Learn Powershell | Achieve More
What is this Powershell of which you speak?

Introducing PoshEventUI: A UI Way to Create Permanent WMI Events

Posted on August 16, 2013

Something that I have been working on the past couple of months (in between other things) was a UI approach to creating permanent WMI events vs. a command line approach.

I really had only a few requirements that I wanted to achieve with this:

- Have a main window showing the current WMI events
- Have Add/Remove buttons for each instance (Filter,Consumer and Binding)
- Use separate windows for each instance
  - The Consumer window will have one window but will change based on the type of Consumer being created

So without further ado, here is my latest project: **PoshEventUI**
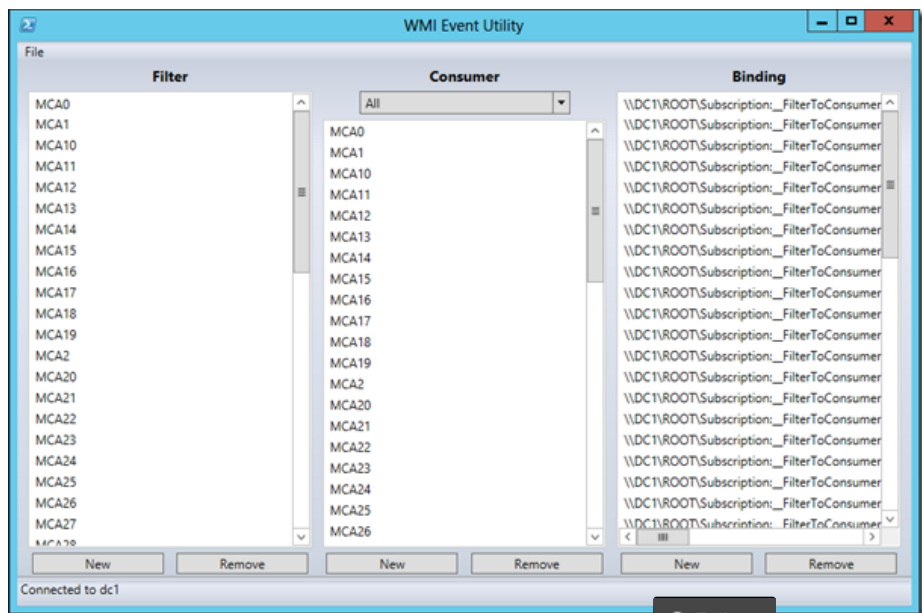
The main UI window is based on ~~...~~ *nthelper.exe available from* http://powerevents.codeplex.co...

The download for this is at https://... m/

Once you download it, unblock th... complain about it and prompt for each script being run (**Unblock-File** works...

Follow

Follow "Learn Powershell | Achieve More"

Get every new post delivered to your Inbox.
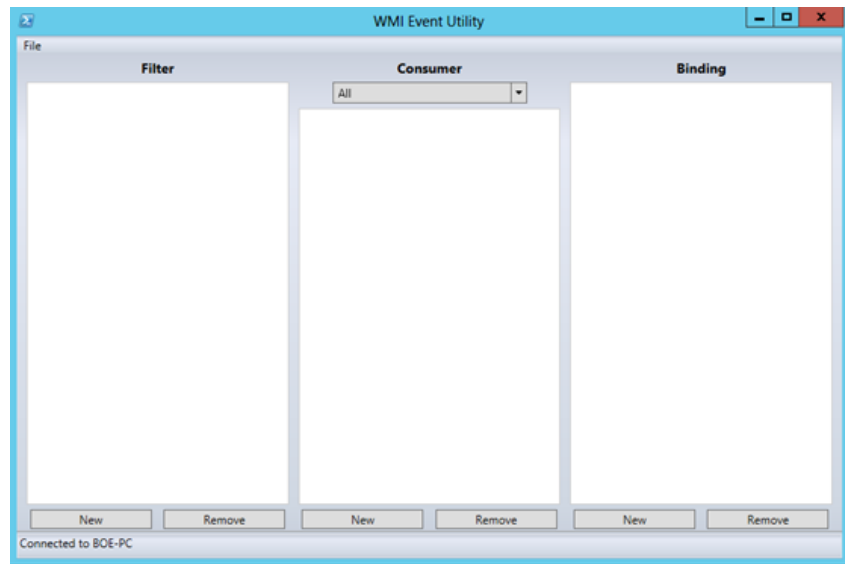
Join 168 other followers

Enter your email address

Sign me up

Powered by WordPress.com

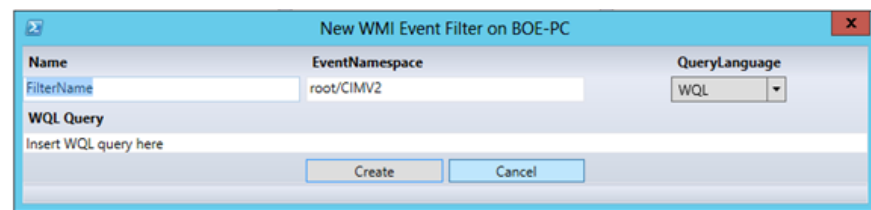Navigate through the unzipped folder and run .\wmieventutility.ps1

```
PS C:\Users\Administrator\Desktop\PoshEventUI_0_9> .\WMIEventUtility.ps1
```

Once the window has been opened, you can choose to to create a **New** instance or to **Remove** and instance.

Let's work through each of these just to get an idea on what you can do.

First off is the **Filter**.

Here you can specify a **Name** and a **WQL Query** that will be used to monitor for a specific event in WMI. You can look at my articles on Temporary Events and Permanent Events for ideas or look at some searches on your favorite search engine. Also available is some great information in the PowerEvents documentation that shows more on working with WQL queries and also provides some great examples.
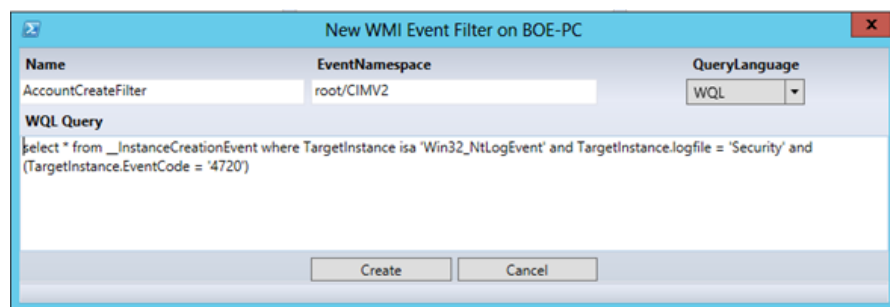
Once completed, click **Create** and it will now be added to the WMI repository.
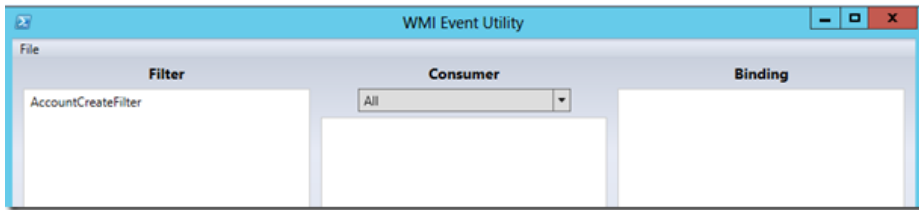
Next up is the **Consumer** creation. There are 5 possible consumers to choose from and you can have multiple consumers setup with a filter via the Binding, so use whatever you wish to accomplish your requirement.
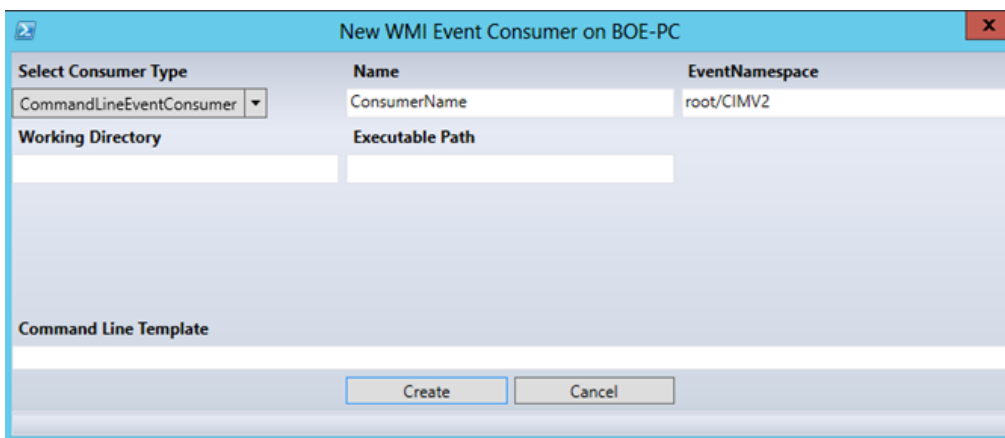
Lets look at all of the possible Consumer choices. You can pick whichever consumer you prefer from the drop down menu in the upper left hand corner labeled **Select Consumer Type**.
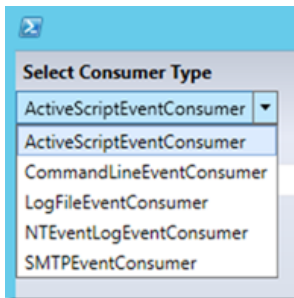
ActiveScriptEventConsumer
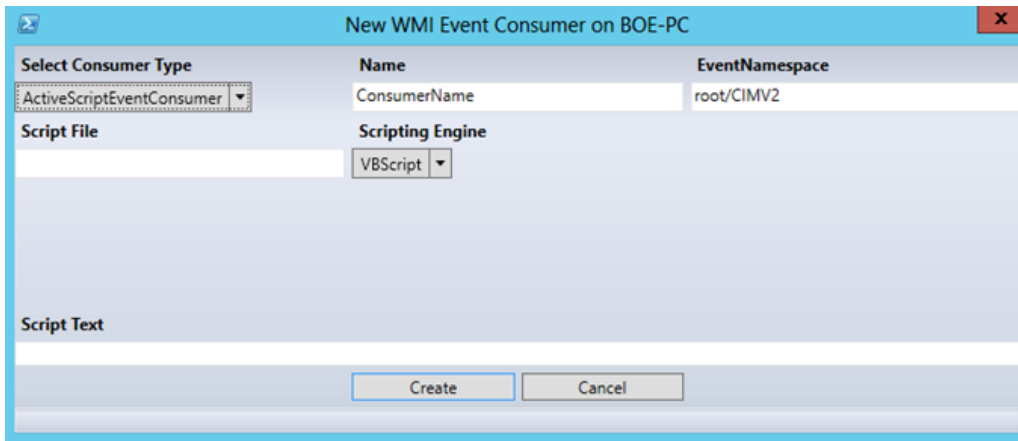
CommandLineEventConsumer

LogFileEventConsumer

New WMI Event Consumer on BOE-PC　　x

Select Consumer Type
LogFileEventConsumer ▼

Name
ConsumerName

EventNamespace
root/CIMV2

File Name

Maximum File Size (0 - 65535)

Encoding
☐ IsUnicode

File Text

　　Create　　　　Cancel

NTEventLogEventConsumer

New WMI Event Consumer on BOE-PC　　x

Select Consumer Type
NTEventLogEventConsumer ▼

Name
ConsumerName

EventNamespace
root/CIMV2

Source

Event ID

Category

UNC Server Name

Event Type
Success ▼

Insertion String Templates

　　Create　　　　Cancel

SMTPEventConsumer

New WMI Event Consumer on BOE-PC　　x

Select Consumer Type
SMTPEventConsumer ▼

Name
ConsumerName

EventNamespace
root/CIMV2

To

From

Subject

Cc

Bcc

Reply To Line

SMTPServer

HeaderFields (separate by ',' or ';')

Message

　　Create　　　　Cancel

In this case I am going with an SMTP consumer.

Note that with the **From** field blank, it will default to <u>WMI@<computername>.com</u> for the address.

Once you have the picked out your consumer and entered the necessary information, click **Create** and you will be taken back to the main window with the new consumer listed.

You can event filter the Consumers based on the type using the drop-down menu.

Last on the list is to bind these together using the **Binding**.


Not really a lot to it. You pick the Filter and Consumer and click **Create.** This will create the binding and enable the WMI event subscription.
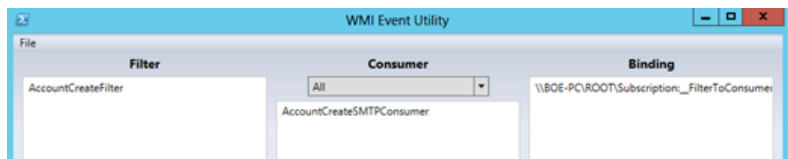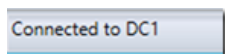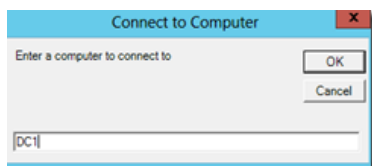
That is really all there is to it! The trick is really knowing the WQL query for the Filter and then determining what kind of consumer fits this best.

Almost forgot… You can even connect to remote systems to add and remove event subscriptions! Just click on the File menu and select **Connect to Another Computer**.
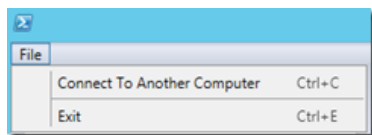
Enter a computername and click **OK** and away you go.

Please keep this mind that it is in **Beta**, so there will most likely be bugs crawling around. Just stop by the Issues page and let me know what they are and I will work to get rid of them. I do have plans for a Template feature which will have pre-determined Filters as well as the option to save custom filters as templates for use later. This may also move over to the Consumers as well at a later date.

Hope everyone enjoys this!

About these ads

Share this:

| Twitter 14 | 8+ Google | f Facebook 5 | ✉ Email | in LinkedIn 2 | Reddit | Pocket |

★ Like

Be the first to like this.

Related

PowerShell and Events: Permanent WMI Event Subscriptions
In "powershell"

PowerShell and Events: WMI Temporary Event Subscriptions
In "powershell"

PowerShell Profile Removal GUI
In "powershell"

About Boe Prox
Microsoft PowerShell MVP working as a Senior Systems Administrator
[View all posts by Boe Prox →](#)

This entry was posted in GUI, powershell, WPF and tagged eventing, permanent events, PoshEventIU, Powershell, wmi. Bookmark the permalink.

3 Responses to *Introducing PoshEventUI: A UI Way to Create Permanent WMI Events*

Rodrigo *says:*
September 17, 2013 at 9:14 am

Yeah, i think this idea very interesting for a new version. I will see this article for try create a consumer for multiples systems.
tks

Reply

Rodrigo *says:*
September 16, 2013 at 5:13 pm

Hi Boe Prox this project really helped me, but now i need create a permanent event to notify sql services to 70 servers, the issue is, i need create one by one or there a way for create to all servers together?

Tks,
Rodrigo Figueiredo

Reply

> **Boe Prox** *says:*
> September 16, 2013 at 9:31 pm
>
> Glad you enjoy the utility! Currently I do not have a way to do a bulk creation against multiple systems. But this sounds like a nice feature request for a future update. I would recommend using PowerEvents
> http://powerevents.codeplex.com to script against for creation of a WMI event consumer.
>
> Reply

Learn Powershell | Achieve More
The Twenty Ten Theme. Ⓦ *Create a free website or blog at WordPress.com.*