Creating User Defined Metrics

By: Rob Zoeteweij
Date: 04 feb 2010
http://oemgc.wordpress.com

This document describes an example of a User Defined Metric

Using User Defined Metrics we can extend the events that are evaluated based on the default "Out of the Box" Metrics with practically any event we can think of.

User Defined Metrics must be defined on target level.

For this example we will select database orcl1
Navigate to the bottom of the page and select User Defined Metrics

Press [Create]

We will name the User Defined Metric in this example "UDM_INVALID_OBJECTS".

The SQL Query that we are going to use in this example will return a number value as result, so we select "Number" as Metric Type

As the SQL Query will return only one row, we select "Single Value" SQL Query Output

We enter the SQL Query:
SELECT COUNT(*) FROM USER_OBJECTS WHERE STATUS != 'VALID'

In this example we want to check the number of invalid objects in the SOE schema, so we need to enter the Database Credentials for this Schema

Now we can test the results of the SQL Query by pressing [Test]
The test proofs to be successful and the value returned is 1, so apparently there is 1 invalid object in the SOE Schema.

Next we need to specify what Alert should be generated in the case there are any invalid objects.

We need to use Comparison Operator >

We entered a message that should be displayed as soon as the metric returns a value that exceeds 0

We would like the first evaluation of the metric as soon as it gets saved and we want to repeat evaluation every 15 minutes.
As you can see the metric was created successful

We have invalidated one of the objects in the SOE schema

Next we select the Alert Message to display it's details

Next we validate the invalid object in the SOE schema and press [Reevaluate Alert]
As this screenshot shows the reevaluation of the Alert caused it to be cleared, now there are no more invalid objects in the SOE schema.
Using a Notification Rule to notify the triggering of the example Alert

We will create an additional Notification Method in this example.

Navigate to Setup --> Notification Methods

Select Add "OS Command" and press [Go]

We name it MY_NOTIFICATION_METHOD
Give it a description
Enter OS Command: date >> /tmp/mynotificationmethod.txt

Press [Test OS Command]

The test of the OS Command was successful

Press [OK]
Press [OK] again

Next we need to have a Notification Rule that will notify for our example User Defined Metric and let it use our newly created Notification Method.

Select Preferences --> Rules

As our example User Defined Metric is created on a Database Instance we will create a new Notification Rule based on the "Out of the Box" Rule "Database Availability and Critical States" and press [Create Like]
We name the rule MY_NOTIFICATION_RULE and select Metrics

Press [Add] to add our example UDM

We enter 'User-Defined' as Search criteria and press [Go]

As you can see the User Defined Metric shows up

Select the second Radio Button and press the Browse Icon
In the Pop Up window we will select our example UDM and press [Select] to copy it to our previous page.

As you can see the name of our example UDM was copied to this page.

Make sure you select Critical as Severity State we want this rule to notify on and press [Continue].
As we browse through the Metrics you will notice our UDM is now included.

Next we select **Actions**

This page automatically show all additional Notification Methods available, which shows the Method we have created some minutes ago.

Select the Assign Method to Rule checkbox

Press [OK]
As you can see our Notification Rule was created successful

The only thing we need to do next is, invalidate one of the objects in the SOE Schema and wait for the UDM to generate it's Alert

After a while our UDM generates it's Alert

We will select the Alert Message
What we see next is that the Alert was generated and that our Notification Rule was executed.