

ExecuteAction

Table of contents

1 Description.....	2
2 Parameters.....	2
3 Parameters specified as nested elements.....	2
4 Examples.....	3

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	<p>The execution strategy specifies which internal dispatcher to use to execute the command. The strategy can be one of the following values:</p> <table border="1"> <tr> <td>localdispatch</td> </tr> <tr> <td>nodedispatch</td> </tr> <tr> <td>localfetch</td> </tr> </table>	localdispatch	nodedispatch	localfetch	No. Defaults to "localdispatch".
localdispatch					
nodedispatch					
localfetch					
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 localfetch 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"/>
```

nodeset

Applicable if nodedispatch execution strategy has been specified. See [NodeSet](#) page for information about its usage.

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:

Example: localdispatch

Call the netutil listening command for the specified port using localdispatch.

```
<controller>  
  <execute strategy="localdispatch">
```

```

    <context depot="\${context.depot}"/>
    <command name="listening" module="netutil"/>
    <arg line="-port 80"/>
  </execute>
</controller>

```

Note, `localdispatch` is the default execution strategy so you don't have to explicitly set it.

Example: `nodedispatch`

Invoke the `netutil listening` command across all hosts that start with the name "web" via `nodedispatch` strategy.

```

<controller>
  <execute strategy="nodedispatch">
    <context depot="\${context.depot}"/>
    <command name="listening" module="netutil"/>
    <arg line="-port 80"/>
    <nodeset>
      <include name="web.*"/>
    </nodeset>
  </execute>
</controller>

```

Example: `localfetch`

Call the `Apache Status` command using the `localfetch` strategy and return the result passed back as the "isUp" property.

```

<controller>
  <execute strategy="localfetch" return="isUp">
    <context depot="ContentApp"
      entityClass="Apache"
      entityName="apache"/>
    <command name="Status"/>
  </execute>
</controller>
<echo message="apache up?: ${isUp}"/>

```

Assumes the `Status` command sets a property named "isUp"

Example: in line properties

Pass property definitions in line to the called command:

```

<controller>
  <execute>
    <context depot="ContentApp"
      entityClass="Apache"
      entityName="isListening"/>
    <command name="Status"/>
  </execute>

```

ExecuteAction

```
<property name="opts.port" value="81"/>  
  <property name="opts.server" value="web1.local"/>  
</execute>  
</controller>
```

Assumes the Status command sets a property named "isUp"