

Plugin Guide

The following describes how to define and package a plugin that is fully compliant with the uptime software Grid, and up.time's Extension Manager. Creating a compliant plugin ensures it is correctly categorized, and can be installed as seamlessly as possible from within up.time. 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

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 up.time'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 up.time (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	
<pre><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>Domai n Services< /category_related> <grid_info> <supported_versions>7 .2,7.3< /supported_versions> <supported_platforms> windows< /supported_platforms> <requires_agent_scrip ts>true< /requires_agent_scripts> <upgrade_in_place>tru e</upgrade_in_place> </grid_info> <help>tooltip text on add monitors page</help> <elements> [plugin input/output settings definition] </elements> </uptime></pre>	▪ <exec>: the Java application executed to collect data ▪ <class>: the name of the plugin as it appears in the list of up.time service monitors ▪ the version-, category-, and grid-related elements define the plugin as a manageable asset within an up.time 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>Scri
  pt Name<
  /short_description>
  <long_description>scri
  pt to execute<
  /long_description>
  <validation_rule><alpha
  numeric/><
  /validation_rule>
  <error_message>
    <alphanumeric>this
    is a required field<
    /alphanumeric>
    </error_message>
  </element>

```

The screenshot shows the 'Enhanced Log Monitor Settings' window. The 'Script Name' field is highlighted with a red rectangle and contains the text 'plugins/scripts/sample/script.php'. Below it, there is a 'Confirm a Variable' checkbox which is currently unchecked. Further down, there are sections for 'Server Response' and 'Usage Rate', each with 'Warning' and 'Critical' levels, and a 'Select a comparison method' dropdown menu.

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>Conf
  irm Variable<
  /short_description>
  <long_description>enabl
  e a variable<
  /long_description>
  <validation_rule/>
  <error_message/>
  </element>

```

The screenshot shows the 'Enhanced Log Monitor Settings' window. The 'Confirm a Variable' checkbox is highlighted with a red rectangle and is checked. The 'Script Name' field above it contains 'plugins/scripts/sample/script.php'. The rest of the interface is the same as in the previous screenshot.

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>Sele
ct an Option<
/short_description>
  <long_description>selec
t from a list<
/long_description>
  <validation_rule><alpha
numeric/><
/validation_rule>
  <error_message>
    <alphanumeric>this
is a required field<
/alphanumeric>
  </error_message>
</element>

```

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_b
asic="1" range_type="0"
units="">
  <control_options>size:
8</control_options>
  <default_value/>
  <short_description>Serv
er Response<
/short_description>
  <long_description>compa
re a string<
/long_description>
  <validation_rule/>
  <error_message/>
  <gui_type><string/><
/gui_type>
</element>

```

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>Us
age Rate<
/short_description>
    <long_description>ret
urned 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>Ra
w Usage<
/short_description>
    <long_description>ret
urned range<
/long_description>
    <validation_rule/>
    <error_message/>
    <gui_type><decimal/><
/gui_type>
  </element>
</group>

```

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 up.time interface. Enabling this allows you to create hidden plugins. optional; default false
class	The name of the plugin as displayed in the up.time 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 up.time Monitoring Station supported by this plugin. The version scheme supported must match up.time 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, up.time users.

Attributes of <elements>	Description
name	The label of the input or output field in the up.time interface, during plugin configuration.
parameter_type	Determines whether this child element is taking data from the up.time user and passing it to the plugin script, or taking data output by the plugin script for use in up.time. 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 up.time 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: <pre> <gui_type> <single_select type="dropdown"> <value>1</value> <value>2</value> <value>3</value> </single_select> </gui_type> </pre>