Example of using Alarm Enrichment in CA SOI 3.1

What is alarm enrichment: An enrichment is a means of selecting a specific alarm that needs more information added to it before it can be further progressed. The enrichment is performed one time on new alarms as they come in from the connectors. It is a means of passing data out of SOI then processing it and sending it back in to be used for further processing.

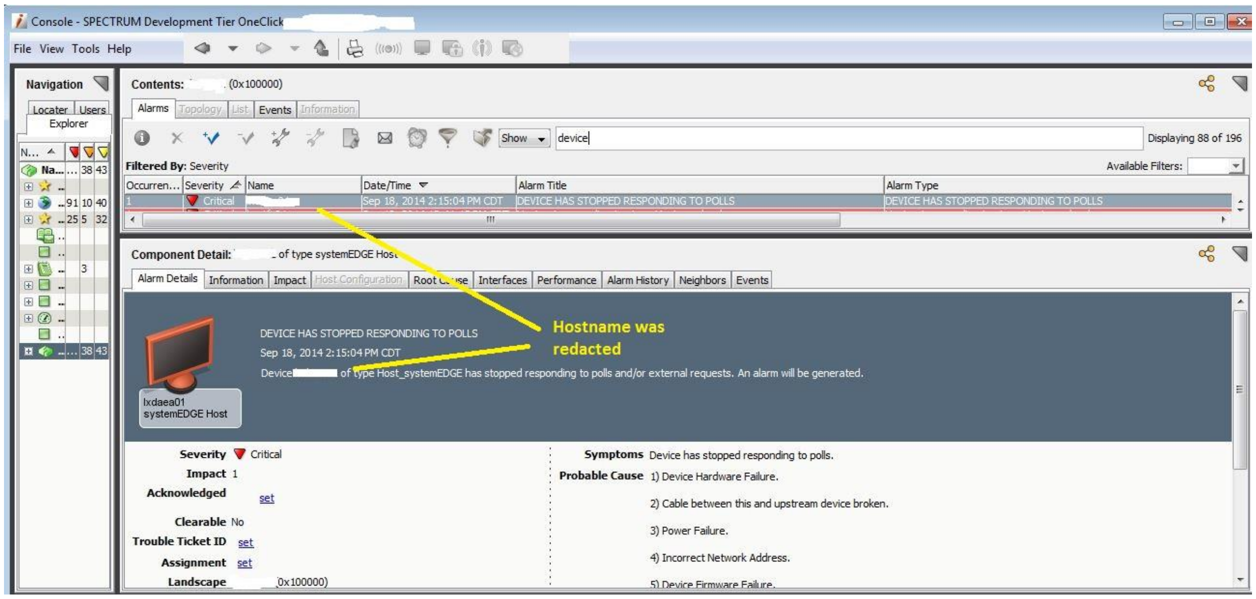
Test case: I have created a hypothetical case as an example to show you how to create an enrichment. I have set this up in a development environment and you will see screen shots for each step that may have redacted information. In our test case we are going to say we have an odd false alarm problem at our site where we are using CA Spectrum to check for hosts down but we get a high number that seem to be responding to pings when the alarm comes in. Thus, dispatching the alarm to technologists that are dealing with hosts that are not down at the time they get the notification. As a Spectrum/SOI administrator we have found that by the time alarm is received the problem has gone away but we can see in some cases there is a delay in the ping response time. We are going to create an enrichment that will do a ping as soon as the alarm gets into CA SOI. See the chart for response time:

Response Time in ms	**Determined Result**
Less than 50	good
50 to 100	marginal
Greater than 100	bad
No response	failed

If we get the results from the ping test back into SOI then we will change the behavior for the alarm so that we minimize the impact on technologists. The enrichment it is only used to perform a ping test then returns the results back to CA SOI.

Response Time in milliseconds	**Determined Result**	**Reactionary steps**
Less than 50	good	Clear the alarm
50 to 100	marginal	Create CASD medium ticket
Greater than 100	Bad	Create CASD high ticket
No response	Failed	Create CASD critical ticket

Step 1: Find the alarm in CA Spectrum. In our case we found two alarm types that are very similar so we will need to account for both conditions with one enrichment. Here is one of the two example alarms.



Console - SPECTRUM Development Tier OneClick [redacted]

Contents: [redacted] (0x100000)

Filtered By: Severity — Show: device — Displaying 88 of 196

Occurren...	Severity	Name	Date/Time	Alarm Title	Alarm Type
1	Critical	[redacted]	Sep 18, 2014 2:15:04 PM CDT	DEVICE HAS STOPPED RESPONDING TO POLLS	DEVICE HAS STOPPED RESPONDING TO POLLS

Component Detail: [redacted] of type systemEDGE Host

DEVICE HAS STOPPED RESPONDING TO POLLS
Sep 18, 2014 2:15:04 PM CDT
Device [redacted] of type Host_systemEDGE has stopped responding to polls and/or external requests. An alarm will be generated.

Hostname was redacted

Severity	Critical
Impact	1
Acknowledged	set
Clearable	No
Trouble Ticket ID	set
Assignment	set
Landscape	[redacted] (0x100000)

Symptoms: Device has stopped responding to polls.

Probable Cause:
1) Device Hardware Failure.
2) Cable between this and upstream device broken.
3) Power Failure.
4) Incorrect Network Address.
5) Device Firmware Failure.

File View Tools Help
Navigation
Locater Users Explorer
Alarms Topology List Events Information
Alarm Details Information Impact Host Configuration Root Cause Interfaces Performance Alarm History Neighbors Events
lxdaea01
systemEDGE Host
Available Filters:

Step 2. Create a PERL script that parses the hostname out of the alarm details, pings the host then returns the results. For this step I created a batch file with the name of a host in this case replace <somehost> with your test host. I have included the PERL script in this document.

Pingtest.bat

- d:\perl\bin\perl.exe ping_test.pl “Device <somehost> of type Host_systemEDGE has stopped responding to polls and/or external requests”
- Take the results of the PERL script and return the parameters back to the command line.
  parm1,0,parm2,good

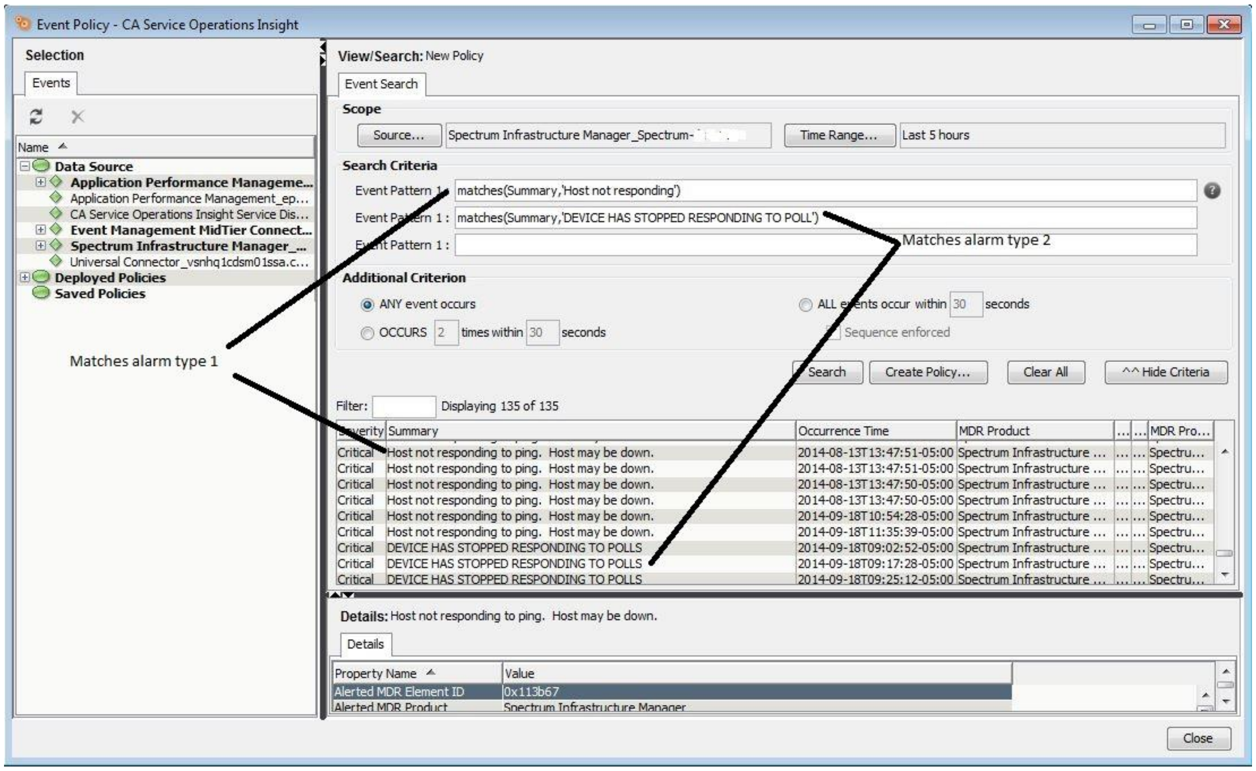
  1 2 3 4

Remember the table above. The first parameter we are going to return for parm1 is going to be the maximum response time and the second parameter parm2 is the word, good,marginal,bad,failed. We use commas to mark parm1,value,parm2,result. SOI needs to see the items comma separated. So the word parm1 tells SOI it’s the first item being returned. The next item after the comma is the numeric value that we got as a result of the ping test. The next item parm2 tells SOI it’s the second item being returned being returned and the last one is the determined result. Take out as many spaces characters as you can a keep the line scrunched together so you always have 4 separate items being returned.

```
#This is an example of and enrichment for host down alarms
# Wtitten by Dan Tadysak, American Family Ins. 9/22/2014
$input = $ARGV[0];
chomp $input;
if ($input =~ /^host/i){   #Look for 1st type of host down alarm
            $input =~ /\(name\s+\-\s+(.*),\s+/;
            $host = $1;}
else{     #Look for 2nd type of host down alarm
            $input =~ /device\s+(.*)\s+of/i;
            $host = $1;}
open (LOG, ">>d:\\scripts\\auto\\test\\pingtest.log");
#Run the ping test and put the results into an array
@ping_cmd = `ping $host`;
$time_max = 0;
#Now iterate thru the array and see what the time delay was
foreach my $ping_test (@ping_cmd){
            chomp $ping_test;
            #print "ping $ping_test\n";
            #Look for the returned data fromn the ping command and just get the time = <number>
            if ($ping_test =~ /time[\=|\>|\<](\d+)$/){
                        $time_val = $1;
                        #Look at each time value and if its the higest set it
                        if ($time_max < $time_val)                    {
                                    #Set the time value because its the highest one we found
                                    $time_max_max = $time_value;}
            }
            elsif ($ping_test =~ /could not find/)       {
                        $result = "failed";           }
}
if (($time_max < 50) && ($result ne "failed")){
            $result = "good";}
elsif (($time_max > 50) && ($time_max < 100)){
            $result = "marginal";}
elsif ($time_max > 100){
            $result = "bad"; }

#We are passing back more than one parameter:
#           "parm1,<some_number>,parm2,<failed|bad|marginal|good>""
print "parm1,$time_max,parm2,$result";
print LOG "input=$host and parm1,$time_max,parm2,$result\n";
close (LOG);
exit;
```

Step 3:Setup the Event Policy in CA SOI. On the Tools menu choose Event Policies. Starting at the top choose the Source connector in this case its coming from CA Spectrum so I choose that connector. In the time range you don't have to pick anything but it helps to have example to make sure that you have set things up correctly so I made sure I went back 5 hours and had plenty of alarm examples. These alarms are what have occurred in the past and can be live alarms or cleared alarms this is where you get the opportunity to look at alarms at each connector. This is helpful for not only the enrichments but just looking for any past event. Then hit Search and you get all of the alarms. In the filter line you can simply type in some or all of the words you want to match and it reduces the number based on what you typed. Once you see the alarms types you are looking for then you can enter search criteria. In our case we have two alarm types that we need to qualify so that these two will be sent to the PERL script on the enrichment. Make sure there is not too little or too much displayed because what you see here is what you get for the enrichment. In most cases you will only have one Event Pattern I just wanted to show how you could get more than one if you had an example like ours.


Event Policy - CA Service Operations Insight
Selection
Events
Name
Data Source
Application Performance Manageme...
Application Performance Management_ep...
CA Service Operations Insight Service Dis...
Event Management MidTier Connect...
Spectrum Infrastructure Manager_...
Universal Connector_vsnhq1cdsm01ssa.c...
Deployed Policies
Saved Policies
Matches alarm type 1
Matches alarm type 2
View/Search: New Policy
Event Search
Scope
Source...
Time Range...
Search Criteria
Additional Criterion
Search
Create Policy...
Clear All
^^ Hide Criteria
Details
Close

Source...: Spectrum Infrastructure Manager_Spectrum-
Time Range...: Last 5 hours

Event Pattern 1 : matches(Summary,'Host not responding')
Event Pattern 1 : matches(Summary,'DEVICE HAS STOPPED RESPONDING TO POLL')
Event Pattern 1 :

(●) ANY event occurs
( ) OCCURS 2 times within 30 seconds
( ) ALL events occur within 30 seconds
[ ] Sequence enforced

Filter: Displaying 135 of 135

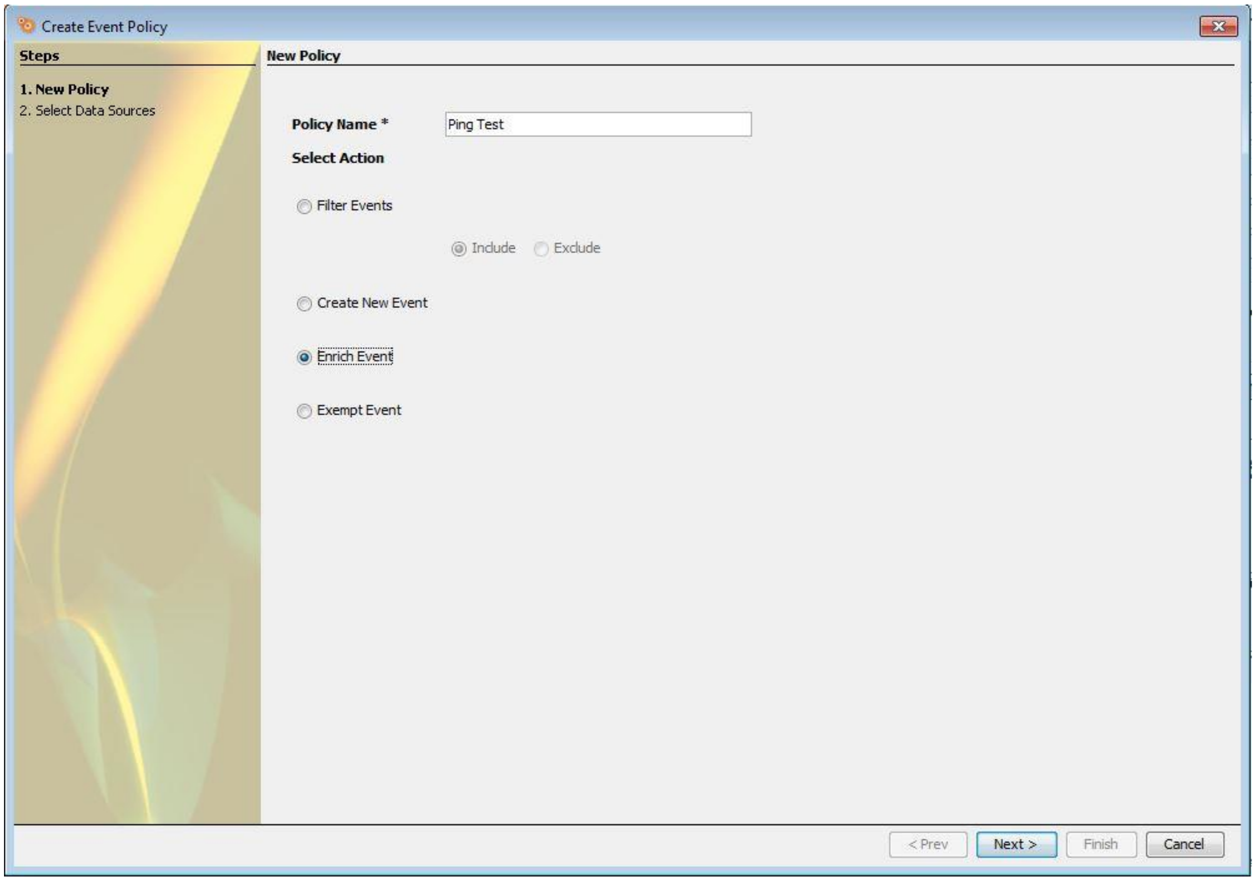
Severity	Summary	Occurrence Time	MDR Product	...	...	MDR Pro...
Critical	Host not responding to ping. Host may be down.	2014-08-13T13:47:51-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	Host not responding to ping. Host may be down.	2014-08-13T13:47:51-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	Host not responding to ping. Host may be down.	2014-08-13T13:47:50-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	Host not responding to ping. Host may be down.	2014-08-13T13:47:50-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	Host not responding to ping. Host may be down.	2014-09-18T10:54:28-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	Host not responding to ping. Host may be down.	2014-09-18T11:35:39-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	DEVICE HAS STOPPED RESPONDING TO POLLS	2014-09-18T09:02:52-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	DEVICE HAS STOPPED RESPONDING TO POLLS	2014-09-18T09:17:28-05:00	Spectrum Infrastructure ...	...	...	Spectru...
Critical	DEVICE HAS STOPPED RESPONDING TO POLLS	2014-09-18T09:25:12-05:00	Spectrum Infrastructure ...	...	...	Spectru...

Details: Host not responding to ping. Host may be down.

Property Name	Value
Alerted MDR Element ID	0x113b67
Alerted MDR Product	Spectrum Infrastructure Manager

Note: When you save the enrichment the event filters you set will remain intact you will not have to worry about the time range. If no source is selected then all connectors is assumed.

Step 4: Click the Create Policy button right next to the Search button. Give your policy a name and select Enrich Event. Then click the Next button.



Policy Name * Ping Test

Select Action

Filter Events

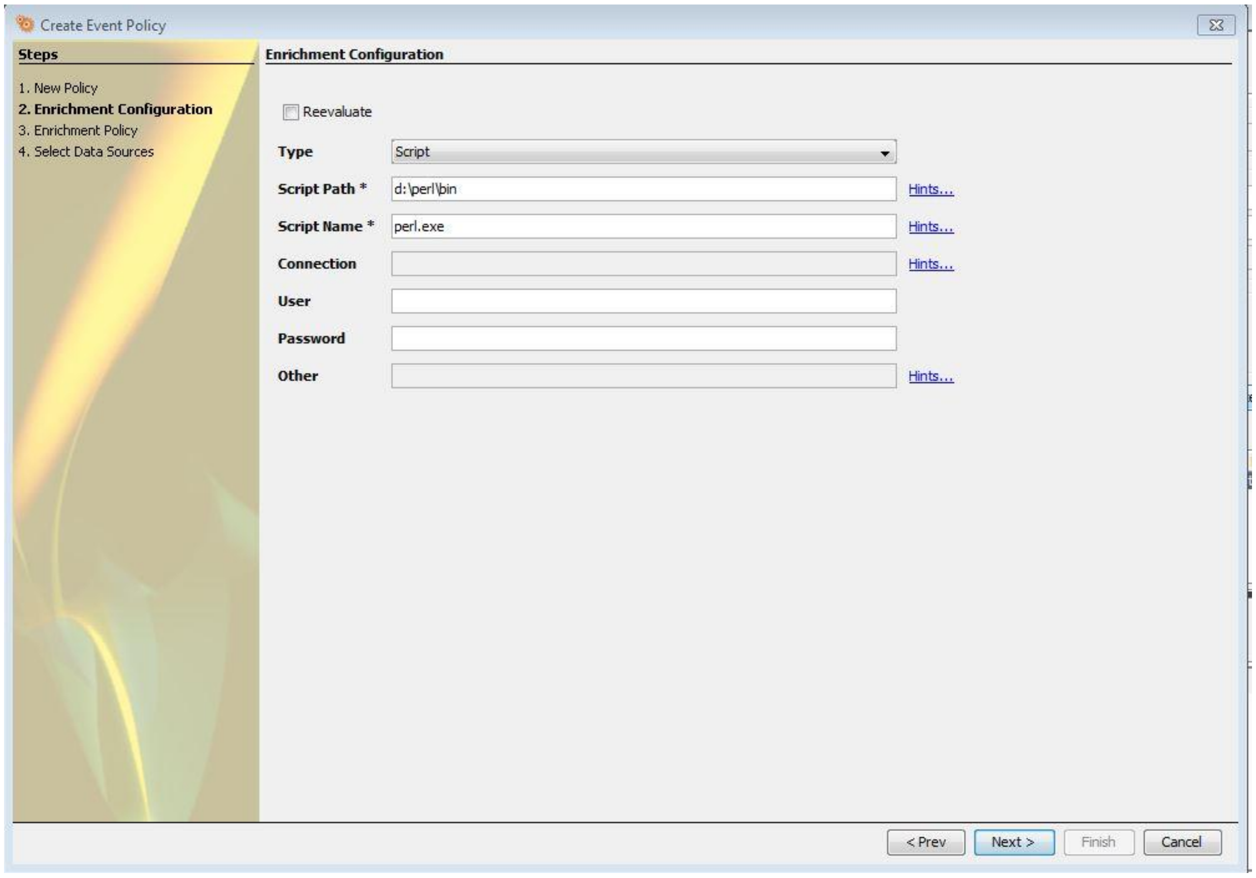
Include Exclude

Create New Event

[✓ Enrich Event]

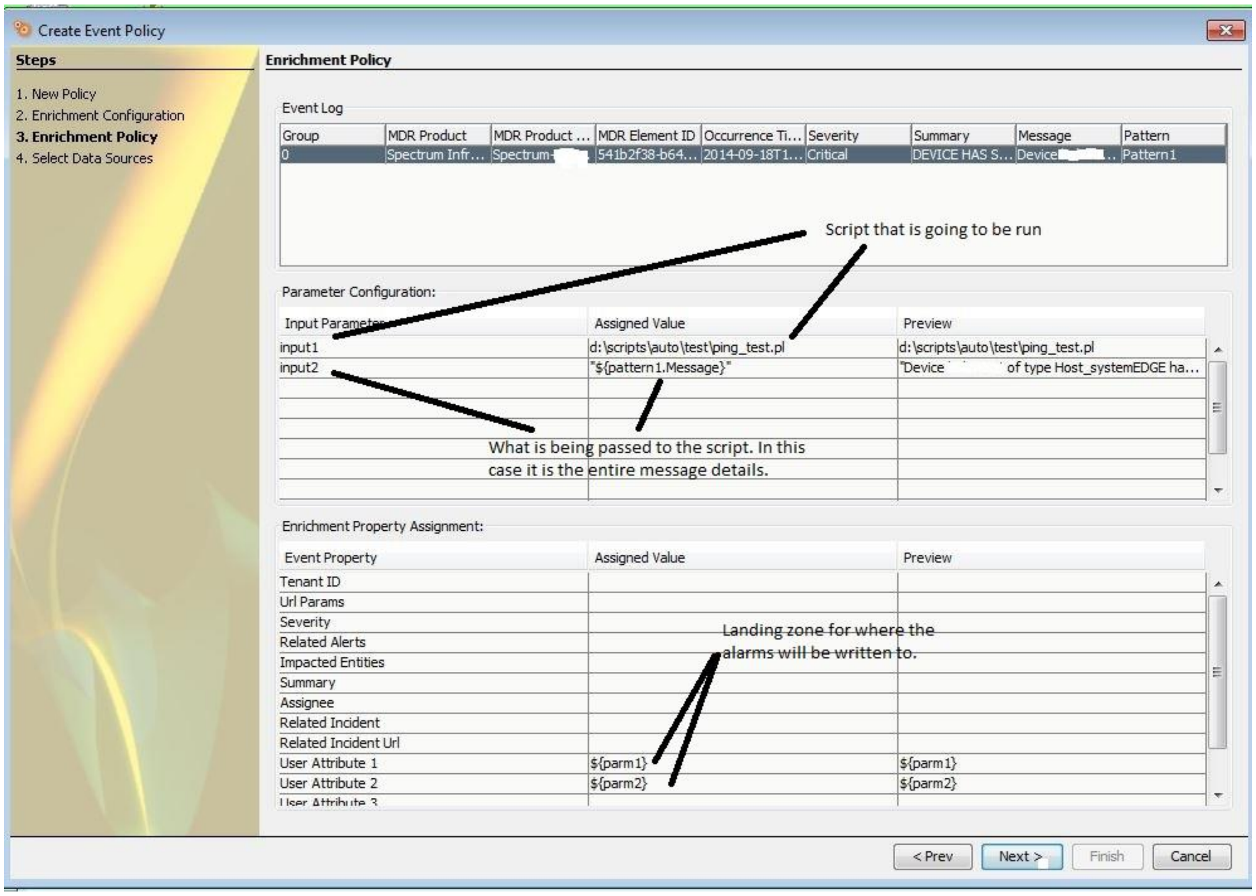
Exempt Event

Step 5: In the Type pulldown select Script, put the full path to the script engine and finally the perl.exe executable. The location of the script will need to be put in one of two places so you need to make sure your machine has your scripting engine on it. For our example we have a SOI system with that has the UI, manager and connector on it as an all-in-one configuration. If you put a check mark in “Reevaluate” it will look at all active alarms and run the enrichment on any that match and on any new ones. Typically, when we are testing it is not checked and when we go live the same is true unless were trying to immediately affect current alarms to get an immediate result. Since the script is going to be placed on the server and requires no other credentials we leave the boxes blank.



Reevaluate	☐
Type	Script
Script Path *	d:\perl\bin
Script Name *	perl.exe
Connection	
User	
Password	
Other	

Step 6: In the Event Log box at the top those are samples from the time range selected that will be used to show you in the preview windows below. When the enrichment is run it will be processed on new alarms and the time range is not in effect it's just in here to give you the ability to see what it looks like based on history events. You need to go to the blank box under Input parameters and type the word input1 just as shown. (It's not very intuitive that the blank boxes can edited). In the assigned value to the right of input1 you put the full path to the PERL script just as shown. In the next blank box down put in input2 then to the right put in the following EXACTLY as shown. This is the alarm details from the alarm. So whatever matches for Event Patterns from step 3 gets passed out to the script here. Down in the bottom Enrichment Properties Assignment this is where we will be putting the results of the PERL script. So scroll down and look for User Attribute 1 then in the box put in ${parm1} and the line below ${parm2} EXACTLY as shown. If you only have 1 parameter you only need one if you have more than you can just keep going. Our example we have two so and we are creating a landing zone for the results.


Create Event Policy
Steps
1. New Policy
2. Enrichment Configuration
3. Enrichment Policy
4. Select Data Sources
Script that is going to be run
What is being passed to the script. In this case it is the entire message details.
Landing zone for where the alarms will be written to.
< Prev
Next >
Finish
Cancel

Enrichment Policy

Event Log

Group	MDR Product	MDR Product ...	MDR Element ID	Occurrence Ti...	Severity	Summary	Message	Pattern
0	Spectrum Infr...	Spectrum-[redacted]	541b2f38-b64...	2014-09-18T1...	Critical	DEVICE HAS S...	Device[redacted]...	Pattern1

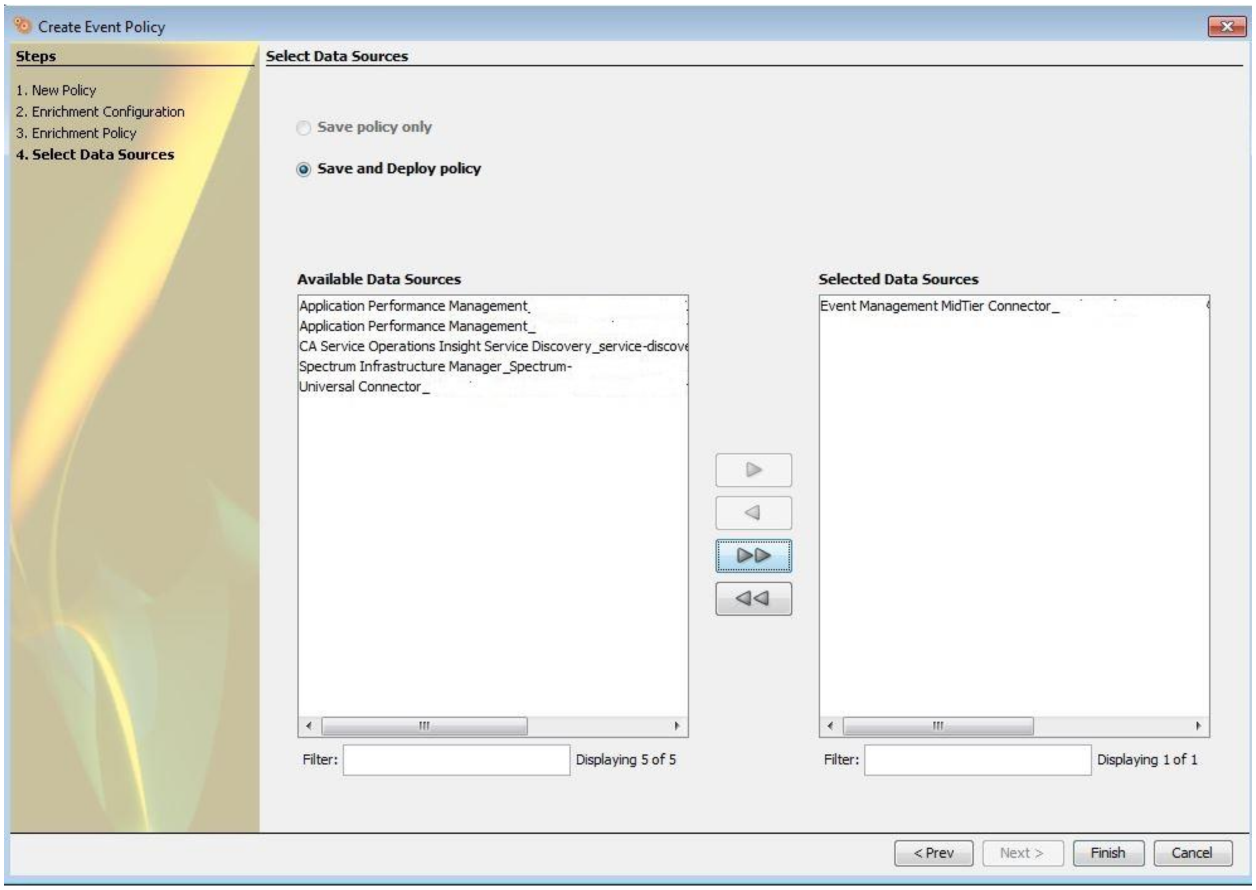
Parameter Configuration:

Input Parameter	Assigned Value	Preview
input1	d:\scripts\auto\test\ping_test.pl	d:\scripts\auto\test\ping_test.pl
input2	"${pattern1.Message}"	"Device [redacted] of type Host_systemEDGE ha...

Enrichment Property Assignment:

Event Property	Assigned Value	Preview
Tenant ID		
Url Params		
Severity		
Related Alerts		
Impacted Entities		
Summary		
Assignee		
Related Incident		
Related Incident Url		
User Attribute 1	${parm1}	${parm1}
User Attribute 2	${parm2}	${parm2}
User Attribute 3		

Step 7: Select the connector for which you want to have the enrichment run on. This seems intuitive but be careful here. If you want the alarm to be check at the global level from any connector then select the Event Manament MidTier Connector. If you select this it will run on any alarm from any connector. If you select Spectrum Infrastructure Manager then it will only check alarms from Spectrum on that connector. So why be careful….if you have more than one Spectrum connector and want it to check all spectrum systems for the same alarm then use the Event Manament MidTier Connector. The PERL script will run from the SOI manager system where the MidTier connector resides. If you select the Spectrum connector and you have a dedicated SOI server for Spectrum connecotrs to run on you need to have PERL and the script on the connector server as that is where the enrichment will be run from. In our example we have an all-in-one SOI system so I just selected the MidTier connector. Highlight the connector and use the arrow button to move it to the box on the right side. Make sure the "Save and Deploy policy" is selected at the top of the screen and click finish.


Create Event Policy

Steps
1. New Policy
2. Enrichment Configuration
3. Enrichment Policy
4. Select Data Sources

Select Data Sources

Save policy only [✓ Save and Deploy policy]

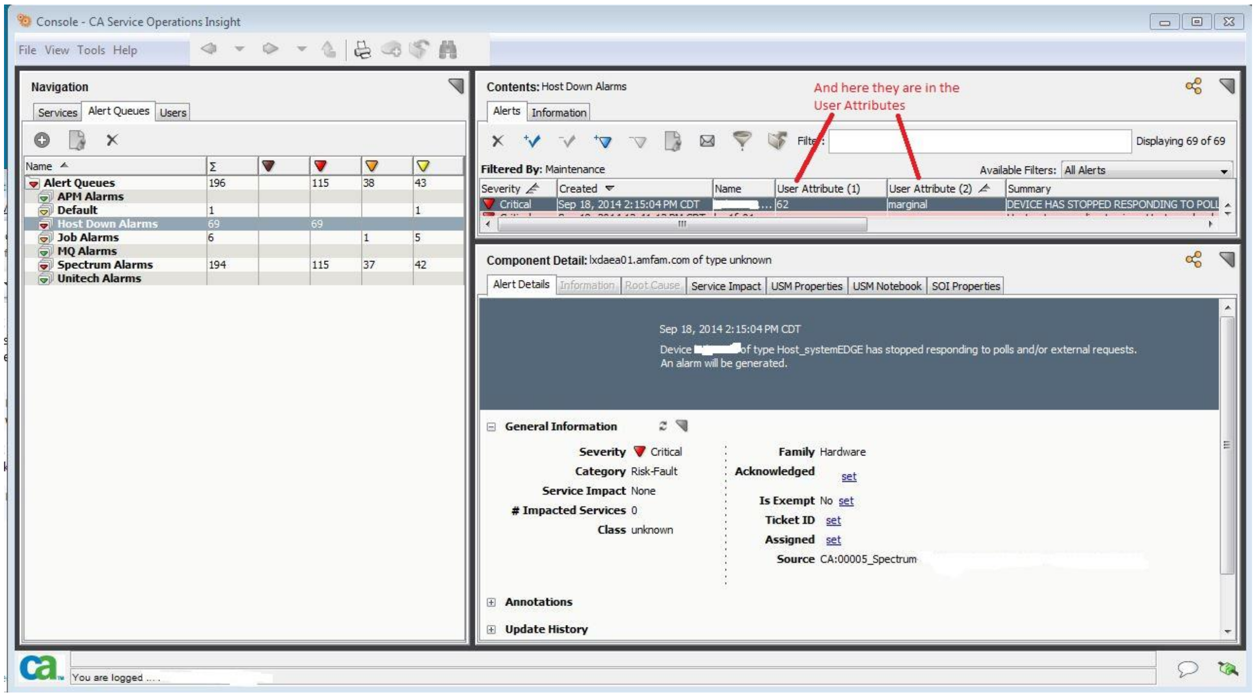
Available Data Sources
Application Performance Management_
Application Performance Management_
CA Service Operations Insight Service Discovery_service-discove
Spectrum Infrastructure Manager_Spectrum-
Universal Connector_

Filter: Displaying 5 of 5

Selected Data Sources
Event Management MidTier Connector_

Filter: Displaying 1 of 1

Step 8: Test your work. Create a test alarm and look at CA SOI to see if the user attributes are populated. Look for the attributes by clicking anywhere in the top row then right click and check mark the User Attributes (there are 5). I am showing in the graphic below the alarms in the alert queue then I see the attributes have a value that was passed in from the enrichment. Remember this important piece of information. The enrichment only runs ONCE and it is one of the first things that is done when a new alarm comes in from a connector.



Name	Σ	▼	🔻	▽	▽
Alert Queues	196		115	38	43
APM Alarms					
Default	1				1
Host Down Alarms	69		69		
Job Alarms	6			1	5
MQ Alarms					
Spectrum Alarms	194		115	37	42
Unitech Alarms					

Contents: Host Down Alarms

Displaying 69 of 69

Filtered By: Maintenance

Severity	Created	Name	User Attribute (1)	User Attribute (2)	Summary
Critical	Sep 18, 2014 2:15:04 PM CDT	[redacted]...	62	marginal	DEVICE HAS STOPPED RESPONDING TO POLL

Component Detail: lxdaea01.amfam.com of type unknown

Sep 18, 2014 2:15:04 PM CDT
Device [redacted] of type Host_systemEDGE has stopped responding to polls and/or external requests.
An alarm will be generated.

General Information

Severity	Critical	**Family**	Hardware
Category	Risk-Fault	**Acknowledged**	set
Service Impact	None	**Is Exempt**	No set
# Impacted Services	0	**Ticket ID**	set
Class	unknown	**Assigned**	set
		Source	CA:00005_Spectrum

Annotations

Update History

There are all kinds of things you can do now that you have this extra piece of information such as creating an Event Policy and Action to clear and alarm, to create a CASD ticket to run a PAM process and so on. You may simply want to move the alarm to another queue just by including more or more of the user attributes in the criteria field. I have about 5 enrichments and I only use them for very necessary processing that can be only done here prior to taking any further action. Also, it keeps nontechnical people from performing actions out of that process by doing the work for them and returning the results. For the reactionary steps from the table above you can easily create an action to look for the word “marginal” on User Attribute and create a CASD ticket with a medium priority level. You get the idea and just apply the response with the Escalation Policies and actions to get the automated outcome.