

# Integrate Windows PowerShell

*EventTracker Enterprise*

# Abstract

This guide provides instructions to enable Microsoft PowerShell logging for EventTracker.

## Scope

The configurations detailed in this guide are consistent with **EventTracker Enterprise** version 7.X and later, and **PowerShell 3.0 and later**.

## Audience

Administrators, who wish to monitor PowerShell command or script execution using EventTracker.

*The information contained in this document represents the current view of Prism Microsystems Inc. on the issues discussed as of the date of publication. Because Prism Microsystems must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Prism Microsystems, and Prism Microsystems cannot guarantee the accuracy of any information presented after the date of publication.*

*This document is for informational purposes only. Prism Microsystems MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, this paper may be freely distributed without permission from Prism, as long as its content is unaltered, nothing is added to the content and credit to Prism is provided.*

*Prism Microsystems may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Prism Microsystems, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.*

*© 2016 Prism Microsystems Corporation. All rights reserved. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

# Table of Contents

Abstract.....	1
Scope.....	1
Audience.....	1
What is PowerShell.....	4
Enable PowerShell logging.....	4
Enable PowerShell logging .....	4
Configure Event viewer.....	8
Configure EventTracker Event Filter .....	11
Event ID - <b>4103</b> .....	13
Event ID - <b>4100</b> .....	13
Event ID - <b>6</b> .....	14
Event ID - <b>8</b> .....	14
Event ID - <b>161</b> .....	15
Event ID - <b>169</b> .....	15
EventTracker Knowledge Pack (KP).....	16
Reports.....	17
Alerts.....	17
Filter .....	17
Import Windows PowerShell Knowledge Pack into EventTracker .....	18
Import Parsing Rules.....	18
Import Alerts.....	19
Import Flex Reports.....	21
Import Filters .....	22
Verify Windows PowerShell knowledge pack in EventTracker.....	23
Verify Parsing Rules .....	23
Verify Alerts .....	24
Verify Flex Reports.....	24
Verify Event Filters .....	25
Create Dashboards in EventTracker.....	26

Schedule Reports..... 26

Create Dashlets..... 29

Sample Dashboards..... 33

Sample Reports ..... 34

# What is PowerShell?

**Windows PowerShell** is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework. PowerShell comes in two versions: Console and Integrated Scripting Environment (ISE). PowerShell also features SSH like remote shell capability through **Windows Remote Management** (WinRM).

EventTracker amasses and examines logs generated by PowerShell to help an administration to monitor remote session's establishment and execution of rogue scripts or commands.

## Enable PowerShell logging

### Enable PowerShell logging

1. Open **Group Policy Editor** in Windows.

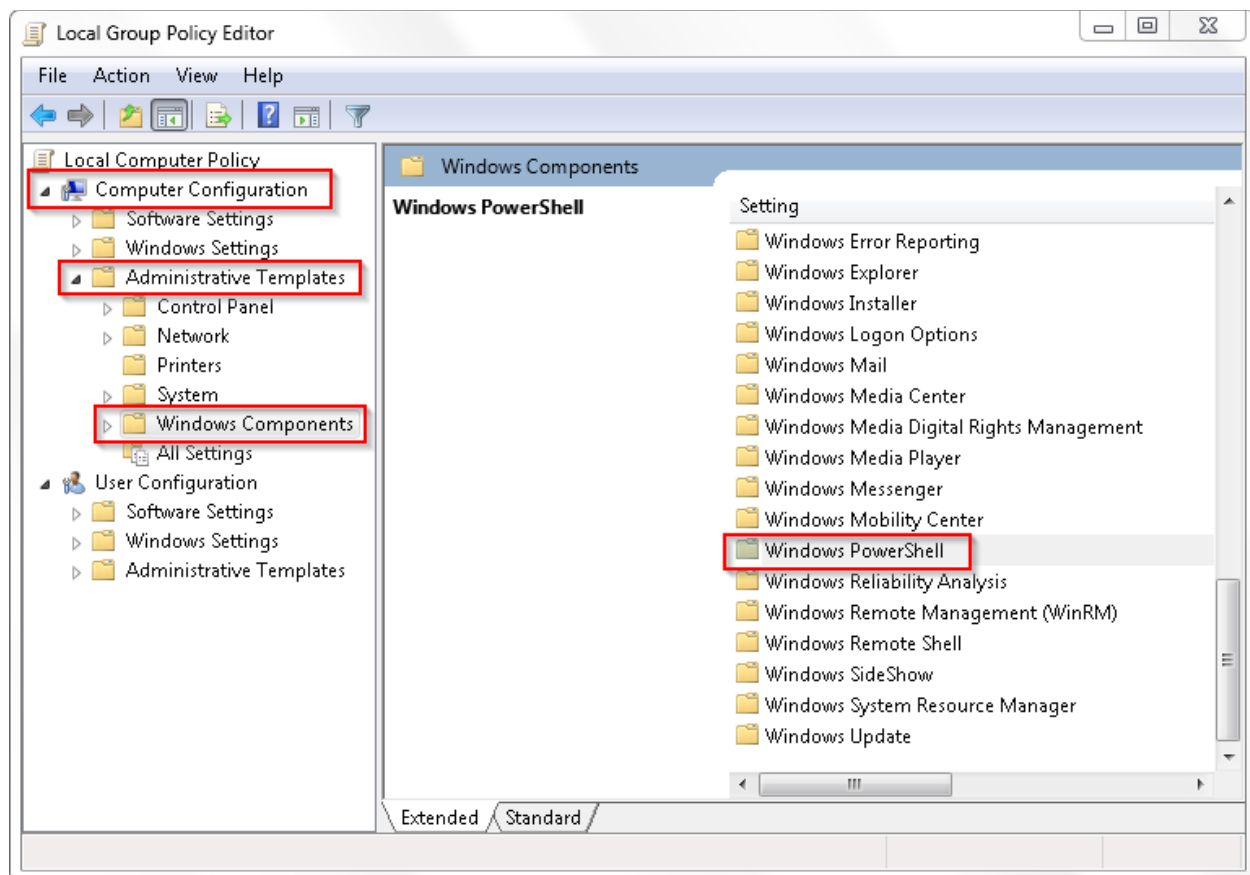


Figure 1

2. Navigate to **Computer Configuration>Administrative Templates>Windows Components>Windows PowerShell**.

For **PowerShell 3.0 and 4.0**, settings are shown as follows:

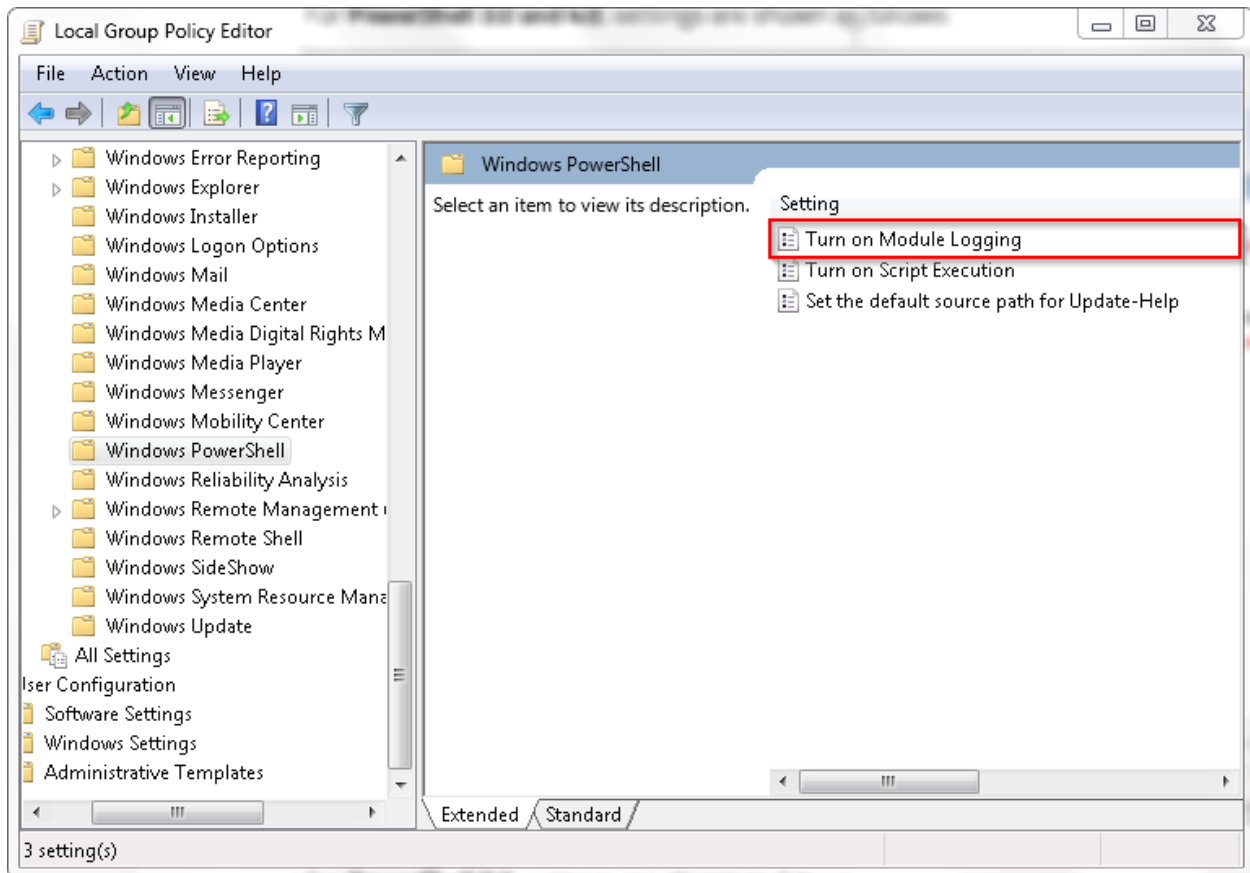


Figure 2

For **PowerShell 5.0**, settings are shown as follows:

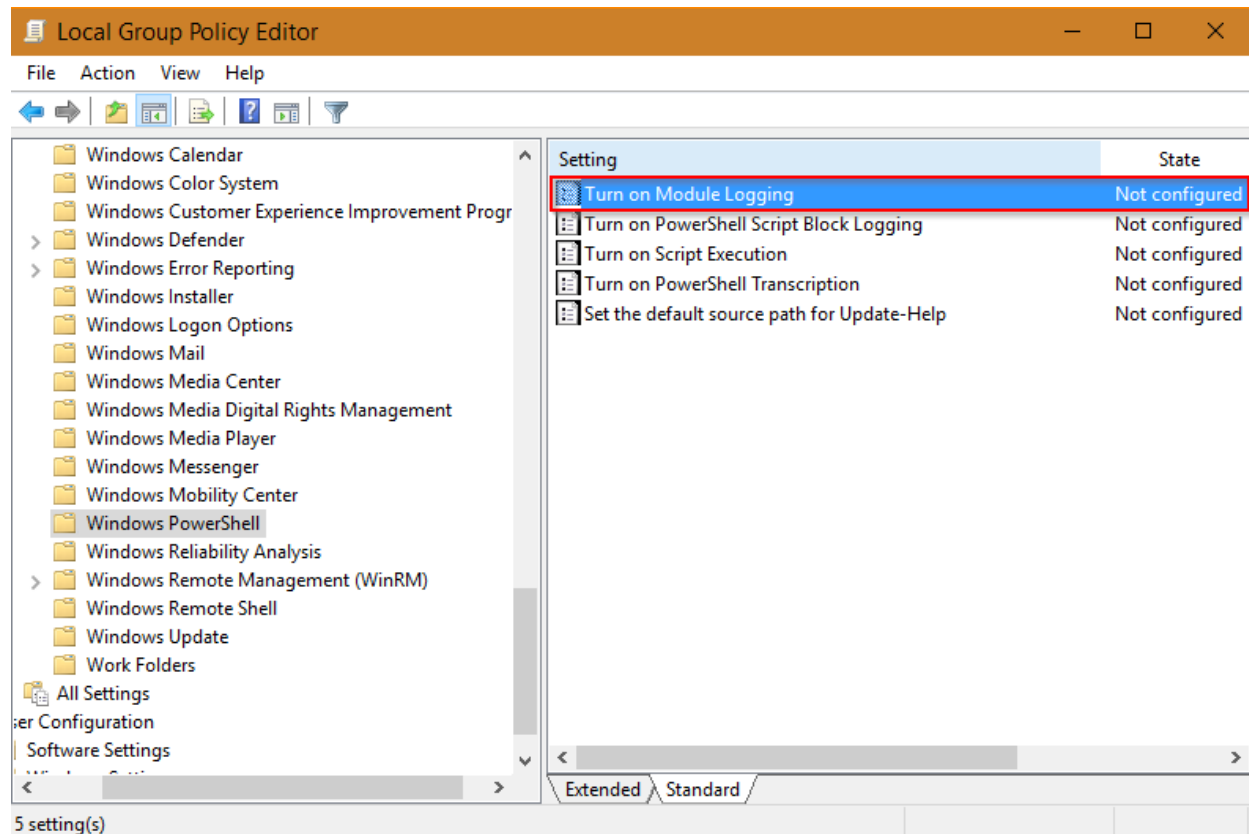


Figure 3

3. Click **Turn on Module Logging** setting.

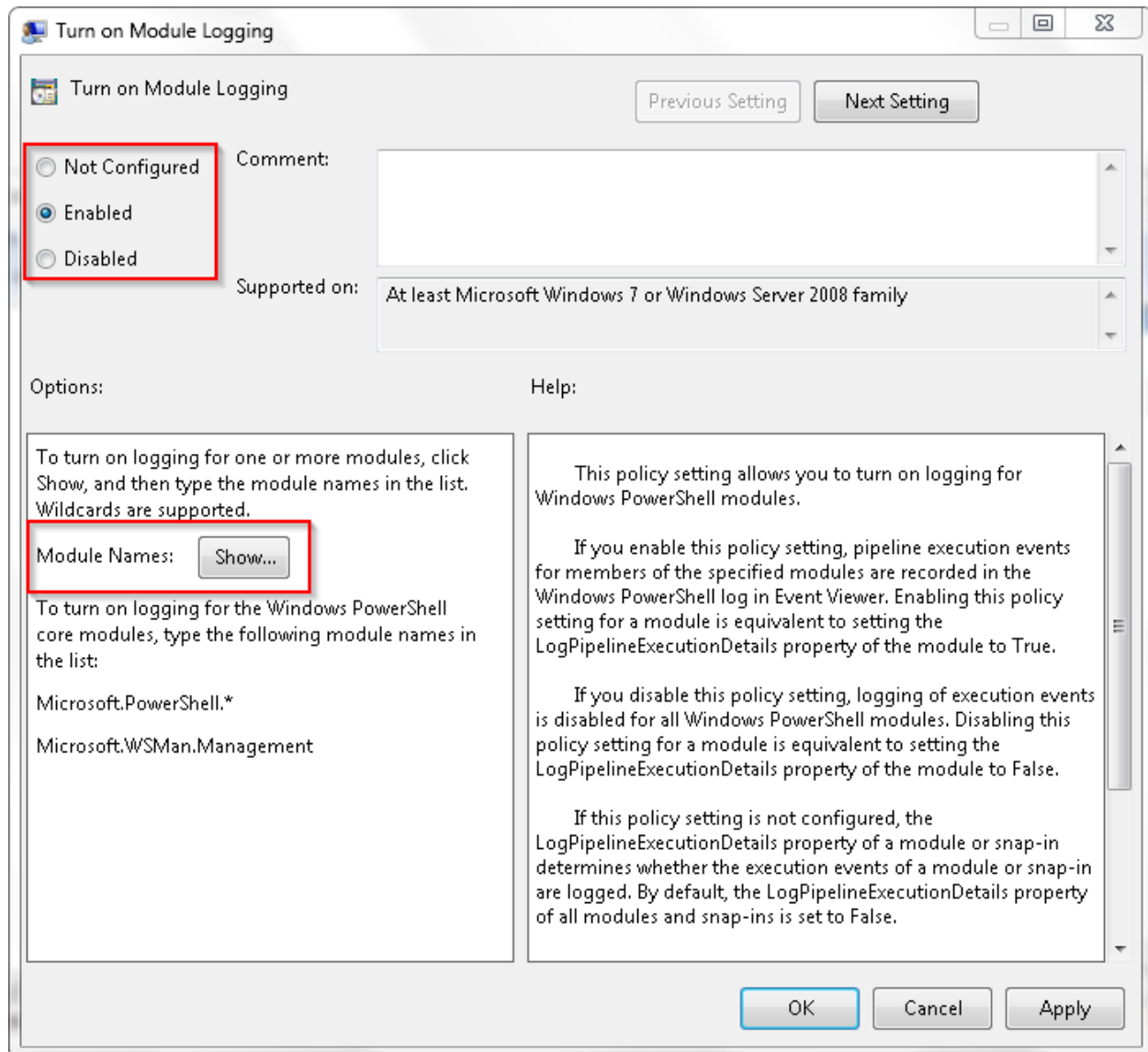


Figure 4

3. Select **Enabled**.
4. In **Module Names** section, select **Show** to enable logging for selected modules.
5. Configure **Module Names** as shown below.



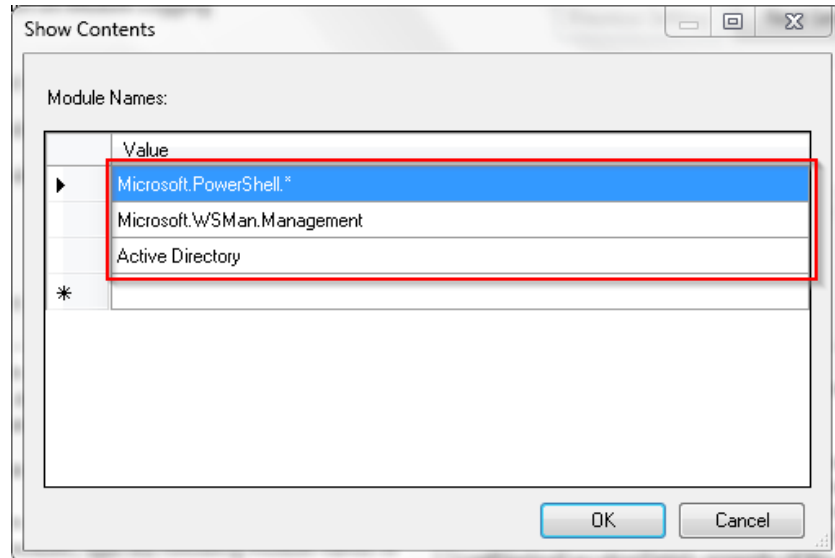


Figure 5

6. Select **OK** and **Apply** to save the changes.

**NOTE-**

- It is not advised to enable script logging options as it might result into high log volume.
- Select value as '\*' in Module Names pane to enable logging for all available modules.

## Configure Event viewer

1. Open **Event Viewer** in Windows.

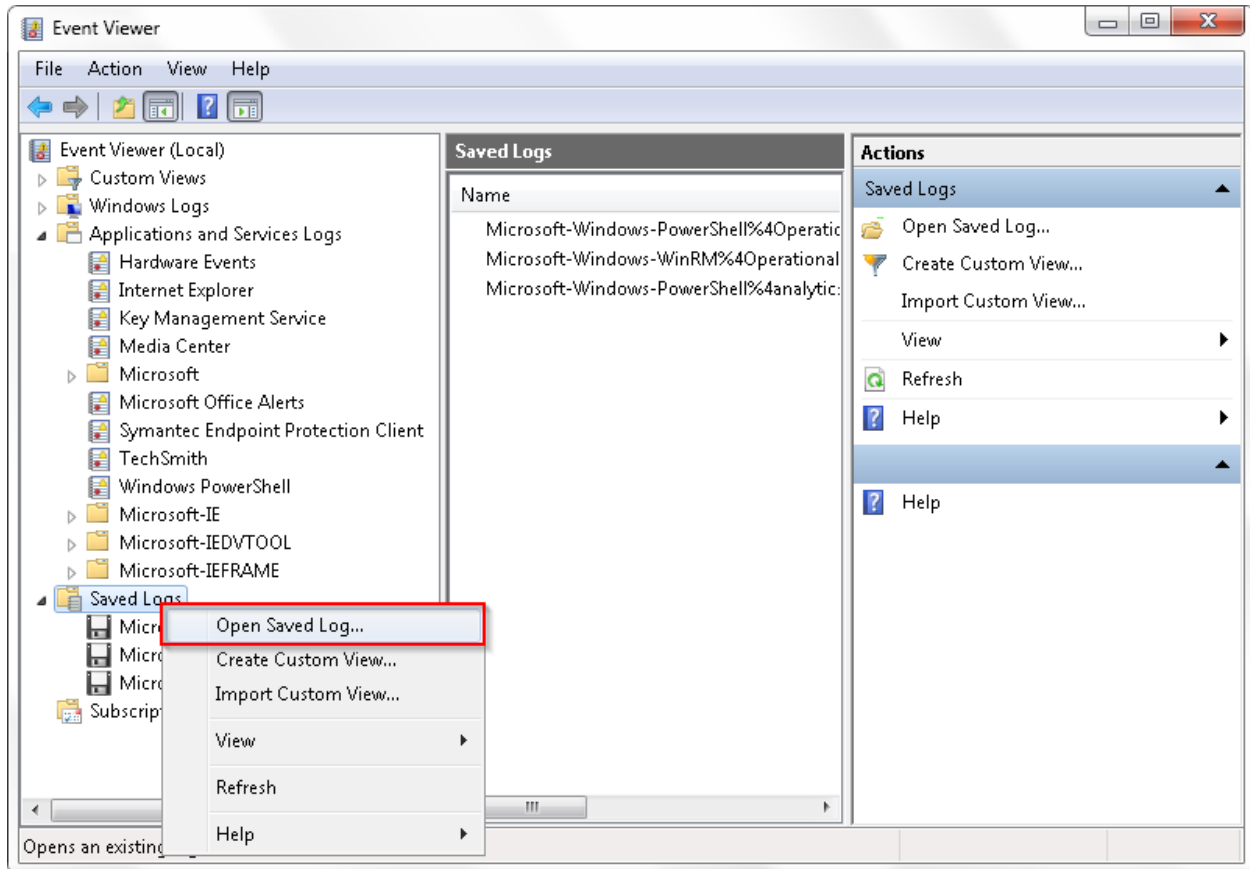


Figure 6

2. Right-click **Saved Logs** and select **Open Saved Log...** option.
3. Navigate to **C:\Windows\System32\winevt\Logs** and select following logs.
  - a. **Microsoft-Windows-PowerShell%4Operational.evtx**
  - b. **Microsoft-Windows-WinRM%4Operational.evtx**

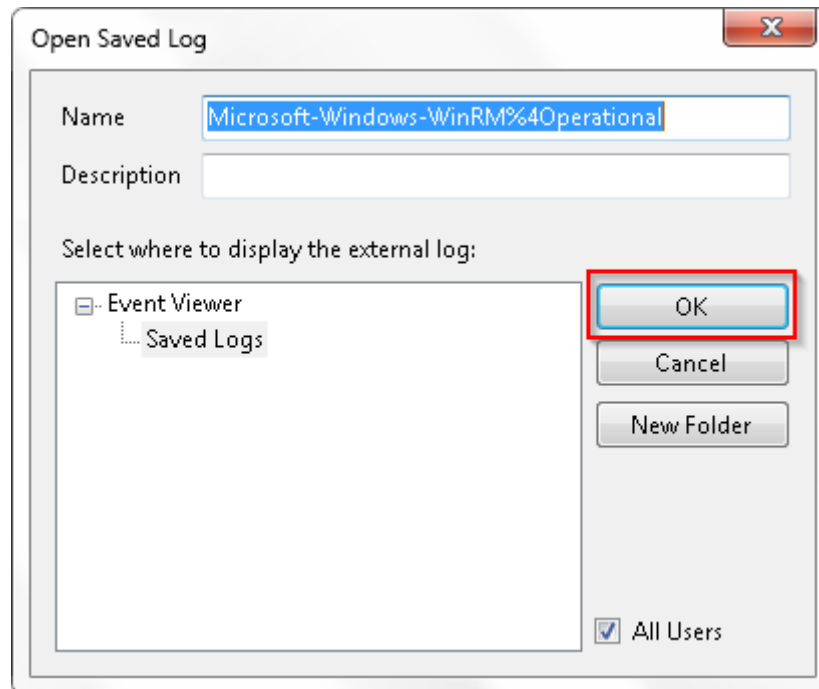


Figure 7

4. For both log types, compose **Open Saved Log** dialog settings per convenience.
5. Select **OK** to confirm.

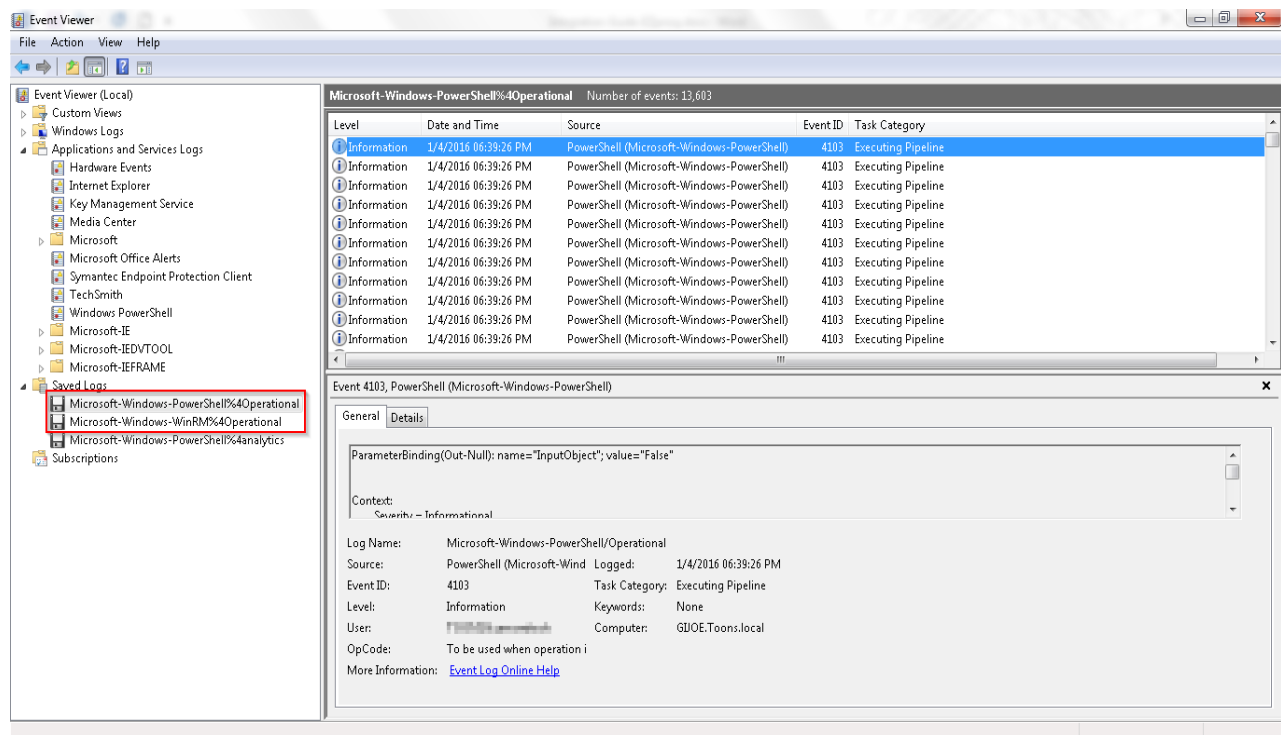


Figure 8

6. PowerShell and WinRM logs can be observed in the right pane.

## Configure EventTracker Event Filter

1. Launch **EventTracker Control Panel**.

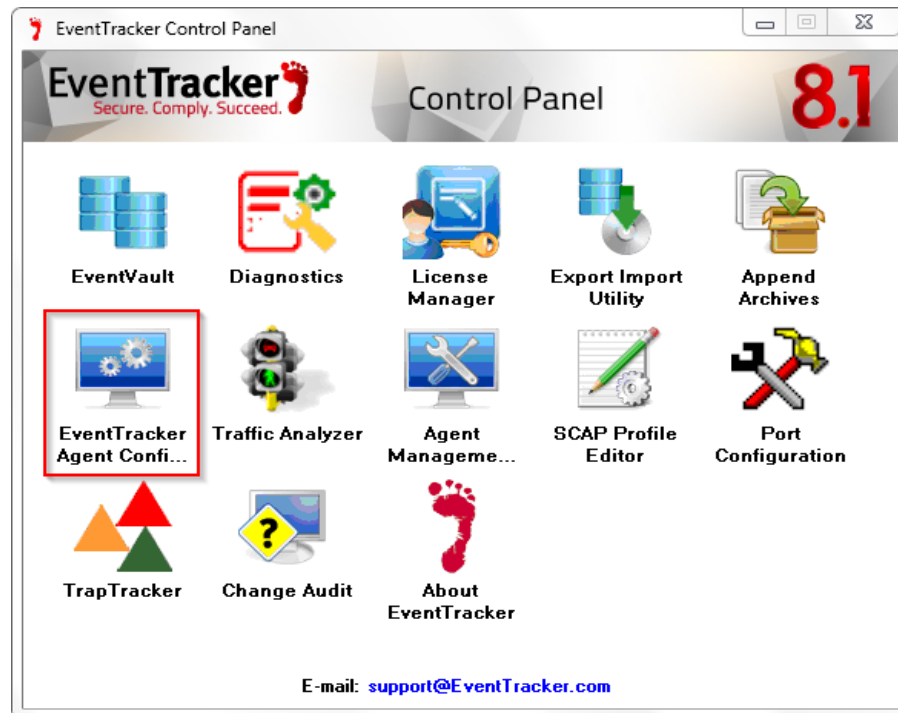


Figure 9

2. Double click **EventTracker Agent Configuration**.

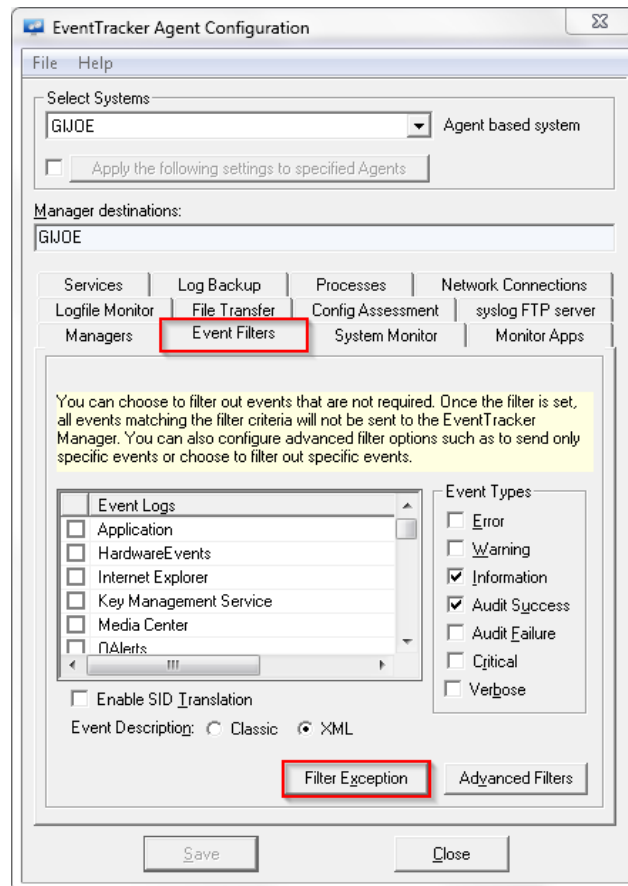


Figure 10

3. Navigate to **Event Filters>Filter Exception**.

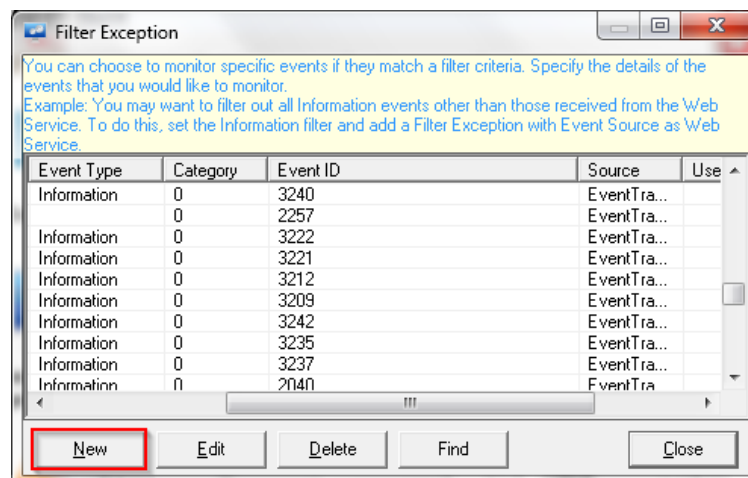


Figure 11

4. Click **New** and compose **Edit Event Details**. Configure settings for relevant events as shown below.

## Event ID - 4103

The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-PowerShell/Operational'. The 'Event Type' dropdown is set to 'Information'. The 'Event ID' text box contains '4103'. The 'Category' text box contains '0'. The 'Match in User', 'Match in Source', and 'Match in Event Descr' text boxes are empty. A yellow information box at the bottom explains the 'Match in Event Descr' field syntax, including AND/OR conditions and negation. The 'OK' button is highlighted with a red box.

Event Details (empty field implies all matches)

Log Type :  
Microsoft-Windows-PowerShell/Operational

Event Type :  
Information

Event ID :  
4103

Category :  
0

Match in User :

Match in Source :

Match in Event Descr :

"Match in Event Descr" field can take multiple strings separated with && or ||.  
- && stands for AND condition. - || stands for OR condition.  
For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string.  
Example:  
The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description.  
[For more information click here.](#)

OK Cancel

Figure 12

## Event ID - 4100

The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-PowerShell/Operational'. The 'Event Type' dropdown is set to 'Warning'. The 'Event ID' text box contains '4100'. The 'Category' text box contains '0'. The 'Match in User', 'Match in Source', and 'Match in Event Descr' text boxes are empty. A yellow information box at the bottom explains the 'Match in Event Descr' field syntax, including AND/OR conditions and negation. The 'OK' button is highlighted with a red box.

Event Details (empty field implies all matches)

Log Type :  
Microsoft-Windows-PowerShell/Operational

Event Type :  
Warning

Event ID :  
4100

Category :  
0

Match in User :

Match in Source :

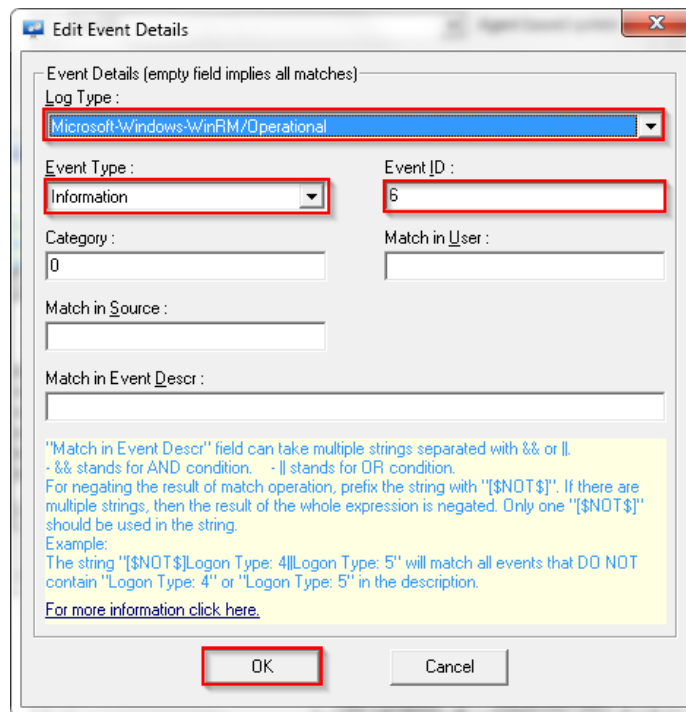
Match in Event Descr :

"Match in Event Descr" field can take multiple strings separated with && or ||.  
- && stands for AND condition. - || stands for OR condition.  
For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string.  
Example:  
The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description.  
[For more information click here.](#)

OK Cancel

Figure 13

## Event ID - 6



The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-WinRM/Operational'. The 'Event Type' dropdown is set to 'Information'. The 'Event ID' text box contains the number '6'. The 'Category' text box contains '0'. The 'Match in User', 'Match in Source', and 'Match in Event Descr' text boxes are empty. A yellow informational box at the bottom contains text about match operations and a link to 'For more information click here.' The 'OK' and 'Cancel' buttons are at the bottom.

Event Details (empty field implies all matches)

Log Type :  
Microsoft-Windows-WinRM/Operational

Event Type :  
Information

Event ID :  
6

Category :  
0

Match in User :

Match in Source :

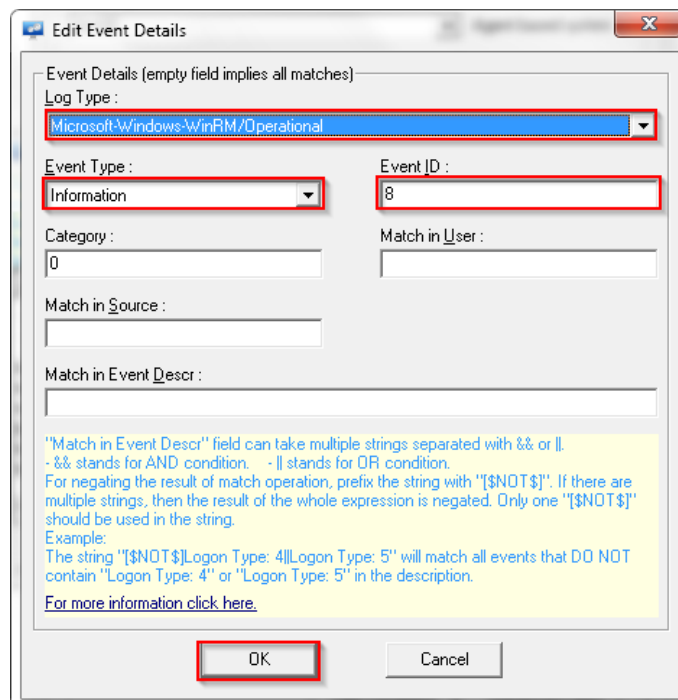
Match in Event Descr :

"Match in Event Descr" field can take multiple strings separated with && or ||.  
- && stands for AND condition. - || stands for OR condition.  
For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string.  
Example:  
The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description.  
[For more information click here.](#)

OK Cancel

Figure 14

## Event ID - 8



The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-WinRM/Operational'. The 'Event Type' dropdown is set to 'Information'. The 'Event ID' text box contains the number '8'. The 'Category' text box contains '0'. The 'Match in User', 'Match in Source', and 'Match in Event Descr' text boxes are empty. A yellow informational box at the bottom contains text about match operations and a link to 'For more information click here.' The 'OK' and 'Cancel' buttons are at the bottom.

Event Details (empty field implies all matches)

Log Type :  
Microsoft-Windows-WinRM/Operational

Event Type :  
Information

Event ID :  
8

Category :  
0

Match in User :

Match in Source :

Match in Event Descr :

"Match in Event Descr" field can take multiple strings separated with && or ||.  
- && stands for AND condition. - || stands for OR condition.  
For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string.  
Example:  
The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description.  
[For more information click here.](#)

OK Cancel

Figure 15

## Event ID - 161

The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-WinRM/Operational'. The 'Event Type' dropdown is set to 'Error'. The 'Event ID' text box contains '161'. The 'Category' text box contains '0'. The 'Match in User' and 'Match in Source' text boxes are empty. The 'Match in Event Descr' text box is empty. A yellow information box at the bottom contains the following text: "Match in Event Descr" field can take multiple strings separated with && or ||. - && stands for AND condition. - || stands for OR condition. For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string. Example: The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description. For more information click here. The 'OK' button is highlighted with a red box.

Figure 16

## Event ID - 169

The screenshot shows the 'Edit Event Details' dialog box. The 'Log Type' dropdown is set to 'Microsoft-Windows-WinRM/Operational'. The 'Event Type' dropdown is set to 'Information'. The 'Event ID' text box contains '169'. The 'Category' text box contains '0'. The 'Match in User' and 'Match in Source' text boxes are empty. The 'Match in Event Descr' text box is empty. A yellow information box at the bottom contains the following text: "Match in Event Descr" field can take multiple strings separated with && or ||. - && stands for AND condition. - || stands for OR condition. For negating the result of match operation, prefix the string with "[NOT\$]". If there are multiple strings, then the result of the whole expression is negated. Only one "[NOT\$]" should be used in the string. Example: The string "[NOT\$]Logon Type: 4||Logon Type: 5" will match all events that DO NOT contain "Logon Type: 4" or "Logon Type: 5" in the description. For more information click here. The 'OK' button is highlighted with a red box.

Figure 17



- Review the changes and click **OK** to confirm.

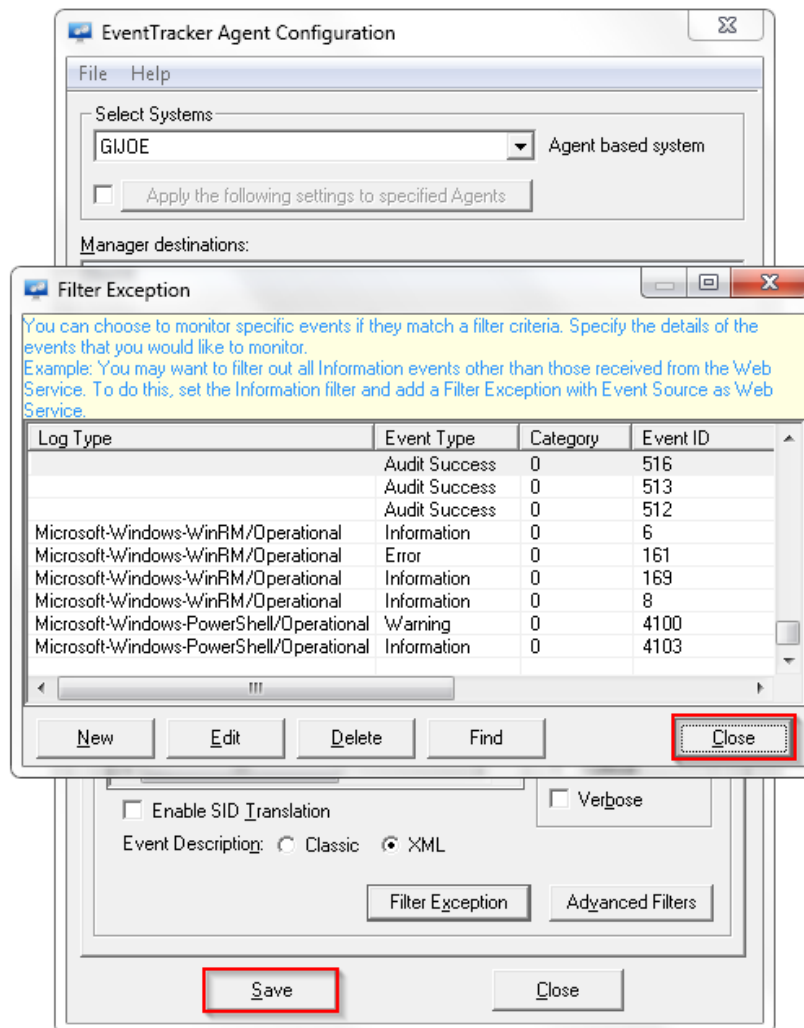


Figure 18

- Click **Close** and **Save** to apply the changes.

## EventTracker Knowledge Pack (KP)

Once logs are received in to EventTracker, Alerts, Reports and Dashboards can be configured into EventTracker. The following Knowledge Packs are available in EventTracker to support Windows PowerShell monitoring.

## Reports

1. **Windows PowerShell-Command execution details**– This report provides information related to command execution on PowerShell which includes User Name, Host Type, Command Executed and Command Parameters fields.
2. **Windows PowerShell-Script execution details**– This report provides information related to command execution through script on PowerShell which includes User Name, Host Type, Script Path, Command Executed and Command Parameters fields.
3. **Windows PowerShell-Command execution error details**– This report provides information related to command execution errors by script or CLI on PowerShell which includes User Name, Host Type, Script Path, Command Executed and Command Parameters fields.
4. **Windows PowerShell-Remote session creation details**– This report provides information related to PowerShell remote session initialization which includes Computer, User Name and Remote Host fields.
5. **Windows PowerShell-Remote session authentication success details**– This report provides information related to successful PowerShell remote session authentication which includes Computer, Remote User Name and Authentication Method fields.
6. **Windows PowerShell-Remote session authentication failure details**– This report provides information related to unsuccessful PowerShell remote session authentication which includes Computer, Event User and Reason fields.

## Alerts

1. **Windows PowerShell: Command execution failed**– This alert is generated when command execution on PowerShell fails.
2. **Windows PowerShell: Remote session initiated**– This alert is generated when PowerShell remote session is initialized.
3. **Windows PowerShell: Remote session user authentication failed**– This alert is generated when PowerShell user authentication fails.

## Filter

1. **Windows PowerShell-EventTracker script filter**– This filter excludes events generated by EventTracker scripts.

# Import Windows PowerShell Knowledge Pack into EventTracker

1. Launch **EventTracker Control Panel**.
2. Double click **Import Export Utility**, and then click the **Import** tab.

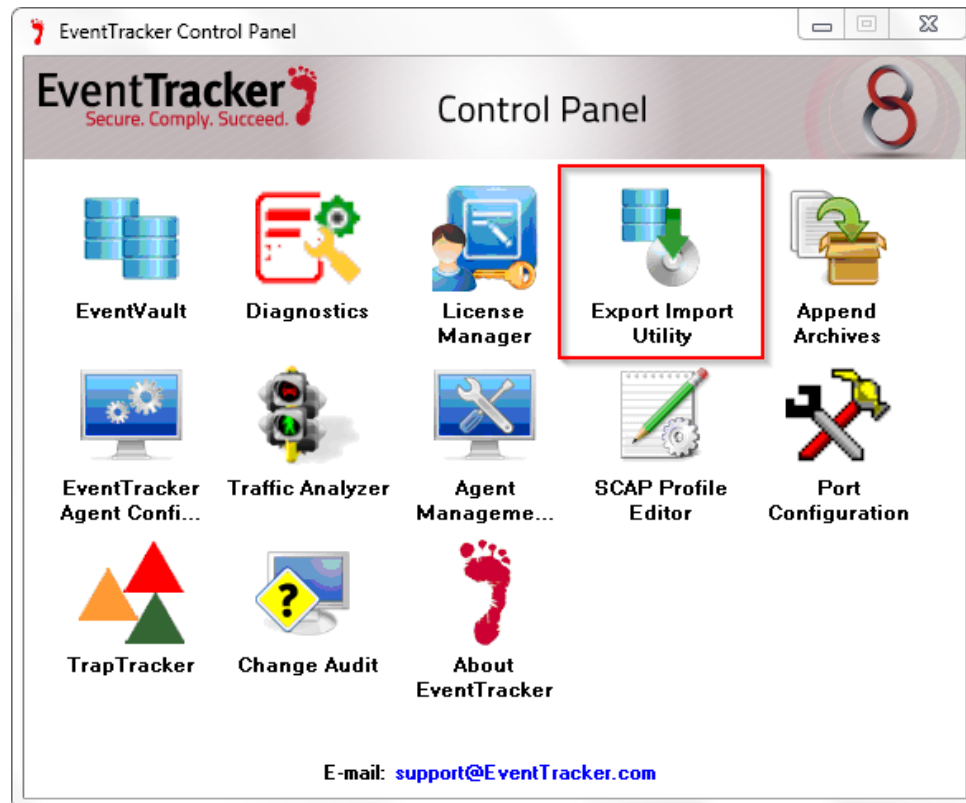



Figure 19

Import **Parsing Rules, Alerts, Reports and Filter** as given below.

## Import Parsing Rules

1. Click **Token Value** option, and then click the browse  button.
2. Locate **All Windows PowerShell** group of **tokens.istoken** file, and then click the **Open** button.

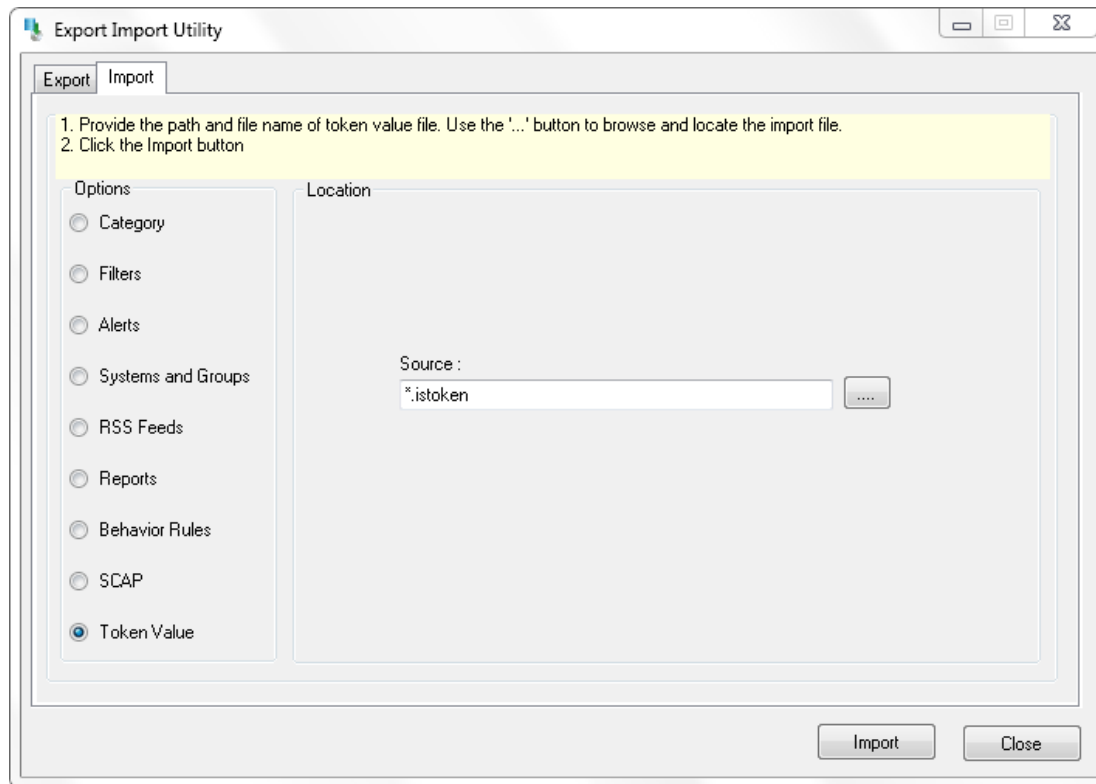


Figure 20

3. To import token value, click the **Import** button.

EventTracker displays success message.

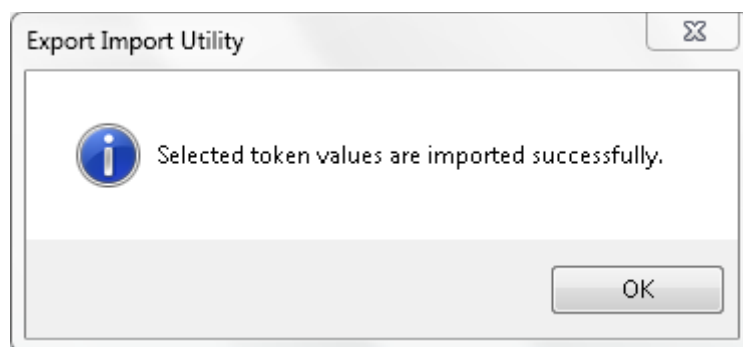



Figure 21

4. Click **OK**, and then click the **Close** button.

## Import Alerts

1. Click **Alerts** option, and then click the '**browse**'  button.

2. Locate **All Windows PowerShell group alerts.isalt** file, and then click the **Open** button.

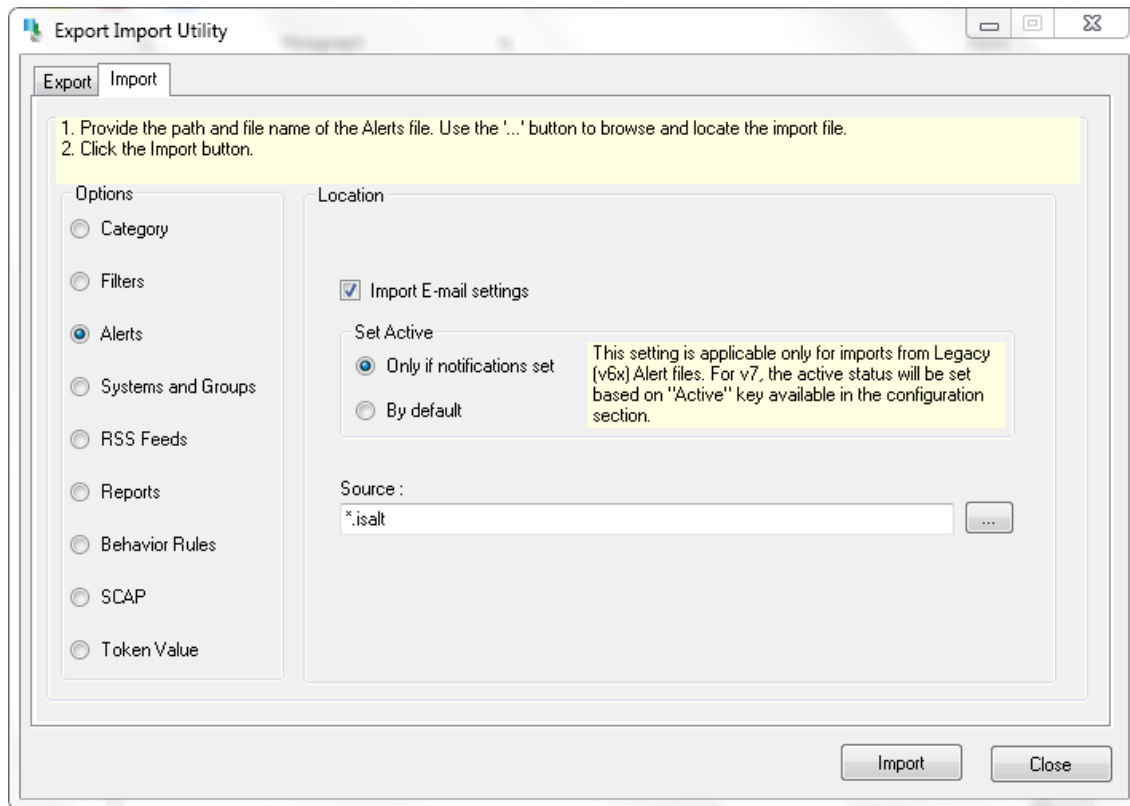


Figure 22

3. To import alerts, click the **Import** button.

EventTracker displays success message.

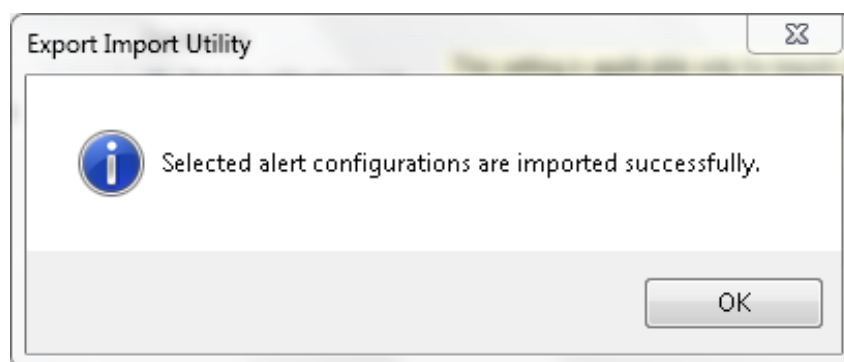



Figure 23

4. Click **OK**, and then click the **Close** button.

## Import Flex Reports

1. Click **Reports** option, and then click the 'browse'  button.
2. Locate **All Windows PowerShell group reports.issch** file, and then click the **Open** button.

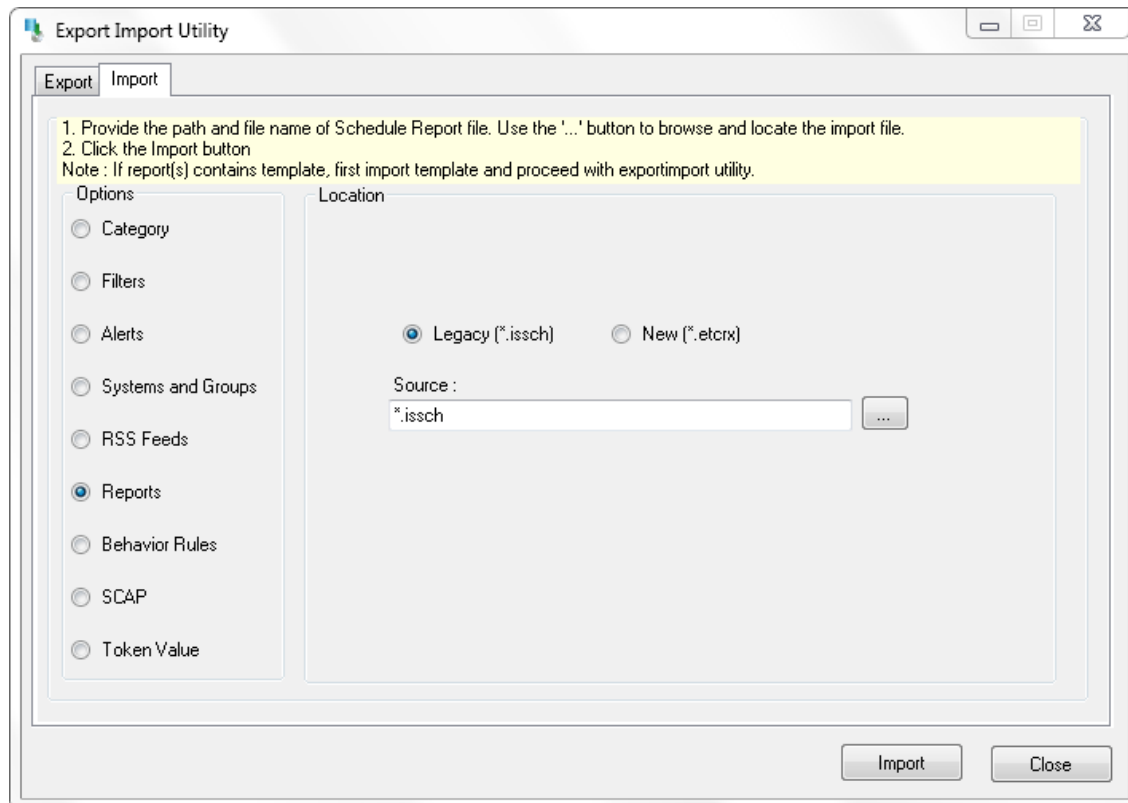


Figure 24

3. To import scheduled reports, click the **Import** button.

EventTracker displays success message.

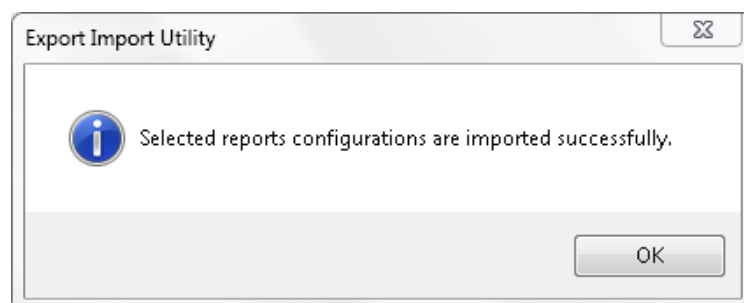



Figure 25

4. Click **OK**, and then click the **Close** button.

## Import Filters

1. Click **Reports** option, and then click the 'browse'  button.
2. Locate **Windows PowerShell Filter.isfil** file, and then click the **Open** button.

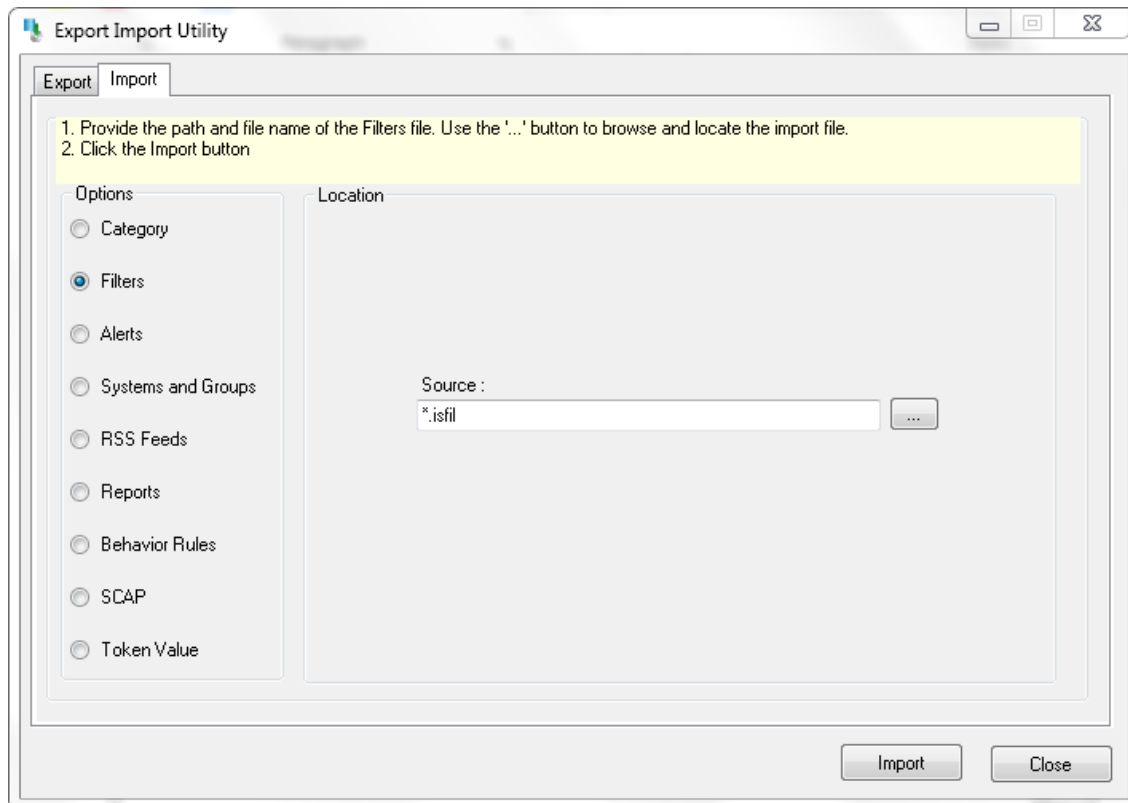


Figure 26

3. To import filters, click the **Import** button.

EventTracker displays success message.

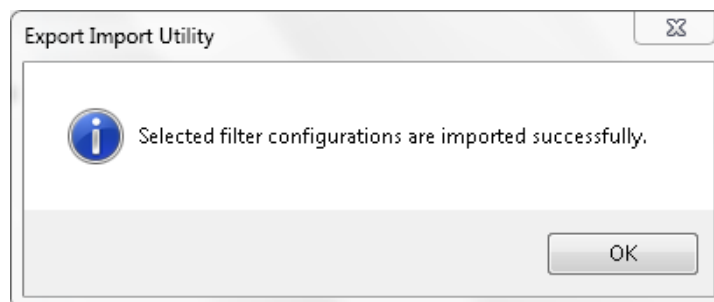


Figure 27

4. Click **OK**, and then click the **Close** button.

# Verify Windows PowerShell knowledge pack in EventTracker

## Verify Parsing Rules

1. Logon to **EventTracker Enterprise**.
2. Click the **Admin** menu, and then click **Parsing Rule**.
3. In **Token Value Group Tree** to view imported token values, scroll down and click **Windows PowerShell** group folder.

Token values are displayed in the token value pane.

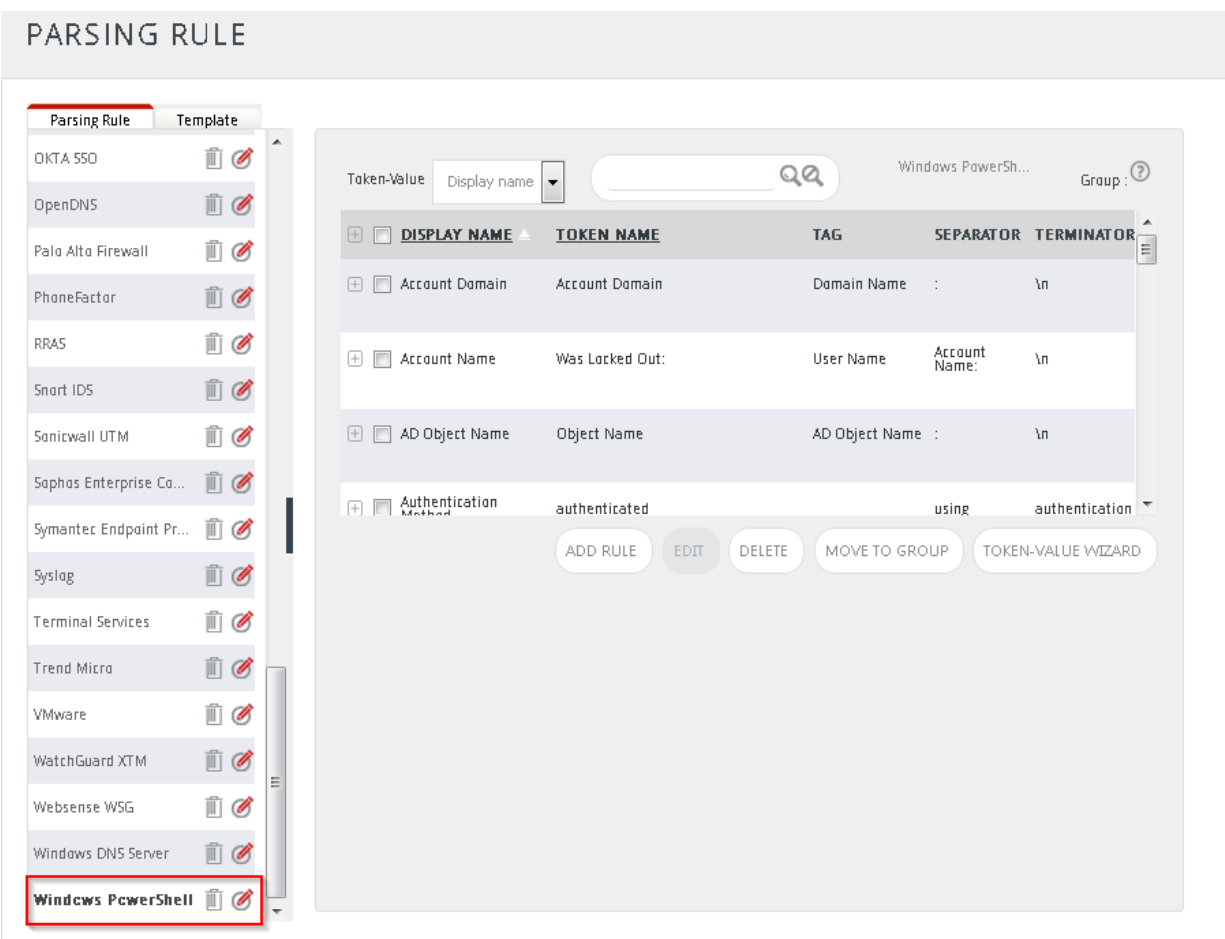



Figure 28



## Verify Alerts

1. Logon to **EventTracker Enterprise**.
2. Click the **Admin** menu, and select **Alerts**.
3. In **Search** field, type '**powershell**', and then click the  button.

Alert Management page will display all the imported PowerShell alerts.

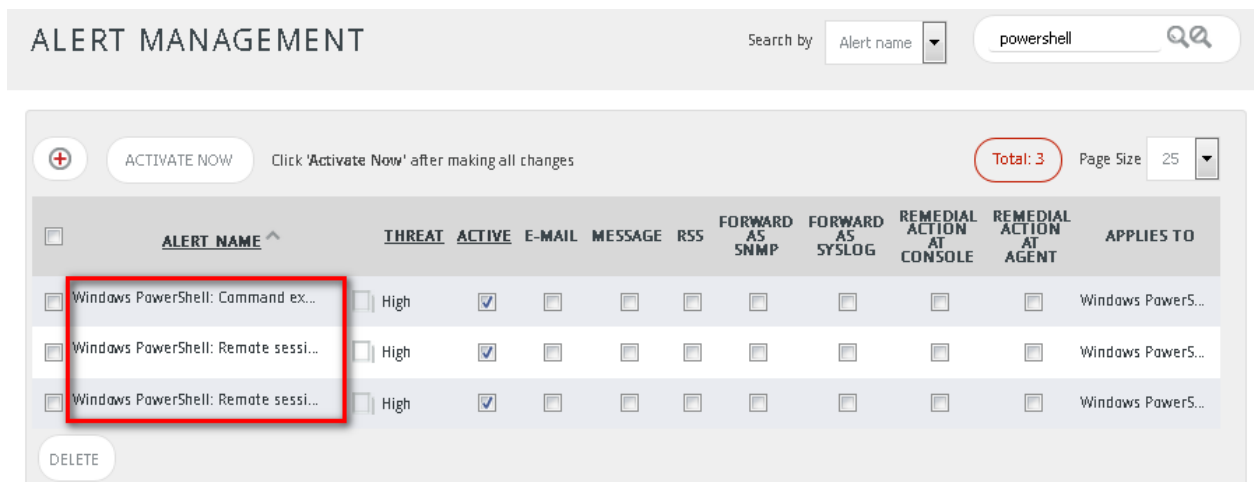


Figure 29

4. To activate the imported alerts, select the respective checkbox in the **Active** column.

EventTracker displays message box.

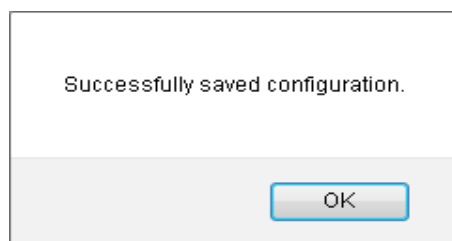


Figure 30

5. Click **OK**, and then click the **Activate Now** button.

**NOTE:** Please specify appropriate **systems** in **alert configuration** for better performance.

## Verify Flex Reports

1. Logon to **EventTracker Enterprise**.

2. Click the **Reports** menu and select **Configuration**.
3. Select **Defined** in report type.
4. To view imported flex reports In **Report Groups Tree**, scroll down and click **Windows PowerShell group** folder.

Imported reports are displayed in the Reports Configuration pane.

The screenshot displays the 'REPORTS CONFIGURATION' interface. At the top, there are tabs for 'Scheduled', 'Queued', and 'Defined', with 'Defined' being the active tab. A search bar is located to the right of these tabs. Below the tabs, the 'REPORT GROUPS' section on the left lists various categories like Security, Compliance, Operations, and Flex. The 'Flex' category is expanded, showing a list of report groups including 'A10 ADC', 'Amazon Web Services', 'Apache Web Server', 'Barracuda Message Ar...', 'Barracuda Spam Fire...', 'Barracuda SSL VPN', 'Centrify Server Suit...', and 'Check Point'. The 'Windows PowerShell' group is highlighted. The main pane shows the 'REPORTS CONFIGURATION' for 'WINDOWS POWERSHELL'. It includes a 'Total: 5' indicator and a table with columns 'TITLE', 'CREATED ON', and 'MODIFIED ON'. The table lists five reports, each with a gear icon for configuration and a plus icon for details. The reports are: 'Windows PowerShell-Remote Session authenticat...', 'Windows PowerShell-Remote session creation de...', 'Windows PowerShell-Command execution error d...', 'Windows PowerShell-Script execution details', and 'Windows PowerShell-Command execution details'.

TITLE	CREATED ON	MODIFIED ON
Windows PowerShell-Remote Session authenticat...	2/3/2016 04:48:24 PM	2/18/2016 02:09:05 PM
Windows PowerShell-Remote session creation de...	2/3/2016 04:30:24 PM	2/18/2016 02:10:14 PM
Windows PowerShell-Command execution error d...	2/3/2016 04:00:24 PM	2/3/2016 05:06:46 PM
Windows PowerShell-Script execution details	2/3/2016 03:51:02 PM	2/3/2016 05:05:32 PM
Windows PowerShell-Command execution details	2/3/2016 03:43:16 PM	2/3/2016 05:04:36 PM

Figure 31

## Verify Event Filters

1. Logon to **EventTracker Enterprise**.
2. Click the **Admin** menu, and select **Event Filters**.
3. In **Search** field, type '**powershell**', and then click the button.

Event Filters page will display all the imported PowerShell filter.

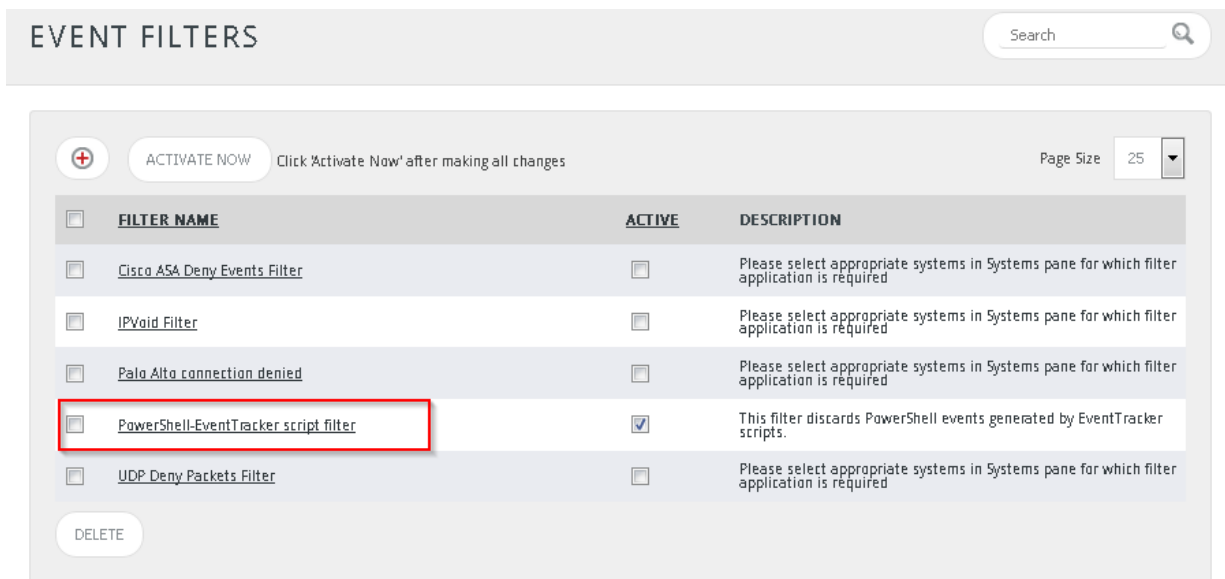


Figure 32

- To activate the imported alerts, select the respective checkbox in the **Active** column.

EventTracker displays message box.

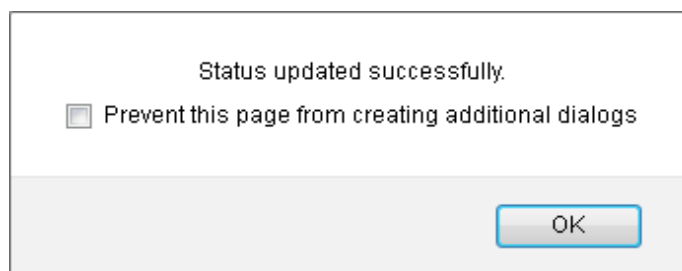


Figure 33

- Click **OK**, and then click the **Activate Now** button.

**NOTE:** Please specify appropriate **systems** in **filter wizard** for better performance.

## Create Dashboards in EventTracker

### Schedule Reports

- Open **EventTracker** in browser and login.

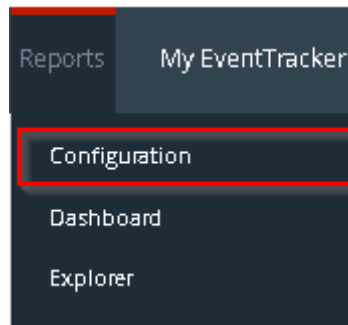


Figure 34

2. Navigate to **Reports>Configuration**.

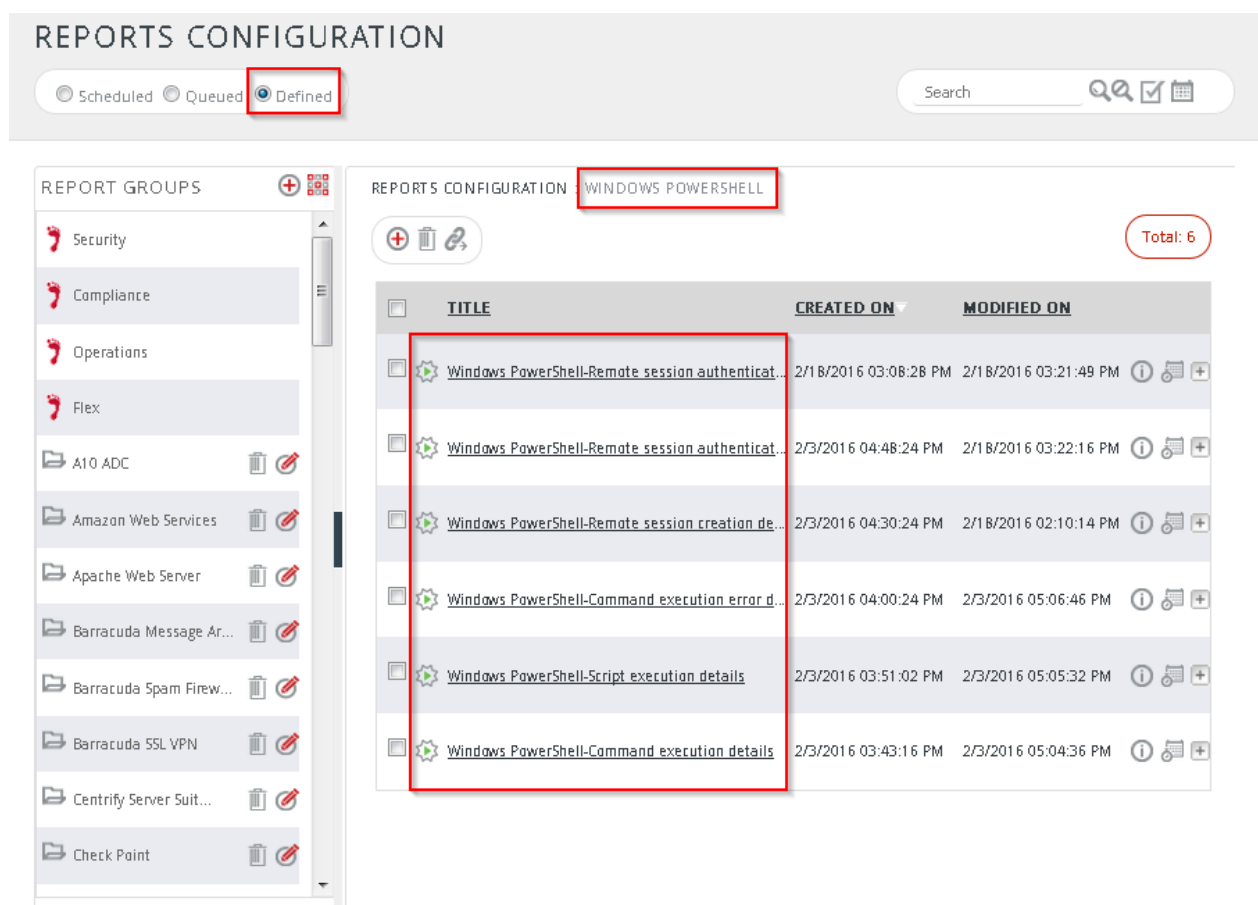



Figure 35

3. Select '**Windows PowerShell**' in report groups. Check **defined** dialog box.
4. Click on '**schedule**'  to plan a report for later execution.

### REPORT WIZARD

TITLE: WINDOWS POWERSHELL-REMOTE SESSION CREATION DETAILS

LOGS

CANCEL < BACK NEXT >

Review cost details and configure the publishing options. Step 8 of 10

#### DISK COST ANALYSIS

Estimated time for completion: 00:00:38(HH:MM:SS)  
Number of tab(s) to be processed: 4  
Available disk space: 184 GB  
Required disk space: 50 MB

☐ Enable publishing option (Configure SMTP Server in manager configuration screen to use this option)  
☒ Deliver results via E-mail  
☐ Notify results via E-mail

To E-mail  [Use comma(,) to separate multiple e-mail recipients]

Update status via RSS

Show in

☒ Persist data in Eventvault Explorer

Figure 36

5. Choose appropriate time for report execution and in **Step 8** check **Persist data in Eventvault explorer** box.

**REPORT WIZARD**

TITLE: WINDOWS POWERSHELL-REMOTE SESSION CREATION DETAILS

DATA PERSIST DETAIL

Cancel < BACK NEXT >

Select columns to persist Step 9 of 10

**RETENTION SETTING**

Retention period: 7 days ⓘ

☐ Persist in database only *[Reports will not be published and will only be stored in the respective database]*

**SELECT COLUMNS TO PERSIST**

COLUMN NAME	PERSIST
Computer	<input checked="" type="checkbox"/>
Event User	<input checked="" type="checkbox"/>
Remote Host	<input checked="" type="checkbox"/>

Figure 37

6. Check column names to persist using **PERSIST** checkboxes beside them. Choose suitable **Retention period**.
7. Proceed to next step and click **Schedule** button.
8. Wait for scheduled time or generate report manually.

## Create Dashlets

1. **EventTracker 8** is required to configure flex dashboard.
2. Open **EventTracker** in browser and logon.

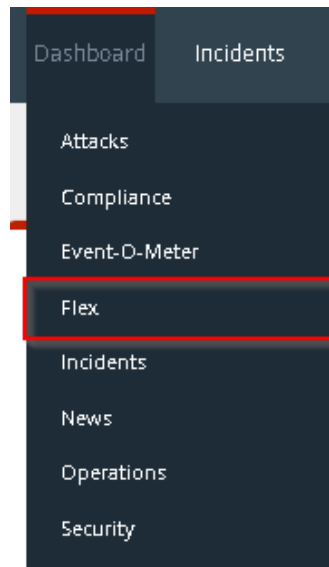


Figure 38

3. Navigate to **Dashboard>Flex**.  
Flex Dashboard pane is shown.

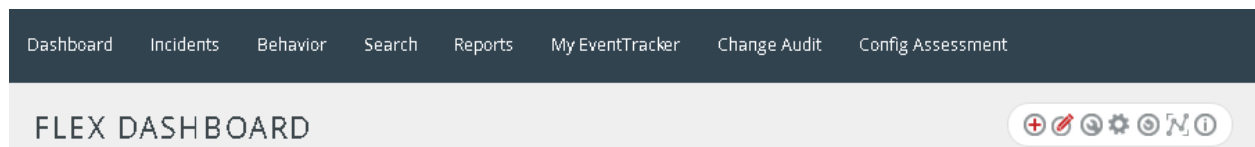




Figure 39

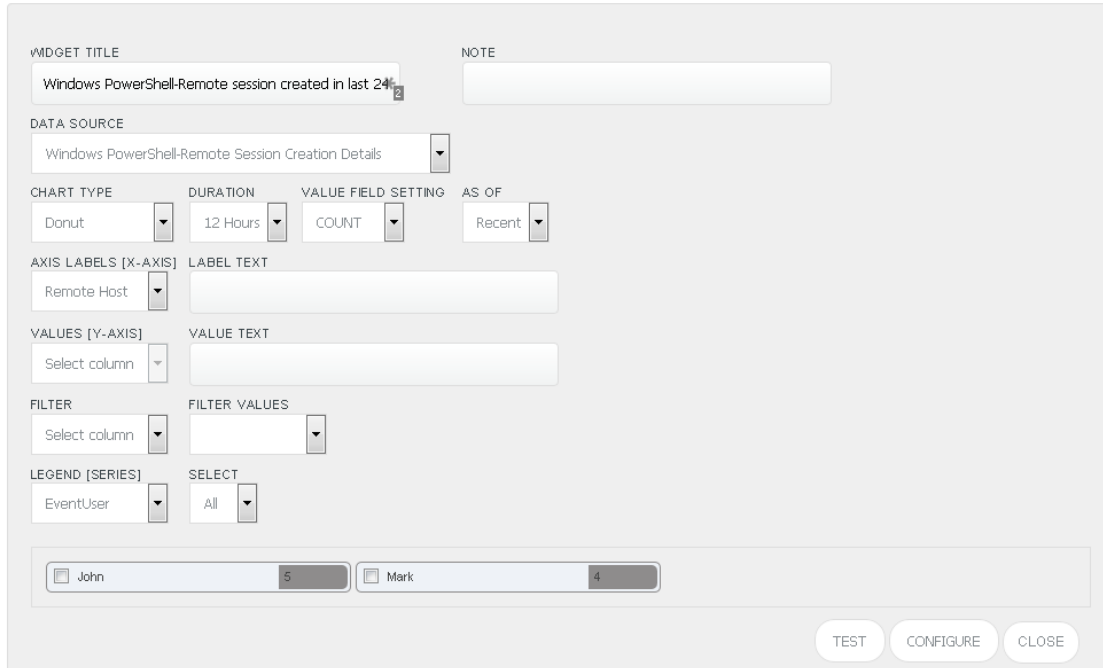
4. Click  to add a new dashboard.  
Flex Dashboard configuration pane is shown.

A screenshot of the 'FLEX DASHBOARD' configuration pane. It has a light gray background. At the top, the text 'FLEX DASHBOARD' is displayed. Below it, there is a form with two input fields: 'Title' with the value 'Windows PowerShell' and 'Description' with the value 'Windows PowerShell 3.0 and later'. At the bottom right of the form are three buttons: 'SAVE', 'DELETE', and 'CANCEL'.

Figure 40

5. Fill fitting title and description and click **Save** button.
6. Click  to configure a new flex dashlet.  
Widget configuration pane is shown.

### WIDGET CONFIGURATION



The Widget Configuration pane is a form with the following sections and controls:

- WIDGET TITLE:** A text input field containing "Windows PowerShell-Remote session created in last 24h".
- NOTE:** A text input field.
- DATA SOURCE:** A dropdown menu showing "Windows PowerShell-Remote Session Creation Details".
- CHART TYPE:** A dropdown menu showing "Donut".
- DURATION:** A dropdown menu showing "12 Hours".
- VALUE FIELD SETTING:** A dropdown menu showing "COUNT".
- AS OF:** A dropdown menu showing "Recent".
- AXIS LABELS [X-AXIS]:** A dropdown menu showing "Remote Host".
- LABEL TEXT:** A text input field.
- VALUES [Y-AXIS]:** A dropdown menu showing "Select column".
- VALUE TEXT:** A text input field.
- FILTER:** A dropdown menu showing "Select column".
- FILTER VALUES:** A dropdown menu.
- LEGEND [SERIES]:** A dropdown menu showing "EventUser".
- SELECT:** A dropdown menu showing "All".
- Preview:** A section showing two bars: "John" with a value of 5 and "Mark" with a value of 4.
- Buttons:** "TEST", "CONFIGURE", and "CLOSE" buttons at the bottom right.

Figure 41

7. Locate earlier scheduled report in **Data Source** dropdown.
8. Select **Chart Type** from dropdown.
9. Select extent of data to be displayed in **Duration** dropdown.
10. Select computation type in **Value Field Setting** dropdown.
11. Select evaluation duration in **As Of** dropdown.
12. Select comparable values in **X Axis** with suitable label.
13. Select numeric values in **Y Axis** with suitable label.
14. Select comparable sequence in **Legend**.
15. Click **Test** button to evaluate.  
Evaluated chart is shown.



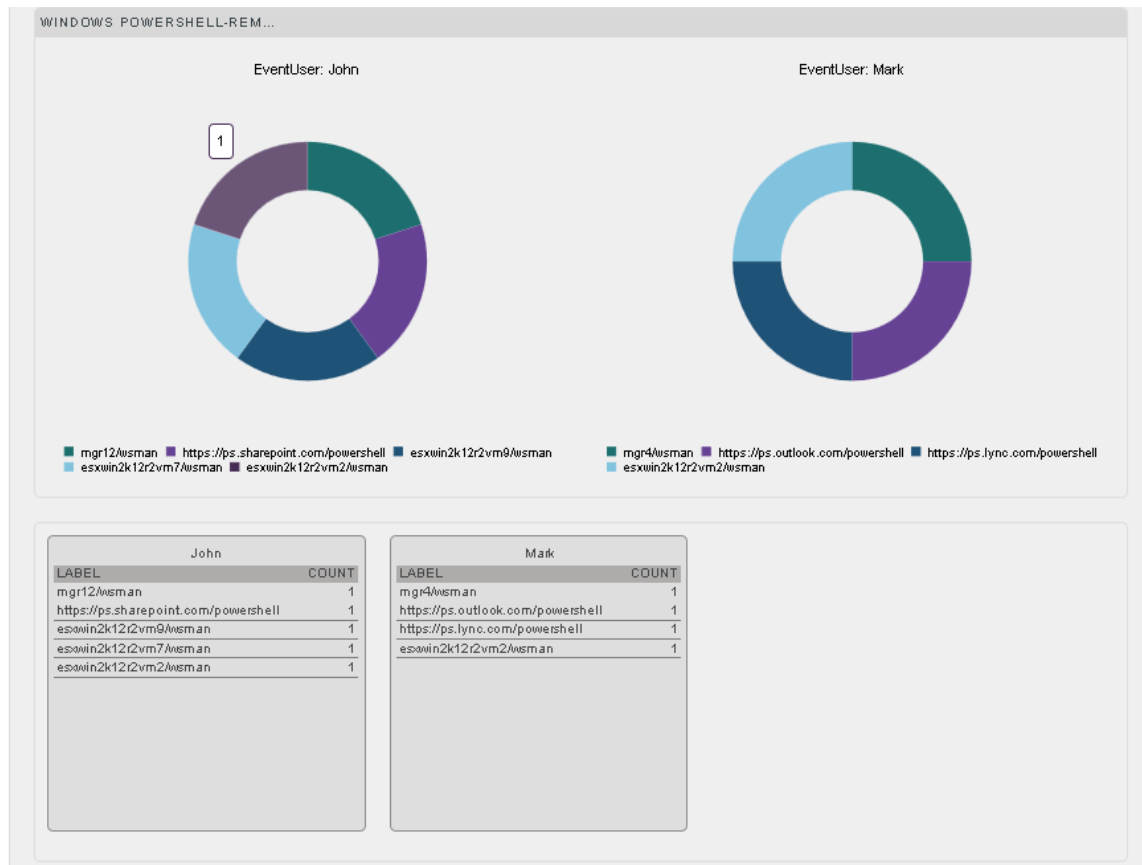


Figure 42

16. If satisfied, click **Configure** button.

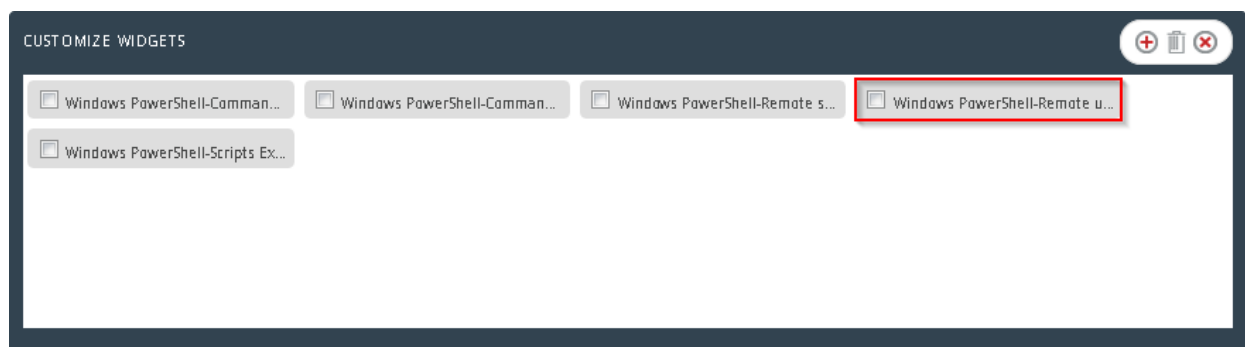



Figure 43

17. Click 'customize'  to locate and choose created dashlet.

18. Click  to add dashlet to earlier created dashboard.

## Sample Dashboards

- **Windows PowerShell-Remote session created in last 24 hrs**

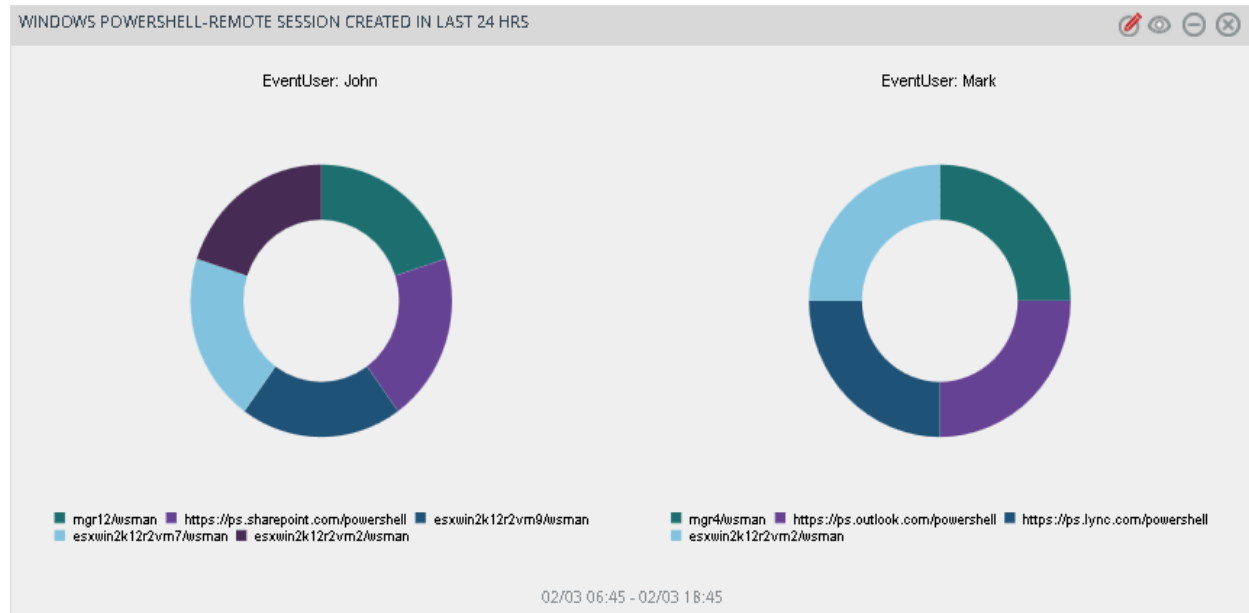


Figure 44

# Sample Reports

- Windows PowerShell-Command execution error details

Windows PowerShell-Command execution error details						
LogTime	Computer	User Name	Host Type	Script Path	Command Executed	Error Details
02/29/2016 02:47:59 PM	MGR23	ADMIN\Jim	Windows PowerShell ISE Host	C:\Downloads\HealthCheck.ps1	Out-File	Error Message = Could not find a part of the path 'C:\Scripts\Test.htm'. Fully Qualified Error ID = FileOpenFailure,Microsoft.PowerShell.Commands.OutFile Command
02/29/2016 02:49:09 PM	MGR23	ADMIN\Jim	Windows PowerShell ISE Host	C:\Downloads\HealthCheck.ps1	Out-File	Error Message = Could not find a part of the path 'C:\Scripts\Test.htm'. Fully Qualified Error ID = FileOpenFailure,Microsoft.PowerShell.Commands.OutFile Command
02/29/2016 02:53:00 PM	HR43	ADMIN\Jim	RemoteHost		Invoke-WebRequest	Error Message = X Network Access Message: The page cannot be displayed  Explanation: There is a problem with the page you are trying to reach and it cannot be displayed.

Figure 45

- Windows PowerShell-Script execution details

Windows PowerShell-Script execution details						
LogTime	Computer	User Name	Host Type	Script Path	Command Executed	Command Parameters
02/29/2016 02:41:46 PM	MGR44	ADMIN\Jim	RemoteHost	C:\Users\tom\Downloads\PowerSpl oit\Recon\PowerView.ps1	New-Object	ParameterBinding(New-Object); name="Property"; value="System.Collectio
02/29/2016 02:41:47 PM	MGR44	ADMIN\Jim	RemoteHost	C:\Users\tom\Downloads\PowerSpl oit\Recon\PowerView.ps1	New-Object	ParameterBinding(New-Object); name="TypeName";
02/29/2016 02:47:59 PM	HR23	ADMIN\Jim	Windows PowerShell Host	C:\HealthCheck.ps1	Get-Content	ParameterBinding(Get-Content); name="Path"; value="C:\scripts\test.ht
02/29/2016 02:47:59 PM	HR23	ADMIN\Jim	Windows PowerShell Host	C:\HealthCheck.ps1	Get-Content	ParameterBinding(Get-Content); name="ErrorAction";
02/29/2016 02:47:59 PM	HR23	ADMIN\Jim	Windows PowerShell Host	C:\HealthCheck.ps1	Out-File	ParameterBinding(Out-File); name="FilePath"; value="C:\Scripts\Test.h
02/29/2016 02:47:59 PM	PRO22	MGMT\Martha	Windows PowerShell ISE Host	C:\Users\tom\Downloads\UpdateCh eck.ps1	New-Object	ParameterBinding(New-Object); name="TypeName";
02/29/2016 02:47:59 PM	PRO22	MGMT\Martha	Windows PowerShell ISE Host	C:\Scripts\UpdateCheck.ps1	New-Object	ParameterBinding(New-Object);
02/29/2016 02:47:59 PM	PRO22	MGMT\Martha	Windows PowerShell ISE Host	C:\Scripts\UpdateCheck.ps1	Invoke-Expression	ParameterBinding(Invoke-Expression); name="Command";

Figure 46

-X-