

Custom Tasks

- [Create a New Task Template Using XML](#)
 - [XML File Types](#)
 - [The Task Library](#)
 - [Create your task.xml file](#)
 - [Create your run.xml file](#)
 - [Package the XML files as a zip archive](#)
 - [Add the zip archive as a task template](#)
- [Create Custom Tasks Using Shell or Batch Scripts](#)
- [Create Your Own Task Launcher Based on a Java Class](#)
- [Learn More](#)

There are three ways to create custom tasks for your workflows.

Create a New Task Template Using XML

AppDynamics provides a set of default task templates. See [Available Task Templates](#). You can also create your own task templates using XML and the default task launcher, AntTask.

XML File Types

You must create two XML files to create a new template. You must use these names.

- task.xml: defines the argument list for the input/output parameter tabs on the UI and links it to the task launcher
- run.xml: defines the syntax for running the task command

The Task Library

Look through the Task Library to find a template similar to the one you are creating and download the template zip file.

1. In the left navigation pane of the UI, select the appropriate application
2. Click **Alert & Respond -> Cloud Auto-Scaling -> Tasks**
3. Click **New+**. The Create Task dialog opens.
4. Click **View All Task Templates** at the bottom left. The Task Template Library opens in the main window.
5. Click Cancel to close the Create Task dialog.

6. Find a similar task template.
7. In the Task Zip File, click the zipfile name.
8. Open the zip file. It should contain both a run.xml and a task.xml. If it doesn't, download a different zip file.

Create your task.xml file

Using the downloaded task.xml file as a model, create your task.xml. See the BZip2 sample below.

- Choose a name, display-name, and a description to show up in the task template list.
- The type is always java.
- The task-argument element contains as many argument elements as the task command has arguments.
 - The name attribute is the name displayed in the Input Parameters tab UI.
 - The description attribute is the content of the tool tip displayed in the Input Parameters tab UI.
 - The is-required attribute indicates if the argument is required
 - Additional attributes, like type, allowed values, etc. depend on the task.
- The java-task and impl-class elements always contain com.singularity.ee.agent.systemagent.task.ant.AntTask. This is the task launcher class.

```
<task>
  <name>BZip2</name>
  <display-name>BZip2</display-name>
  <description>Zips to a bzip file using BZip2 algorithm</description>
  <type>java</type>
  <task-arguments>
    <argument name="source-file" description="Source tar file to zip"
      is-required="true" />
    <argument name="destination-file" description="Destination file to create"
      is-required="true" />
  </task-arguments>
  <java-task>
    <impl-class>com.singularity.ee.agent.systemagent.task.ant.AntTask
    </impl-class>
  </java-task>
</task>
```

Create your run.xml file

Using the downloaded run.xml file as a model, create your run.xml. See the BZip2 sample below.

- You must begin this file with an XML declaration.
- The project, property, and target names must be as they are in the sample. The file `args.properties` is populated based on input from the Input Parameters UI
- The element inside the target element is the syntax to run the task command. The arguments come from `args.properties`.

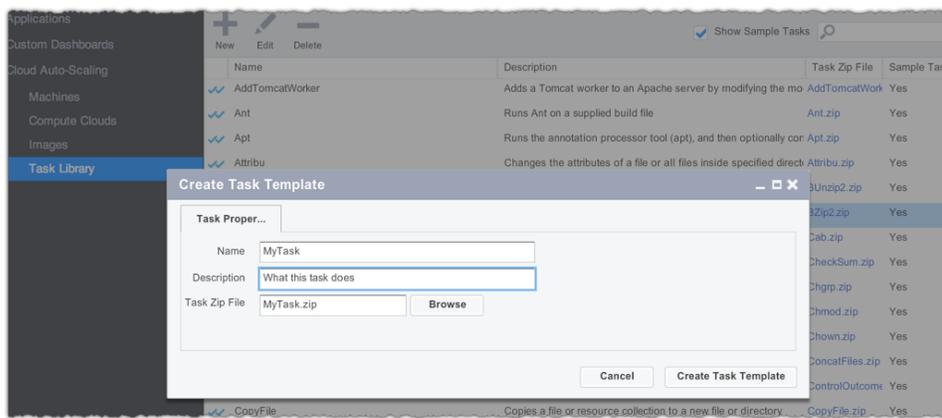
```
<?xml version="1.0" encoding="UTF-8"?>
<project name="run" default="run" basedir=".>
  <property file="args.properties" />
  <target name="run">
    <bzip2 src="\${source-file}" destfile="\${destination-file}" />
  </target>
</project>
```

Package the XML files as a zip archive

Add the task.xml and run.xml files to a zip archive and name this zip archive the name of the task.

Add the zip archive as a task template

1. From the **Cloud Auto-Scaling** menu select **Task Library**.
2. Click **New+**. The Create Task Template dialog appears.



3. Provide the name and description for the task.
4. Upload the Zip file containing the XML task files.
5. Click **Create Task Template**.

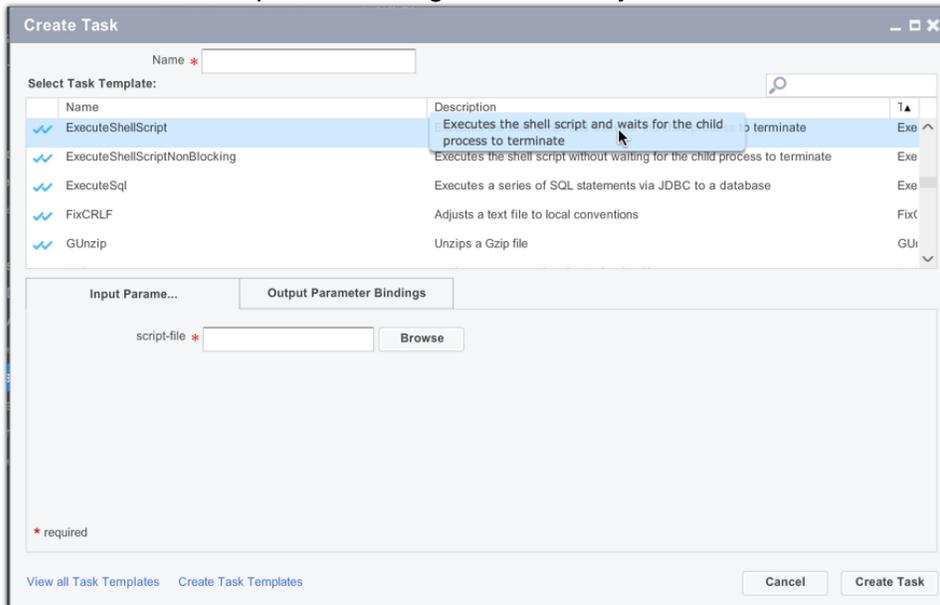
AppDynamics adds the task template to the Task Library. See [Add Tasks for Workflow Steps](#) for using this template.

Create Custom Tasks Using Shell or Batch Scripts

You can also create custom tasks by using a shell or batch script.

1. Create the shell or batch script file on the system.
2. In the left nav bar of the UI, select an application.
3. Click **Alert & Respond -> Cloud Auto-Scaling -> Tasks**.
4. Click **New +**. The Create Task window opens.

5. Give the task a name.
6. From the task list, select the task template `ExecuteShellScript` or `ExecuteShellScriptNonBlocking`, as necessary.



7. Browse to find the script file.
8. Click **Create Task**.

Create Your Own Task Launcher Based on a Java Class

If you need to create a completely new task that cannot be accomplished by using the `AntTask` launcher or by using a shell or batch script, you must create a Java class that implements the `public com.singularity.ee.agent.systemagent.api.ITask` interface bundled with the Machine Agent.

Your class must implement the following two methods:

- `execute()`, which takes input in the form of name-value pairs and the execution context for the task, and returns the task output
- `stop()`, which stops the process immediately

This class is referenced in the `task.xml` file, in the `<java-task>` element.

AppDynamics highly recommends using shell/batch scripts instead whenever possible. See for more information [Create Custom Tasks Using Shell or Batch Scripts](#).

Contact the [AppDynamics Support Team](#) if you need additional help in creating a Java task.

Learn More

- [Workflow Overview](#)
- [Add Tasks for Workflow Steps](#)