

# Report Commands

## Table of contents

1 Report Commands.....	2
------------------------	---

## 1. Report Commands

The purpose of these commands is to explicitly write something to the report. Usually (with assertion commands) only a failed assertion is reported in the output document. The commands explained here are used to add more data and arbitrary text to the output.

### 1.1. ReportBlock

This command can be used everywhere in an instruction file. It can be indefinitely nested and can always contain all tags its parent tag would allow. The effect of ReportBlock is that in the output file a corresponding output tag <Block> is added. The tag contains a copy of all attributes from the ReportBlock command. So the ReportBlock can be used to add any arbitrary text to the output file. Apart from that the <Block> in the output will also contain all output tags corresponding to the inner tags of the ReportBlock instruction.

An additional feature is, that ReportBlock supports conditional execution of its inner tags. With the attributes *if* and *ifNot* it is possible to make the execution of the inner tags depending on the value of a variable or an even more complex expression.

#### 1.1.1. Attributes

The following table lists all attributes that can be used with the *ReportBlock* command.

Attribute Name	Description	required/optional
if	The inner tags of ReportBlock will only be executed if the condition in this attribute evaluates to <b>true</b> . Here <b>true</b> actually means any of the following string values: <ul style="list-style-type: none"> <li>• true</li> <li>• yes</li> <li>• on</li> <li>• 1</li> </ul>	optional
ifNot	The inner tags of ReportBlock will only be executed if the condition in this attribute is <b>NOT</b> true. For a description of the syntax of conditions see the <i>if</i> attribute.	optional
beforeVersion	The inner tags of ReportBlock	optional

	<p>will only be executed if the condition in this attribute is true. The condition specifies the name of a variable and a version number. The value of the variable then will be compared against the version number. The condition evaluates to true if the version value of the variable is a lower version than the specified value.</p> <p>The variable name and the version to compare with must be separated by a colon (':').</p> <p>Example:  <b>beforeVersion="build.version:</b></p> <p>The following values for variable <i>build.version</i> will evaluate the example to true:</p> <ul style="list-style-type: none"> <li>• 1.0.45</li> <li>• 4.3.27</li> <li>• 4.2</li> </ul> <p>The following values for variable <i>build.version</i> will evaluate the example to false:</p> <ul style="list-style-type: none"> <li>• 5.1.3</li> <li>• 4.3.28</li> <li>• 4.10.2</li> <li>• 12.2.7</li> <li>• 4.3.102</li> </ul>	
sinceVersion	<p>The inner tags of ReportBlock will only be executed if the condition in this attribute is true. The condition specifies the name of a variable and a version number. The value of the variable then will be compared against the version number. The condition evaluates to true if the version value of the variable is a higher or equal version compared to the specified value.</p> <p>The variable name and the version to compare with must</p>	optional

	<p>be separated by a colon (':').  Example:  <b>sinceVersion="build.version:3.</b>  The following values for variable <i>build.version</i> will evaluate the example to true:</p> <ul style="list-style-type: none"> <li>• 3.0</li> <li>• 3.21.7</li> <li>• 3.21.11</li> <li>• 11.0.1</li> <li>• 3.123.2</li> </ul> <p>The following values for variable <i>build.version</i> will evaluate the example to false:</p> <ul style="list-style-type: none"> <li>• 2.4.5</li> <li>• 3.21.6</li> <li>• 3.20.46.5</li> </ul>	
--	--	--

### 1.1.2. Examples

Example	Description
<pre>&lt;ReportBlock label="First test set"&gt;   &lt;AssertExistence element="....." /&gt; &lt;/ReportBlock&gt;</pre>	Simply creates a <Block> element in the output around the report results of the included child elements.

## 1.2. ReportVariable

The purpose of this command is to write the current value of a variable to the output.

### 1.2.1. Attributes

The following table lists all attributes that can be used with this command.

Attribute Name	Description	required/optional
name	Specifies the name of the variable to be reported. The specified variable must exist. Otherwise an error msgid="SERR002" will reported.	<b>required</b>

### 1.2.2. Examples

```
<Set name="title" value="Example"/>
<SettingsFile type="properties" name="config.properties">
  <Set name="title" value="inside"/>
  <ReportVariable name="title"/>
</SettingsFile>
<ReportVariable name="title"/>
```

**Description:** The first <ReportVariable> command will add

```
<Variable name="title">inside</Variable>
```

and the second <ReportVariable> command will add

```
<Variable name="title">Example</Variable>
```

to the output report file.

### 1.3. ReportValue

With this command it is possible to write the current value(s) of a configuration element to the report.

**Note:**

This command must be used only inside a *Data Source Adapter* tag.

#### 1.3.1. Attributes

The following table lists all attributes that can be used with this command.

Attribute Name	Description	required/optional
element	Specifies the element to be reported. Be aware that the syntax of this attribute depends on the <i>Data Source Adapter</i> this command is used with. Refer to the appropriate <i>Data Source Adapter</i> documentation for details.	<b>required</b>

#### 1.3.2. Examples

```
<SettingsFile type="manifest" name="base.jar/META-INF/MANIFEST.MF">  
  <ReportValue element="Specification-Title"/>  
  <ReportValue element="Specification-Version"/>  
  <ReportValue element="Specification-Vendor"/>  
</SettingsFile>
```

**Description:** Writes the values of the "Specification" settings in the manifest file of base.jar to the report.