Greg Watson

PTP User/Developer Meeting, Chicago, September 2012

# Adding Support For a New Resource Manager

# Introduction

- Based on The (JAXB) Configurable Resource Manager for PTP by Albert L. Rossi
  - http://wiki.eclipse.org/images/2/28/JAXBDemo.pdf
- Documentation describing how to create a JAXB configuration
  - http://help.eclipse.org/juno/topic/org.eclipse.ptp.rm.jaxb.doc.isv/html/toc.html

# Overview

- Allows you to launch and monitor applications on local or remote resources
- Configured from an XML file via Java Architecture for XML Binding (JAXB)
  - Converts XML to Java classes
- Maximum adaptability
  - Allow users to fit the resource manager to a class of systems, to a single host, or even to special application usage
- No Java coding is necessary
  - Users should be able to accommodate new systems without writing and loading additional Eclipse plugins
- Partition the client functionality
  - Eliminate the need for special server-side proxies
  - Scale more successfully in the updating of job and resource information

# Resource Manager Schema (XSD)

- Schema definition
  – http://eclipse.org/ptp/schemas/rm.xsd
- Three main elements
  <site-data>
    - Used to provide fixed or default URLs for the control and monitor connections (optional)

  <control-data>
    - Supplies the job-control actions (submission, cancellation, status update, etc.)
    - Defines the launch configuration user interface

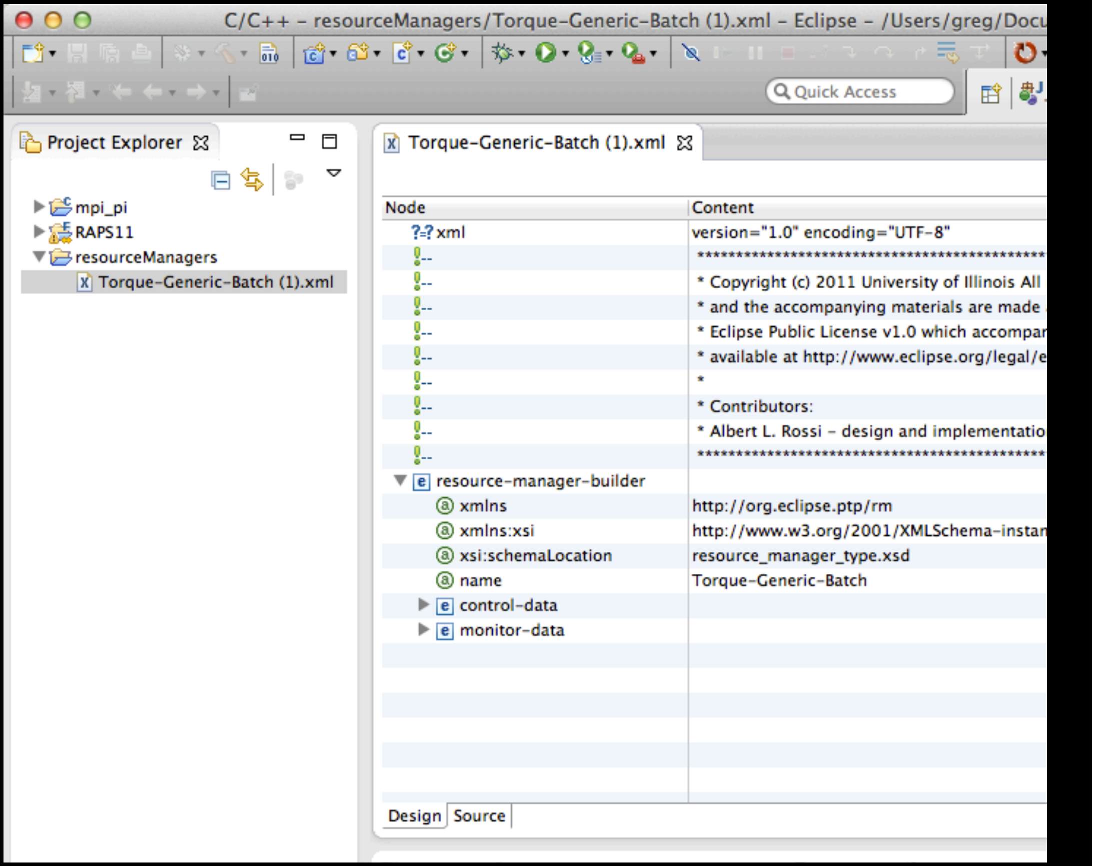
  <monitor-data>
    - Associates a monitoring client with the resource manager (optional)

# Importing an Existing Configuration

- In the C/C++ Perspective, right click in Project Explorer
- Select **Import…**
- Open the **Resource Managers** folder
- Select **Resource Manager Definition from Plug-in**
- Click **Next >**
- From the dropdown, select the desired configuration (e.g. Torque-Generic-Batch)
- Click **Finish**
- Choose **Yes** if you are asked “Do you wish to create a ‘resourceManagers’ project now?”

# Customizing an Existing Configuration

- Open the “resourceManagers” folder
- Double-click to open XML editor
- Edit this file and make your changes
- Valid XML files in the “resourceManagers” folder will appear in the Target System Configurations drop-down


C/C++ – resourceManagers/Torque-Generic-Batch (1).xml – Eclipse – /Users/greg/Docu
Quick Access
Project Explorer
mpi_pi
RAPS11
resourceManagers
Torque-Generic-Batch (1).xml
Torque-Generic-Batch (1).xml
Design
Source

| Node | Content |
|---|---|
| ?=? xml | version="1.0" encoding="UTF-8" |
| !-- | ******************************************** |
| !-- | * Copyright (c) 2011 University of Illinois All |
| !-- | * and the accompanying materials are made |
| !-- | * Eclipse Public License v1.0 which accompar |
| !-- | * available at http://www.eclipse.org/legal/e |
| !-- | * |
| !-- | * Contributors: |
| !-- | * Albert L. Rossi – design and implementatio |
| !-- | ******************************************** |
| resource-manager-builder | |
| xmlns | http://org.eclipse.ptp/rm |
| xmlns:xsi | http://www.w3.org/2001/XMLSchema-instan |
| xsi:schemaLocation | resource_manager_type.xsd |
| name | Torque-Generic-Batch |
| control-data | |
| monitor-data | |

# Attributes

- Job submission occurs in an “environment” that consists of
  - Attributes defined in the XML
  - Predefined attributes
- Attributes can be referenced in two ways
  - Some XML elements have attributes which take the name of an attribute
    - E.g. parser adds entry to the value field (a List) of the queues property:

      ```
      <target ref="queues">
      ... <add field="value"> ...
      ```
  - A string value for the attribute’s fields can be obtained using the Eclipse variable resolution syntax
    - E.g. tooltip on widget references that field of destination attribute

      ```
      <tooltip>${ptp_rm:destination#tooltip}</tooltip>
      ```
    - ptp_rm – namespace
    - destination – name of attribute
    - tooltip – attribute field

# Attribute Properties

- name
  - The name used to refer to the attribute
- type
  - The type of attribute (any primitive type, lists or maps of strings)
- status
  - Optional property indicating the validity of the attribute
- visible
  - Indicates that its value will not be changed via the Launch Tab interface
  - Certain widgets (such as the attribute table or tree) check this property to see if the attribute should be included automatically in its list
- readOnly
  - The attribute value cannot be modified
- min,max
  - Upper and lower bounds for integer attributes
- translateBooleanAs
  - Translates boolean attribute values to other values, such as “YES, NO”

# Attribute Elements

- description
  - – Brief description of the attribute for display in the UI
- tooltip
  - – Description  of the attribute to be displayed in a tooltip
- choice
  - – Fixed set of values that the attribute can take, and that the user can choose from
- items-from
  - – Links the attribute to another attribute (of type java.lang.Collection) containing values
- default
  - – Default value of the attribute if no value has been set
- value
  - – Current value of the attribute
- link-value-to
  - – Links to another attribute from which the value of this attribute will be obtained
- validator
  - – Specifies a regular expression used to validate the attribute

# Special Attributes

- The @jobId attribute is a special attribute designating the runtime id for a job launch
- In the lifecycle of the run/launch call, this value begins as an internally generated unique id (UUID) which is replaced with the actual job ID returned by the scheduler
- The @jobId, along with the target paths for <managed-file> elements, are not known at configuration time (i.e., before the user hits “Run”)
- @jobId is made visible to the parsers and the returned status object of the submit command
- Neither @jobId nor target paths are available for reference in other managed files or in the <script> element, because these elements are generated prior to the actual submission
- If the <script> needs to refer to the @jobId, it must do so via an environment variable set at the time of job submission

# <control-data> section

<attribute>

Environment values defined by the definition

<managed-file>

Local files, either pre-existing or written out from the definition itself, meant to be staged in conjunction with a job submission

<script>

Definition of a “batch script” to be staged in conjunction with a job submission

<start-up-command>

Commands run when the resource manager is started

<shut-down-command>

Commands run when the resource manager is stopped

<submit-interactive[-debug]> | <submit-batch[-debug]>

Commands for submitting [debug] jobs

<terminate-job, suspend-job, resume-job, hold-job, release-job>

Commands for controlling job submission

<get-job-status>

On-demand check of status of job

<button-action>

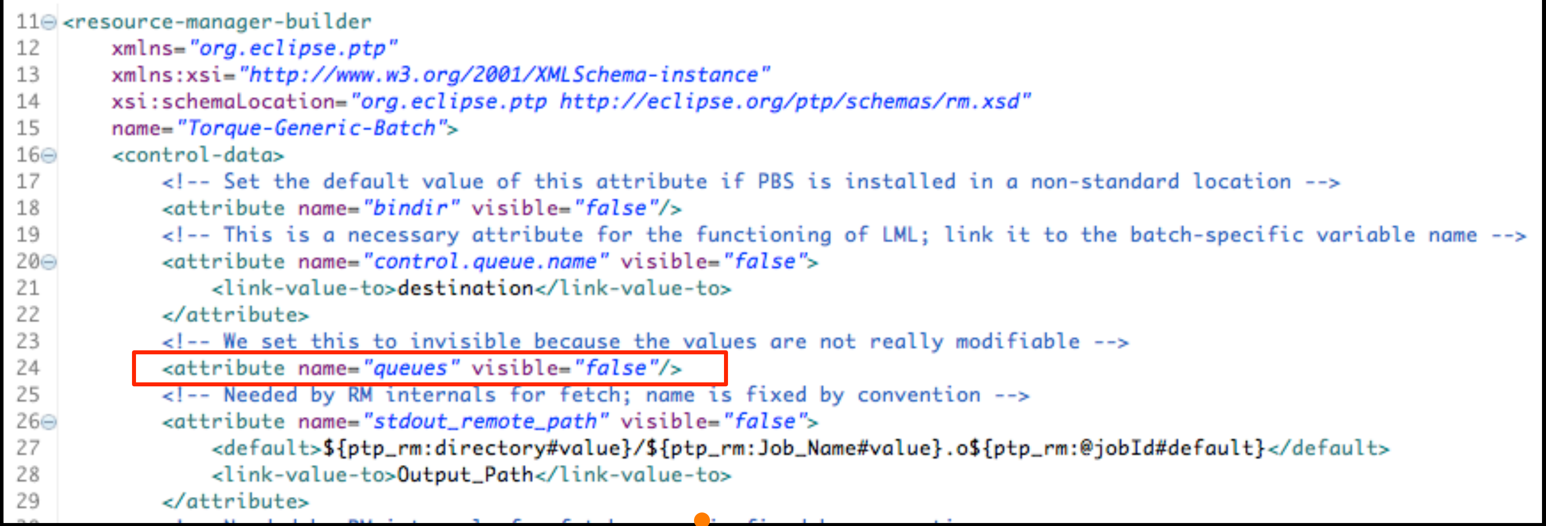
Special command run via a Launch-Tab push-button

<launch-tab>

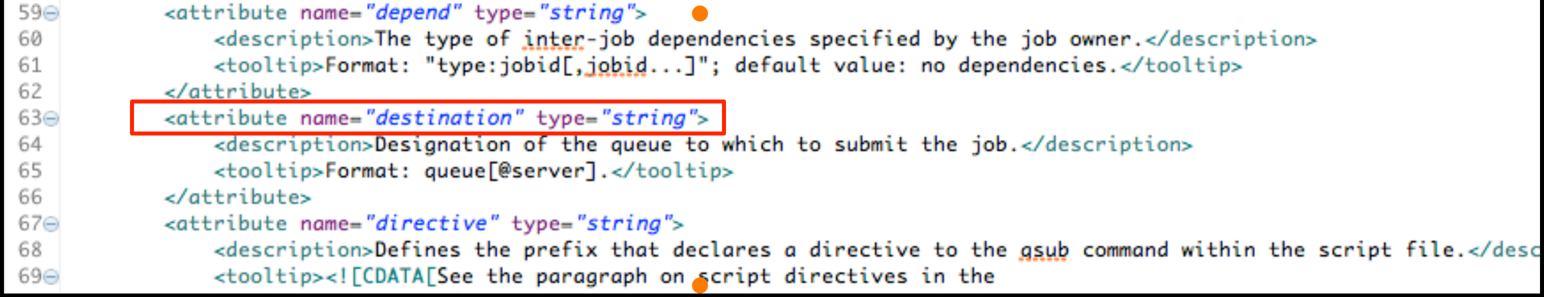
Section describing the parts and disposition of the UI used to configure and launch a job

# Example: Torque-Generic-Batch

## <attribute>



```
<resource-manager-builder
    xmlns="org.eclipse.ptp"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="org.eclipse.ptp http://eclipse.org/ptp/schemas/rm.xsd"
    name="Torque-Generic-Batch">
    <control-data>
        <!-- Set the default value of this attribute if PBS is installed in a non-standard location -->
        <attribute name="bindir" visible="false"/>
        <!-- This is a necessary attribute for the functioning of LML; link it to the batch-specific variable name -->
        <attribute name="control.queue.name" visible="false">
            <link-value-to>destination</link-value-to>
        </attribute>
        <!-- We set this to invisible because the values are not really modifiable -->
        <attribute name="queues" visible="false"/>
        <!-- Needed by RM internals for fetch; name is fixed by convention -->
        <attribute name="stdout_remote_path" visible="false">
            <default>${ptp_rm:directory#value}/${ptp_rm:Job_Name#value}.o${ptp_rm:@jobId#default}</default>
            <link-value-to>Output_Path</link-value-to>
        </attribute>
```

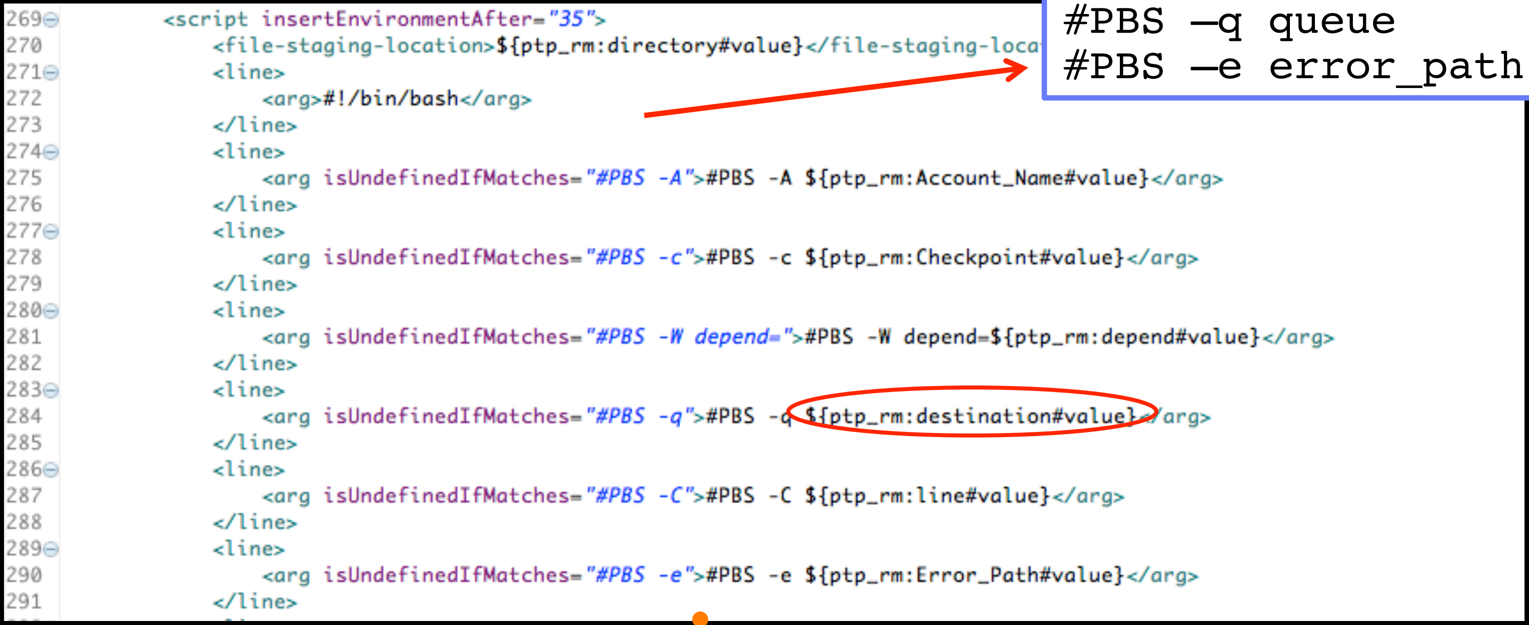


```
        <attribute name="depend" type="string">
            <description>The type of inter-job dependencies specified by the job owner.</description>
            <tooltip>Format: "type:jobid[,jobid...]"; default value: no dependencies.</tooltip>
        </attribute>
        <attribute name="destination" type="string">
            <description>Designation of the queue to which to submit the job.</description>
            <tooltip>Format: queue[@server].</tooltip>
        </attribute>
        <attribute name="directive" type="string">
            <description>Defines the prefix that declares a directive to the qsub command within the script file.</desc
            <tooltip><![CDATA[See the paragraph on script directives in the
```

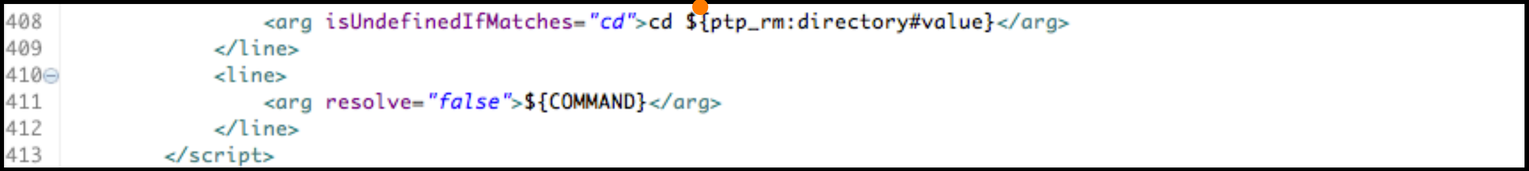
# Example: Torque-Generic-Batch

## <script>

```
#!/bin/bash
#PBS –A account
#PBS –q queue
#PBS –e error_path
```



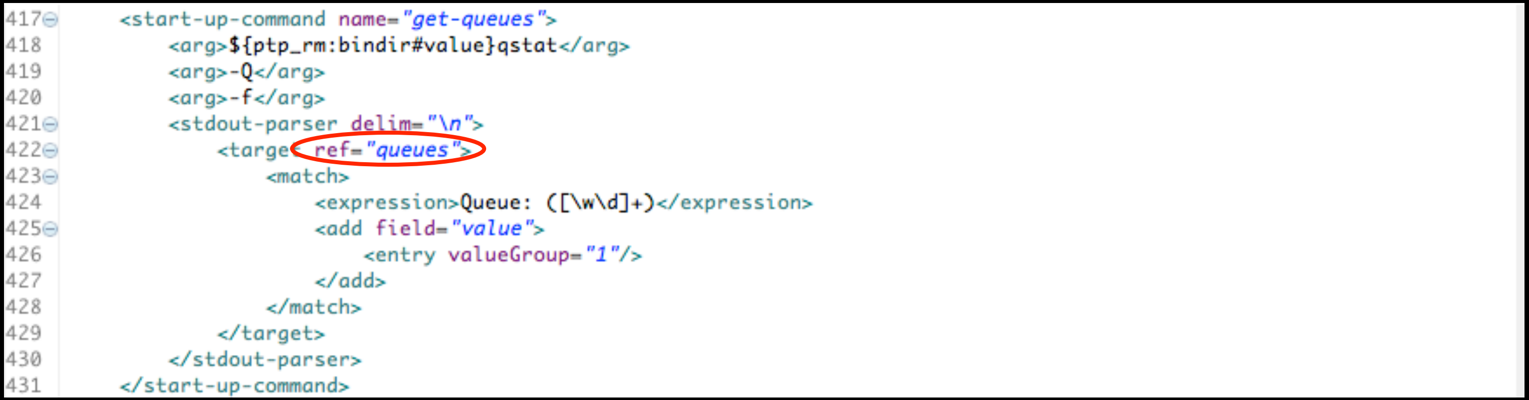
```
<script insertEnvironmentAfter="35">
    <file-staging-location>${ptp_rm:directory#value}</file-staging-loca
    <line>
        <arg>#!/bin/bash</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -A">#PBS -A ${ptp_rm:Account_Name#value}</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -c">#PBS -c ${ptp_rm:Checkpoint#value}</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -W depend=">#PBS -W depend=${ptp_rm:depend#value}</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -q">#PBS -q ${ptp_rm:destination#value}</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -C">#PBS -C ${ptp_rm:line#value}</arg>
    </line>
    <line>
        <arg isUndefinedIfMatches="#PBS -e">#PBS -e ${ptp_rm:Error_Path#value}</arg>
    </line>
```



```
        <arg isUndefinedIfMatches="cd">cd ${ptp_rm:directory#value}</arg>
    </line>
    <line>
        <arg resolve="false">${COMMAND}</arg>
    </line>
</script>
```

# Example: Torque-Generic-Batch

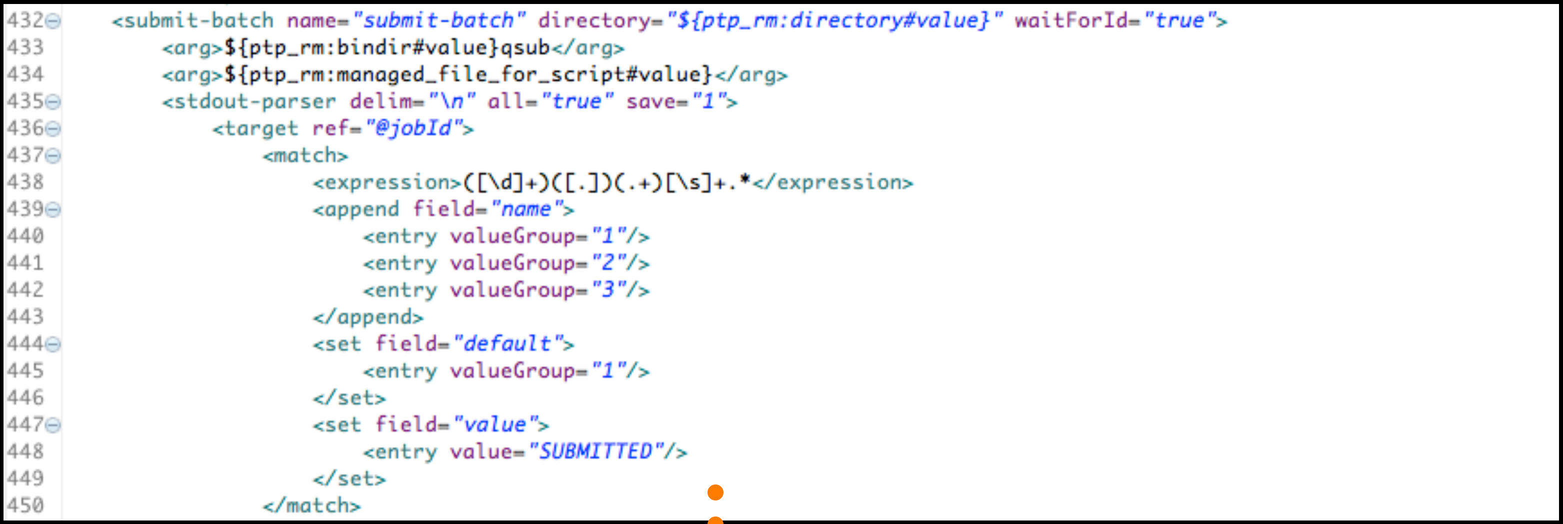
## <startup-up-command>



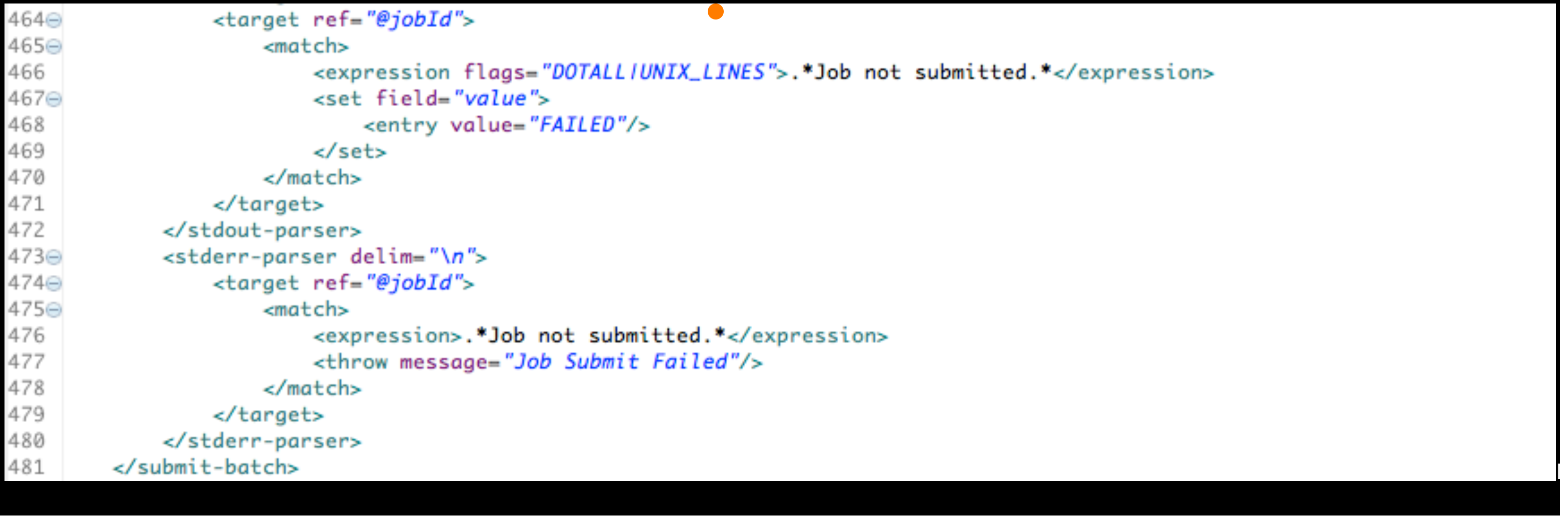
```
<start-up-command name="get-queues">
    <arg>${ptp_rm:bindir#value}qstat</arg>
    <arg>-Q</arg>
    <arg>-f</arg>
    <stdout-parser delim="\n">
        <target ref="queues">
            <match>
                <expression>Queue: ([\w\d]+)</expression>
                <add field="value">
                    <entry valueGroup="1"/>
                </add>
            </match>
        </target>
    </stdout-parser>
</start-up-command>
```

# Example: Torque-Generic-Batch

## <submit-batch>



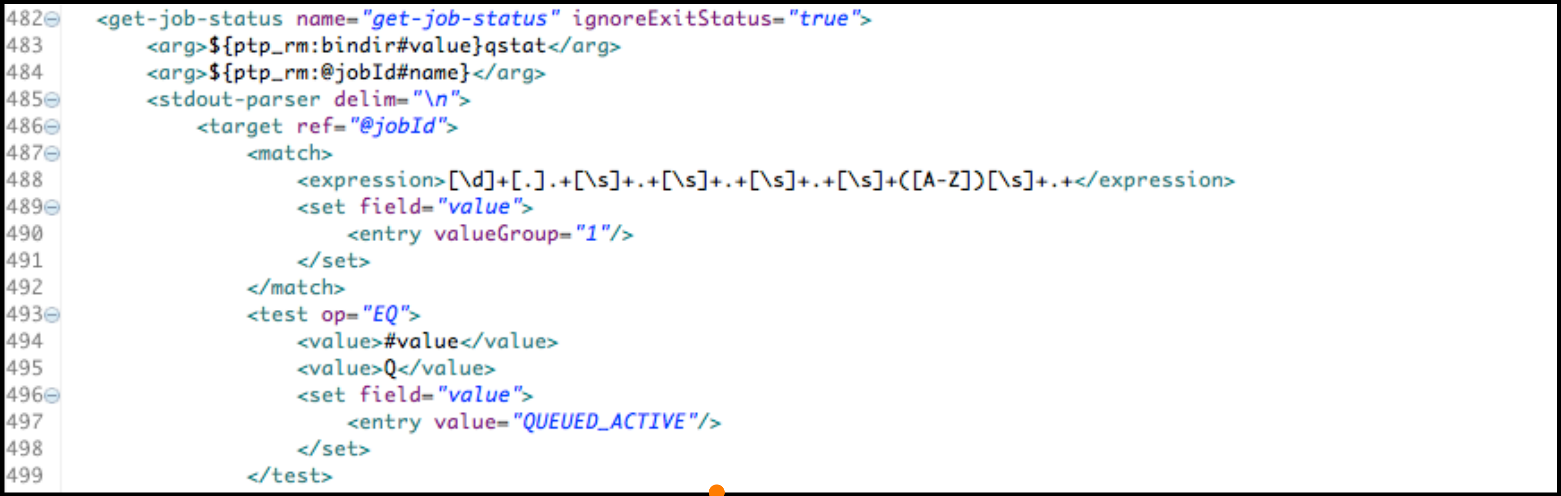
```
<submit-batch name="submit-batch" directory="${ptp_rm:directory#value}" waitForId="true">
    <arg>${ptp_rm:bindir#value}qsub</arg>
    <arg>${ptp_rm:managed_file_for_script#value}</arg>
    <stdout-parser delim="\n" all="true" save="1">
        <target ref="@jobId">
            <match>
                <expression>([\d]+)([.])(.+)[\s]+.*</expression>
                <append field="name">
                    <entry valueGroup="1"/>
                    <entry valueGroup="2"/>
                    <entry valueGroup="3"/>
                </append>
                <set field="default">
                    <entry valueGroup="1"/>
                </set>
                <set field="value">
                    <entry value="SUBMITTED"/>
                </set>
            </match>
```



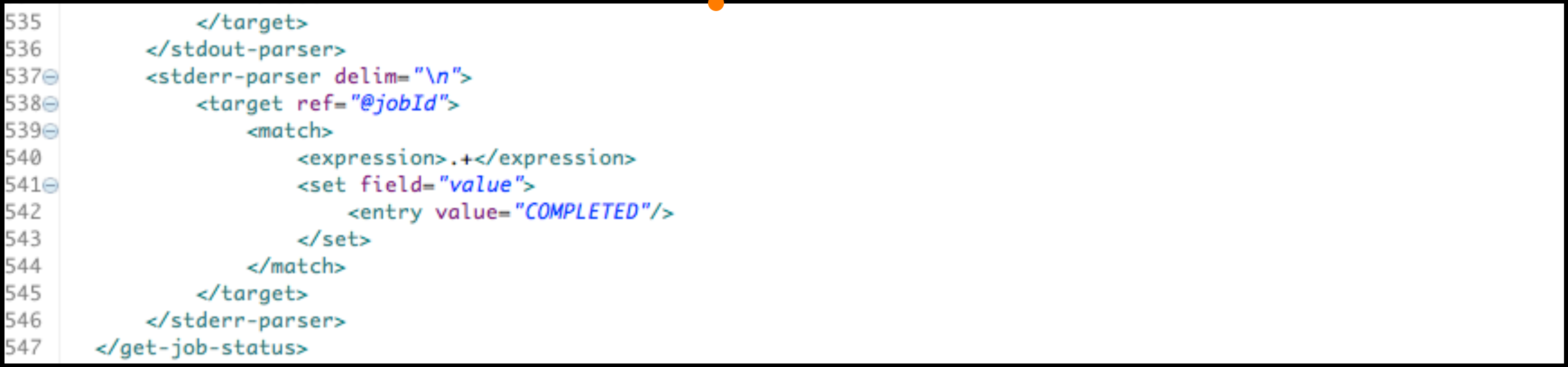
```
        <target ref="@jobId">
            <match>
                <expression flags="DOTALL|UNIX_LINES">.*Job not submitted.*</expression>
                <set field="value">
                    <entry value="FAILED"/>
                </set>
            </match>
        </target>
    </stdout-parser>
    <stderr-parser delim="\n">
        <target ref="@jobId">
            <match>
                <expression>.*Job not submitted.*</expression>
                <throw message="Job Submit Failed"/>
            </match>
        </target>
    </stderr-parser>
</submit-batch>
```

# Example: Torque-Generic-Batch

## <get-job-status>



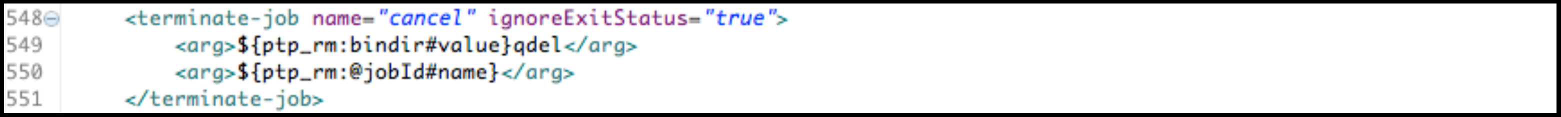
```
<get-job-status name="get-job-status" ignoreExitStatus="true">
    <arg>${ptp_rm:bindir#value}qstat</arg>
    <arg>${ptp_rm:@jobId#name}</arg>
    <stdout-parser delim="\n">
        <target ref="@jobId">
            <match>
                <expression>[\d]+[.].+[\s]+.+[\s]+.+[\s]+.+[\s]+([A-Z])[\s]+.+</expression>
                <set field="value">
                    <entry valueGroup="1"/>
                </set>
            </match>
            <test op="EQ">
                <value>#value</value>
                <value>Q</value>
                <set field="value">
                    <entry value="QUEUED_ACTIVE"/>
                </set>
            </test>
```



```
        </target>
    </stdout-parser>
    <stderr-parser delim="\n">
        <target ref="@jobId">
            <match>
                <expression>.+</expression>
                <set field="value">
                    <entry value="COMPLETED"/>
                </set>
            </match>
        </target>
    </stderr-parser>
</get-job-status>
```

# Example: Torque-Generic-Batch

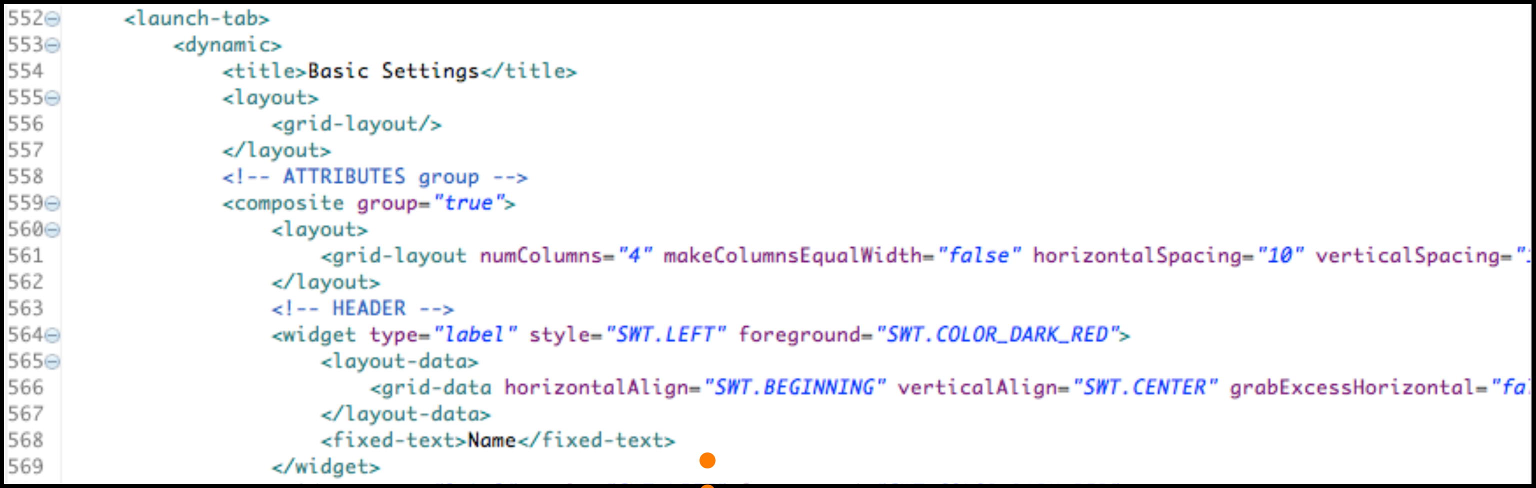
## <terminate-job>



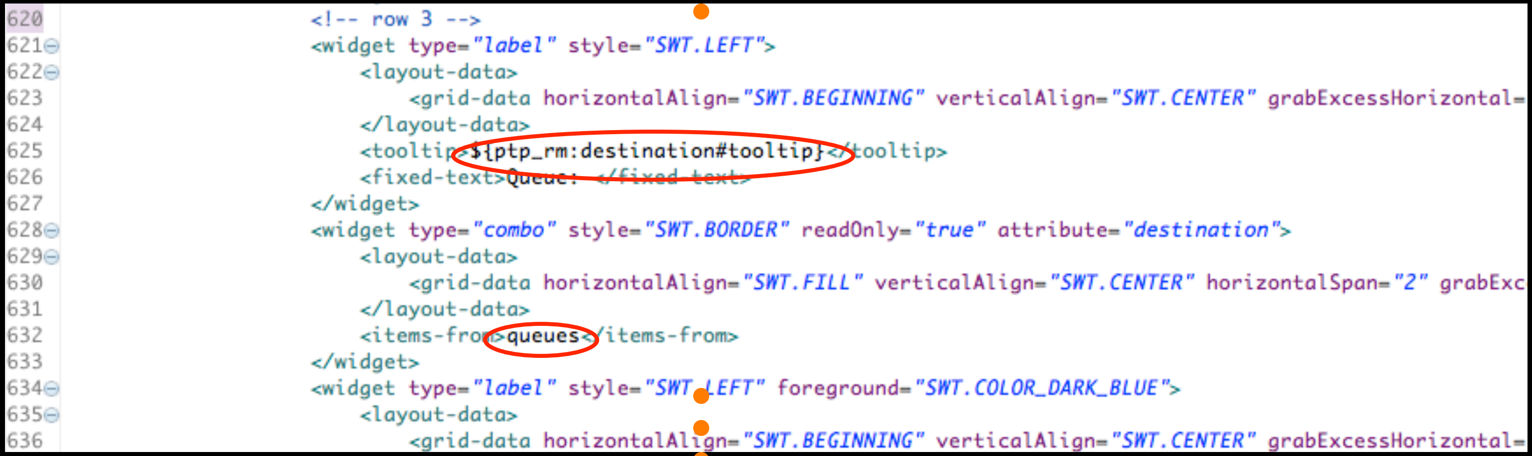
```
    <terminate-job name="cancel" ignoreExitStatus="true">
        <arg>${ptp_rm:bindir#value}qdel</arg>
        <arg>${ptp_rm:@jobId#name}</arg>
    </terminate-job>
```

# Example: Torque-Generic-Batch

## <launch-tab>



```
    <launch-tab>
        <dynamic>
            <title>Basic Settings</title>
            <layout>
                <grid-layout/>
            </layout>
            <!-- ATTRIBUTES group -->
            <composite group="true">
                <layout>
                    <grid-layout numColumns="4" makeColumnsEqualWidth="false" horizontalSpacing="10" verticalSpacing="
                </layout>
                <!-- HEADER -->
                <widget type="label" style="SWT.LEFT" foreground="SWT.COLOR_DARK_RED">
                    <layout-data>
                        <grid-data horizontalAlign="SWT.BEGINNING" verticalAlign="SWT.CENTER" grabExcessHorizontal="fa
                    </layout-data>
                    <fixed-text>Name</fixed-text>
                </widget>
```



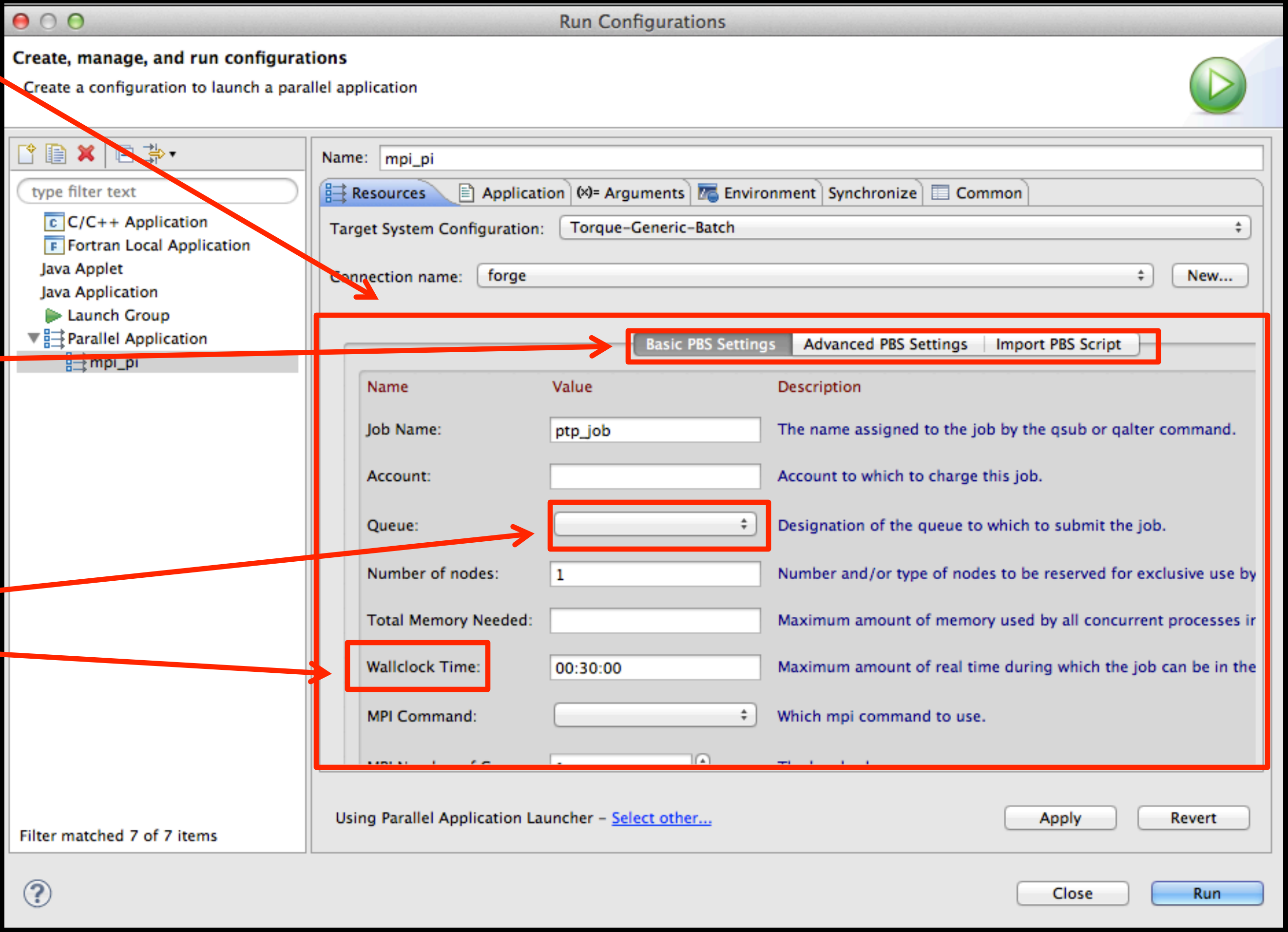
```
                    <!-- row 3 -->
                    <widget type="label" style="SWT.LEFT">
                        <layout-data>
                            <grid-data horizontalAlign="SWT.BEGINNING" verticalAlign="SWT.CENTER" grabExcessHorizontal=
                        </layout-data>
                        <tooltip>${ptp_rm:destination#tooltip}</tooltip>
                        <fixed-text>Queue: </fixed-text>
                    </widget>
                    <widget type="combo" style="SWT.BORDER" readOnly="true" attribute="destination">
                        <layout-data>
                            <grid-data horizontalAlign="SWT.FILL" verticalAlign="SWT.CENTER" horizontalSpan="2" grabExc
                        </layout-data>
                        <items-from>queues</items-from>
                    </widget>
                    <widget type="label" style="SWT.LEFT" foreground="SWT.COLOR_DARK_BLUE">
                        <layout-data>
                            <grid-data horizontalAlign="SWT.BEGINNING" verticalAlign="SWT.CENTER" grabExcessHorizontal=
```

# Example: Torque-Generic-Batch

<launch-tab>

<dynamic>

<widget>



Run Configurations

**Create, manage, and run configurations**

Create a configuration to launch a parallel application

Name: mpi_pi

Target System Configuration: Torque-Generic-Batch

Connection name: forge

| Name | Value | Description |
|---|---|---|
| Job Name: | ptp_job | The name assigned to the job by the qsub or qalter command. |
| Account: | | Account to which to charge this job. |
| Queue: | | Designation of the queue to which to submit the job. |
| Number of nodes: | 1 | Number and/or type of nodes to be reserved for exclusive use by |
| Total Memory Needed: | | Maximum amount of memory used by all concurrent processes ir |
| Wallclock Time: | 00:30:00 | Maximum amount of real time during which the job can be in the |
| MPI Command: | | Which mpi command to use. |

Using Parallel Application Launcher – Select other...

# <launch-tab>

- Based on SWT widgets
- Most widgets connected directly to attributes
- Uses SWT composite, layout and layout-data elements to arrange widgets
- Control-state element used to enable/disable or show/hide widgets
- Special widgets available, such as
  - browse
  - attribute-viewer