

ExecuteAction

1. Description

When the ExecuteAction type is specified to the [controller task](#) the described command handler is executed within the specified context.

2. Parameters

Attribute	Description	Required
strategy	The execution strategy specifies which internal dispatcher to use to execute the command. The strategy can be one of the following values:	No. Defaults to ant.
	ant	Run the command using Ant task
	antfetch	Run the command using ant-task
	exec	Run the command using ExecT via the ad command.
	nodedispatch	Run the command but first look the deployments.properties file matches one on the local host executed using the <i>ant</i> strategy found to be on another node, then is dispatched over ssh.
failonerror	Set false to not cause the command to fail if there is an error.	No.
return	Property to set with return values	No. Useful only with antfetch strategy.
adExecutable	Path to the ad executable.	No. Useful only with exec strategy.
adArgs	Arguments to pass to the ad	No. Useful only with exec

	command.	strategy.
--	----------	-----------

3. Parameters specified as nested elements

command

A [command type](#) describes what command name to run. Only the command's name is a required attribute. The other attributes are ignored by ExecuteAction.

The example below shows how to specify to run a command named Status

```
<command name="Status" />
```

context

A [context type](#) describes in what context the command handler should run. There are three primary components of the context relevant to executed actions: depot, entityClass and entityName. The depot attribute describes which project to find the object or type in. The entityClass attribute describes what type (and indirectly which module) to run in. The entityName attribute specifies a particular object's environment to run in.

The example below shows how to specify a fully qualified object context.

```
<context depot="ContentApp" entityClass="Apache" entityName="apache" />
```

If it is desirable to invoke a command handler but not execute the handler within the context of a specific object, a type-level context can be specified. The example below suggests there is a type named Utility used to run various administrative procedures. Note the entityName attribute is omitted.

```
<context depot="ContentApp" entityClass="Utility" />
```

property

Additional properties can be passed into the context of the called command by using the property element.

The example below shows how one would define the property named foo with the value bar.

```
<property name="foo" value="bar" />
```

workflow

A [workflow type](#) describes a task sequence to execute protected by a configurable [errorhandler](#) element.

4. Examples

Given the choice of ExecuteAction strategies and related attributes, there are a variety of methods to execute a command. Several examples are shown below:

ExecuteAction

Call the Mysql Start command within the specified object context.

```
<controller>
  <execute>
    <context depot="ContentApp"
             entityClass="Mysql"
             entityName="mysql" />
    <command name="Start" />
  </execute>
</controller>
```

Call the Apache Status command using the Antfetch task and return the result passed back as the "isUp" property.

```
<controller>
  <execute strategy="antfetch" return="isUp">
    <context depot="ContentApp"
             entityClass="Apache"
             entityName="apache" />
    <command name="Status" />
  </execute>
</controller>
<property name="apache.up" value="${isUp}" />
```

Run the Status command using the exec strategy, specifying the path to the ad executable and saving the output in the property named "output".

```
<controller outputproperty="output">
  <execute strategy="exec"
           adexecutable="/usr/local/antdepo-1.2.8/bin/ad">
    <context depot="${depot.name}"
             entityClass="Mysql"
             entityName="mysql" />
    <command name="Status" />
  </execute>
</controller>
```