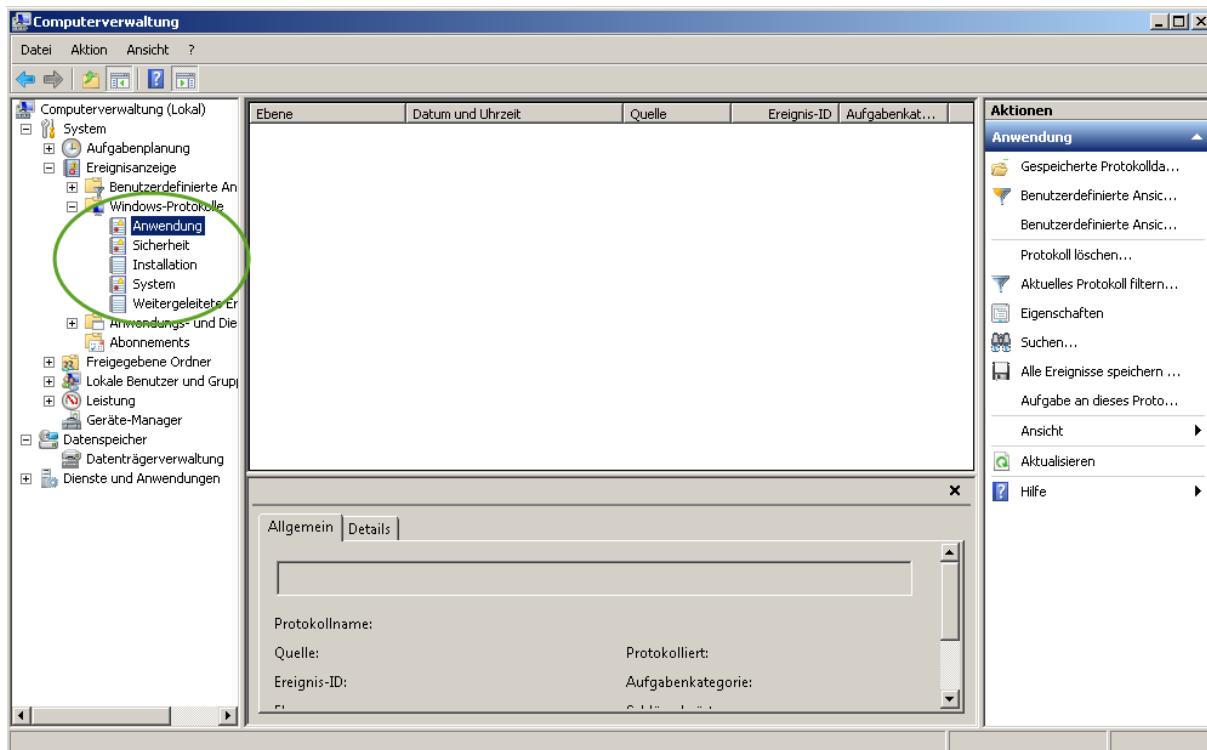


Windows Event-Logs per Batch und Powershell erstellen

Windows erstellt zu allen möglichen Aktionen/Ereignissen detaillierte **Protokolle**. Diese kann man über die > **Computerverwaltung** zur Fehlerbehebung oder Recherche einsehen.

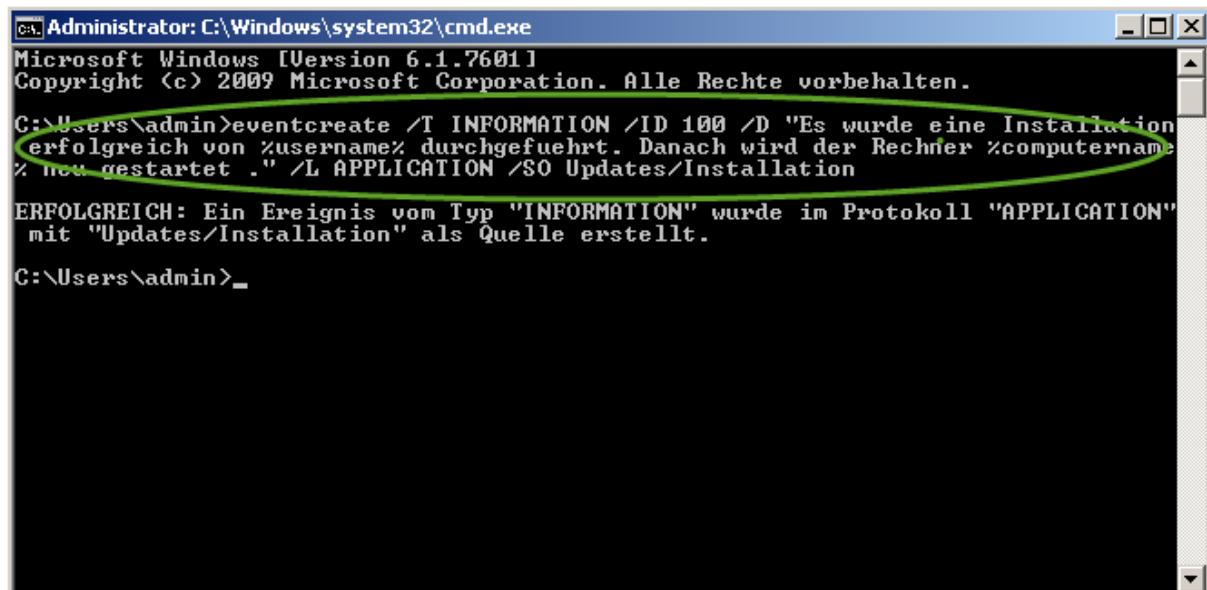


Heute zeige ich Ihnen wie Sie per Batch und Powershell ein solches Event-Log selbst erstellen können.

Entweder über die **CMD** oder per **Batchdatei** auszuführen wäre folgende Codezeile:

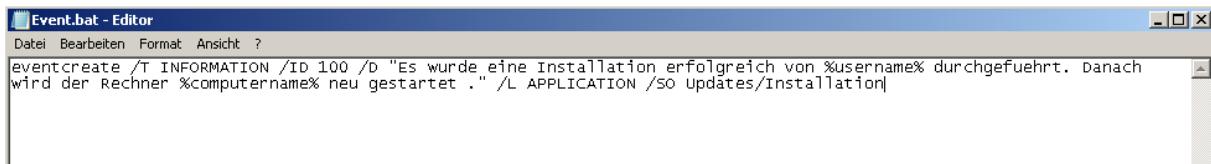
```
eventcreate /T INFORMATION /ID 100 /D "Es wurde eine Installation erfolgreich von %username% durchgefuehrt. Danach wird der Rechner %computername% neu gestartet." /L APPLICATION /SO Updates/Installation
```

Über die CMD ausgeführt:



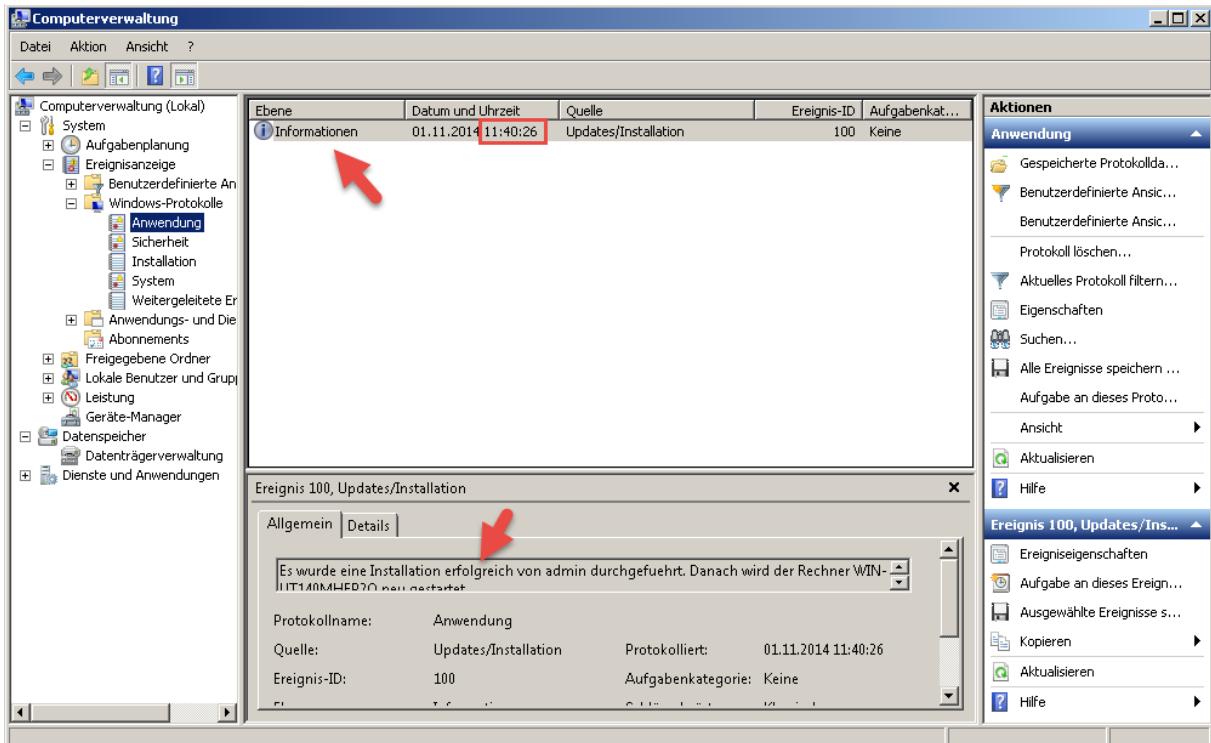
Windows Event-Logs per Batch und Powershell erstellen

Über eine Batchdatei ausgeführt:

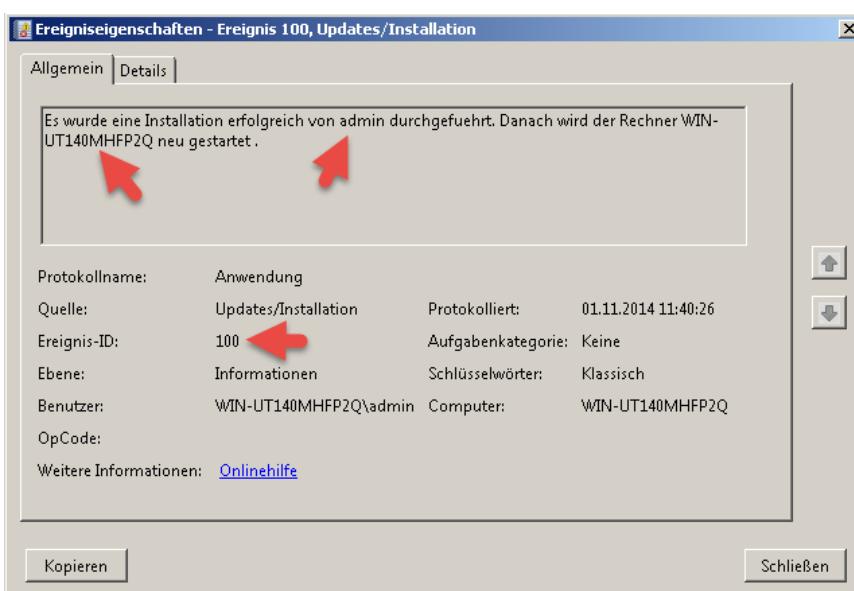


```
eventcreate /T INFORMATION /ID 100 /D "Es wurde eine Installation erfolgreich von %username% durchgefuehrt. Danach wird der Rechner %computername% neu gestartet ." /L APPLICATION /SO Updates/Installation
```

Nach der Ausführung der Codezeile sehen wir unter dem Windows-Protokoll (Anwendung) folgenden Eintrag.



Die Detailansicht des Logs:



Windows Event-Logs per Batch und Powershell erstellen

Ein individuelles Event-Log über die Powershell erstellen wir wie folgt. Wir öffnen die Powershell und führen folgenden Befehl zum Erstellen eines eigenen Event-Logs aus.

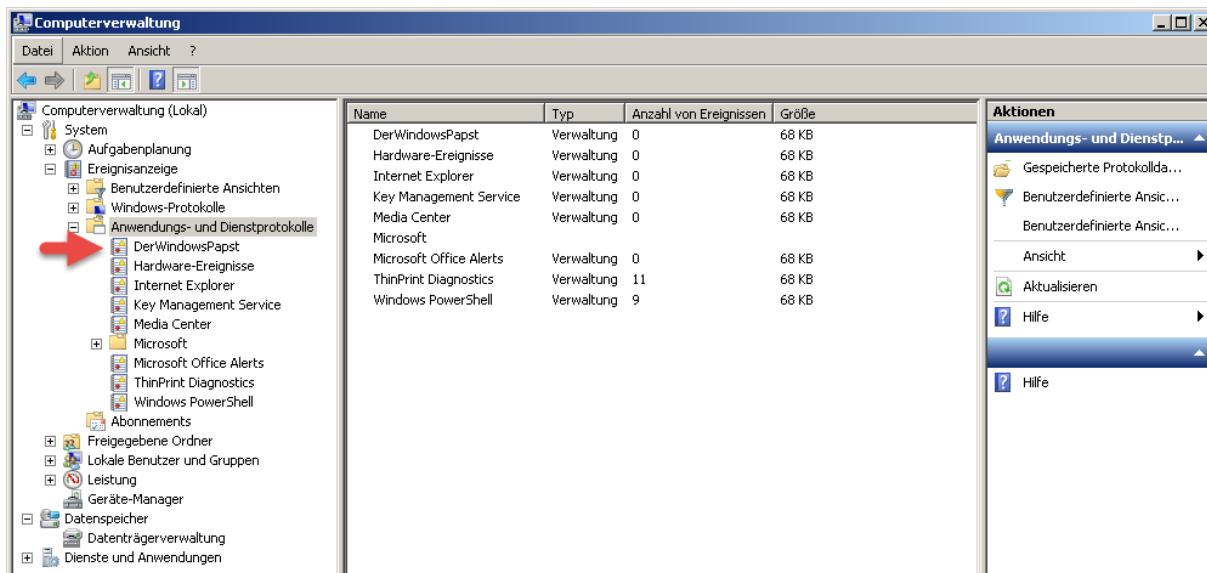
Event-Log Name: DerWindowsPapst
Sourcen: Test, Sonstiges und Routine

new-eventlog -LogName DerWindowsPapst -Source Test,Sonstiges,Routine



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright <C> 2009 Microsoft Corporation. Alle Rechte vorbehalten.
PS C:\Users\admin> new-eventlog -LogName DerWindowsPapst -Source Test,Sonstiges,Routine
PS C:\Users\admin>
```

Danach öffnen wir die Ereignisanzeige und finden unter **Anwendungs- und Dienstprotokolle** unser eigenes Event-Log wieder.



Nachdem wir unser individuelles Event-log erstellt haben, werden wir es noch limitieren (512kb Größe, alte Einträge werden überschrieben und eine Vorhaltezeit von 180 Tagen). Dazu führen wir diesen Befehl aus:

Limit-EventLog -LogName DerWindowsPapst -MaximumSize 524288 -OverFlowAction OverwriteOlder -RetentionDays 180



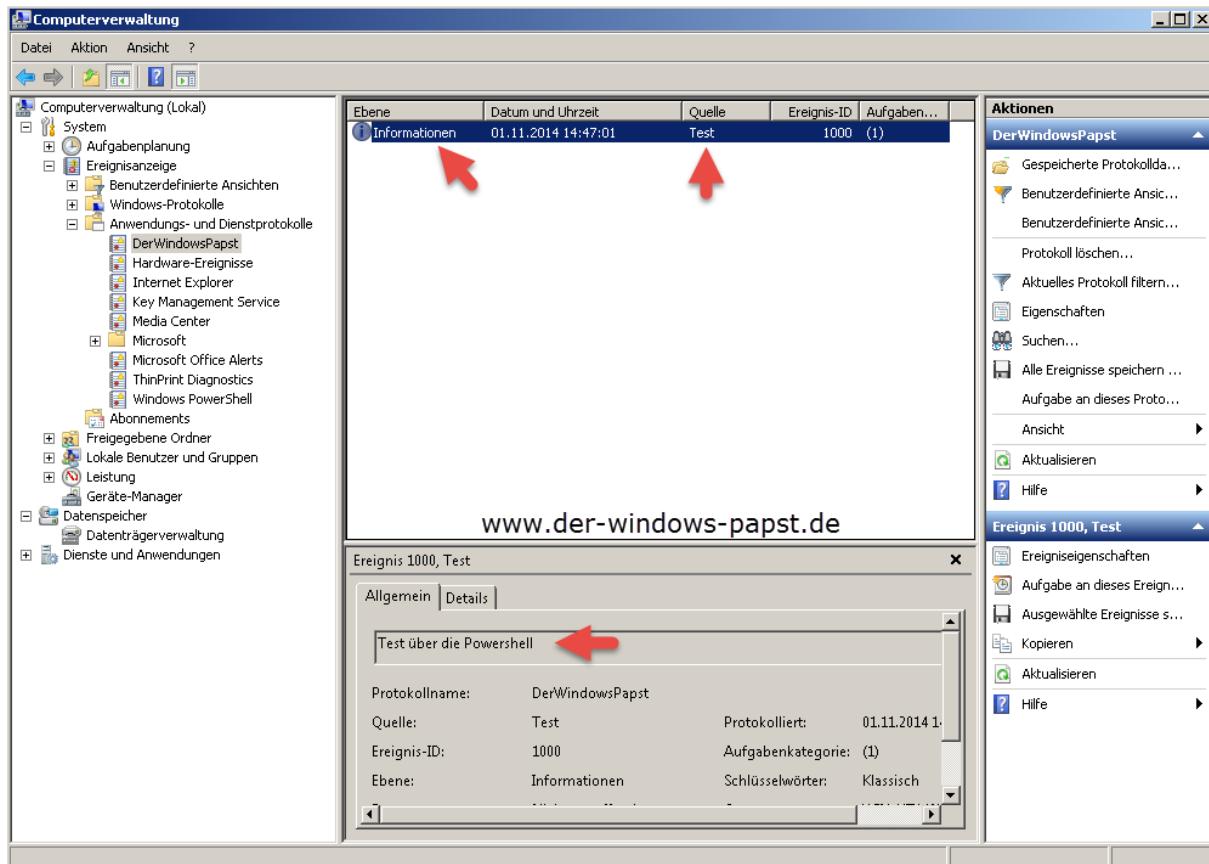
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright <C> 2009 Microsoft Corporation. Alle Rechte vorbehalten.
PS C:\Users\admin> new-eventlog -LogName DerWindowsPapst -Source Test,Sonstiges,Routine
PS C:\Users\admin> Limit-EventLog -LogName DerWindowsPapst -MaximumSize 524288 -OverFlowAction OverwriteOlder -RetentionDays 180
PS C:\Users\admin>
```

Mit folgendem Befehl werden wir unseren ersten Eintrag in unser Event-Log schreiben.

Write-EventLog DerWindowsPapst -Source Test -eventID 1000 -Message "Test über die Powershell"

Windows Event-Logs per Batch und Powershell erstellen

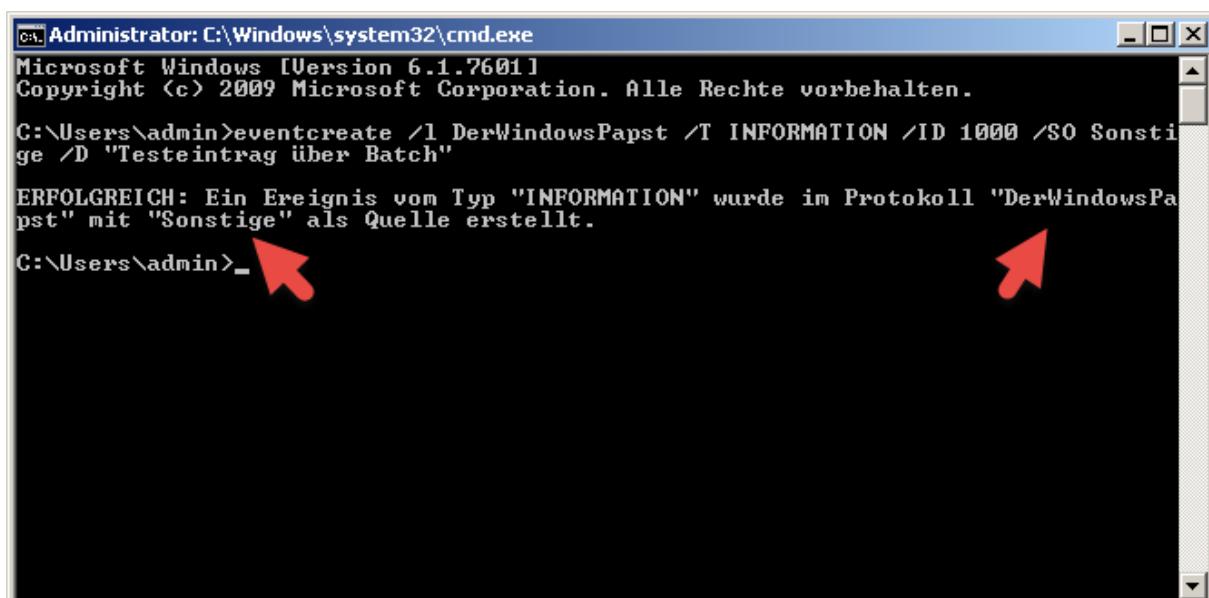
Das Ergebnis sieht dann so aus:



Nun werde ich über eine Batchzeile einen Eintrag in unser Event-Log „DerWindowsPapst“ erstellen.

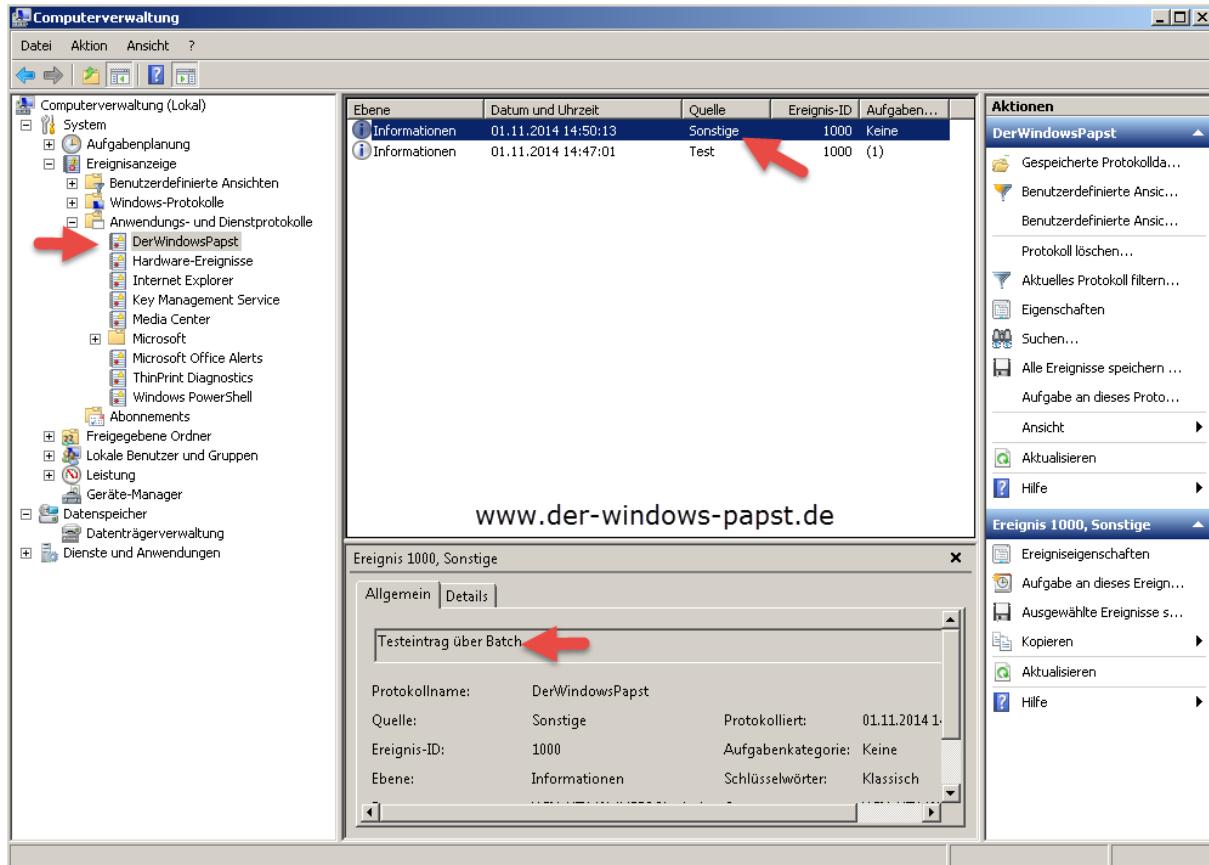
Dazu führe ich folgende Befehlszeile aus:

```
eventcreate /l DerWindowsPapst /T INFORMATION /ID 1000 /SO Sonstige /D "Testeintrag über Batch"
```



Windows Event-Logs per Batch und Powershell erstellen

Das Ergebnis sieht dann so aus:



Wenn wir uns jetzt über die Powershell unsere Event-Logs anzeigen lassen möchten gehen wir wie folgt vor.

Wir Listet die Event-Logs mit dem Befehl **get-eventlog -list** auf.

```
Administrator: Windows PowerShell
PS C:\Users\admin> get-eventlog -list
Max<(K) Retain OverflowAction      Entries Log
20.480    0 OverwriteAsNeeded     1.808 Application
512      180 OverwriteOlder       2 DerWindowsPapst
20.480    0 OverwriteAsNeeded     0 HardwareEvents
512      7 OverwriteOlder        0 Internet Explorer
20.480    0 OverwriteAsNeeded     0 Key Management Service
8.192      0 OverwriteAsNeeded     0 Media Center
128      0 OverwriteAsNeeded      0 Alerts
20.480    0 OverwriteAsNeeded     2.360 Security
20.480    0 OverwriteAsNeeded     3.866 System
512      0 OverwriteAsNeeded      11 ThinPrint Diagnostics
15.360    0 OverwriteAsNeeded      9 Windows PowerShell

PS C:\Users\admin> get-eventlog -newest 5 -logname DerWindowsPapst
Index Time          EntryType   Source           InstanceID Message
2 Nov 01 14:50  Information  Sonstige          1000 Testeintrag über Batch
1 Nov 01 14:47  Information  Test             1000 Test über die Powershell

PS C:\Users\admin>
```

Und filtern die Anzeige der neuesten 5 Einträge bezogen auf den Logname „DerWindowsPapst“ mit folgendem Befehl.

Windows Event-Logs per Batch und Powershell erstellen

Get-eventlog -newest 5 -logname DerWindowsPapst

```
Administrator: Windows PowerShell
PS C:\Users\admin> get-eventlog -list
Max<(K) Retain OverflowAction      Entries Log
20.480    0 OverwriteAsNeeded     1.808 Application
512       180 OverwriteOlder      2 DerWindowsPapst
20.480    0 OverwriteAsNeeded     0 HardwareEvents
512       7 OverwriteOlder       0 Internet Explorer
20.480    0 OverwriteAsNeeded     0 Key Management Service
8.192     0 OverwriteAsNeeded     0 Media Center
128      0 OverwriteAsNeeded     0 Alerts
20.480    0 OverwriteAsNeeded     2.360 Security
20.480    0 OverwriteAsNeeded     3.866 System
512      0 OverwriteAsNeeded     11 ThinPrint Diagnostics
15.360   0 OverwriteAsNeeded     9 Windows PowerShell

PS C:\Users\admin> get-eventlog -newest 5 -logname DerWindowsPapst
Index Time          EntryType  Source           InstanceID Message
----- ----          -----    -----           -----      -----
2 Nov 01 14:50  Information Sonstige          1000 Testeintrag über Batch
1 Nov 01 14:47  Information Test             1000 Test über die Powershell

PS C:\Users\admin>
```

Zur detaillierten Einsicht in einen Event führen wir folgenden Befehl aus:

```
$a = get-eventlog -log DerWindowsPapst -newest 1
$a | format-list -property *
```

Mit dem ersten Befehl wird das neueste Ereignis aus dem Systemereignisprotokoll abgerufen in der Variablen "\$a" gespeichert.

Beim zweiten Befehl wird das Ereignis in "\$a" mit einem Pipelineoperator (|) an den Befehl "Format-List" gesendet, der alle (*) der Ereigniseigenschaften anzeigt.

```
Administrator: Windows PowerShell
PS C:\Users\admin> $a = get-eventlog -log DerWindowsPapst -newest 1
PS C:\Users\admin> $a | format-list -property *
EventID : 1000
MachineName : WIN-UT140MHFP2Q
Data : {}
Index : 2
Category : <0>
CategoryNumber : 0
EntryType : Information
Message : Testeintrag über Batch
Source : Sonstige
ReplacementStrings : {Testeintrag über Batch}
InstanceId : 1000
TimeGenerated : 01.11.2014 14:50:13
TimeWritten : 01.11.2014 14:50:13
UserName : WIN-UT140MHFP2Q\admin
Site :
Container :

PS C:\Users\admin>
```