** blade is in the **Needs Attention** section of Update Compliance. This blade contains a list of queries with a description and a link to the query. These queries contain important meta-information that did not fit within any specific section or were listed to serve as a good starting point for modification into custom queries.

Security Update Status

6/12/2019 • 4 minutes to read • [Edit Online](#)



The Security Update Status section provides information about [security updates](#) across all devices. The section tile within the [Overview Blade](#) lists the percentage of devices on the latest security update available. Meanwhile, the blades within show the percentage of devices on the latest security update for each Windows 10 version and the deployment progress toward the latest two security updates.

The **Overall Security Update Status** blade provides a visualization of devices that are and do not have the latest security updates. Below the visualization are all devices further broken down by operating system version and a count of devices that are up to date and not up to date. The **Not up to date** column also provides a count of update failures.

The **Latest Security Update Status** and **Previous Security Update Status** tiles are stacked to form one blade. The **Latest Security Update Status** provides a visualization of the different deployment states devices are in regarding the latest update for each build (or version) of Windows 10, along with the revision of that update. The **Previous Security Update Status** blade provides the same information without the accompanying visualization.

The various deployment states reported by devices are as follows:

Deployment status

Deployment status summarizes detailed status into higher-level states to get a quick sense of the status the given device was last reported to be in relative to this specific update. Note that with the latency of deployment data, devices might have since moved on from the reported deployment status.

DEPLOYMENT STATUS	DESCRIPTION
Failed	The device encountered a failure during the update process. Note that due to latency, devices reporting this status may have since retried the update.

DEPLOYMENT STATUS	DESCRIPTION
Progress stalled	The device started the update process, but no progress has been reported in the last 7 days.
Deferred	The device is currently deferring the update process due to Windows Update for Business policies.
In progress	The device has begun the updating process for this update. This status appears if the device is in any stage of the update process including and after download, but before completing the update. If no progress has been reported in the last 7 days, devices will move to Progress stalled .**
Update completed	The device has completed the update process.
Update paused	The device is prevented from being offered the update due to updates being paused on the device.
Unknown	No record is available for this device relative to this update. This is a normal status if an update has recently been released or if the device does not use Windows Update.

Detailed status

Detailed status provides a detailed stage-level representation of where in the update process the device was last reported to be in relative to this specific update. Note that with the latency of deployment data, devices might have since moved on from the reported detailed status.

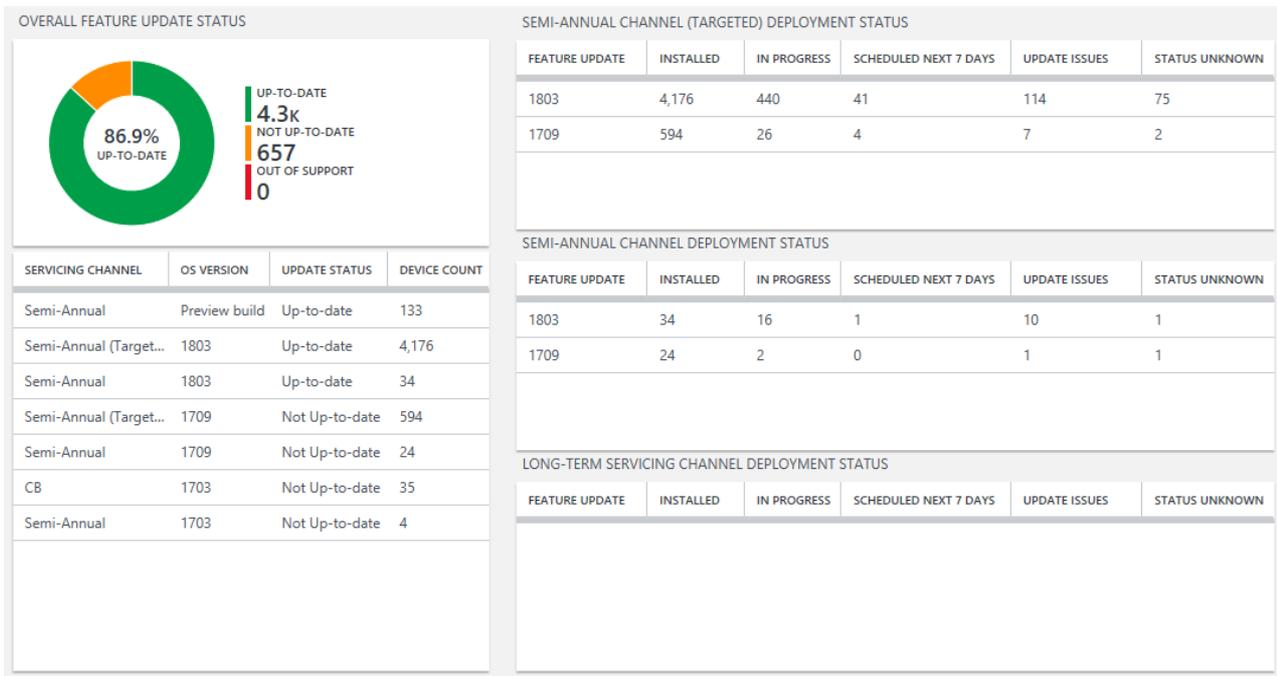
DETAILED STATUS	DESCRIPTION
Scheduled in next X days	The device is currently deferring the update with Windows Update for Business policies but will be offered the update within the next X days.
Compatibility hold	The device has been placed under a <i>compatibility hold</i> to ensure a smooth feature update experience and will not resume the update until the hold has been cleared. For more information see Feature Update Status report
Update deferred	The device is currently deferring the update with Windows Update for Business policies.
Update paused	The device is prevented from being offered the update due to updates being paused on the device.
Update offered	The device has been offered the update by Windows Update but has not yet begun to download it.
Download started	The device has begun downloading the update.
Download succeeded	The device has finished downloading the update but has not yet begun installing the update.
Install started	The device has begun installing the update.

DETAILED STATUS	DESCRIPTION
PreInstall task passed	The device has passed checks prior to beginning the rest of the installation process after a restart.
Reboot required	The device requires a restart to install the update, but one has not yet been scheduled.
Reboot pending	The device is pending a restart to install the update.
Reboot initiated	The device reports "Reboot initiated" just before actually restarting specifically to apply the update.
Commit	The device, after a restart, is committing changes relevant to the update.
Finalize succeeded	The device has finished final tasks after a restart to apply the update.
Update successful	The device has successfully applied the update.
Cancelled	The update was cancelled at some point in the update process.
Uninstalled	The update was successfully uninstalled from the device.
Rollback	The update failed to apply during the update process, causing the device to roll back changes and revert to the previous update.

The rows of each tile in this section are interactive; selecting them will navigate you to the query that is representative of that row and section.

Feature Update Status

5/31/2019 • 2 minutes to read • [Edit Online](#)



The Feature Update Status section provides information about the status of [feature updates](#) across all devices. This section tile in the [Overview Blade](#) gives a percentage of devices that are on the latest applicable feature update; [Servicing Channel](#) is considered in determining applicability. Within this section are two blades; one providing a holistic view of feature updates, the other containing three **Deployment Status** tiles, each charged with tracking the deployment for a different [Servicing Channel](#).

Overall Feature Update Status

The Overall Feature Update Status blade breaks down how many devices are up-to-date or not, with a special callout for how many devices are running a build that is not supported (for a full list of feature updates, check out the [Windows 10 Release Information](#) page). The table beneath the visualization breaks devices down by Servicing Channel and operating system version, then defining whether this combination is *up-to-date*, *not up-to-date* or *out of support*. Finally, the table provides a count of devices that fall into this category.

Deployment Status by Servicing Channel

To effectively track deployment, **Deployment Status Blades** are divided into each Servicing Channel chosen for the device. This is because Deployment for each channel will happen at different periods in time and feature updates are targeted separately for each channel. Within each Deployment Status tile, devices are aggregated on their feature update distribution, and the columns list the states each device is in.

Refer to the following list for what each state means:

- **Installed** devices are devices that have completed installation for the given update.
- When a device is counted as **In Progress**, it has begun the feature update installation.
- Devices that are **scheduled next 7 days** are all devices that were deferred from installing the Feature update using [Windows Update for Business Settings](#) and are set to begin installation in the next 7 days.
- Devices that have failed the given feature update installation are counted as **Update failed**.

- If a device should be, in some way, progressing toward this security update, but its status cannot be inferred, it will count as **Status Unknown**. Devices not using Windows Update are the most likely devices to fall into this category.

Compatibility holds

Microsoft uses diagnostic data to determine whether devices that use Windows Update are ready for a feature update in order to ensure a smooth experience. When Microsoft determines a device is not ready to update due to a known issue, a *compatibility hold* is generated to delay the device's upgrade and safeguard the end-user experience. Holds are released over time as diagnostic data is analyzed and fixes are addressed. Details are provided on some, but not all compatibility holds on the Windows 10 release information page for any given release.

To learn how compatibility holds are reflected in the experience, see [Update compliance perspectives](#).

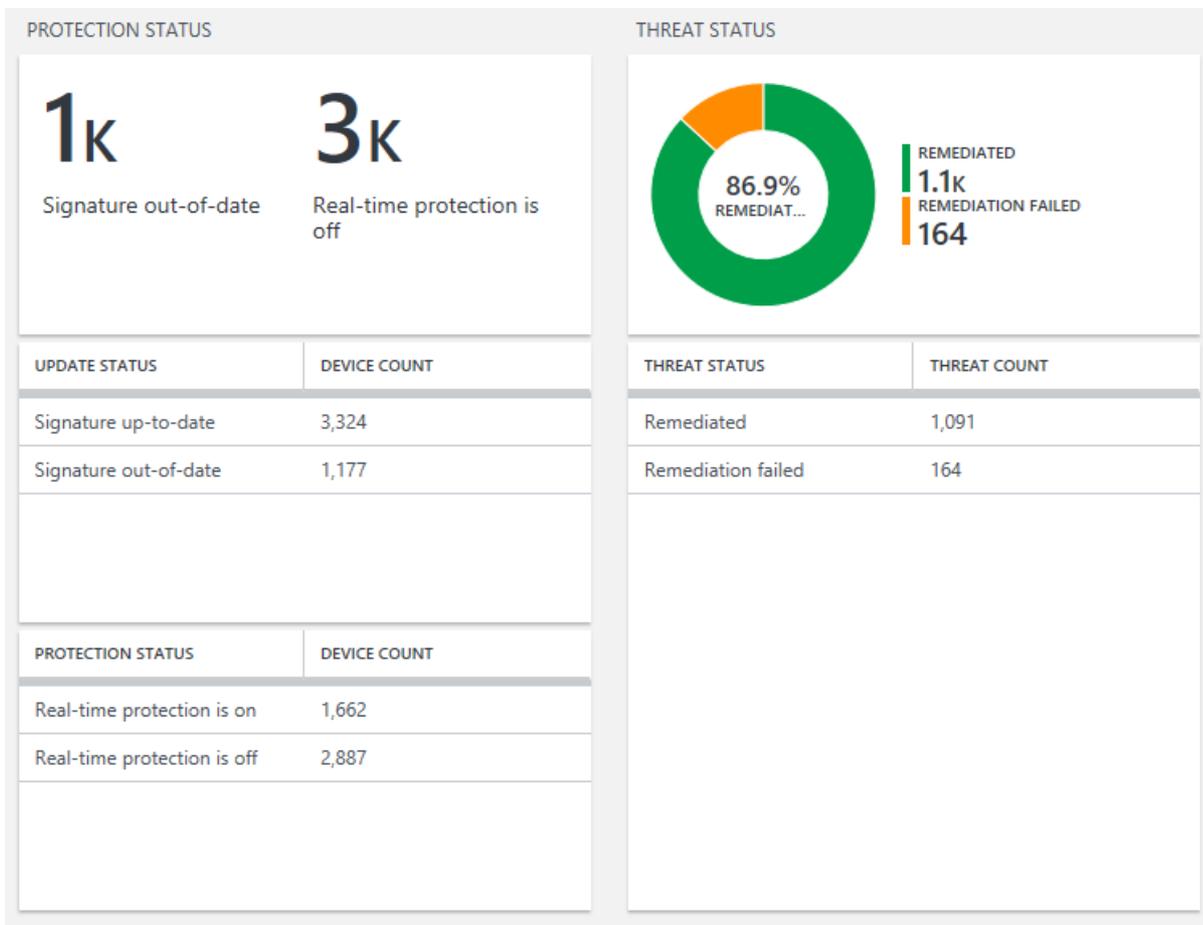
Opting out of compatibility hold

Microsoft will release a device from a compatibility hold when it has determined it can safely and smoothly install a feature update, but you are ultimately in control of your devices and can opt out if desired. To opt out, set the registry key **HKLM\Software\Microsoft\Windows NT\CurrentVersion\502505fe-762c-4e80-911e-0c3fa4c63fb0** to a name of **DataRequireGatedScanForFeatureUpdates** and a value of **0**.

Setting this registry key to **0** will force the device to opt out from *all* compatibility holds. Any other value, or deleting the key, will resume compatibility protection on the device.

Windows Defender AV Status

6/26/2019 • 2 minutes to read • [Edit Online](#)



The Windows Defender AV Status section deals with data concerning signature and threat status for devices that use Windows Defender Antivirus. The section tile in the [Overview Blade](#) provides the percentage of devices with insufficient protection – this percentage only considers devices using Windows Defender Antivirus.

NOTE

Update Compliance's Windows Defender Antivirus status is compatible with E3, B, F1, VL Professional and below licenses. Devices with an E5 license are not shown here; devices with an E5 license can be monitored using the [Windows Defender ATP portal](#). If you'd like to learn more about Windows 10 licensing, see the [Windows 10 product licensing options](#).

Windows Defender AV Status sections

The **Protection Status** blade gives a count for devices that have either out-of-date signatures or real-time protection turned off. Below, it gives a more detailed breakdown of the two issues. Selecting any of these statuses will navigate you to a Log Search view containing the query.

The **Threat Status** blade shows, among devices that have encountered threats, how many were and were not remediated successfully. It also provides a detailed count. Selecting either of these will take you to the respective query in Log Search for further investigation.

Here are some important terms to consider when using the Windows Defender AV Status section of Update Compliance:

- **Signature out of date** devices are devices with a signature older than 14 days.
- **No real-time protection** devices are devices that are using Windows Defender AV but have turned off real-time protection.
- **Recently disappeared** devices are devices that were previously seen by Windows Defender AV and are no longer seen in the past 7 days.
- **Remediation failed** devices are devices where Windows Defender AV failed to remediate the threat. This could be due to a number of reasons, including a full disk, network error, operation aborted, etc. Manual intervention might be needed from IT team.
- **Not assessed** devices are devices where either a non-Microsoft AV solution is used or it has been more than 7 days since the device recently disappeared.

Windows Defender data latency

Because of the way Windows Defender is associated with the rest of Windows device data, Defender data for new devices might take much longer to appear than other data types. This process could take up to 28 days.

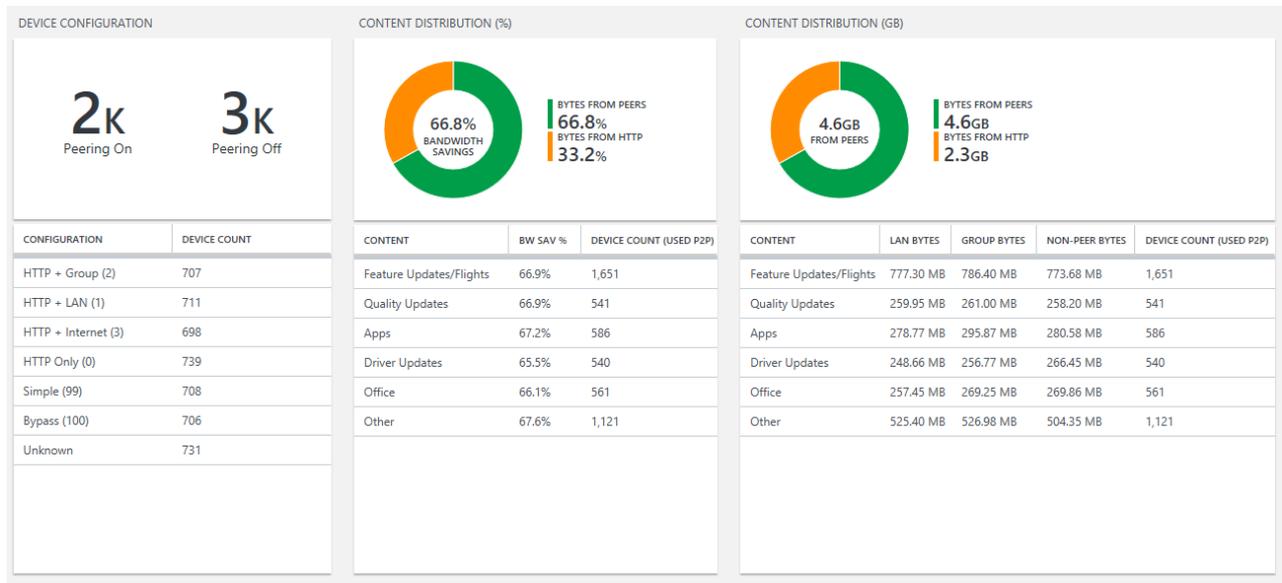
Related topics

- [Windows Defender Antivirus pre-requisites](#)

Delivery Optimization in Update Compliance

5/31/2019 • 2 minutes to read • [Edit Online](#)

The Update Compliance solution of Windows Analytics provides you with information about your Delivery Optimization configuration, including the observed bandwidth savings across all devices that used peer-to-peer distribution over the past 28 days.



IMPORTANT

There are currently two known issues affecting the Delivery Optimization status displayed in these blades:

- Devices running Windows 10, version 1803 or older versions are not sending the correct configuration profile. As a result, the information in the Device Configuration blade might not accurately reflect the settings in your environment.
- Some devices running Windows 10, version 1809 report the Delivery Optimization DownloadMode configuration value as the sequential value in the list of possible configurations rather than the actual configured value. For example, a device that is configured as HTTP + Group (2), will be shown as HTTP + Internet (3) in Update Compliance.

Look for fixes for both of these issues in a forthcoming update.

Delivery Optimization Status

The Delivery Optimization Status section includes three blades:

- The **Device Configuration** blade shows a breakdown of download configuration for each device
- The **Content Distribution (%)** blade shows the percentage of bandwidth savings for each category
- The **Content Distribution (GB)** blade shows the total amount of data seen from each content type broken down by the download source (peers vs non-peers).

Device Configuration blade

Devices can be set to use different download modes; these download modes determine in what situations Delivery Optimization will use peer-to-peer distribution to accomplish the downloads. The top section shows the number of devices configured to use peer-to-peer distribution in *Peering On* compared to *Peering Off* modes. The table shows a breakdown of the various download mode configurations seen in your environment. For more

information about the different configuration options, see [Set up Delivery Optimization for Windows 10 updates](#) for recommendations for different scenarios or [Delivery Optimization reference](#) for complete details of this setting.

Content Distribution (%) blade

The first of two blades showing information on content breakdown, this blade shows a ring chart summarizing **Bandwidth Savings %**, which is the percentage of data received from peer sources out of the total data downloaded (for any device that used peer-to-peer distribution). The table breaks down the Bandwidth Savings % into specific content categories along with the number of devices seen downloading the given content type that used peer-to-peer distribution.

Content Distribution (GB) blade

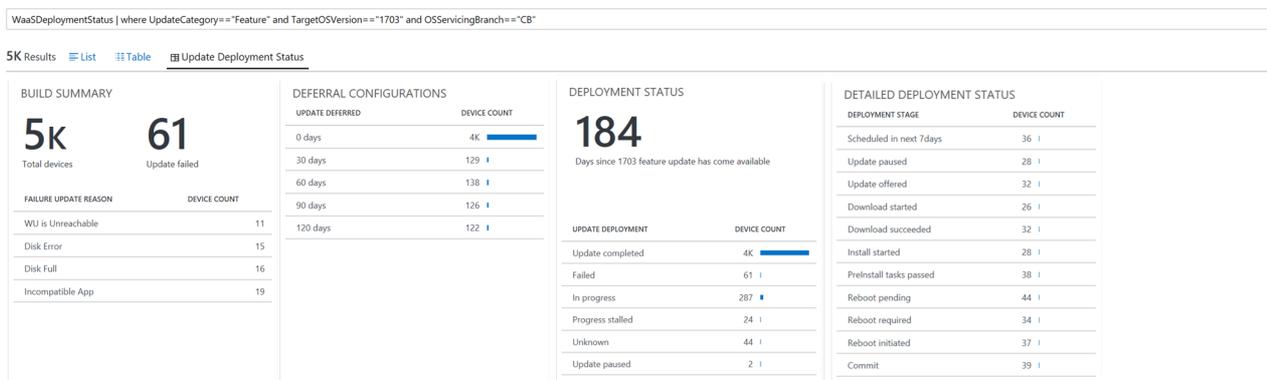
The second of two blades showing information on content breakdown, this blade shows a ring chart summarizing the total bytes downloaded by using peer-to-peer distribution compared to HTTP distribution. The table breaks down the number of bytes from each download source into specific content categories, along with the number of devices seen downloading the given content type that used peer-to-peer distribution.

The download sources that could be included are:

- LAN Bytes: Bytes downloaded from LAN Peers which are other devices on the same local network
- Group Bytes: Bytes downloaded from Group Peers which are other devices that belong to the same Group (available when the "Group" download mode is used)
- HTTP Bytes: Non-peer bytes. The HTTP download source can be Microsoft Servers, Windows Update Servers, a WSUS server or an SCCM Distribution Point for Express Updates.

Perspectives

5/31/2019 • 3 minutes to read • [Edit Online](#)



Perspectives are elaborations on specific queries hand-crafted by developers which data views that provide deeper insight into your data. Perspectives are loaded whenever clicking into more detailed views from both the Security Update Status section and Feature Update Status section of Update Compliance.

There is only one perspective framework; it is for **Update Deployment Status**. The same framework is utilized for both feature and quality updates.

The first blade is the **Build Summary** blade. This blade summarizes the most important aspects of the given build being queried, listing the total number of devices, the total number of update failures for the build, and a breakdown of the different errors encountered.

The second blade is the **Deferral Configurations** blade, breaking down Windows Update for Business deferral settings (if any).

Deployment status

The third blade is the **Deployment Status** blade. This defines how many days it has been since the queried version has been released, and breaks down the various states in the update funnel each device has reported to be in. The possible states are as follows:

STATE	DESCRIPTION
Update Completed	When a device has finished the update process and is on the queried update, it will display here as Update completed.
In Progress	Devices that report they are "In Progress" are one of the various stages of installing an update; these stages are reported in the Detailed Deployment Status blade.
Deferred	When a device's Windows Update for Business deferral policy dictates that the update is not yet applicable due to deferral, it will report as such in this blade.

STATE	DESCRIPTION
Progress stalled	Devices that report as "Progress stalled" have been stuck at "In progress" for more than 7 days.
Cancelled	The update was cancelled.
Blocked	There is a hard block on the update being completed. This could be that another update must be completed before this one, or some other task is blocking the installation of the update.
Unknown	Devices that do not report detailed information on the status of their updates will report Unknown. This is most likely devices that do not use Windows Update for deployment.
Update paused	These devices have Windows Update for Business pause enabled, preventing this update from being installed.
Failed	A device is unable to install an update. This failure could be linked to a serious error in the update installation process or, in some cases, a compatibility hold .

Detailed deployment status

The final blade is the **Detailed Deployment Status** blade. This blade breaks down the detailed stage of deployment a device is in, beyond the generalized terms defined in Deployment Status. The following are the possible stages a device can report:

STATE	DESCRIPTION
Update deferred	When a device's Windows Update for Business policy dictates the update is deferred.
Update paused	The device's Windows Update for Business policy dictates the update is paused from being offered.
Update offered	The device has been offered the update, but has not begun downloading it.
Pre-Download tasks passed	The device has finished all necessary tasks prior to downloading the update.
Compatibility hold	The device has been placed under a <i>compatibility hold</i> to ensure a smooth feature update experience and will not resume the update until the hold has been cleared. For more information see Feature Update Status report
Download Started	The update has begun downloading on the device.
Download Succeeded	The update has successfully completed downloading.
Pre-Install Tasks Passed	Tasks that must be completed prior to installing the update have been completed.

STATE	DESCRIPTION
Install Started	Installation of the update has begun.
Reboot Required	The device has finished installing the update, and a reboot is required before the update can be completed.
Reboot Pending	The device has a scheduled reboot to apply the update.
Reboot Initiated	The scheduled reboot has been initiated.
Update Completed/Commit	The update has successfully installed.

NOTE

Interacting with any rows in the perspective view will automatically apply the given value to the query and execute it with the new parameter, narrowing the perspective to devices that satisfy that criteria. For example, clicking "Not configured (-1)" devices in Deferral Configurations will filter the query to only contain devices that do not have a deferral configuration. These filters can also be applied to queries via the filter sidebar.

Monitor the health of devices with Device Health

5/31/2019 • 2 minutes to read • [Edit Online](#)

Introduction

Device Health is the newest Windows Analytics solution that complements the existing Upgrade Readiness and Update Compliance solutions by providing IT with reports on some common problems the end users might experience so they can be proactively remediated, thus saving support calls and improving end-user productivity.

Like Upgrade Readiness and Update Compliance, Device Health is a solution built in Azure Portal, a cloud-based monitoring and automation service that has a flexible servicing subscription based on data usage and retention. This release is free for customers to try and will not incur charges on your Azure Portal workspace for its use. For more information about Azure Portal, see [Windows Analytics in the Azure Portal](#).

Device Health uses Windows diagnostic data that is part of all Windows 10 devices. If you have already employed Upgrade Readiness or Update Compliance solutions, all you need to do is select Device Health from the Azure Portal solution gallery and add it to your Azure Portal workspace. Device Health requires enhanced diagnostic data, so you might need to implement this policy if you've not already done so.

Device Health provides the following:

- Identification of devices that crash frequently, and therefore might need to be rebuilt or replaced
- Identification of device drivers that are causing device crashes, with suggestions of alternative versions of those drivers that might reduce the number of crashes
- Notification of Windows Information Protection misconfigurations that send prompts to end users
- No need for new complex customized infrastructure, thanks to cloud-connected access using Windows 10 diagnostic data

See the following topics in this guide for detailed information about configuring and using the Device Health solution:

- [Get started with Device Health](#): How to add Device Health to your environment.
- [Using Device Health](#): How to begin using Device Health.

An overview of the processes used by the Device Health solution is provided below.

Device Health licensing

Use of Windows Analytics Device Health requires one of the following licenses:

- Windows 10 Enterprise or Windows 10 Education per-device with active Software Assurance
- Windows 10 Enterprise E3 or E5 per-device or per-user subscription (including Microsoft 365 F1, E3, or E5)
- Windows 10 Education A3 or A5 (including Microsoft 365 Education A3 or A5)
- Windows VDA E3 or E5 per-device or per-user subscription

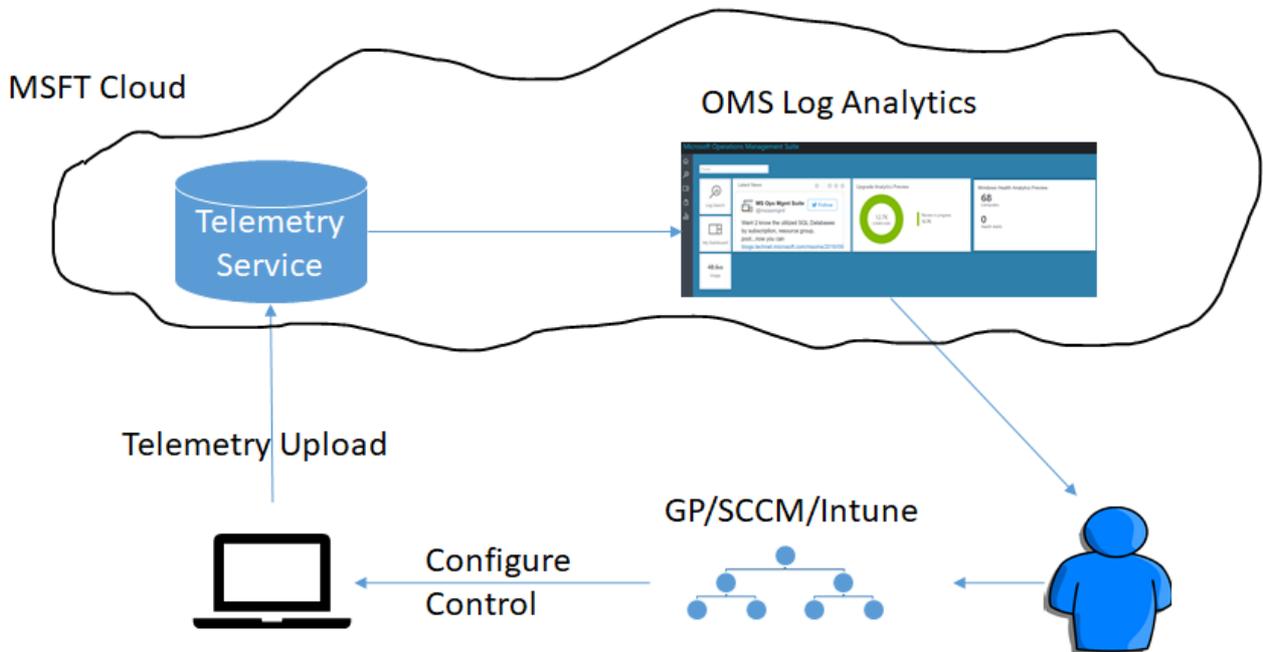
You don't have to install Windows 10 Enterprise on a per-device basis--you just need enough of the above licenses for the number of devices using Device Health.

Device Health architecture

The Device Health architecture and data flow is summarized by the following five-step process:

- (1) User computers send diagnostic data to a secure Microsoft data center using the Microsoft Data Management Service.
- (2) Diagnostic data is analyzed by the Microsoft Telemetry Service.
- (3) Diagnostic data is pushed from the Microsoft Telemetry Service to your Azure Portal workspace.
- (4) Diagnostic data is available in the Device Health solution.
- (5) You are now able to proactively monitor Device Health issues in your environment.

These steps are illustrated in following diagram:



NOTE

This process assumes that Windows diagnostic data is enabled and data sharing is enabled as described in [Enrolling devices in Windows Analytics](#).

Related topics

[Get started with Device Health](#)

[Use Device Health to monitor frequency and causes of device crashes](#)

For the latest information on Windows Analytics, including new features and usage tips, see the [Windows Analytics blog](#)

Get started with Device Health

6/10/2019 • 3 minutes to read • [Edit Online](#)

This topic explains the steps necessary to configure your environment for Windows Analytics Device Health.

- [Get started with Device Health](#)
 - [Add the Device Health solution to your Azure subscription](#)
 - [Enroll devices in Windows Analytics](#)
 - [Use Device Health to monitor device crashes, app crashes, sign-in failures, and more](#)
 - [Related topics](#)

Add the Device Health solution to your Azure subscription

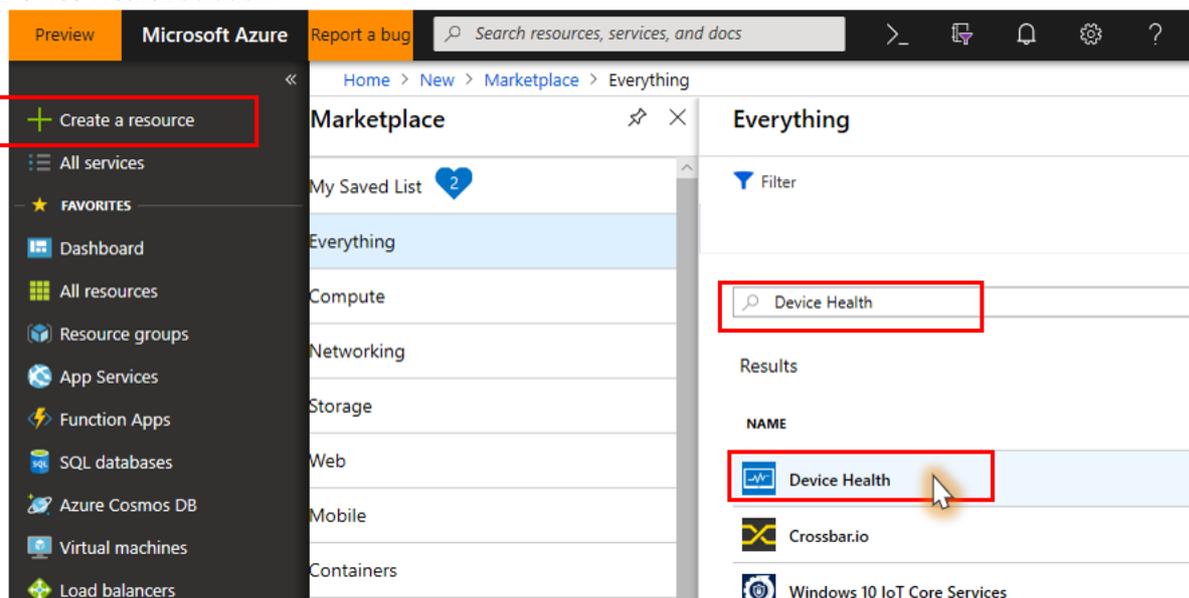
Device Health is offered as a *solution* which you link to a new or existing [Azure Monitor workspace](#) within your Azure *subscription*. To configure this, follows these steps:

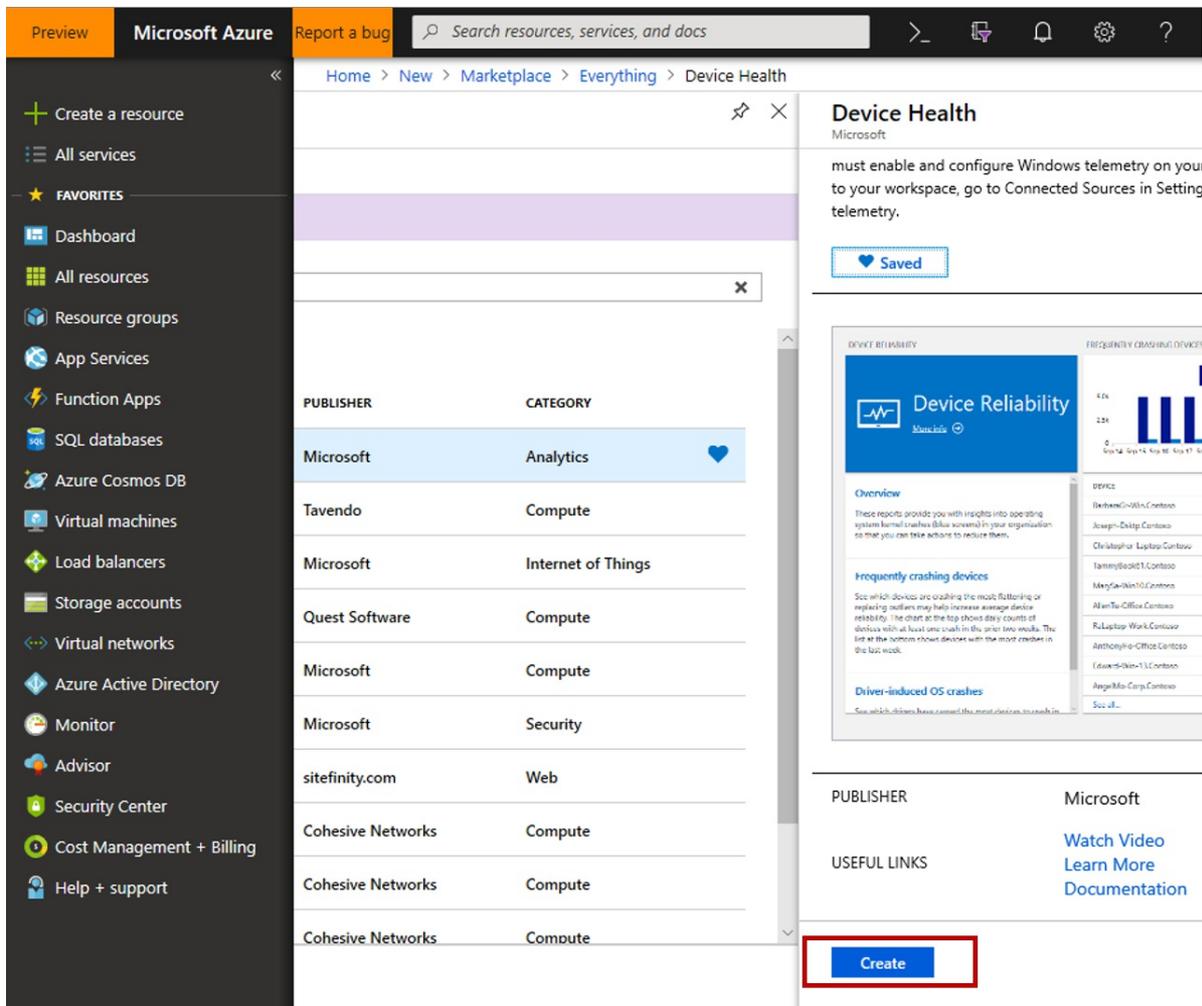
1. Sign in to the [Azure Portal](#) with your work or school account or a Microsoft account. If you don't already have an Azure subscription you can create one (including free trial options) through the portal.

NOTE

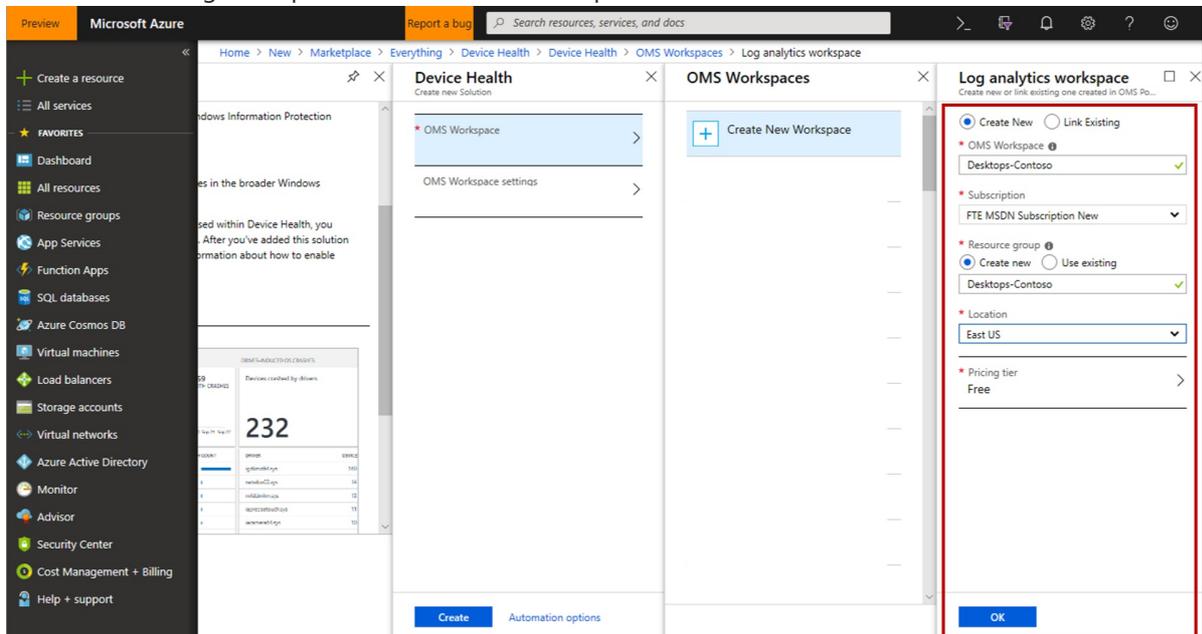
Device Health is included at no additional cost with Windows 10 [education and enterprise licensing](#). An Azure subscription is required for managing and using Device Health, but no Azure charges are expected to accrue to the subscription as a result of using Device Health.

2. In the Azure portal select **Create a resource**, search for "Device Health", and then select **Create** on the **Device Health** solution.





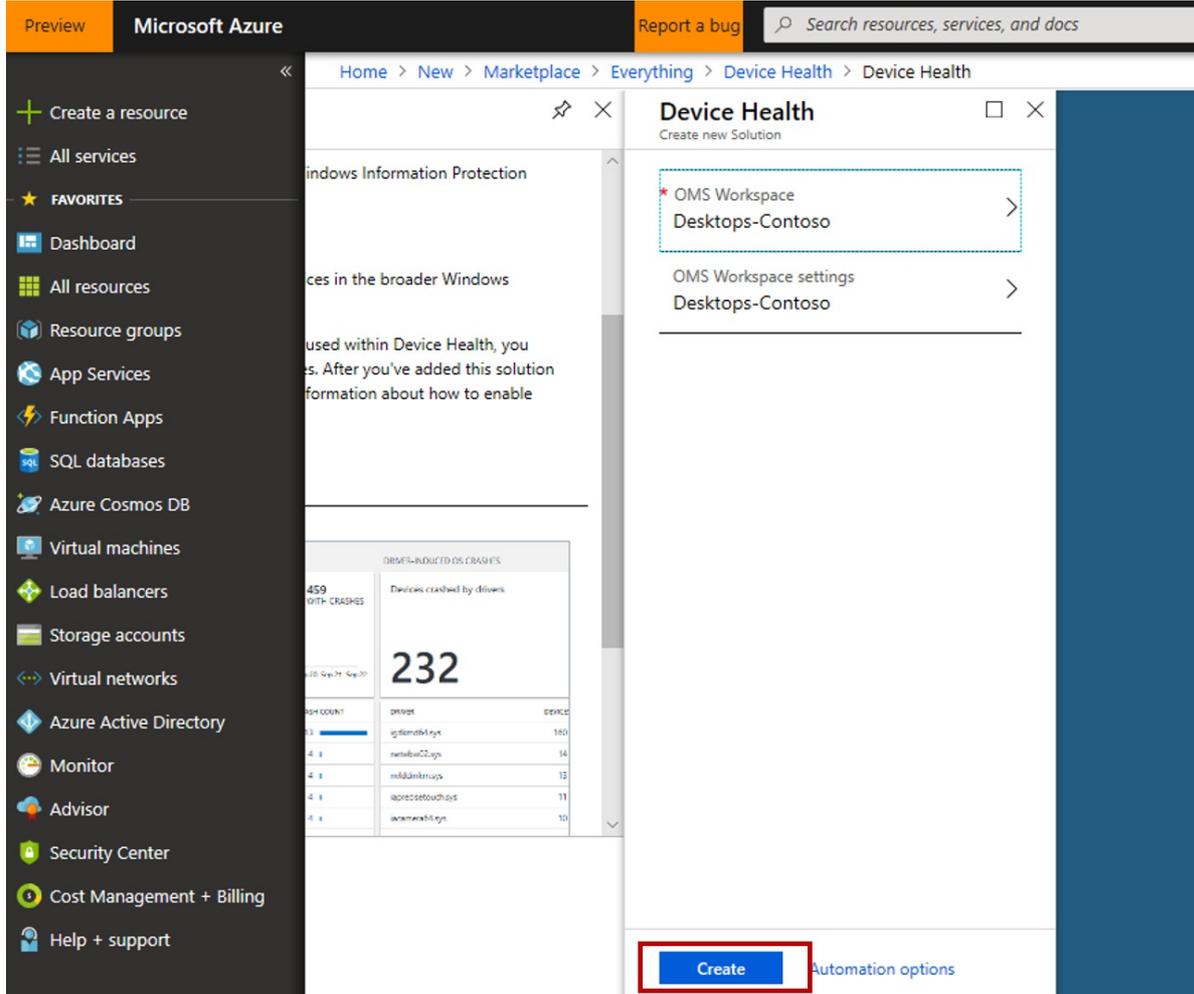
3. Choose an existing workspace or create a new workspace to host the Device Health solution.



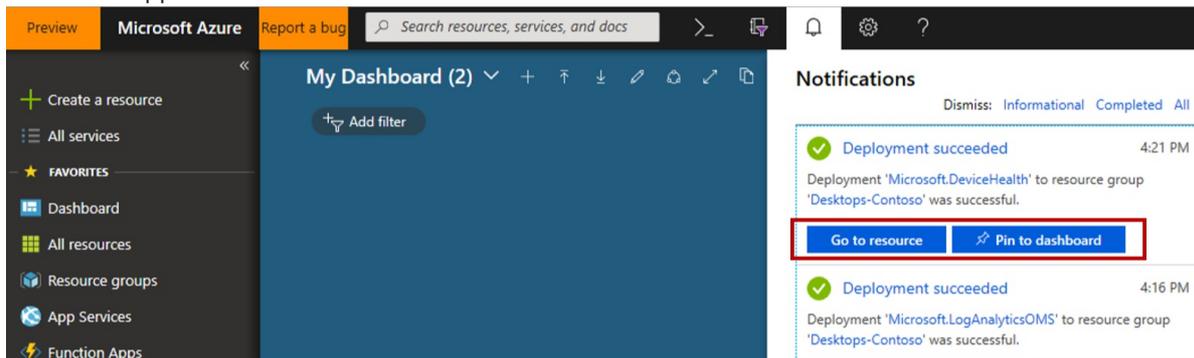
- If you are using other Windows Analytics solutions (Upgrade Readiness or Update Compliance) you should add Device Health to the same workspace.
- If you are creating a new workspace, and your organization does not have policies governing naming conventions and structure, consider the following workspace settings to get started:
 - Choose a workspace name which reflects the scope of planned usage in your organization, for example *PC-Analytics*.
 - For the resource group setting select **Create new** and use the same name you chose for your new workspace.

- For the location setting, choose the Azure region where you would prefer the data to be stored.
- For the pricing tier select **per GB**.

4. Now that you have selected a workspace, you can go back to the Device Health blade and select **Create**.



5. Watch for a Notification (in the Azure portal) that "Deployment 'Microsoft.DeviceHealth' to resource group 'YourResourceGroupName' was successful." and then select **Go to resource**. This might take several minutes to appear.



- Suggestion: Choose the **Pin to Dashboard** option to make it easy to navigate to your newly added Device Health solution.
- Suggestion: If a "resource unavailable" error occurs when navigating to the solution, try again after one hour.

Enroll devices in Windows Analytics

Once you've added Device Health to a workspace in your Azure subscription, you can start enrolling the devices in your organization. For Device Health there are two key steps for enrollment:

1. Deploy your CommercialID (from Device Health Settings page) to your Windows 10 devices (typically using

Group Policy or similar)

2. Ensure the Windows Diagnostic Data setting on devices is set to Enhanced or Full (typically using Group Policy or similar). Note that the [Limit Enhanced](#) policy can substantially reduce the amount of diagnostic data shared with Microsoft while still allowing Device Health to function. For full enrollment instructions and troubleshooting, see [Enrolling devices in Windows Analytics](#).

After enrolling your devices (by deploying your CommercialID and Windows Diagnostic Data settings), it may take 48-72 hours for the first data to appear in the solution. Until then, the Device Health tile will show "Performing Assessment."

Use Device Health to monitor device crashes, app crashes, sign-in failures, and more

Once your devices are enrolled and data is flowing, you can move on to [Using Device Health](#).

NOTE

You can remove the Device Health solution from your workspace if you no longer want to monitor your organization's devices. Windows diagnostic data will continue to be shared with Microsoft as normal as per the diagnostic data sharing settings on the devices.

Related topics

[Use Device Health to monitor frequency and causes of device crashes](#)

For the latest information on Windows Analytics, including new features and usage tips, see the [Windows Analytics blog](#)

Using Device Health

6/26/2019 • 17 minutes to read • [Edit Online](#)

This section describes how to use Device Health to monitor devices deployed on your network and troubleshoot the causes if they crash.

Device Health provides IT Pros with reports on some common problems that users might experience so that they can be proactively remediated. This decreases support calls and improves productivity.

Device Health provides the following benefits:

- Identification of devices that crash frequently and therefore might need to be rebuilt or replaced
- Identification of device drivers that are causing device crashes, with suggestions of alternative versions of those drivers that might reduce the number of crashes
- Notification of Windows Information Protection misconfigurations that send prompts to end users

NOTE

Information is refreshed daily so that health status can be monitored. Changes will be displayed about 24-48 hours after their occurrence, so you always have a recent snapshot of your devices.

In Azure Portal, the aspects of a solution's dashboard are usually divided into *blades*. Blades are a slice of information, typically with a summarization tile and an enumeration of the items that makes up that data. All data is presented through *queries*. *Perspectives* are also possible, wherein a given query has a unique view designed to display custom data. The terminology of blades, tiles, and perspectives will be used in the sections that follow.

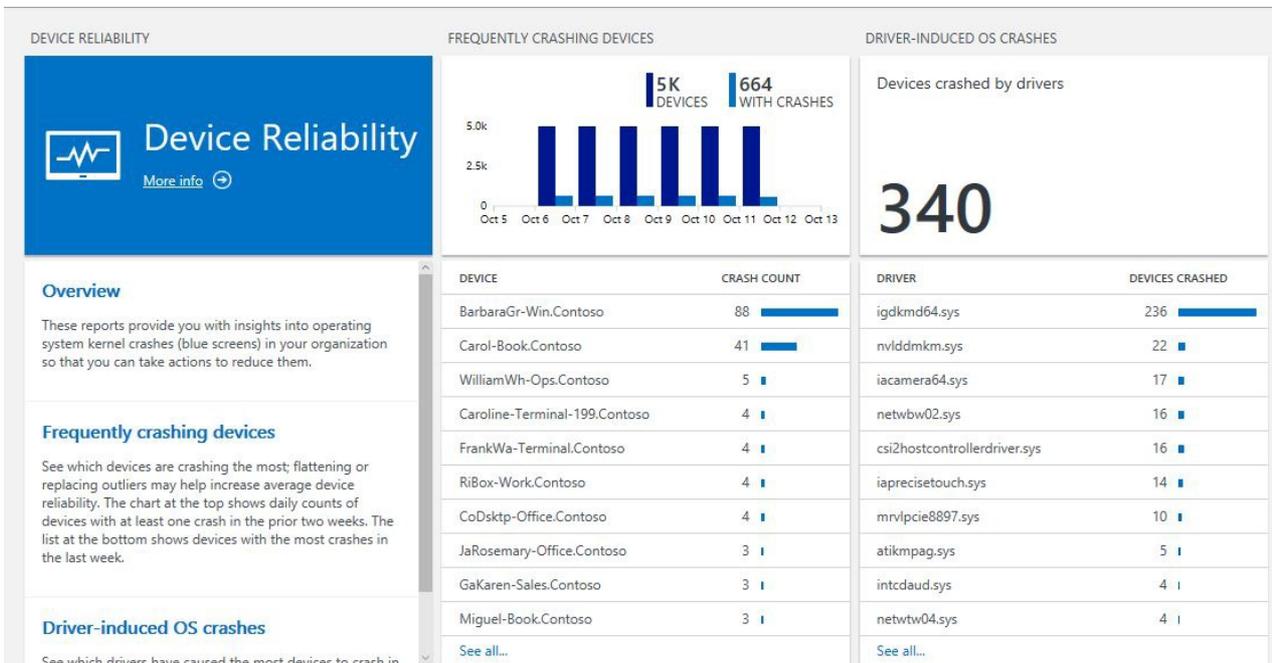
Device Reliability

- [Frequently crashing devices](#)
- [Driver-induced OS crashes](#)

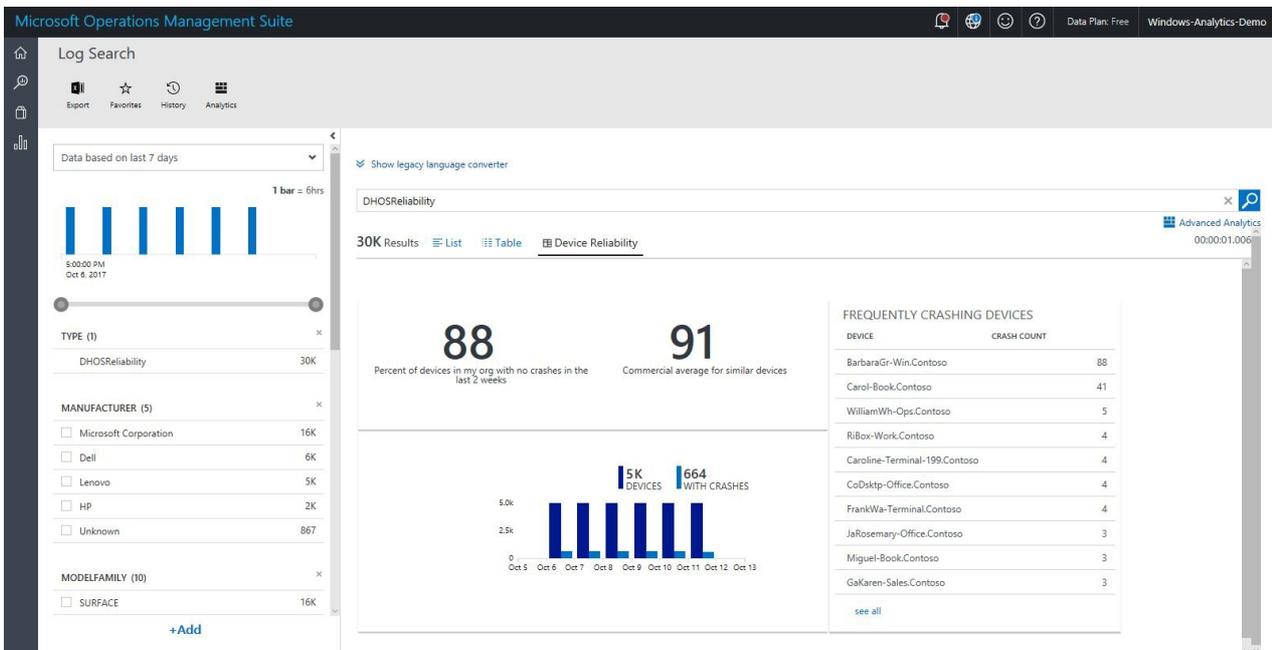
Frequently Crashing Devices

This middle blade in Device Reliability displays the devices that have crashed the most often in the last week. This can help you identify unhealthy devices that might need to be rebuilt or replaced.

See the following example:



Clicking the header of the Frequently Crashing Devices blade opens a reliability perspective view, where you can filter data (by using filters in the left pane), see trends, and compare to commercial averages:



"Commercial averages" here refers to data collected from deployments with a mix of operating system versions and device models that is similar to yours. If your crash rate is higher, there are opportunities for improvement, for example by moving to newer driver versions.

Notice the filters in the left pane; they allow you to filter the crash rate shown to a particular operating system version, device model, or other parameter.

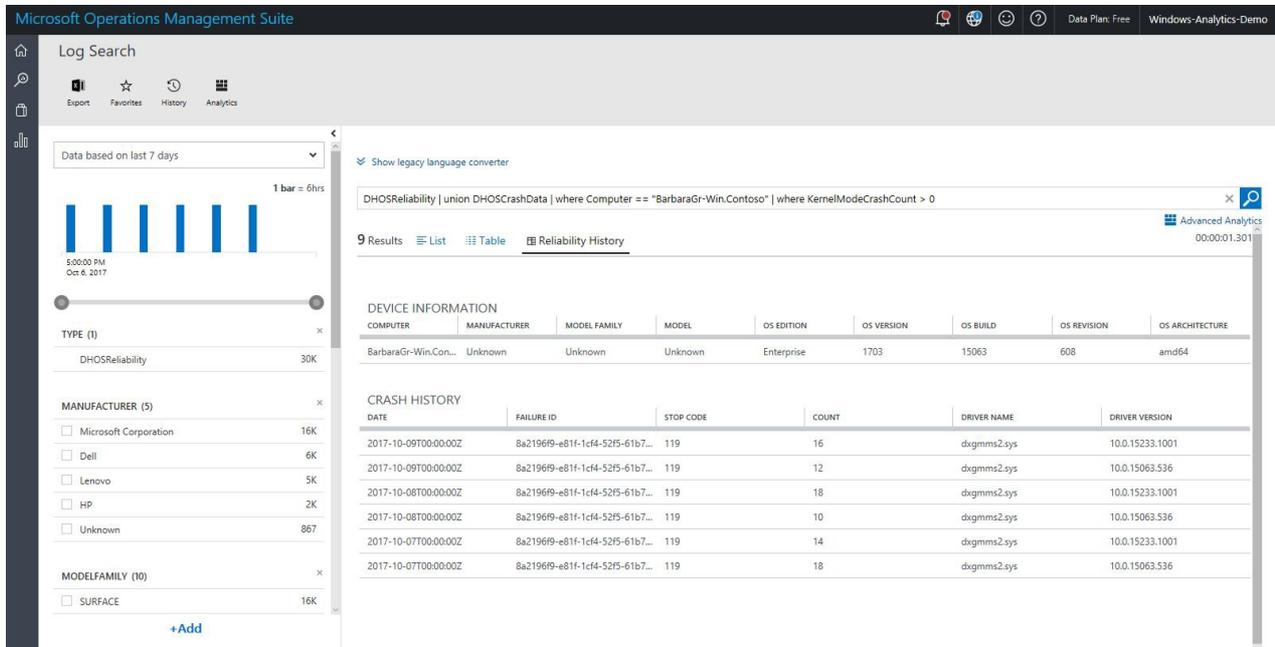
NOTE

Use caution when interpreting results filtered by model or operating system version. This is very useful for troubleshooting, but might not be accurate for *comparisons* because the crashes displayed could be of different types. The overall goal for working with crash data is to ensure that most devices have the same driver versions and that the version has a low crash rate.

TIP

Once you've applied a filter (for example setting OSVERSION=1607) you will see the query in the text box change to append the filter (for example, with "(OSVERSION=1607)"). To undo the filter, remove that part of the query in the text box and click the search button to the right of the text box to run the adjusted query."

If you click through a particular device from the view blade or from the Device Reliability perspective, it will take you to the Crash History perspective for that device.

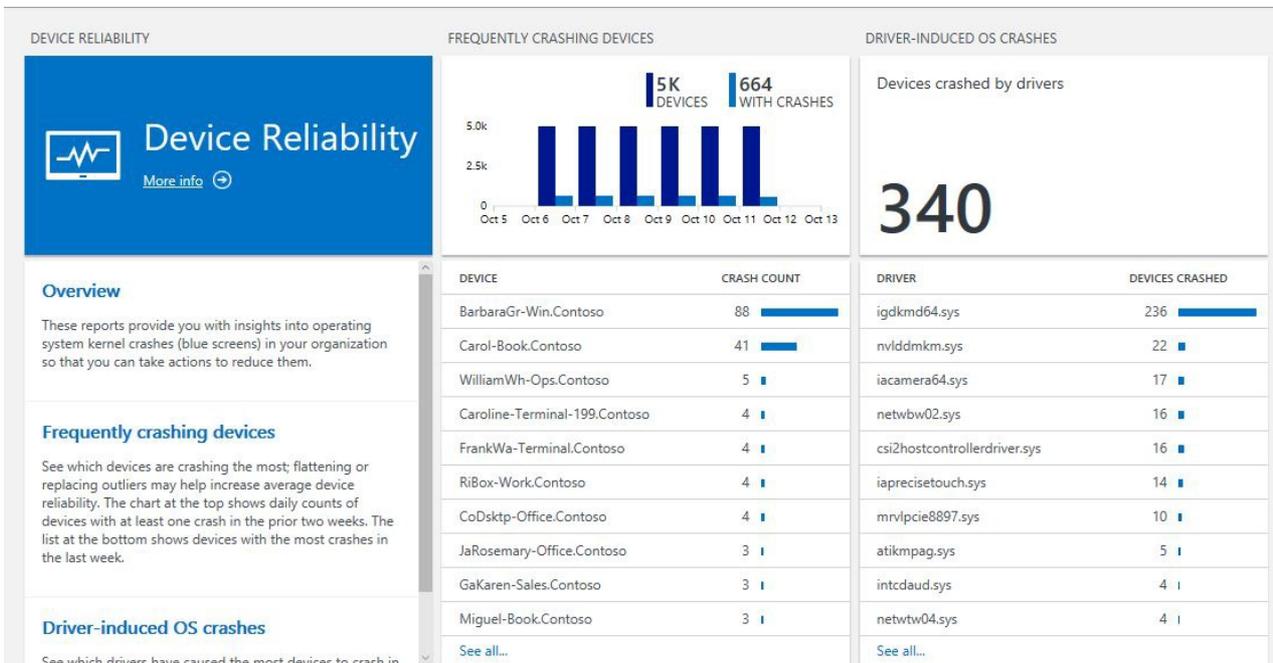


This displays device records sorted by date and crash details by failure ID, also sorted by date. In this view are a number of useful items:

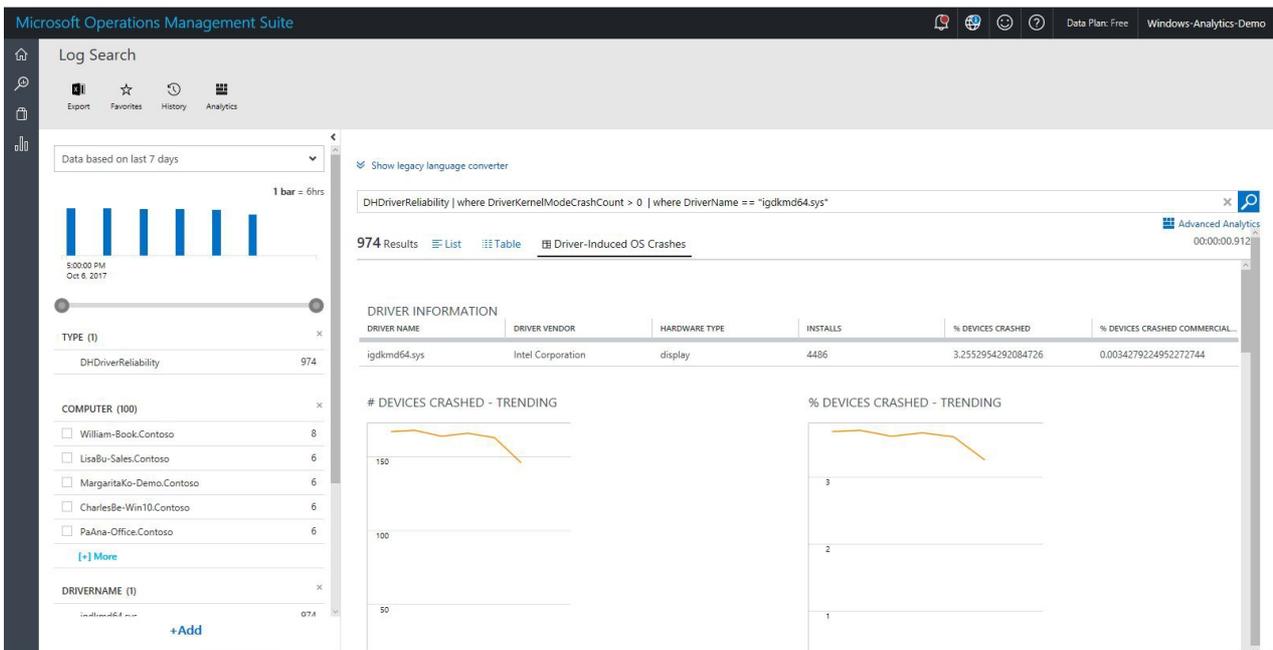
- Crash history records by date, aggregated by Failure ID. The Failure ID is an internal number that is used to group crashes that are related to each other. Eventually over time, you can use the Failure ID to provide additional info. If a crash was caused by driver, some driver fields will also be populated.
- StopCode: this is hex value that would be displayed on a bluescreen if you were looking directly at the affected device.
- Count: the number times that particular Failure ID has occurred on that specific device *on that date*.

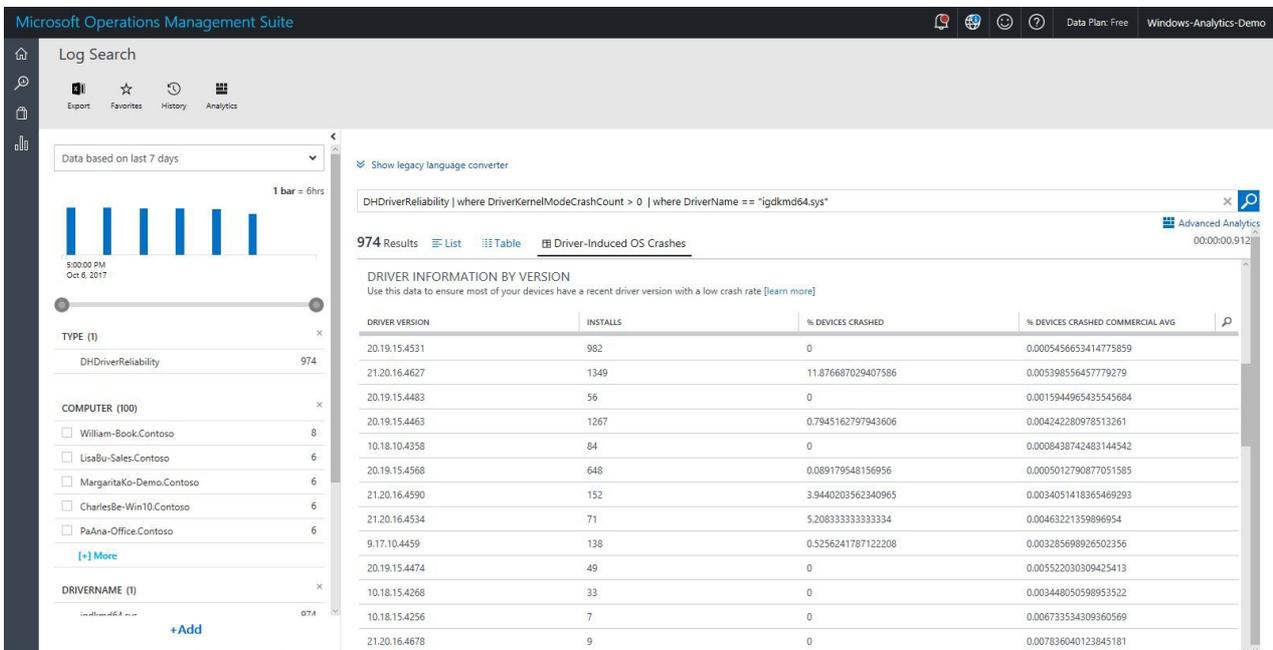
Driver-induced crashes

This blade (on the right) displays drivers that have caused the most devices to crash in the last two weeks. If your crash rate is high, you can reduce the overall operating system crashes in your deployment by upgrading those drivers with a high crash rate.



Clicking a listed driver on the Driver-Induced OS Crashes blade opens a driver perspective view, which shows the details for the responsible driver, trends and commercial averages for that driver, and alternative versions of the driver.





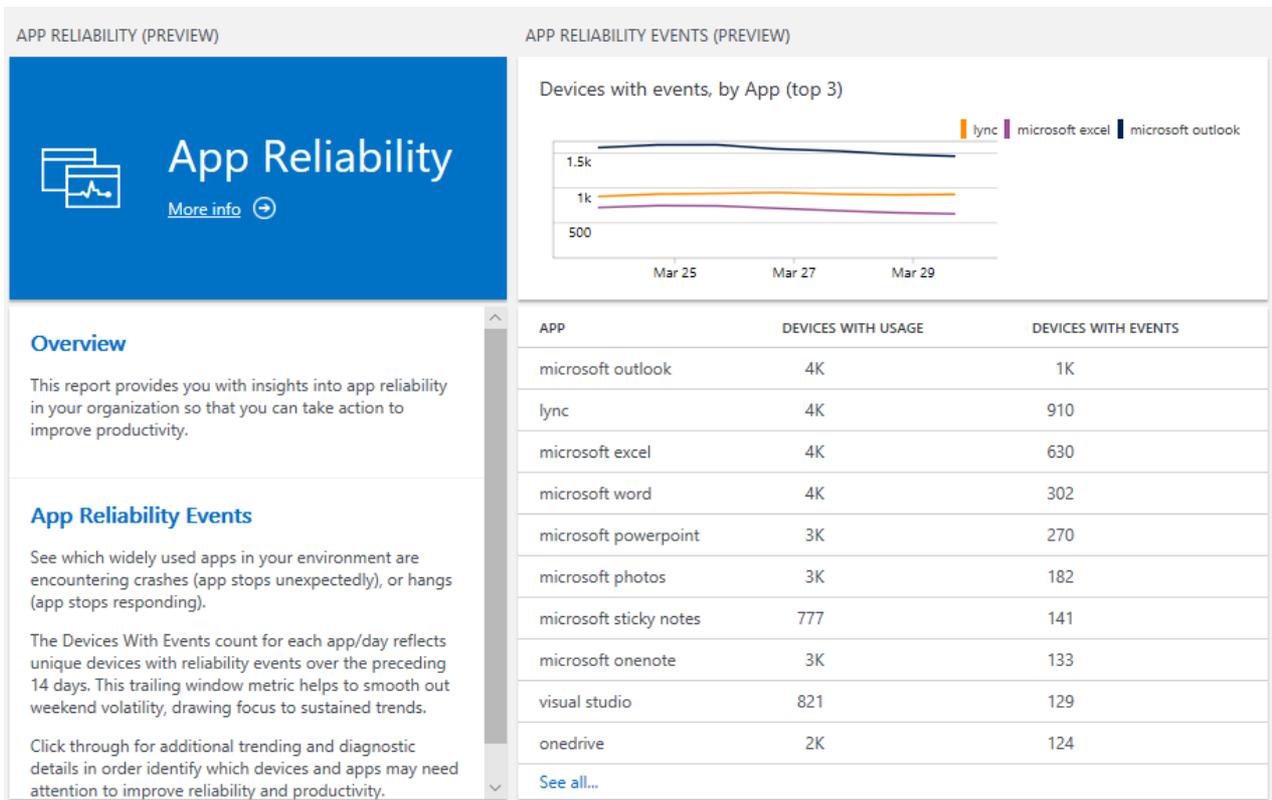
The driver version table can help you determine whether deploying a newer version of the driver might help you reduce the crash rate. In the example shown above, the most commonly installed driver version (19.15.1.5) has a crash rate of about one-half of one percent--this is low, so this driver is probably fine. However, driver version 19.40.0.3 has a crash rate of almost 20%. If that driver had been widely deployed, updating it would substantially reduce the overall number of crashes in your organization.

App Reliability

The App Reliability report shows you useful data on app usage and behavior so that you can identify apps that are misbehaving and then take steps to resolve the problem.

App reliability events

The default view includes the **Devices with events** count, which shows the number of devices in your organization that have logged a reliability event for a given app over the last 14 days. A "reliability event" occurs when an app either exits unexpectedly or stops responding. The table also includes a **Devices with Usage** count. This enables you to see how widely used the app was over the same period to put the Devices with Events count into perspective.



When you click a particular app, the detailed **App reliability** view opens. The first element in the view is the App Information summary:

APP	PUBLISHER	DEVICES WITH USAGE	DEVICES WITH EVENTS	% WITH EVENTS	% WITH EVENTS (COMMERCIAL...
Mail & Calendar	Microsoft Corporation	1873	152	8.1	0.9

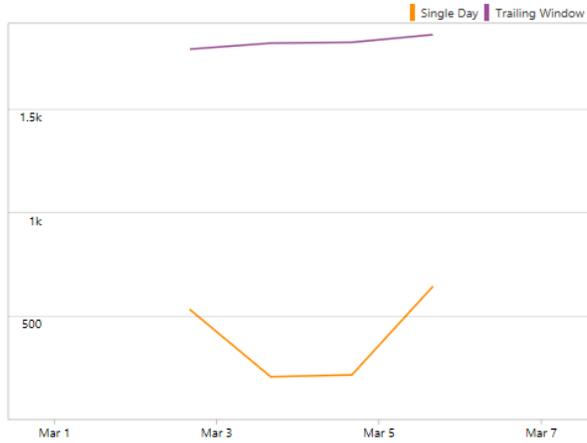
This table contains:

- App name
- Publisher
- Devices with usage: the number of unique devices that logged any usage of the app
- Devices with events: the number of unique devices that logged any reliability event for the app
- % with events: the ratio of "devices with events" to "devices with usage"
- % with events (commercial average): the ratio of "devices with events" to "devices with usage" in data collected from deployments with a mix of operating system versions and device models that is similar to yours. This can help you decide if a given app is having problems specifically in your environment or more generally in many environments.

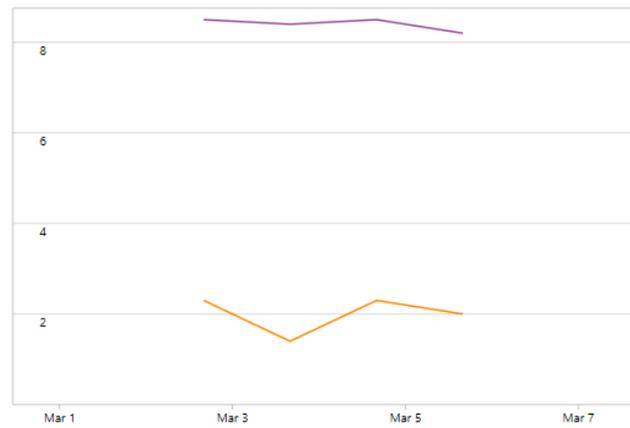
Trend section

Following the App Information summary is the trend section:

DEVICES WITH EVENTS



% USED DEVICES WITH EVENTS



With these trend graphs you can more easily detect if an issue is growing, shrinking, or steady. The trend graph on the left shows the number of devices that logged any reliability event for the app. The trend graph on the right shows the ratio of "devices with events" to "devices with usage."

Each graph displays two lines:

- Trailing window: in this line, each day's value reflects reliability events that occurred in the 14 days leading up to that day. This is useful for gauging the long-term trend with reduced volatility due to weekends and small populations.
- Single day: Each day's value reflects reliability events that occurred in a single day. This is useful if an issue is quickly emerging (or being resolved).

App and OS versions table

The next element in the view is the App and OS versions table:

APP VERSION	OS VERSION	DEVICES WITH USAGE	DEVICES WITH EVENTS	% WITH EVENTS	% WITH EVENTS (COMMER...
17.8827.21855.0	1709	1590	129	8.1	1.2
17.9029.21675.0	1709	757	17	2.2	0.4
17.8827.21855.0	1703	91	7	7.7	1.2
17.9029.21675.0	1703	38	1	2.6	0.4
17.9029.21735.0	1709	18	0	0	NaN

This table breaks out the metrics by combinations of App and OS version. This enables you to identify patterns in that might indicate devices needing an update or configuration change.

For example, if the table shows that a later version of an app is more reliable than an earlier version in your environment, then prioritizing deployment of the later version is likely the best path forward. If you are already running the latest version of the app, but reliability events are increasing, then you might need to do some troubleshooting, or seek support from Microsoft or the app vendor.

By default the table is limited to the most-used version combinations in your environment. To see all version combinations click anywhere in the table.

Reliability event history table

The next element in the view is the reliability event history table:

DATE	COMPUTER	PROCESS NAME	APP VERSION	OS VERSION	DIAGNOSTIC SIGNATURE	
2018-03-06T08:00:00Z	JudyHo-Box.Contoso	microsoft.windowscommuni...	17.9029.21675.0	1709	NULL_POINTER_WRITE_c000...	
2018-03-06T08:00:00Z	PatriciaHe-Work.Contoso	microsoft.windowscommuni...	17.9029.21675.0	1709	INVALID_POINTER_READ_c0...	
2018-03-06T08:00:00Z	JovitaMc-Box.Contoso	microsoft.windowscommuni...	17.9029.21675.0	1709	APPLICATION_FAULT_59379...	
2018-03-06T08:00:00Z	OlgaPa-Office.Contoso	microsoft.windowscommuni...	17.9029.21675.0	1709	NULL_POINTER_WRITE_c000...	
2018-03-06T08:00:00Z	FIDon-Work.Contoso	microsoft.windowscommuni...	17.9029.21675.0	1709	NULL_POINTER_WRITE_c000...	

This table shows the most detailed information. Although Device Health is not a debugging tool, the details available in this table can help with troubleshooting by providing the specific devices, versions, and dates of the reliability events.

This view also includes the **Diagnostic Signature** column. This value can be helpful when you are working with product support or troubleshooting on your own. The value (also known as Failure ID or Failure Name) is the same identifier used to summarize crash statistics for Microsoft and partner developers.

The Diagnostic Signature value contains the type of reliability event, error code, DLL name, and function name involved. You can use this information to narrow the scope of troubleshooting. For example, a value like `APPLICATION_HANG_ThreadHang_Contoso-Add-In.dll!GetRegistryValue()` implies that the app stopped responding when Contoso-Add-In was trying to read a registry value. In this case you might prioritize updating or disabling the add-in, or using Process Monitor to identify the registry value it was trying to read, which could lead to a resolution through antivirus exclusions, fixing missing keys, or similar remedies.

By default the table is limited to a few recent rows. To see all rows click anywhere in the table.

FAQs and limitations

Why does a particular app not appear in the views?

When we allow reliability events from all processes, the list of apps fills with noisy processes which don't feel like meaningful end-user apps (for example, `taskhost.exe` or `odd-test-thing.exe`). In order to draw focus to the apps which matter most to users, App Reliability uses a series of filters to limit what appears in the list. The filter criteria include the following:

- Filter out background processes which have no detected user interaction.
- Filter out operating system processes which, despite having user interaction, do not feel like apps (for example, `Lgonui.exe`, `Winlogon.exe`). **Known limitation:** Some processes which may feel like apps are not currently detected as such (and are therefore filtered out as OS processes). These include `Explorer.exe`, `l explore.exe`, `Microsoftedge.exe`, and several others.
- Remove apps which are not widely used in your environment. **Known limitation:** This might result in an app that you consider important being filtered out when that app is not among the 30 most widely used in your environment.

We welcome your suggestions and feedback on this filtering process at the [Device Health Tech Community](#).

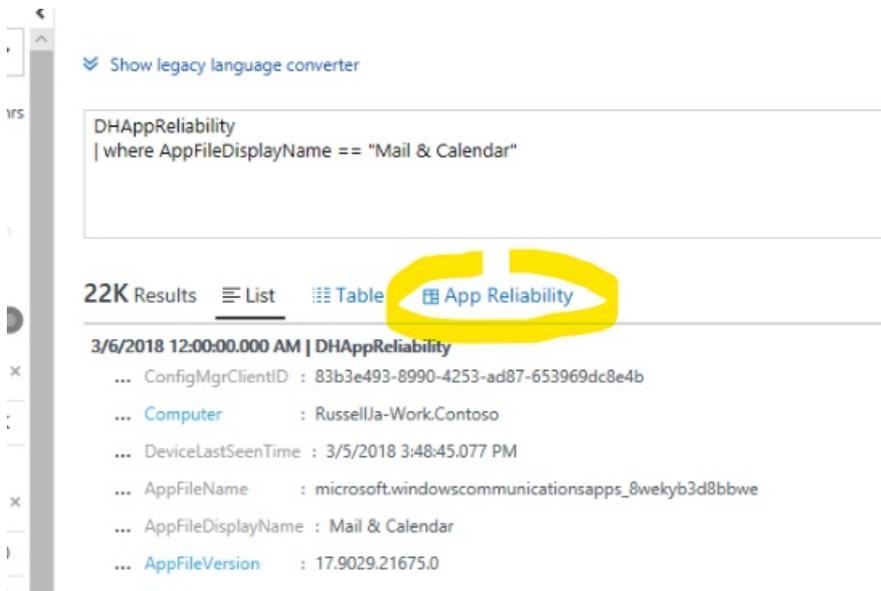
Why are there multiple names and entries for the same app?

For example, you might see *Skype for Business*, *skype for business*, and *Lync* listed separately, but you only use *Skype for Business*. Or you might see *MyApp Pro* and *MyApp Professional* listed separately, even though they feel like the same thing.

Apps have many elements of metadata which describe them. These include an Add/Remove programs title ("Contoso Suite 12"), executable file names ("ContosoCRM.exe"), executable display name ("Contoso CRM"), and others. App publishers (and in some cases app re-packagers) set these values. For the most part we leave the data as set by the publisher which can lead to some report splitting. In certain cases we apply transformations to reduce splitting, for example we (by design) convert many values to lower case so that incoming data such as "Contoso CRM" and "CONTOSO CRM" become the same app name for reporting.

Clicking an app in the App Reliability Events blade sometimes results a List view of records instead of the App Reliability view

To work around this, click the **App Reliability** tab above the results to see the expected view.



Clicking "See all..." from the App Reliability Events blade followed by clicking an app from the expanded list results in raw records instead of the App Reliability view

To work around this, replace all of the text in the Log Search query box with the following:

DHAppReliability | where AppFileDisplayName == "<Name of app as it appeared in the list>"

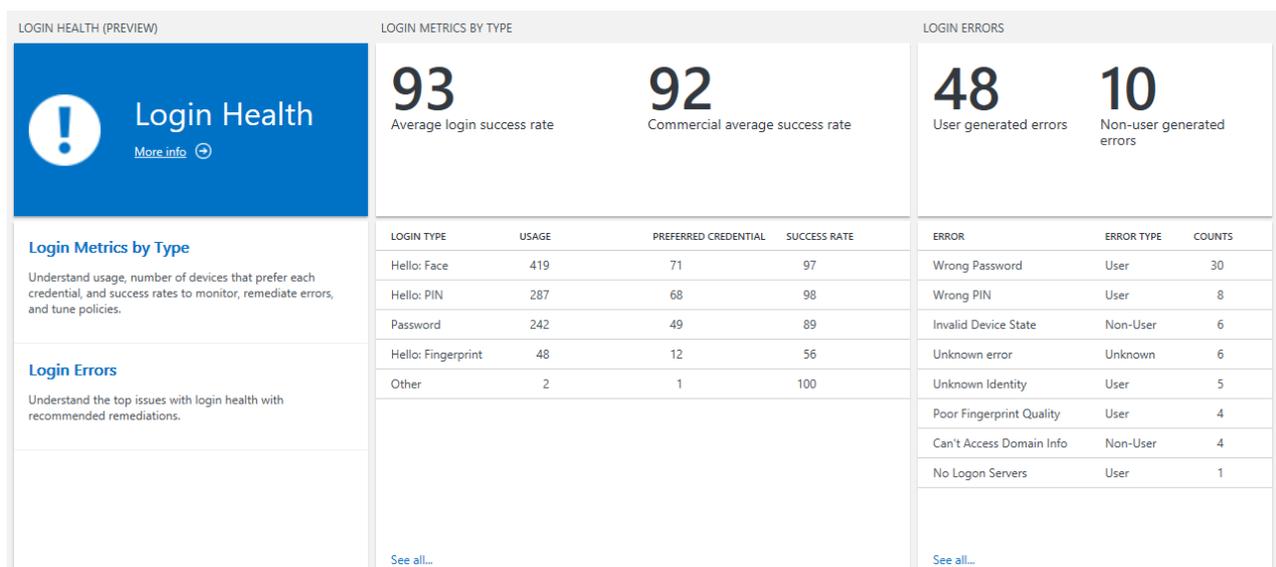
For example:

DHAppReliability | where AppFileDisplayName == "Microsoft Outlook"

Login Health

Login Health provides reports on Windows login attempts in your environment, including metrics on the login methods being used (such as Windows Hello, face recognition, fingerprint recognition, PIN, or password), the rates and patterns of login success and failure, and the specific reasons logins have failed.

The Login Health blades appear in the Device Health dashboard:



Login Errors

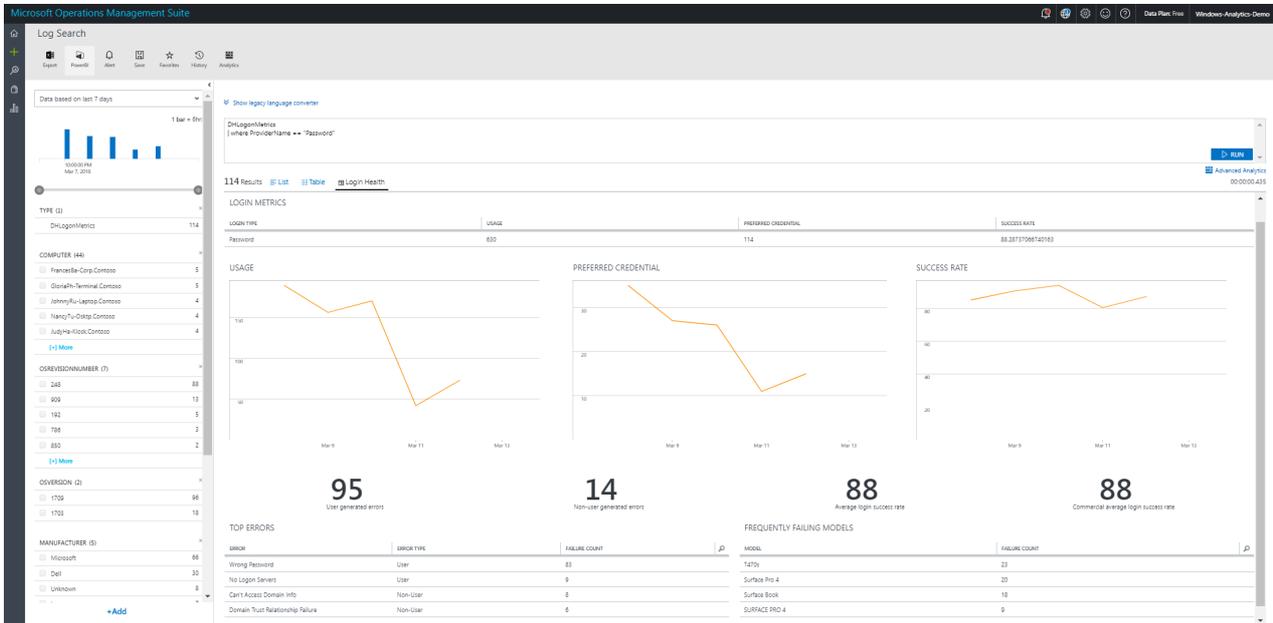
The **Login errors** blade displays data on the frequency and type of errors, with statistics on specific errors. They are generally categorized into user-generated (caused by bad input) or non-user-generated (might need IT intervention) errors. Click any individual error to see all instances of the error's occurrence for the specified time period.

Login Metrics by Type

The **Login metrics by type** blade shows the success rate for your devices, as well as the success rate for other environments with a mix of operating system versions and device models similar to yours (the **Commercial average success rate**).

In the table (by type) you can gauge how broadly each login type is attempted, the number of devices that prefer the type (most used), and the success rate. If migration from passwords to an alternative such as Hello: PIN is going well, you would see high usage and high success rates for the new type.

Click any of the login types to see detailed login health data for that type:



This view shows trends over time of usage, preferred credentials, and success rate along with the most frequent errors and frequently failing devices for that login type.

Click a specific login error in this view to see a list of all instances for that error and login type within the specified time range:

The screenshot shows a detailed view of login failures. The table below lists three instances of the 'Can't Access Domain Info' error.

TimeGenerated	Computer	Manufacturer	ModelFamily	Model	OSArchitecture	OSEdition	OSVersion	OSBuildNumber	OSRevisionNumber	ProviderName
3/8/2018 12:00:00.000 AM	TabathaHe-Corp.Contoso	Microsoft	Surface	Surface Pro 4	amd64	Enterprise	1709	16,299	125	Password
3/8/2018 12:00:00.000 AM	CarmenCr-Win.Contoso	Microsoft	Surface	Surface Pro 4	amd64	Enterprise	1709	16,299	248	Password
3/12/2018 1:00:00.000 AM	WalterTh-Sales.Contoso	Microsoft	Surface	Surface Pro 4	amd64	Enterprise	1709	16,299	248	Password

Included in this view are device attributes and error attributes such as the following:

- LogonStatus/LogonSubStatus: Status code for the login attempt
- SignInFailureReason: Known failure reasons evaluated from status or sub-status

- SuggestedSignInRemediation: Suggested remediation that was presented to the user at the time of error

The filters in the left pane allow you to filter errors to a particular operating system, device model, or other parameters. Alternatively, clicking the most frequently failing models from the Login Health perspective will take you to a list of error instances filtered to the login type and specified device model within the specified time range.

NOTE

Windows Hello: Face authentication errors are not currently included in the login health reports.

Windows Information Protection

Windows Information Protection (WIP) helps protect work data from accidental sharing. Users might be disrupted if WIP rules are not aligned with real work behavior. WIP App Learning shows which apps on which computers are attempting to cross policy boundaries.

For details about deploying WIP policies, see [Protect your enterprise data using Windows Information Protection \(WIP\)](#).

Once you have WIP policies in place, by using the WIP section of Device Health, you can:

- Reduce disruptive prompts by adding rules to allow data sharing from approved apps.
- Tune WIP rules, for example by confirming that certain apps are allowed or disallowed by current policy.

WINDOWS INFORMATION PROTECTION



WIP App Learning

[More info](#) ➔

Overview

Windows Information Protection (WIP) helps protect work data from accidental sharing.

Your IT department configures which apps are allowed to access work data, and controls the level of protection.

App Learning

Display details about all unconfigured apps on managed devices that try to access work data, and reduce disruptive prompts by adding rules to allow data sharing from approved apps.

APP LEARNING

94

Apps with WIP access events

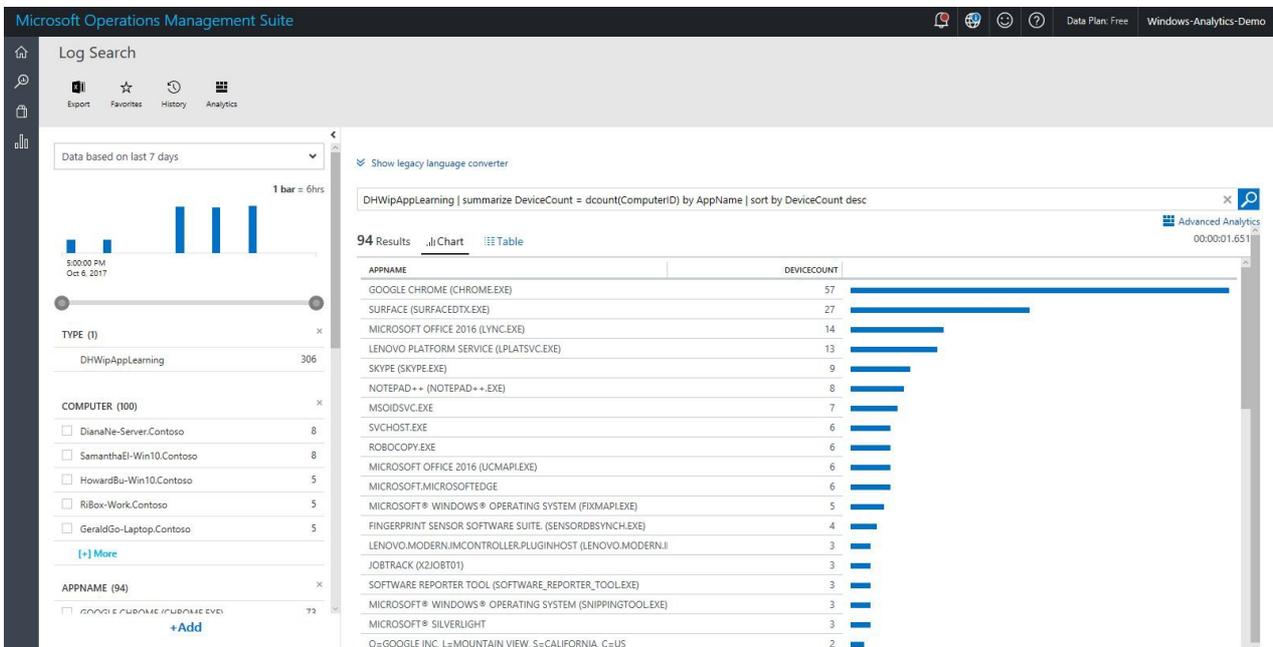
165

Devices reporting WIP access events

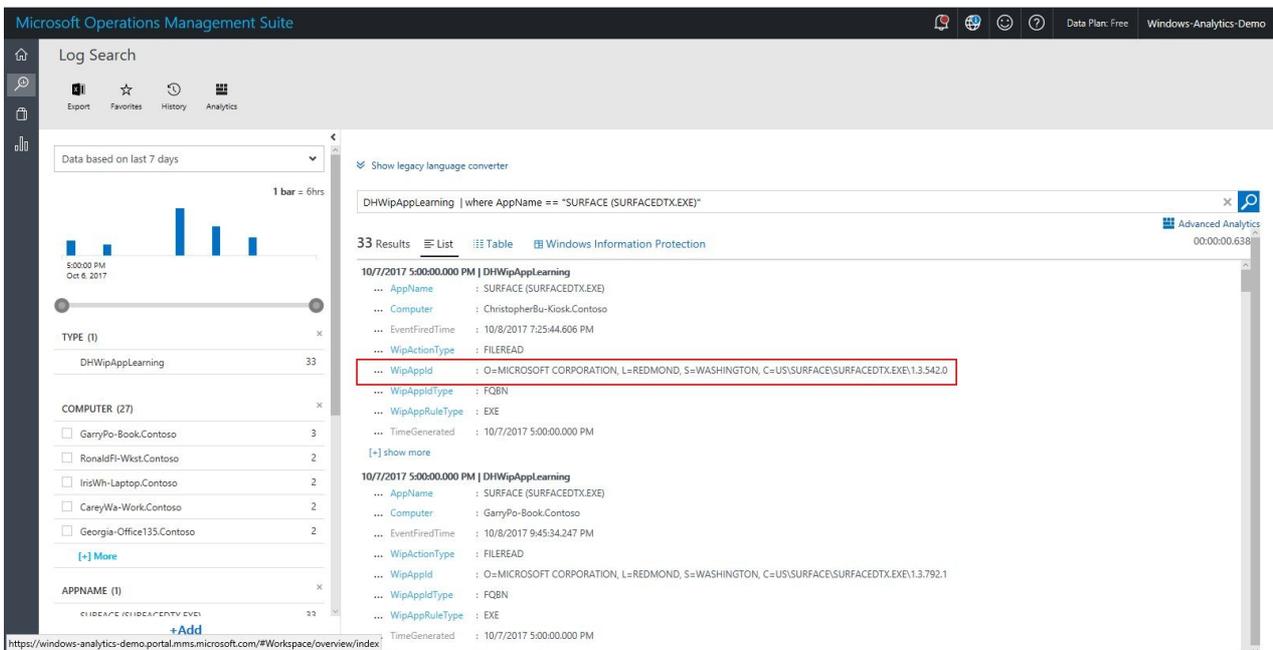
APP	DEVICES
GOOGLE CHROME (CHROME.E...	57 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
SURFACE (SURFACEDTX.EXE)	27 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
MICROSOFT OFFICE 2016 (LYN...	14 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
LENOVO PLATFORM SERVICE (...)	13 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
SKYPE (SKYPE.EXE)	9 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
NOTEPAD++ (NOTEPAD++.EXE)	8 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
MSOIDSVC.EXE	7 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
ROBOCOPY.EXE	6 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
MICROSOFT.MICROSOFTEDGE	6 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>
SVCHOST.EXE	6 <div style="width: 100%; height: 10px; background-color: #0070C0;"></div>

[See all...](#)

Clicking through the **APP LEARNING** tile shows details of app statistics that you can use to explore each incident and update app policies by using AppLocker or WIP AppIDs.



In this chart view, you can click a particular app listing, which will open additional details on the app in question, including details you need to adjust your Windows Information Protection Policy:



Here you can copy the WipAppid and use that for = adjusting the WIP policy.

Data model and built-in extensibility

All of the views and blades display slices of the most useful data by using pre-formed queries. You have access to the full set of data collected by Device Health, which means you can construct your own queries to expose any data that is of interest to you. For documentation on working with log searches, see [Find data using log searches](#). This topic section provides information about the data types being populated specifically by Device Health.

Example queries

You can run these queries from the Azure Portal **Log Search** interface (available at several points in the Device Health interface) by just typing them in. There are few details to be aware of:

- After running a query, make sure to set the date range (which appears upper left after running initial query) to "7 days" to ensure you get data back.
- If you see the search tutorial dialog appearing frequently, it's likely because you have read-only access to

the Azure Portal workspace. Ask a workspace administrator to grant you "contributor" permissions (which is required for the "completed tutorial" state to persist).

- If you use the search filters in the left pane, you might notice there is no control to undo a filter selection. To undo a selection, delete the (FilterName="FilterValue") element that is appended to the search query and then click the search button again. For example, after you run a base query of *Type = DHOSReliability KernelModeCrashCount > 0*, a number of filter options appear on the left. If you then filter on **Manufacturer** (for example, by setting *Manufacturer="Microsoft Corporation"* and then clicking **Apply**), the query will change to *Type = DHOSReliability KernelModeCrashCount > 0 (Manufacturer="Microsoft Corporation")*. Delete (*Manufacturer="Microsoft Corporation"*) and then click the **search** button again to re-run the query without that filter.

Device reliability query examples

DATA	QUERY
Total devices	Type = DHOSReliability measure countdistinct(ComputerID) by Type
Number of devices that have crashed in the last three weeks	Type = DHOSReliability KernelModeCrashCount > 0 measure countdistinct(ComputerID) by Type
Compare the percentage of your devices that have not crashed with the percentage of similar devices outside your organization ("similar" here means other commercial devices with the same mix of device models, operating system versions and update levels).	Type=DHOSReliability measure avg(map(KernelModeCrashCount, 1, 10000, 0, 1)) as MyOrgPercentCrashFreeDevices, avg(KernelModeCrashFreePercentForIndustry) as CommercialAvgPercentCrashFreeDevices by Type Display Table
As above, but sorted by device manufacturer	Type=DHOSReliability measure avg(map(KernelModeCrashCount, 1, 10000, 0, 1)) as MyOrgPercentCrashFreeDevices, avg(KernelModeCrashFreePercentForIndustry) as CommercialAvgPercentCrashFreeDevices, countdistinct(ComputerID) as NumberDevices by Manufacturer sort NumberDevices desc Display Table
As above, but sorted by model	Type=DHOSReliability measure avg(map(KernelModeCrashCount, 1, 10000, 0, 1)) as MyOrgPercentCrashFreeDevices, avg(KernelModeCrashFreePercentForIndustry) as CommercialAvgPercentCrashFreeDevices, countdistinct(ComputerID) as NumberDevices by ModelFamily sort NumberDevices desc Display Table
As above, but sorted by operating system version	Type=DHOSReliability measure avg(map(KernelModeCrashCount, 1, 10000, 0, 1)) as MyOrgPercentCrashFreeDevices, avg(KernelModeCrashFreePercentForIndustry) as CommercialAvgPercentCrashFreeDevices, countdistinct(ComputerID) as NumberDevices by OSVersion sort NumberDevices desc Display Table
Crash rate trending in my organization compared to the commercial average. Each interval shows percentage of devices that crashed at least once in the trailing two weeks	Type=DHOSReliability measure avg(map(KernelModeCrashCount, 1, 10000, 0, 1)) as MyOrgPercentCrashFreeDevices, avg(KernelModeCrashFreePercentForIndustry) as CommercialAvgPercentCrashFreeDevices by TimeGenerated Display LineChart

DATA	QUERY
Table of devices that have crashed the most in the last two weeks	Type = DHOSReliability KernelModeCrashCount > 0 Dedup ComputerID select Computer, KernelModeCrashCount sort TimeGenerated desc, KernelModeCrashCount desc Display Table
Detailed crash records, most recent first	Type = DHOSCrashData sort TimeGenerated desc, Computer asc display Table
Number of devices that crashed due to drivers	Type = DHDriverReliability DriverKernelModeCrashCount > 0 measure countdistinct(ComputerID) by Type
Table of drivers that have caused the most devices to crash	Type = DHDriverReliability DriverKernelModeCrashCount > 0 measure countdistinct(ComputerID) by DriverName Display Table
Trend of devices crashed by driver by day	* Type=DHOSCrashData DriverName!="ntkrnlmp.exe" DriverName IN {Type=DHOSCrashData measure count() by DriverName
Crashes for different versions of a given driver (replace netwtw04.sys with the driver you want from the previous list). This lets you get an idea of which <i>versions</i> of a given driver work best with your devices	Type = DHDriverReliability DriverName="netwtw04.sys" Dedup ComputerID sort TimeGenerated desc measure countdistinct(ComputerID) as InstallCount, sum(map(DriverKernelModeCrashCount,1,10000, 1)) as DevicesCrashed by DriverVersion Display Table
Top crashes by FailureID	Type =DHOSCrashData measure count() by KernelModeCrashFailureId Display Table

Windows Information Protection (WIP) App Learning query examples

DATA	QUERY
Apps encountering policy boundaries on the most computers (click on an app in the results to see details including computer names)	Type=DHWipAppLearning measure countdistinct(ComputerID) as ComputerCount by AppName
Trend of App Learning activity for a given app. Useful for tracking activity before and after a rule change	Type=DHWipAppLearning AppName="MICROSOFT.SKYPEAPP"

Exporting data and configuring alerts

Azure Portal enables you to export data to other tools. To do this, in any view that shows **Log Search** just click the **Export** button. Similarly, clicking the **Alert** button will enable you to run a query automatically on a schedule and receive email alerts for particular query results that you set. If you have a PowerBI account, then you will also see a **PowerBI** button that enables you to run a query on a schedule and have the results automatically saved as a PowerBI data set.

Related topics

[Get started with Device Health](#)

For the latest information on Windows Analytics, including new features and usage tips, see the [Windows Analytics blog](#)

Enrolling devices in Windows Analytics

6/14/2019 • 14 minutes to read • [Edit Online](#)

If you have not already done so, consult the topics for any of the three Windows Analytics solutions (Update Compliance, Upgrade Readiness, and Device Health) you intend to use and follow the steps there to add the solutions to Azure Portal.

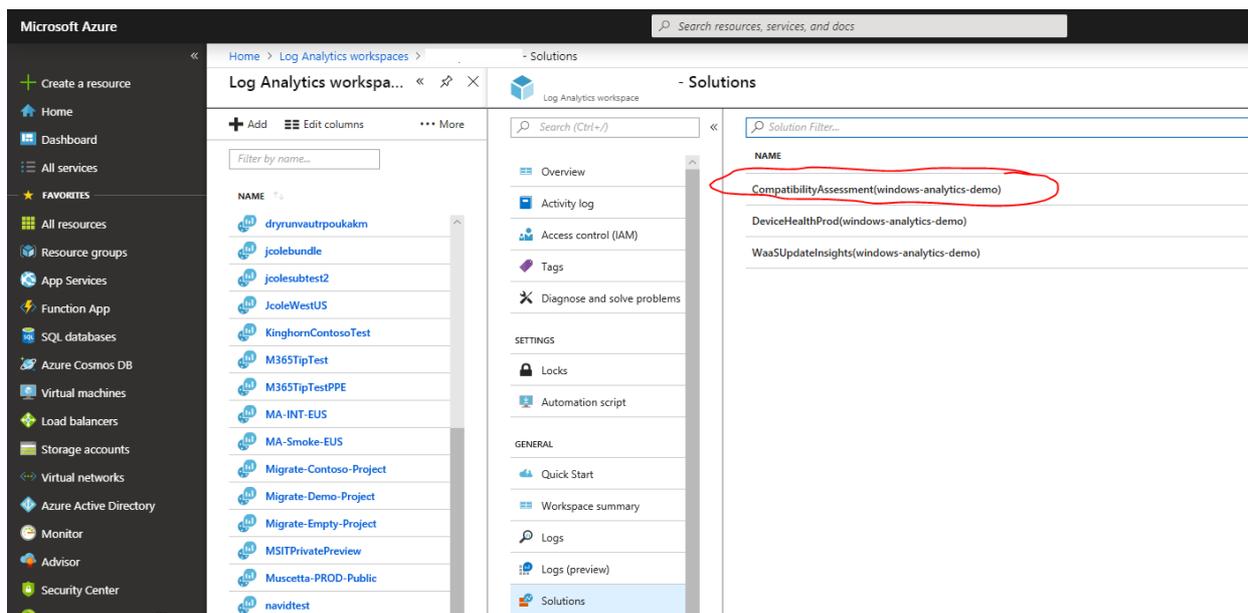
- [Get started with Device Health](#)
- [Get started with Update Compliance](#)
- [Get started with Upgrade Readiness](#)

If you've already done that, you're ready to enroll your devices in Windows Analytics by following these steps:

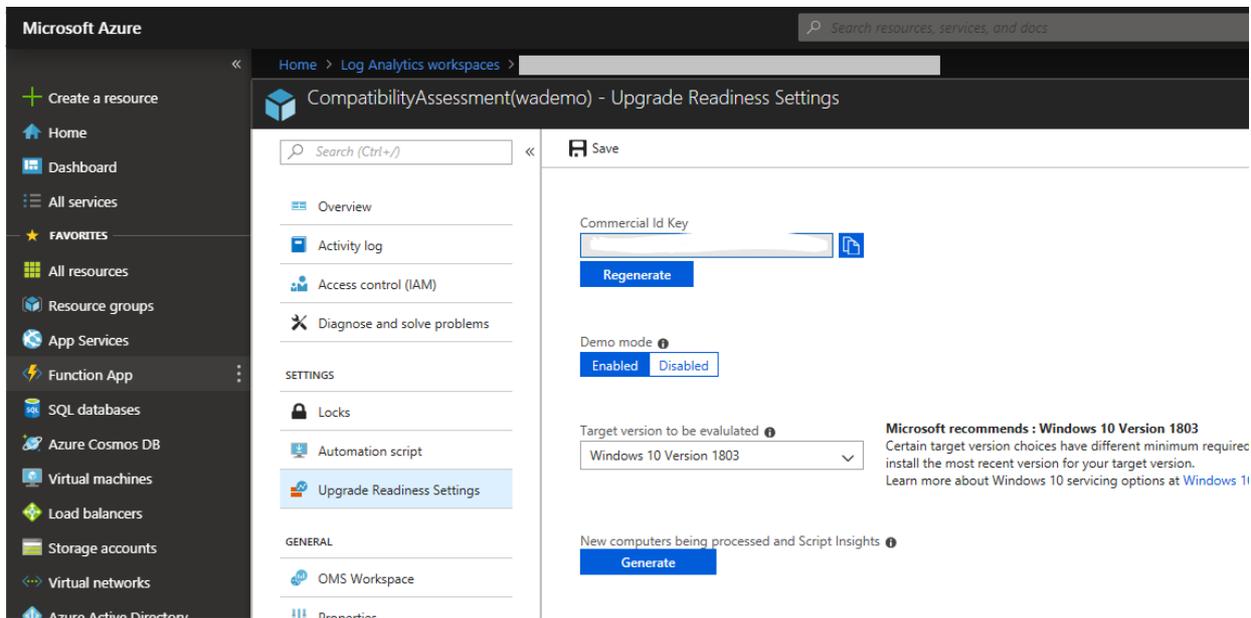
Copy your Commercial ID key

Microsoft uses a unique commercial ID to map information from user computers to your Azure workspace. This should be generated for you automatically. Copy your commercial ID key from any of the Windows Analytics solutions you have added to your Windows Portal, and then deploy it to user computers.

To find your commercial ID, first navigate to the **Solutions** tab for your workspace, and then select the solution. In this example, Upgrade Readiness is being adjusted by selecting **CompatibilityAssessment**:



From there, select the settings page, where you can find and copy your commercial ID:



Important

Regenerate a Commercial ID key only if your original ID key can no longer be used. Regenerating a commercial ID key resets the data in your workspace for all solutions that use the ID. Additionally, you'll need to deploy the new commercial ID key to user computers again.

Enable data sharing

To enable data sharing, configure your proxy server to whitelist the following endpoints. You might need to get approval from your security group to do this.

ENDPOINT	FUNCTION
<code>https://ceuswatcab01.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://ceuswatcab02.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://eaus2watcab01.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://eaus2watcab02.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://weus2watcab01.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://weus2watcab02.blob.core.windows.net</code>	Windows Error Reporting (WER); required for Device Health reports in Windows 10, version 1809 or later. Not used by Upgrade Readiness or Update Compliance AV reports.

ENDPOINT	FUNCTION
<code>https://v10c.events.data.microsoft.com</code>	Connected User Experience and Diagnostic component endpoint for use with devices running Windows 10, version 1803 or later that also have the 2018-09 Cumulative Update (KB4458469, KB4457136, KB4457141) or later installed
<code>https://v10.events.data.microsoft.com</code>	Connected User Experience and Diagnostic component endpoint for use with Windows 10, version 1803 <i>without</i> the 2018-09 Cumulative Update installed
<code>https://v10.vortex-win.data.microsoft.com</code>	Connected User Experience and Diagnostic component endpoint for Windows 10, version 1709 or earlier
<code>https://vortex-win.data.microsoft.com</code>	Connected User Experience and Diagnostic component endpoint for operating systems older than Windows 10
<code>https://settings-win.data.microsoft.com</code>	Enables the compatibility update to send data to Microsoft.
<code>http://adl.windows.com</code>	Allows the compatibility update to receive the latest compatibility data from Microsoft.
<code>https://watson.telemetry.microsoft.com</code>	Windows Error Reporting (WER); required for Device Health reports. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://oca.telemetry.microsoft.com</code>	Online Crash Analysis; required for Device Health reports. Not used by Upgrade Readiness or Update Compliance AV reports.
<code>https://login.live.com</code>	This endpoint is required by Device Health to ensure data integrity and provides a more reliable device identity for all of the Windows Analytics solutions on Windows 10. If you want to disable end-user managed service account (MSA) access, you should apply the appropriate policy instead of blocking this endpoint.

NOTE

Proxy authentication and SSL inspections are frequent challenges for enterprises. See the following sections for configuration options.

IMPORTANT

For privacy and data integrity, Windows checks for a Microsoft SSL certificate when communicating with the diagnostic data endpoints. SSL interception and inspection aren't possible. To use Desktop Analytics, exclude these endpoints from SSL inspection.

Configuring endpoint access with SSL inspection

To ensure privacy and data integrity Windows checks for a Microsoft SSL certificate when communicating with the diagnostic data endpoints. Accordingly SSL interception and inspection is not possible. To use Windows Analytics services you should exclude the above endpoints from SSL inspection.

Configuring endpoint access with proxy server authentication

If your organization uses proxy server authentication for outbound traffic, use one or more of the following approaches to ensure that the diagnostic data is not blocked by proxy authentication:

- **Best option: Bypass** Configure your proxy servers to **not** require proxy authentication for traffic to the diagnostic data endpoints. This is the most comprehensive solution and it works for all versions of Windows 10.
- **User proxy authentication:** Alternatively, you can configure devices to use the logged on user's context for proxy authentication. First, update the devices to Windows 10, version 1703 or later. Then, ensure that users of the devices have proxy permission to reach the diagnostic data endpoints. This requires that the devices have console users with proxy permissions, so you couldn't use this method with headless devices.
- **Device proxy authentication:** Another option--the most complex--is as follows: First, configure a system level proxy server on the devices. Then, configure these devices to use machine-account-based outbound proxy authentication. Finally, configure proxy servers to allow the machine accounts access to the diagnostic data endpoints.

Deploy the compatibility update and related updates

The compatibility update scans your devices and enables application usage tracking. If you don't already have these updates installed, you can download the applicable version from the Microsoft Update Catalog or deploy it using Windows Server Update Services (WSUS) or your software distribution solution, such as System Center Configuration Manager.

OPERATING SYSTEM	UPDATES
Windows 10	Windows 10 includes the compatibility update, so you will automatically have the latest compatibility update so long as you continue to keep your Windows 10 devices up to date with cumulative updates.
Windows 8.1	The compatibility update is included in monthly quality updates for Windows 8.1. We recommend installing the latest Windows Monthly Rollup before attempting to enroll devices into Windows Analytics.
Windows 7 SP1	The compatibility update is included in monthly quality updates for Windows 7. We recommend installing the latest Windows Monthly Rollup before attempting to enroll devices into Windows Analytics.

Connected User Experiences and Telemetry service

With Windows diagnostic data enabled, the Connected User Experience and Telemetry service (DiagTrack) collects system, application, and driver data. Microsoft analyzes this data, and shares it back to you through Windows Analytics. For the best experience, install these updates depending upon the operating system version.

- For Windows 10, install the latest Windows 10 cumulative update.
- For Windows 8.1, install the October 2018 monthly rollup, [KB4462926](#)
- For Windows 7, install the October 2018 monthly rollup, [KB4462923](#)

IMPORTANT

Restart devices after you install the compatibility updates for the first time.

NOTE

We recommend you configure your update management tool to automatically install the latest version of these updates. There is a related optional update, [KB 3150513](#), which can provide updated configuration and definitions for older compatibility updates. For more information about this optional update, see <https://support.microsoft.com/kb/3150513>.

If you are planning to enable IE Site Discovery in Upgrade Readiness, you will need to install a few additional updates.

SITE DISCOVERY	UPDATE
Review site discovery	<p>KB3080149 Updates the Diagnostic and Telemetry tracking service to existing devices. This update is only necessary on Windows 7 and Windows 8.1 devices. For more information about this update, see https://support.microsoft.com/kb/3080149</p> <p>Install the latest Windows Monthly Rollup. This functionality has been included in Internet Explorer 11 starting with the July 2016 Cumulative Update.</p>

NOTE

IE site discovery is disabled on devices running Windows 7 and Windows 8.1 that are in Switzerland and EU countries.

Set diagnostic data levels

You can set the diagnostic data level used by monitored devices either with the [Upgrade Readiness deployment script](#) or by policy (by using Group Policy or Mobile Device Management).

The basic functionality of Upgrade Readiness will work at the Basic diagnostic data level, you won't get usage or health data for your updated devices without enabling the Enhanced level. This means you won't get information about health regressions on updated devices. So it is best to enable the Enhanced diagnostic data level, at least on devices running Windows 10, version 1709 (or later) where the Enhanced diagnostic data setting can be paired with "limited enhanced" data level (see [Windows 10 enhanced diagnostic data events and fields used by Windows Analytics](#)). For more information, see [Windows Analytics and privacy](#).

Enroll a few pilot devices

You can use the Upgrade Readiness deployment script to automate and verify your deployment. We always recommend manually running this script on a few representative devices to verify things are properly configured and the device can connect to the diagnostic data endpoints. Make sure to run the pilot version of the script, which will provide extra diagnostics.

See the [Upgrade Readiness deployment script](#) topic for information about obtaining and running the script, and for a description of the error codes that can be displayed. See "[Understanding connectivity scenarios and the deployment script](#)" on the Windows Analytics blog for a summary of setting the ClientProxy for the script, which will enable the script properly check for diagnostic data endpoint connectivity.

After data is sent from devices to Microsoft, it generally takes 48-56 hours for the data to populate in Windows Analytics. The compatibility update takes several minutes to run. If the update does not get a chance to finish running or if the computers are inaccessible (turned off or sleeping for example), data will take longer to populate in Windows Analytics. For this reason, you can expect most of your devices to be populated in

Windows Analytics in about 1-2 weeks after deploying the update and configuration to user computers. As described in the Windows Analytics blog post "[You can now check on the status of your computers within hours of running the deployment script](#)", you can verify that devices have successfully connected to the service within a few hours. Most of those devices should start to show up in the Windows Analytics console within a few days.

Deploy additional optional settings

Certain Windows Analytics features have additional settings you can use.

- **Update Compliance** is only compatible with Windows 10 desktop devices (workstations and laptops). To use the Windows Defender Antivirus Assessment, devices must be protected by Windows Defender AV (and not a partner antivirus application), and must have enabled cloud-delivered protection, as described in [Utilize Microsoft cloud-delivered protection in Windows Defender Antivirus](#). See the [Troubleshoot Windows Defender Antivirus reporting in Update Compliance](#) topic for help with ensuring that the configuration is correct.
- For devices running Windows 10, version 1607 or earlier, Windows diagnostic data must also be set to Enhanced (see [Configure Windows diagnostic data in your organization](#)) in order to be compatible with Windows Defender Antivirus. See the [Windows Defender Antivirus in Windows 10 and Windows Server 2016](#) for more information about enabling, configuring, and validating Windows Defender AV.
- **Device Health** is only compatible with Windows 10 desktop devices (workstations and laptops) and Windows Server 2016. The solution requires that at least the Enhanced level of diagnostic data is enabled on all devices that are intended to be displayed in the solution. In Windows 10, version 1709, a new policy was added to "limit enhanced telemetry to the minimum required by Windows Analytics". To learn more about Windows diagnostic data, see [Configure Windows diagnostic data in your organization](#).
- **IE site discovery** is an optional feature of Upgrade Readiness that provides an inventory of websites that are accessed by client devices using Internet Explorer on Windows 7, Windows 8.1, and Windows 10. To enable IE site discovery, make sure the required updates are installed (per previous section) and enable IE site discovery in the deployment script batch file.

Deploying Windows Analytics at scale

When you have completed a pilot deployment, you are ready to automate data collection and distribute the deployment script to the remaining devices in your organization.

Automate data collection

To ensure that user computers are receiving the most up-to-date data from Microsoft, we recommend that you establish the following data sharing and analysis processes:

- Enable automatic updates for the compatibility update and related updates. These updates include the latest application and driver issue information as we discover it during testing.
- Schedule the Upgrade Readiness deployment script to automatically run monthly. Scheduling the script ensures that full inventory is sent monthly even if devices were not connected or had low battery power at the time the system normally sends inventory. Make sure to run the production version of the script, which is lighter weight and non-interactive. The script also has a number of built-in error checks, so you can monitor the results. If you can't run the deployment script at scale, another option is to configure things centrally via Group Policy or Mobile Device Management (MDM). Although we recommend using the deployment script, both options are discussed in the sections below.

When you run the deployment script, it initiates a full scan. The daily scheduled task to capture the changes is created when the update package is installed. For Windows 10 devices, this task is already included in the

operating system. A full scan averages about 2 MB, but the scans for changes are very small. The scheduled task is named "Windows Compatibility Appraiser" and can be found in the Task Scheduler Library under Microsoft > Windows > Application Experience. Changes are invoked via the nightly scheduled task. It attempts to run around 3:00AM every day. If the system is powered off at that time, the task will run when the system is turned on.

Distribute the deployment script at scale

Use a software distribution system such as System Center Configuration Manager to distribute the Upgrade Readiness deployment script at scale. For more information, see [Upgrade Readiness deployment script](#). For information on how to deploy PowerShell scripts by using Windows Intune, see [Manage PowerShell scripts in Intune for Windows 10 devices](#).

Distributing policies at scale

There are a number of policies that can be centrally managed to control Windows Analytics device configuration. All of these policies have *preference* registry key equivalents that can be set by using the deployment script. Policy settings override preference settings if both are set.

NOTE

You can only set the diagnostic data level to Enhanced by using policy. For example, this is necessary for using Device Health.

These policies are under Microsoft\Windows\DataCollection:

POLICY	VALUE
CommercialId	In order for your devices to show up in Windows Analytics, they must be configured with your organization's Commercial ID.
AllowTelemetry (in Windows 10)	1 (Basic), 2 (Enhanced) or 3 (Full) diagnostic data. Windows Analytics will work with basic diagnostic data, but more features are available when you use the Enhanced level (for example, Device Health requires Enhanced diagnostic data and Upgrade Readiness only collects app usage and site discovery data on Windows 10 devices with Enhanced diagnostic data). For more information, see Configure Windows diagnostic data in your organization .
LimitEnhancedDiagnosticDataWindowsAnalytics (in Windows 10)	Only applies when AllowTelemetry=2. Limits the Enhanced diagnostic data events sent to Microsoft to just those needed by Windows Analytics. For more information, see Windows 10, version 1709 enhanced diagnostic data events and fields used by Windows Analytics .
AllowDeviceNameInTelemetry (in Windows 10)	In Windows 10, version 1803, a separate opt-in is required to enable devices to continue to send the device name. Allowing device names to be collected can make it easier for you to identify individual devices that report problems. Without the device name, Windows Analytics can only label devices by a GUID that it generates.
CommercialDataOptIn (in Windows 7 and Windows 8)	1 is required for Upgrade Readiness, which is the only solution that runs on Windows 7 or Windows 8.

You can set these values by using Group Policy (in Computer Configuration > Administrative Templates > Windows Components > Data Collection and Preview Builds) or by using Mobile Device Management (in

Provider/*Provider ID*/CommercialID). (If you are using Microsoft Intune, use `MS DM Server` as the provider ID.) For more information about deployment using MDM, see the [DMClient CSP](#) topic in MDM documentation.

The corresponding preference registry values are available in

HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\DataCollection and can be configured by the deployment script. If a given setting is configured by both preference registry settings and policy, the policy values will override. However, the **IEDataOptIn** setting is different--you can only set this with the preference registry keys:

- IEOptInLevel = 0 Internet Explorer data collection is disabled
- IEOptInLevel = 1 Data collection is enabled for sites in the Local intranet + Trusted sites + Machine local zones
- IEOptInLevel = 2 Data collection is enabled for sites in the Internet + Restricted sites zones
- IEOptInLevel = 3 Data collection is enabled for all sites

For more information about Internet Explorer Security Zones, see [About URL Security Zones](#).

Distribution at scale without using the deployment script

We recommend using the deployment script to configure devices. However if this is not an option, you can still manage settings by policy as described in the previous section. However, if you don't run the deployment script, you won't benefit from its error checking, and you might have to wait a long time (possibly weeks) before devices send the initial full inventory scan.

Note that it is possible to initiate a full inventory scan on a device by calling these commands:

- `CompatTelRunner.exe -m:general.tel.dll -f:DoCensusRun`
- `CompatTelRunner.exe -m:appraiser.dll -f:DoScheduledTelemetryRun ent`

For details on how to run these and how to check results, see the deployment script.

Frequently asked questions and troubleshooting Windows Analytics

6/14/2019 • 15 minutes to read • [Edit Online](#)

IMPORTANT

The OMS portal has been deprecated; you should start using the Azure portal instead as soon as possible. Many experiences are the same in the two portals, but there are some key differences. See [Windows Analytics in the Azure Portal](#) for steps to use Windows Analytics in the Azure portal. For much more information about the transition from OMS to Azure, see [OMS portal moving to Azure](#).

This topic compiles the most common issues encountered with configuring and using Windows Analytics, as well as general questions. This FAQ, along with the [Windows Analytics Technical Community](#), are recommended resources to consult before contacting Microsoft support.

Troubleshooting common problems

If you've followed the steps in the [Enrolling devices in Windows Analytics](#) topic and are still encountering problems, you might find the solution here.

[Devices not appearing in Upgrade Readiness](#)

[Devices not appearing in Device Health Device Reliability](#)

[Device crashes not appearing in Device Health Device Reliability](#)

[Apps not appearing in Device Health App Reliability](#)

[Upgrade Readiness shows many "Computers with outdated KB"](#)

[Upgrade Readiness shows many "Computers with incomplete data"](#)

[Upgrade Readiness doesn't show app inventory data on some devices](#)

[Upgrade Readiness doesn't show IE site discovery data from some devices](#)

[Device names not appearing for Windows 10 devices](#)

[Custom log queries using the AbnormalShutdownCount field of Device Health show zero or lower than expected results](#)

[Disable Upgrade Readiness](#)

[Exporting large data sets](#)

Devices not appearing in Upgrade Readiness

In Log Analytics, go to **Settings > Connected sources > Windows telemetry** and verify that you are subscribed to the Windows Analytics solutions you intend to use.

Even though devices can take 2-3 days after enrollment to show up due to latency in the system, you can now verify the status of your devices within a few hours of running the deployment script as described in [You can now check on the status of your computers within hours of running the deployment script](#) on the Tech Community Blog.

NOTE

If you generate the status report and get an error message saying "Sorry! We're not recognizing your Commercial Id," go to **Settings > Connected sources > Windows telemetry** remove the Upgrade Readiness solution, and then re-add it.

If devices are not showing up as expected, find a representative device and follow these steps to run the latest pilot version of the Upgrade Readiness deployment script on it to troubleshoot issues:

1. Download and extract the [Upgrade Readiness Deployment Script](#). Ensure that the **Pilot/Diagnostics** folder is included.
2. Edit the script as described in [Upgrade Readiness deployment script](#).
3. Check that `isVerboseLogging` is set to `$true`.
4. Run the script again. Log files will be saved to the directory specified in the script.
5. Check the output of the script in the command window and/or log **UA_dateTime_machineName.txt** to ensure that all steps were completed successfully.
6. If you are still seeing errors you can't diagnose, then consider open a support case with Microsoft Support through your regular channel and provide this information.

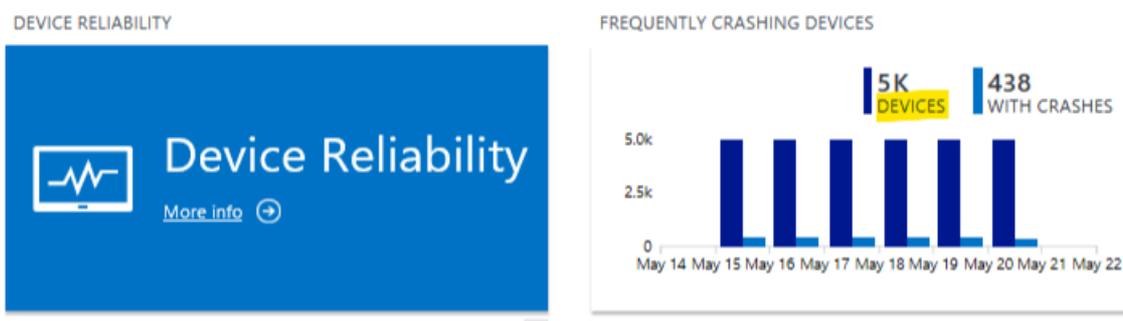
If you want to check a large number of devices, you should run the latest script at scale from your management tool of choice (for example, System Center Configuration Manager) and check the results centrally.

If you think the issue might be related to a network proxy, check "Enable data sharing" section of the [Enrolling devices in Windows Analytics](#) topic. Also see [Understanding connectivity scenarios and the deployment script](#) on the Windows Analytics blog.

If you have deployed images that have not been generalized, then many of them might have the same ID and so Windows Analytics will see them as one device. If you suspect this is the issue, then you can reset the IDs on the non-generalized devices by performing these steps:

1. Net stop diagtrack
2. Reg delete hklm\software\microsoft\sqmclient /v MachineId /f
3. Net start diagtrack

Devices not appearing in Device Health Device Reliability

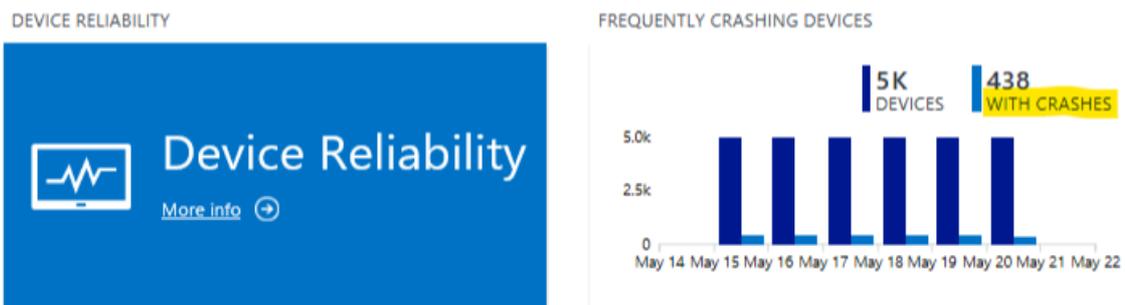


If you have devices that appear in other solutions, but not Device Health (the Device Health overview tile shows "Performing Assessment" or the device count is lower than expected), follow these steps to investigate the issue:

1. Using the Azure portal, remove the Device Health (appears as DeviceHealthProd on some pages) solution from your Log Analytics workspace. After completing this, add the Device Health solution to your workspace again.
2. Confirm that the devices are running Windows 10.
3. Verify that the Commercial ID is present in the device's registry. For details see <https://gpsearch.azurewebsites.net/#13551>.
4. Confirm that devices are opted in to send diagnostic data by checking in the registry that **AllowTelemetry** is set to either 2 (Enhanced) or 3 (Full).

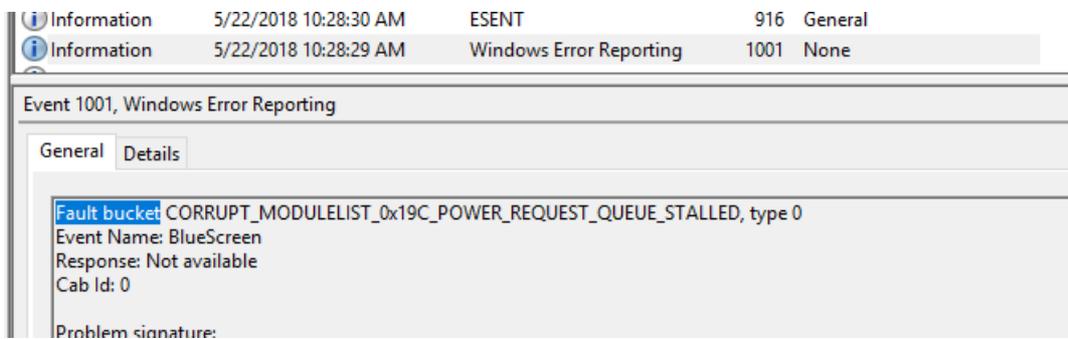
- **AllowTelemetry** under **HKLM\Software\Policies\Microsoft\Windows\DataCollection** is the IT policy path.
 - **AllowTelemetry** under **HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\DataCollection** is the user preference (Settings app) path.
 - IMPORTANT: By convention (and in earlier versions of Windows 10) the IT policy would take precedence over any user preference. Starting with Windows 10, version 1803, the user can lower the device's effective value even when an IT policy is set. This change assists organizations in complying with regional or organizational expectations about user control over privacy settings. For organizations where user control of privacy settings is not required, the previous behavior (IT policy path always wins) can be enabled using the new policy **Computer Configuration\Administrative Templates\Windows Components\Data Collection and Preview Builds\Configure telemetry opt-in setting user interface**.
5. Verify that devices can reach the endpoints specified in [Enrolling devices in Windows Analytics](#). Also check settings for SSL inspection and proxy authentication; see [Configuring endpoint access with SSL inspection](#) for more information.
 6. Wait 48 hours for activity to appear in the reports.
 7. If you need additional troubleshooting, contact Microsoft Support.

Device crashes not appearing in Device Health Device Reliability



If you know that devices are experiencing stop error crashes that do not seem to be reflected in the count of devices with crashes, follow these steps to investigate the issue:

1. Verify that devices are reporting data properly by following the steps in the [Devices not appearing in Device Health Device Reliability](#) section of this topic.
2. Trigger a known crash on a test device by using a tool such as [NotMyFault](#) from Windows Sysinternals.
3. Verify that Windows Error Reporting (WER) is not disabled or redirected by confirming the registry settings in **HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting** (or **HKLM\Software\Policies\Microsoft\Windows\DataCollection**, which will take precedence if set):
 - Verify that the value "Disabled" (REG_DWORD), if set, is 0.
 - Verify that the value "DontSendAdditionalData" (REG_DWORD), if set, is 0.
 - Verify that the value "CorporateWERServer" (REG_SZ) is not configured.
4. Verify that WER can reach all diagnostic endpoints specified in [Enrolling devices in Windows Analytics](#)--if WER can only reach some of the endpoints, it could be included in the device count while not reporting crashes.
5. Check that crash reports successfully complete the round trip with Event 1001 and that BucketID is not blank. A typical such event looks like this:



You can use the following Windows PowerShell snippet to summarize recent occurrences of Event 1001. Most events should have a value for BucketID (a few intermittent blank values are OK, however).

```
$limitToMostRecentNEvents = 20
Get-WinEvent -FilterHashTable @{ProviderName="Windows Error Reporting"; ID=1001} |
?{ $_.Properties[2].Value -match "crash|blue" } |
% { [pscustomobject]@{
    TimeCreated=$_.TimeCreated
    WEREvent=$_.Properties[2].Value
    BucketId=$_.Properties[0].Value
    ContextHint = $(
        if($_.Properties[2].Value -eq "bluescreen"){ "kernel" }
        else{ $_.Properties[5].Value }
    )
}} | Select-Object -First $limitToMostRecentNEvents
```

The output should look something like this:

TimeCreated	WEREvent	BucketId	ContextHint
5/24/2018 11:25:10 AM	BlueScreen	0x1E_c0000005_X_nt!KiExceptionDispatch	kernel
5/24/2018 11:24:59 AM	BlueScreen		kernel
5/24/2018 11:18:22 AM	APPCRASH	1984268883	CrashMe.exe
5/24/2018 11:18:16 AM	APPCRASH	1916561429	CrashMe.exe
5/21/2018 3:21:08 PM	BlueScreen	0x139_3_CORRUPT_LIST_ENTRY_nt!ExAllocatePoolWithTag	kernel
5/21/2018 3:20:59 PM	BlueScreen		kernel
5/21/2018 3:14:23 PM	BlueScreen	AV_myfault!unknown_function	kernel
5/21/2018 3:14:09 PM	BlueScreen		kernel
5/21/2018 3:07:35 PM	APPCRASH	1911820819159639774	CrashMe.exe
5/21/2018 3:07:27 PM	APPCRASH	1916561429	CrashMe.exe

6. Check that some other installed device, app, or crash monitoring solution is not intercepting crash events.
7. Wait 48 hours for activity to appear in the reports.
8. If you need additional troubleshooting, contact Microsoft Support.

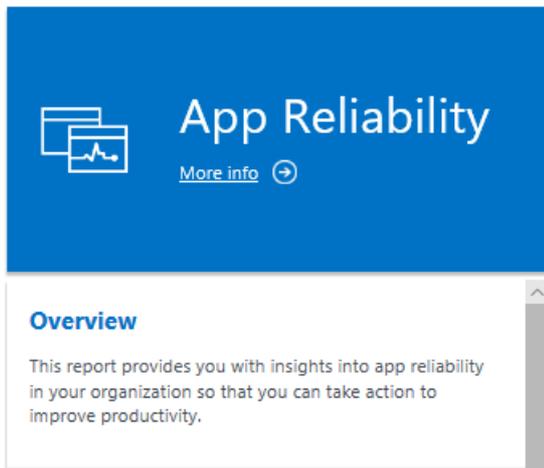
Endpoint connectivity

Devices must be able to reach the endpoints specified in [Enrolling devices in Windows Analytics](#).

If you are using proxy server authentication, it's worth taking extra care to check the configuration. Prior to Windows 10, version 1703, WER only uploads error reports in the machine context, so whitelisting endpoints to allow non-authenticated access was typically used. In Windows 10, version 1703 and later versions, WER will attempt to use the context of the user that is logged on for proxy authentication such that only the user account requires proxy access.

For more information, see [Enrolling devices in Windows Analytics](#).

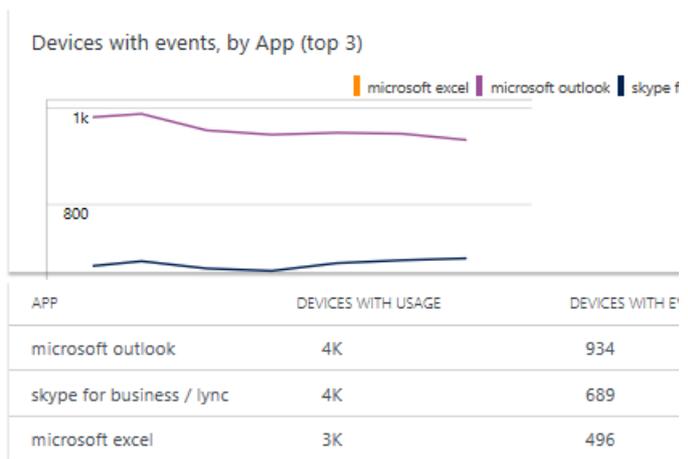
Apps not appearing in Device Health App Reliability



App Reliability
More info →

Overview

This report provides you with insights into app reliability in your organization so that you can take action to improve productivity.

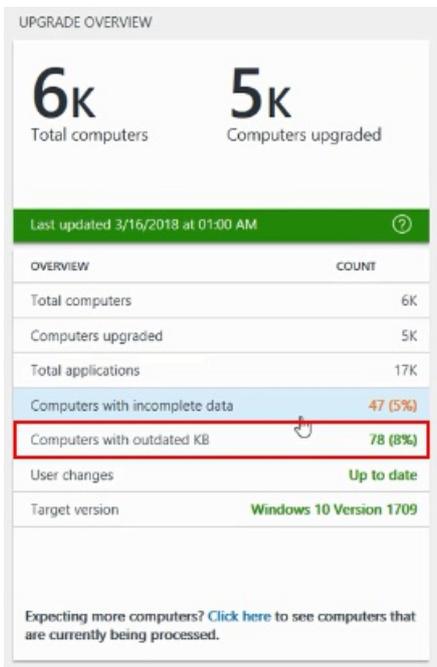


If apps that you know are crashing do not appear in App Reliability, follow these steps to investigate the issue:

1. Double-check the steps in the [Devices not appearing in Device Health Device Reliability](#) and [Device crashes not appearing in Device Health Device Reliability](#) sections of this topic.
2. Confirm that an in-scope application has crashed on an enrolled device. Keep the following points in mind:
 - Not all user-mode crashes are included in App Reliability, which tracks only apps that have a GUI, have been used interactively by a user, and are not part of the operating system.
 - Enrolling more devices helps to ensure that there are enough naturally occurring app crashes.
 - You can also use test apps which are designed to crash on demand.
3. Verify that *per-user* Windows Error Reporting (WER) is not disabled or redirected by confirming the registry settings in **HKCU\SOFTWARE\Microsoft\Windows\Windows Error Reporting** (or **HKCU\Software\Policies\Microsoft\Windows\DataCollection**, which will take precedence if set):
 - Verify that the value "Disabled" (REG_DWORD), if set, is 0.
 - Verify that the value "DontSendAdditionalData" (REG_DWORD), if set, is 0.
 - Verify that the value "CorporateWERServer" (REG_SZ) is not configured.
4. Check that some other installed device, app, or crash monitoring solution is not intercepting crash events.
5. Wait 48 hours for activity to appear in the reports.
6. If you need additional troubleshooting, contact Microsoft Support.

Upgrade Readiness shows many "Computers with outdated KB"

If you see a large number of devices reported as shown in this screenshot of the Upgrade Readiness tile:

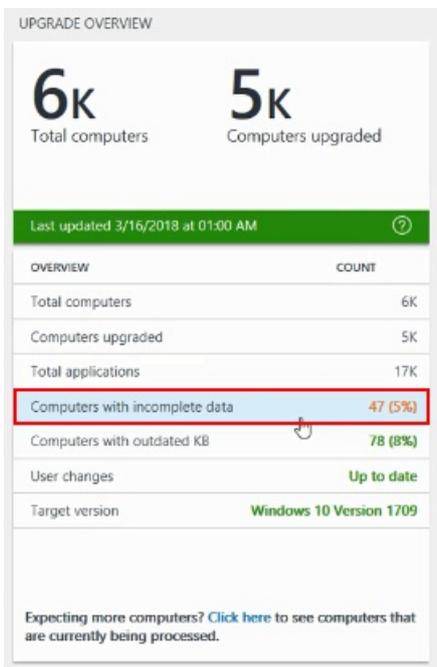


On Windows 7 SP1 and Windows 8.1 devices, you must deploy the compatibility update as described in [Enrolling devices in Windows Analytics](#).

Note that the compatibility update retains the same KB number when a new version is released, so even if the update is installed on your devices, *they might not be running the latest version*. The compatibility update is now a critical update, so you can check that the latest version is installed from your management tool.

Upgrade Readiness shows many "Computers with incomplete data"

If you see a large number of devices reported as shown in this screenshot of the Upgrade Readiness tile:



Download the latest deployment script and run it on an affected device to check for issues. See the [Upgrade Readiness deployment script](#) topic for information about obtaining and running the script, and for a description of the error codes that can be displayed. Remember to wait up to 48-72 hours to see the results. See "[Understanding connectivity scenarios and the deployment script](#)" on the Windows Analytics blog for a summary of setting the ClientProxy for the script, which will enable the script properly check for diagnostic data endpoint connectivity.

If this becomes a recurring issue, schedule a full inventory scan monthly, as per the device enrollment guidelines for deployment at scale.

Upgrade Readiness doesn't show app inventory data on some devices

Upgrade Readiness only collects app inventory on devices that are not yet upgraded to the target operating system version specified in the Upgrade Readiness Overview blade. This is because Upgrade Readiness targets upgrade planning (for devices not yet upgraded).

Upgrade Readiness doesn't show IE site discovery data from some devices

Double-check that IE site discovery opt-in has been configured in the deployment script. (See the [Upgrade Readiness deployment script](#) topic for information about obtaining and running the script, and for a description of the error codes that can be displayed. See "[Understanding connectivity scenarios and the deployment script](#)" on the Windows Analytics blog for a summary of setting the ClientProxy for the script, which will enable the script properly check for diagnostic data endpoint connectivity.)

Also, on Windows 10 devices remember that IE site discovery requires data diagnostics set to the Enhanced level.

There are two additional configurations to check:

1. Make sure Flip Ahead with Page Prediction is enabled. It can be configured at Internet Options -> Advanced -> Browsing -> Enable flip ahead with page prediction.
2. Make sure IE is not running in InPrivate mode.

Finally, Upgrade Readiness only collects IE site discovery data on devices that are not yet upgraded to the target operating system version specified in the Upgrade Readiness Overview blade. This is because Upgrade Readiness targets upgrade planning (for devices not yet upgraded).

NOTE

IE site discovery is disabled on devices running Windows 7 and Windows 8.1 that are in Switzerland and EU countries.

Device names not appearing for Windows 10 devices

Starting with Windows 10, version 1803, the device name is no longer collected by default and requires a separate opt-in. For more information, see [Enrolling devices in Windows Analytics](#). Allowing device names to be collected can make it easier for you to identify individual devices that report problems. Without the device name, Windows Analytics can only label devices by a GUID that it generates.

Custom log queries using the AbnormalShutdownCount field of Device Health show zero or lower than expected results

This issue affects custom queries of the Device Health data by using the **Logs > Search page** or API. It does not impact any of the built-in tiles or reports of the Device Health solution. The **AbnormalShutdownCount** field of the **DHOSReliability** data table represents abnormal shutdowns other than crashes, such as sudden power loss or holding down the power button.

We have identified an incompatibility between AbnormalShutdownCount and the Limited Enhanced diagnostic data level on Windows 10, versions 1709, 1803, and 1809. Such devices do not send the abnormal shutdown signal to Microsoft. You should not rely on AbnormalShutdownCount in your custom queries unless you use any one of the following workarounds:

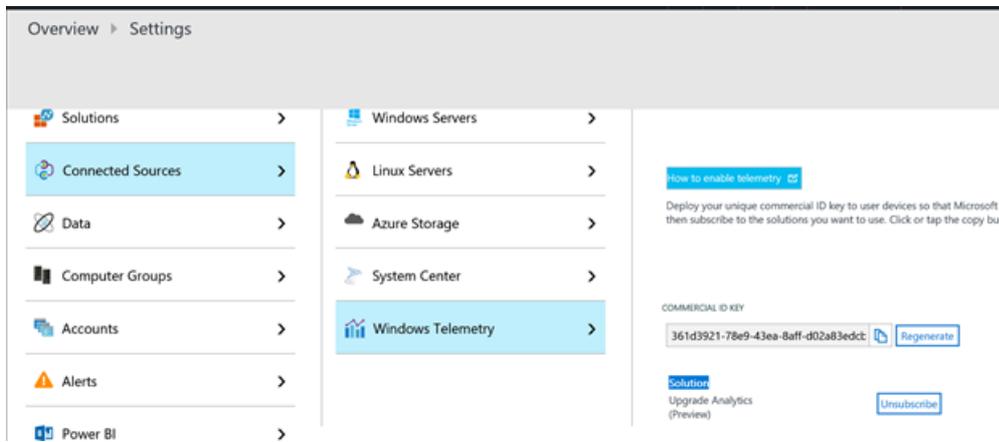
- Upgrade devices to Windows 10, version 1903 when available. Participants in the Windows Insider program can preview this change using Windows Insider builds.
- Change the diagnostic data setting from devices running Windows 10, versions 1709, 1803, and 1809 normal Enhanced level instead of Limited Enhanced.
- Use alternative data from devices to track abnormal shutdowns. For example, you can forward abnormal shutdown events from the Windows Event Log to your Log Analytics workspace by using the Log Analytics agent. Suggested events to forward include:
 - Log: System, ID: 41, Source: Kernel-Power

- o Log System, ID: 6008, Source: EventLog

Disable Upgrade Readiness

If you want to stop using Upgrade Readiness and stop sending diagnostic data to Microsoft, follow these steps:

1. Unsubscribe from the Upgrade Readiness solution in Azure Portal. In Azure Portal, go to **Settings > Connected Sources > Windows Telemetry** and choose the **Unsubscribe** option.



2. Disable the Commercial Data Opt-in Key on computers running Windows 7 SP1 or 8.1. On computers running Windows 10, set the diagnostic data level to **Security**:

Windows 7 and Windows 8.1: Delete CommercialDataOptIn registry property from *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection*

Windows 10: Follow the instructions in [Configure Windows diagnostic data in your organization](#).

3. If you enabled **Internet Explorer Site Discovery**, you can disable Internet Explorer data collection by setting the *IEDataOptIn* registry key to value "0". The IEDataOptIn key can be found under: *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection*.
4. **Optional step:** You can also remove the "CommercialId" key from: *"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection"*.

Exporting large data sets

Azure Log Analytics is optimized for advanced analytics of large data sets and can efficiently generate summaries and analytics for them. The query language is not optimized (or intended) for returning large raw data sets and has built-in limits to protect against overuse. There are times when it might be necessary to get more data than this, but that should be done sparingly since this is not the intended way to use Azure Log Analytics. The following code snippet shows how to retrieve data from UApp one "page" at a time:

```
let snapshot = toscalar(UApp | summarize max(TimeGenerated));
let pageSize = 100000;
let pageNumber = 0;

UApp
| where TimeGenerated == snapshot and IsRollup==true and RollupLevel=="Granular" and Importance == "Low install count"
| order by AppName, AppVendor, AppVersion desc
| serialize
| where row_number(0) >= (pageSize * pageNumber)
| take pageSize
```

Other common questions

What are the requirements and costs for Windows Analytics solutions?

WINDOWS ANALYTICS SOLUTION	WINDOWS LICENSE REQUIREMENTS	WINDOWS VERSION REQUIREMENTS	MINIMUM DIAGNOSTIC DATA REQUIREMENTS
Upgrade Readiness	No additional requirements	Windows 7 with Service Pack 1, Windows 8.1, Windows 10	Basic level in most cases; Enhanced level to support Windows 10 app usage data and IE site discovery
Update Compliance	No additional requirements	Windows 10	Basic level
Device Health	Any of the following licenses: <ul style="list-style-type: none"> - Windows 10 Enterprise or Windows 10 Education per-device with active Software Assurance - Windows 10 Enterprise E3 or E5 per-device or per-user subscription (including Microsoft 365 F1, E3, or E5) - Windows 10 Education A3 or A5 (including Microsoft 365 Education A3 or A5) - Windows VDA E3 or E5 per-device or per-user subscription - Windows Server 2016 or later 	Windows 10	<ul style="list-style-type: none"> - For Windows 10 version 1709 or later: Enhanced (Limited) - For earlier versions: Enhanced

NOTE

Regarding licensing requirements for Device Health, you do not need per-seat licensing, but only enough licenses to cover your total device usage. For example, if you have 100 E3 licenses, you can monitor 100 devices with Device Health.

Beyond the cost of Windows operating system licenses, there is no additional cost for using Windows Analytics. Within Azure Log Analytics, Windows Analytics is "zero-rated;" this means it is excluded from data limits and costs regardless of the Azure Log Analytics pricing tier you have chosen. To be more specific, Azure Log Analytics is available in different pricing tiers as described in [Pricing - Log Analytics](#).

- If you are using the free tier, which has a cap on the amount of data collected per day, the Windows Analytics data will not count towards this cap. You will be able to collect all the Windows Analytics data from your devices and still have the full cap available for collecting additional data from other sources.
- If you are using a paid tier that charges per GB of data collected, the Windows Analytics data will not be charged. You will be able to collect all the Windows Analytics data from your devices and not incur any costs.

Note that different Azure Log Analytics plans have different data retention periods, and the Windows Analytics solutions inherit the workspace's data retention policy. So, for example, if your workspace is on the free plan then Windows Analytics will retain the last week's worth of "daily snapshots" that are collected in the workspace.

Why do SCCM and Upgrade Readiness show different counts of devices that are ready to upgrade?

System Center Configuration Manager (SCCM) considers a device ready to upgrade if *no installed app* has an upgrade decision of "not ready" (that is, they are all "ready" or "in progress"), while Upgrade Readiness considers a device ready to upgrade only if *all* installed apps are marked "ready".

Currently, you can choose the criteria you wish to use:

- To use the SCCM criteria, create the collection of devices ready to upgrade within the SCCM console (using the analytics connector).

- To use the Upgrade Readiness criteria, export the list of ready-to-upgrade devices from the corresponding Upgrade Readiness report, and then build the SCCM collection from that spreadsheet.

How does Upgrade Readiness collect the inventory of devices and applications?

For details about this process and some tips, see [How does Upgrade Readiness in WA collect application inventory for your OMS workspace?](#) on the Windows Analytics blog.