Plugin Guide

The following describes how to define and package a plugin that is fully compliant with The Grid, and Uptime Infrastructure Monitor's Extension Manager. Creating a compliant plugin ensures it is correctly categorized, and can be installed as seamlessly as possible from within Uptime Infrastructure Monitor. A plugin has the following guidelines:

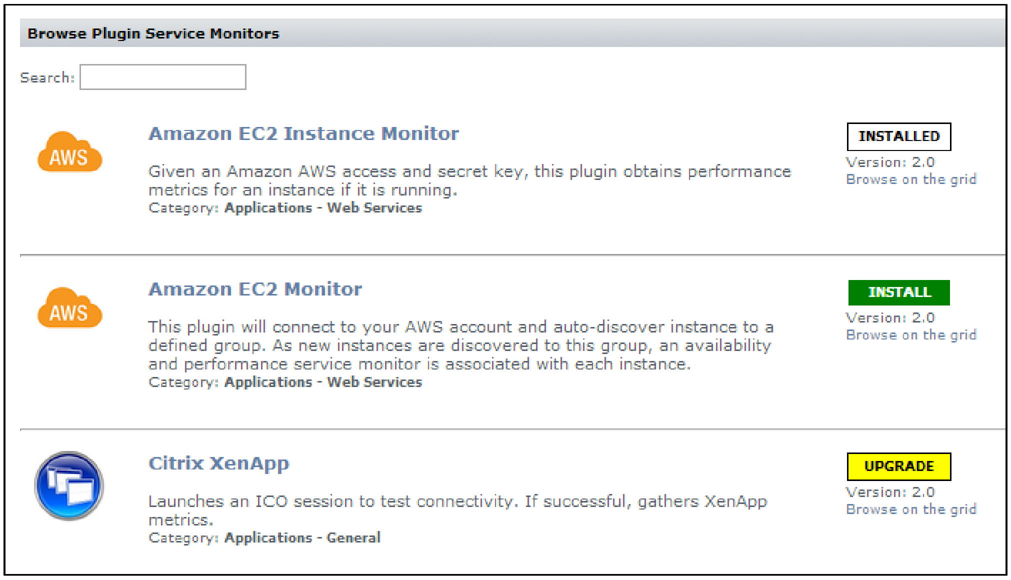
- the main distributable is in .ZIP format
- the plugin scripts and other contents are structured in subdirectories:
  - `/files`: contains all files commonly used across different platforms (required)
  - `/files-posix`: contains Linux-specific files (optional)
  - `/files-win`: contains Windows-specific files (optional)
- an XML file at the root of the distributable that defines the plugin

Be sure to check out the following series of topics that help you create unique service monitors in Uptime Infrastructure Monitor:

- Creating Custom Service Monitors in Uptime Infrastructure Monitor
- Creating Custom Service Monitors with Retained Data Collection
- Creating Plugin Service Monitors in Uptime Infrastructure Monitor

Extension Manager: Seamless Plugin and Gadget Management

Extension Manager allows you to browse and manage Grid-hosted plugin monitors and dashboard gadgets from within the Uptime Infrastructure Monitor Web interface. You can seamlessly check for updates, and install new plugins or updates without requiring the installation of the Plugin Manager.



Browse Plugin Service Monitors

Search:

Amazon EC2 Instance Monitor

Given an Amazon AWS access and secret key, this plugin obtains performance metrics for an instance if it is running.
Category: **Applications - Web Services**

INSTALLED
Version: 2.0
Browse on the grid

Amazon EC2 Monitor

This plugin will connect to your AWS account and auto-discover instance to a defined group. As new instances are discovered to this group, an availability and performance service monitor is associated with each instance.
Category: **Applications - Web Services**

INSTALL
Version: 2.0
Browse on the grid

Citrix XenApp

Launches an ICO session to test connectivity. If successful, gathers XenApp metrics.
Category: **Applications - General**

UPGRADE
Version: 2.0
Browse on the grid

Understanding the Plugin's XML Definition

A plugin's XML definition performs several key tasks:

- provides metadata for the plugin for categorization and management in Uptime Infrastructure Monitor's Extension Manager and Add Service Monitor page
- indicates the core script and auxiliary files that are required to run the plugin monitor
- defines the input fields on the plugin's configuration page that allow users to pass data to the plugin when it is run
- defines the output that the plugin returns to Uptime Infrastructure Monitor (to be included in alert messages, or for retention in the DataStore for graphing and reporting), and allows users to define comparative criteria for returned output that triggers alerts

Each input or output field on the plugin's configuration page is built from instances of <element>, which is a child element of <elements>. The following is an example of an XML definition that includes three input fields and three output fields.

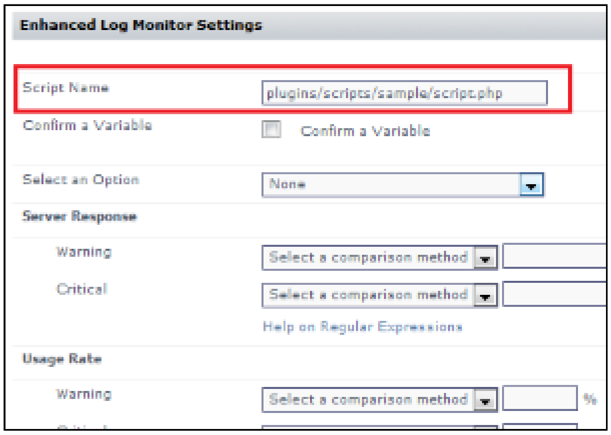
Plugin XML Definition

```
<uptime ...
  ... target_grouping="other">
<exec>com.uptimesoftware.uptime.
CustomScriptRunner</exec>
  <class>Enhanced Log Monitor</class>
  <erdc_version>1.0</erdc_version>?
  <category>Operating System Monitors<
/category>
  <category_related>Domain Services<
/category_related>
  <grid_info>
    <supported_versions>7.2,7.3<
/supported_versions>
    <supported_platforms>windows<
/supported_platforms>
    <requires_agent_scripts>true<
/requires_agent_scripts>
    <upgrade_in_place>true<
/upgrade_in_place>
  </grid_info>
  <help>tooltip text on add monitors
page</help>
  <elements>
    [plugin input/output settings
definition]
  </elements>
</uptime>
```

- <exec>: the Java application executed to collect data
- <class>: the name of the plugin as it appears in the list of Uptime Infrastructure Monitor service monitors
- the version-, category-, and grid-related elements define the plugin as a manageable asset within an Uptime Infrastructure Monitor deployment
- <help>: elaborate on the plugin purpose; appears as a tool tip
- <elements>: use <group> and <element> child elements to create configuration fields (see below for more information)

Plugin Input Settings Definition

```
<element name="mandatoryInputField"
parameter_type="input"
 data_type="string" gui_basic="1"
range_type="0">
  <gui_type> <string/> </gui_type>
  <control_options>size:40 <
/control_options>
  <default_value>plugins/scripts/sample
   /script.php</default_value>
  <short_description>Script Name<
/short_description>
  <long_description>script to execute<
/long_description>
  <validation_rule><alphanumeric/><
/validation_rule>
  <error_message>
    <alphanumeric>this is a required
field</alphanumeric>
  </error_message>
</element>
```



Enhanced Log Monitor Settings

Script Name: plugins/scripts/sample/script.php

Confirm a Variable: □ Confirm a Variable

Select an Option: None

Server Response

Warning: Select a comparison method

Critical: Select a comparison method

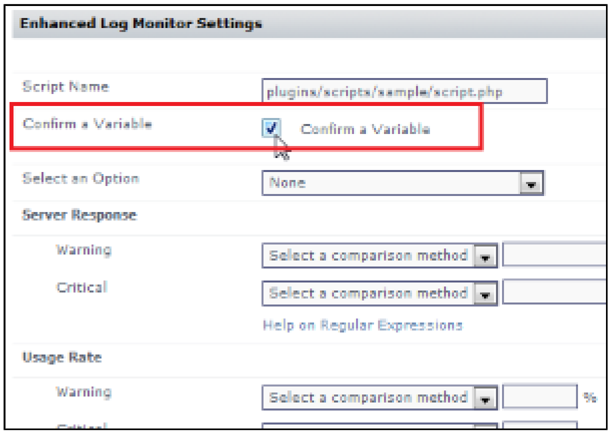
Help on Regular Expressions

Usage Rate

Warning: Select a comparison method %

This first input element is a mandatory input field that accepts a string value, denoting a script name that will be executed by the plugin.

```
<element name="CheckBox"
parameter_type="input"
 data_type="boolean" gui_basic="1"
range_type="0">
  <gui_type> <boolean/> </gui_type>
  <control_options>size:40 <
/control_options>
  <default_value/>
  <short_description>Confirm Variable<
/short_description>
  <long_description>enable a variable<
/long_description>
  <validation_rule/>
  <error_message/>
</element>
```



Enhanced Log Monitor Settings

Script Name: plugins/scripts/sample/script.php

Confirm a Variable: ☑ Confirm a Variable

Select an Option: None

Server Response

Warning: Select a comparison method

Critical: Select a comparison method

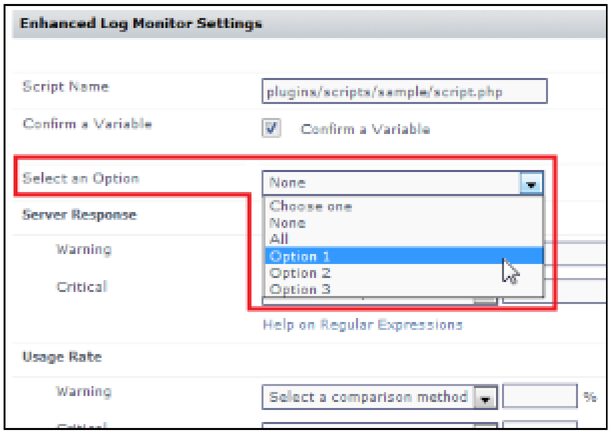
Help on Regular Expressions

Usage Rate

Warning: Select a comparison method %

The second input element is a check box that passes a Boolean to the executed script.

```
<element name="mandatoryInputField"
parameter_type="input"
 data_type="string" gui_basic="1"
range_type="0">
  <gui_type>
    <single_select type="dropdown">
      <value>None</value>
      <value>All</value>
      <value>Option 1</value>
      <value>Option 2</value>
      <value>Option 3</value>
    </single_select>
  </gui_type>
  <control_options>size:8 <
/control_options>
  <default_value>None</default_value>
  <short_description>Select an Option<
/short_description>
  <long_description>select from a list<
/long_description>
  <validation_rule><alphanumeric/><
/validation_rule>
  <error_message>
    <alphanumeric>this is a required
field</alphanumeric>
  </error_message>
</element>
```



Enhanced Log Monitor Settings

Script Name: plugins/scripts/sample/script.php

Confirm a Variable: ☑ Confirm a Variable

Select an Option: None

- Choose one
- None
- All
- Option 1
- Option 2
- Option 3

Server Response

Warning

Critical

Help on Regular Expressions

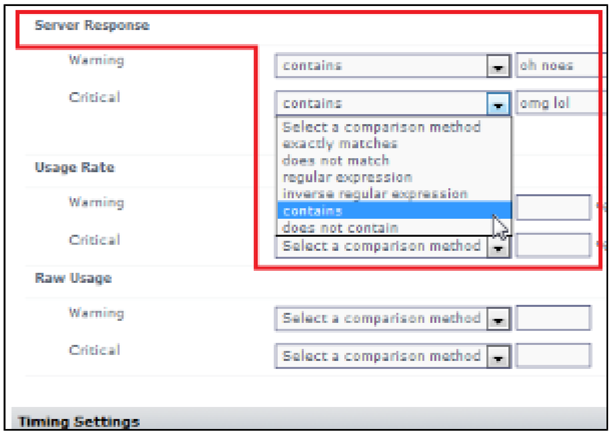
Usage Rate

Warning: Select a comparison method %

The third input element is a drop-down that passes one of several string-based options to the executed script.

Plugin Outputs Settings Definition

```
<element name="
serverResponse"
parameter_type="output"
 data_type="string" gui_ba
sic="1" range_type="0"
units="">
  <control_options>size:
8</control_options>
  <default_value/>
  <short_description>Serve
r Response<
/short_description>
  <long_description>compar
e a string<
/long_description>
  <validation_rule/>
  <error_message/>
  <gui_type><string/><
/gui_type>
</element>
```



Server Response

Warning: contains | oh noes

Critical: contains | omg lol

- Select a comparison method
- exactly matches
- does not match
- regular expression
- inverse regular expression
- contains
- does not contain

Usage Rate

Warning

Critical: Select a comparison method

Raw Usage

Warning: Select a comparison method

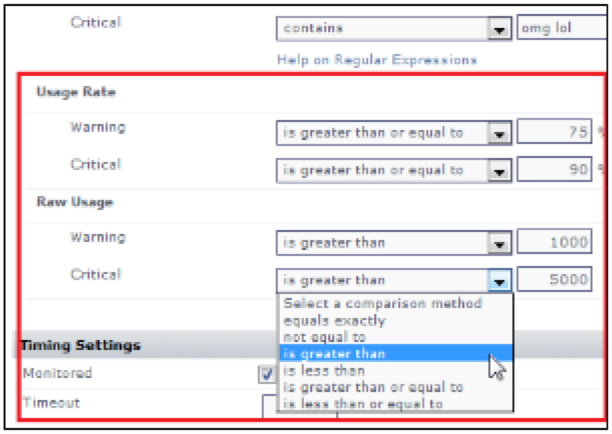
Critical: Select a comparison method

Timing Settings

For the first output element, the plugin accepts a string value that will be compared against output from the executed script. The string data type determines the comparative options to determine warning and critical conditions.

```
<group name="Specific
Metric Type Check"
switch="0">
  <short_description/>
  <long_description/>
  <element name="
usageRate",
parameter_type="output"
   data_type="integer"
range_type="0" units="%">
    <control_options>size:
8</control_options>
    <default_value/>
    <short_description>Usa
ge Rate<
/short_description>
    <long_description>retu
rned integer<
/long_description>
    <validation_rule/>
    <error_message/>
   <gui_type><integer/><
/gui_type>
  </element>

  <element name="
usageRaw" parameter_type="
output"
   data_type="ranged"
gui_basic="1" range_type="
0"
   units="">
    <control_options>size:
8</control_options>
    <default_value/>
    <short_description>Raw
Usage</short_description>
    <long_description>retu
rned range<
/long_description>
    <validation_rule/>
    <error_message/>
    <gui_type><decimal/><
/gui_type>
  </element>
</group>
```



Critical: contains | omg lol

Help on Regular Expressions

Usage Rate

Warning: is greater than or equal to | 75 %

Critical: is greater than or equal to | 90 %

Raw Usage

Warning: is greater than | 1000

Critical: is greater than | 5000

Timing Settings

Monitored ☑

Timeout

The next two output elements are similar enough to be paired on the configuration screen using <group> as a parent element. The name attribute is the grouping label in the user interface. The units attribute adds a label after the field to assist the user.

The first of the two elements accepts integer values and compares them to integer values returned through the script.

The second of the two elements accepts decimal values and compares them to a range of values returned through the script.

Definition XML Reference

The following table summarizes valid values for the various XML elements, child elements, and attributes that define a plugin.

Child element of <elements>	Description
`element hidden`	Determines whether the plugin is hidden in the Uptime Infrastructure Monitor interface. Enabling this allows you to create hidden plugins. optional; default false
`class`	The name of the plugin as displayed in the Uptime Infrastructure Monitor interface (for example, the Add Service Monitor page).
`exec`	References to any support classes that are required to run the plugin.
`erdc_version`	The version scheme supported is 0.0 (major and minor version). A minor-minor version (for example, 1.2.1) is not processed.

`category`	The main category this plugin belongs to. This will help others find the plugin on the Grid, and this is where the plugin will be listed on the Add Monitors page. - Advanced and Script Monitors - Applications - All Types - Applications - Databases - Applications - Email - Applications - General - Applications - Web Services - Domain Services - End User Experience Monitors - Network Device Monitors - Network Service Monitors - Operating System Monitors - Storage Monitors - VMware Monitors
`category_related`	Other categories the plugin is related to, separated by commas.
`grid_info`	This element wraps child elements that describe the plugin specifically for the Grid.

Child element of <grid_info>	Description
`supported_versions`	The versions of the Uptime Infrastructure Monitor Monitoring Station supported by this plugin. The version scheme supported must match Uptime Infrastructure Monitor releases (in 0.0 format), and it needs to be a comma-separated list of every version supported.
`supported_platforms`	possible values: - linux - windows - linux, windows
`requires_agent_scripts`	If "true," indicates that the administrator is also going to have to install or deploy agent-side scripts as a follow-up step to installing the plugin on the Monitoring Station. If this is enabled, the user will see a related message on the Extension Manager, directing them to the plugin's Grid page for more information. optional; default false
`upgrade_in_place`	If "true," indicates this version will overwrite previous versions of the plugin as part of an automatic upgrade. optional; default false
`help`	Help text for the plugin that is displayed when the user screams "help!" as loud as they can. lol j/k – it's what appears as a tool tip for the plugin on the Add Service Monitor page.
`elements`	Describes what the plugin processes and inputs and outputs, and how these are presented to, and configured by, Uptime Infrastructure Monitor users.

Attributes of <elements>	Description
`name`	The label of the input or output field in the Uptime Infrastructure Monitor interface, during plugin configuration.
`parameter_type`	Determines whether this child element is taking data from the Uptime Infrastructure Monitor user and passing it to the plugin script, or taking data output by the plugin script for use in Uptime Infrastructure Monitor. Possible values: - input - output
`data_type`	The expected data type that will be sent to, or returned by, the plugin script. - boolean - datetime - decimal - integer - ranged (output only) - string

`gui_basic`	Determines whether the element appears in a standard or advanced view in the configuration screen. default enabled (1)
`range_type`	default is disabled (0)
`units`	adds a label in the GUI to help the user determine what to input

Child elements of <element>	Description
`control_options`	Determines the size of the text input field during plugin configuration.
`default_value`	Populates the field with a default value.
`short_description`	The name of the configuration field in the Uptime Infrastructure Monitor interface.
`long_description`	The description that appears as a tool tip.
`validation_rule`	Optionally use a child element (for example, <integer/> or <alphanumeric/>) to indicate how to validate the input data.
`error_message`	Message provided to the user if the validation rule fails.

`gui_type`	Indicates how the element is presented and configured in the UI. Use one of the following child elements:
	• boolean
	• decimal
	• integer
	• password
	• single_select
	• string
	The single_select child element uses a type attribute (for example, dropdown or radio) that indicates the type of GUI selection widget that will be used. A value child element then defines each of the selector options. For example:
	`<gui_type>`
	`  <single_select type="dropdown">`
	`    <value>1</value>`
	`    <value>2</value>`
	`    <value>3</value>`
	`  </single_select>`
	`</gui_type>`

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, such as 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
```

Creating the XML definition for your plug-in service monitor

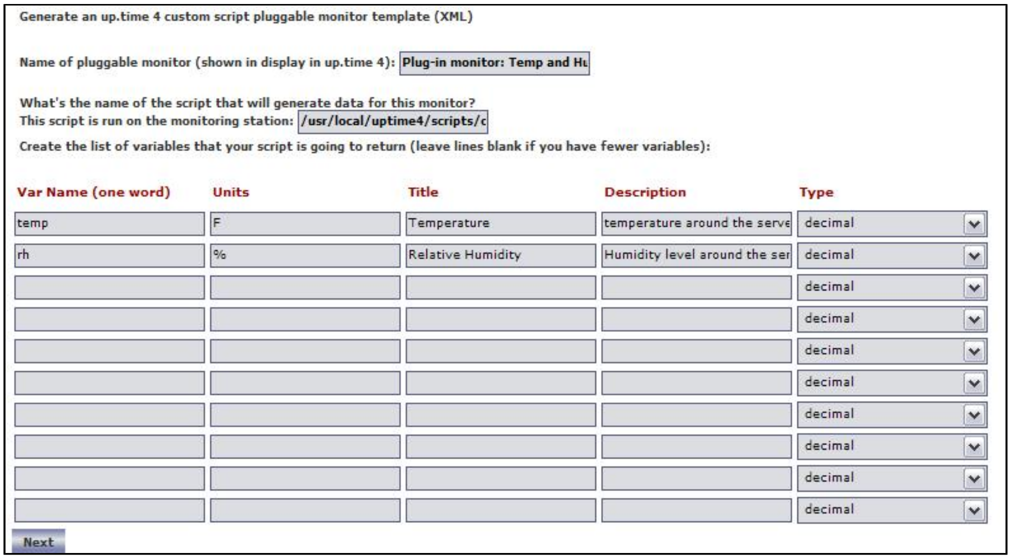
To integrate the plug-in monitor with Uptime Infrastructure Monitor, you must produce an XML definition that Uptime Infrastructure Monitor will use to understand how to process your custom script and what options should be displayed within the Uptime Infrastructure Monitor interface. To create your XML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



Generate an up.time 4 custom script pluggable monitor template (XML)

Name of pluggable monitor (shown in display in up.time 4): Plug-in monitor: Temp and Hu

What's the name of the script that will generate data for this monitor?
This script is run on the monitoring station: /usr/local/uptime4/scripts/c

Create the list of variables that your script is going to return (leave lines blank if you have fewer variables):

Var Name (one word)	Units	Title	Description	Type
temp	F	Temperature	temperature around the serve	decimal
rh	%	Relative Humidity	Humidity level around the ser	decimal
				decimal
				decimal
				decimal
				decimal
				decimal
				decimal
				decimal
				decimal

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.

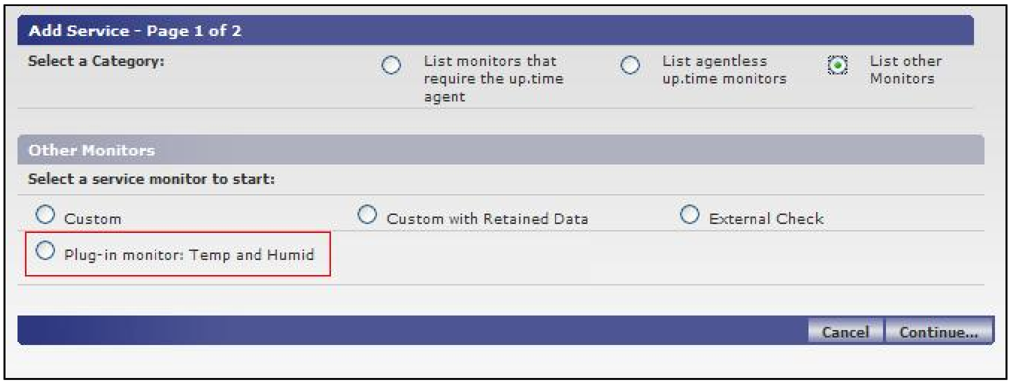
Example Execution:

```
> 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:



Add Service - Page 1 of 2

Select a Category: ○ List monitors that require the up.time agent ○ List agentless up.time monitors ● List other Monitors

Other Monitors

Select a service monitor to start:

○ Custom ○ Custom with Retained Data ○ External Check

○ Plug-in monitor: Temp and Humid