Creating Plugin Service Monitors in Uptime Infrastructure Monitor

Contents

Overview

This article is part of a series:
Part 1 - Creating Custom Service Monitors in Uptime Infrastructure Monitor
Part 2 - Creating Custom Service Monitors with Retained Data Collection
Part 3 - Creating Plugin Service Monitors in Uptime Infrastructure Monitor

Plug-in service monitors allow you to add service monitors directly into the Uptime Infrastructure Monitor interface. You can use plug-in service monitors to check, alert on and graph your own custom application or business metrics. Plug-in in monitors have the following benefits:

- Plug-in monitors enable you create more meaningful graphs with customized headings for your retained application and business metrics.
- They are installed directly into the Uptime Infrastructure Monitor interface, beside the standard Uptime Infrastructure Monitor service monitors. Your plug-in monitors are as transparent as other monitors to your end users.
- Creating instances of your monitor much easier using a standard service monitor template that you define. Once you create your monitor template you never have to remember the exact arguments or location of your custom script, they are included right in the interface automatically.

Example graph produced using a plug-in monitor.
blocked URL

The process of updating your existing scripts and service monitors to be plug-in monitors can usually be completed in just a few minutes. This article builds on the scripts and knowledge that were developed in previous articles. Take some time to review the previous articles before continuing.

Formatting your monitoring station script for retained data tracking

To format your custom script for use with a plug-in monitor, you will only need to make changes to the way that it outputs information when the script is run. Instead of printing a number to screen on individual lines -- like the Custom with Retained Data -- you must also print a variable name along with the numerical or string value. Choosing a variable name is the most important part of setting up the plug-in monitor. The variable name must be exactly the same within your script output and the XML definition for your service monitor. This is discussed later in this article.

The format of plug-in monitor output is included below along with an example set of output that includes two integer variables (`transactions` and `users`) and two string values (`lasterror` and `connectmsg`). The output format is a simple name and value pairing, the variable naming coming first on a line and the value for that variable following it.

Expected format:
variable value
variableX valueX

Example Script Execution and Output
> gather_data.sh
transactions 2398
users 5
lasterror Error opening user connection!
connectmsg Connection refused

Changing the check_temp script for use within a plug-in service monitor

Using the `check_temp.sh` script as an example, change the output format so that it can be used within a plug-in monitor. You must first decide what to call the variables that the script will output. Currently, the script outputs temperature and relative humidity -- name the two variables `temp` and `rh`.

Next, change the output of the script `check_temp.sh` as detailed below:

Previous Format - For a custom service monitor with retained performance data.
> ./check_temp.sh test-agent 9998
64.5
32.8

New Format - For use within a plug-in service monitor, notice that we have added our variable names to the output lines:
> ./check_temp.sh test-agent 9998
temp 64.5
rh 32.8

If you look back through the examples in previous articles, you will notice that the output format from the monitoring station script above now matches that of the agent side script. Note that this is not always the case, but works well for this example.

In order for the `check_temp.sh` script to produce the output listed above, edit the script so that it matches the example below:

```
    #!/bin/sh

# This script takes the following arguments:
# check_temp.sh hostname port
# Example execution:
# ./check_temp.sh my-agent 9998

# This script can be placed anywhere on the monitoring station system as long as it is
# executable by the uptime user.

#First, collect our arguments
AGENT=$1
PORT=$2

TMPFILE=/tmp/$$.temp

# now use the info above to contact the agent, store the output in a file for parsing
`echo -n rexec secretpassword /opt/uptime-agent/my-scripts/show_temp.sh my-arguments | /usr/local/uptime/bin
/netcat $AGENT $PORT > $TMPFILE`
```

Note

If you are using agentcmd instead of netcat, replace netcat with agentcmd in the command above. For example:

```
    `/usr/local/uptime/scripts/agentcmd -p $PORT $AGENT rexec secretpassword /opt/uptime-agent/my-scripts
/show_temp.sh my-arguments > $TMPFILE`
```

For more information see Using the agentcmd utility.

```
    # we have the output from the agent. If it is ERR that means there was a problem running the script on the
agent
`grep ERR $TMPFILE`
if [ $? -eq 0 ]
then
echo "Could not execute agent side script!"
# by exiting with a 2 we are forcing a CRIT service outage
exit 2
fi

# because our agent side script produces output that fits into the plug-in monitor format
# we don't have to format our output at all, simply print it to the screen and we are finished.

cat $TMPFILE

exit 0
```

Creating the XML definition for your plug-in service monitor

To integrate the plug-in monitor with Uptime Infrastructure Monitor, you must produce an XML definition that Uptime Infrastructure Monitor will use to understand how to process your custom script and what options should be displayed within the Uptime Infrastructure Monitor interface. To create your XML definition browse to the Plug-in Service Monitor XML Generation Tool and follow the steps on screen.

This page will be your primary tool to create and edit your XML definition files. If you want to further customize the XML definition of your plug-in monitor, please contact uptime-support@idera.com for assistance. Before you use the XML Generation Tool, have the following information about your plug-in monitor available:

- The name of your monitor, which will appear in the Uptime Infrastructure Monitor services list. This name must be unique.
- The full path to your custom script
- For each variable that your script produces:
  - Var Name: this must match the output variable name and be only one word.
  - Title: for instance if your variable was named 'temp' an appropriate title may be 'Temperature'
  - Description: Full description of the variable.
  - Unit: if the variable should have a unit associated with it for graphing
  - Type: Either String or Decimal based on the value for your variable.

Once you have generated your XML file, save it to your monitoring station in the `UPTIME_DIRxml` folder and continue to the next step. Below is an example screen shot of the options used to create an XML definition for the `check_temp.sh` script. The XML file that these options produce is attached to this article.

Example options used to produce XML for `check_temp.sh`.
blocked URL

Importing and managing your XML plug-in monitor definition

Now that you have both the XML definition for your custom script and your custom script in place, you can import your plug-in monitor into Uptime Infrastructure Monitor. You use the `erdcloader` and `erdcdeleter` to import and export your plug-in monitor. Examples of the options for these commands are included below.

The `UPTIME_DIR/scripts/erdcloader` utility is used to import your plug-in monitor XML definition as a service monitor template within Uptime Infrastructure Monitor.

Option Name	Description
-c	Changes the default launch configuration file
-h, --help	Prints help information
-x, --xml	Defines the XML file to load to create your plug-in monitor template.

Example Execution:
> cd UPTIME_DIR
> scripts/erdcloader -x MyMonitor.xml

The `UPTIME_DIR/scripts/erdcdeleter` utility is used to remove your plug-in monitor template from Uptime Infrastructure Monitor.

Option Name	Description
-l, --list	Lists all service monitor templates that can be removed from Uptime Infrastructure Monitor
-h, --help	Prints help information
-n, --name	Deletes the service monitor template with the given name. A template must have no service monitors currently using it in order to be deleted.

Example Execution:

```
> cd UPTIME_DIR
> scripts/erdcdeleter --list
.
.
.
My test monitor template
> scripts/erdcdeleter --name "My test monitor template"
```

While working with your XML definition and ensuring that your service monitor works correctly, you will probably run each of these commands a number of times. Below is a quick example of the output expected when importing the `check_temp.xml` template using the `erdcloader` utility.

```
> cd UPTIME_DIR
> ls scripts/check_temp.sh
scripts/check_temp.sh
> ls xml/check_temp.xml
xml/check_temp.xml
> scripts/erdcloader -x xml/check_temp.xml
2006-05-12 17:09:03,376 DEBUG (HibernateManager:178) - Configuring database for: mysql
2006-05-12 17:09:08,743 DEBUG (ERDCXmlParser:56) - Plug-in monitor: Temp and Humid
```

Now that the plug-in monitor has been imported, you can browse to the Add Service Instance page in the Uptime Infrastructure Monitor user interface and see the plug-in monitor listed, as shown below:

blocked URL

Adding Instances of your plug-in service monitor

To add an instance of your plug-in service monitor, browse to the **Add Service Instance** list, select your plug-in service monitor, and click **Continue**. The options displayed on the monitor template match the definition that you created with the XML generation tool. The service monitor settings that would appear based on the example XML definition are shown below:

blocked URL