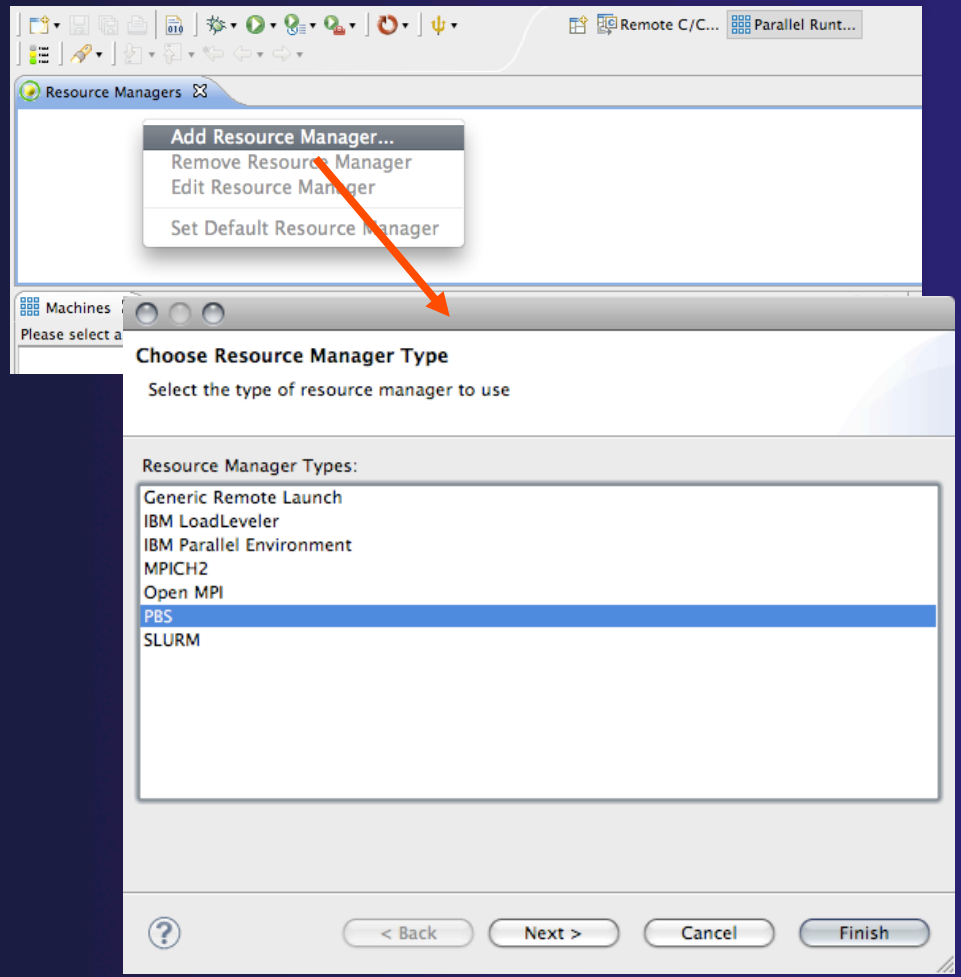
Adding a PBS Resource Manager

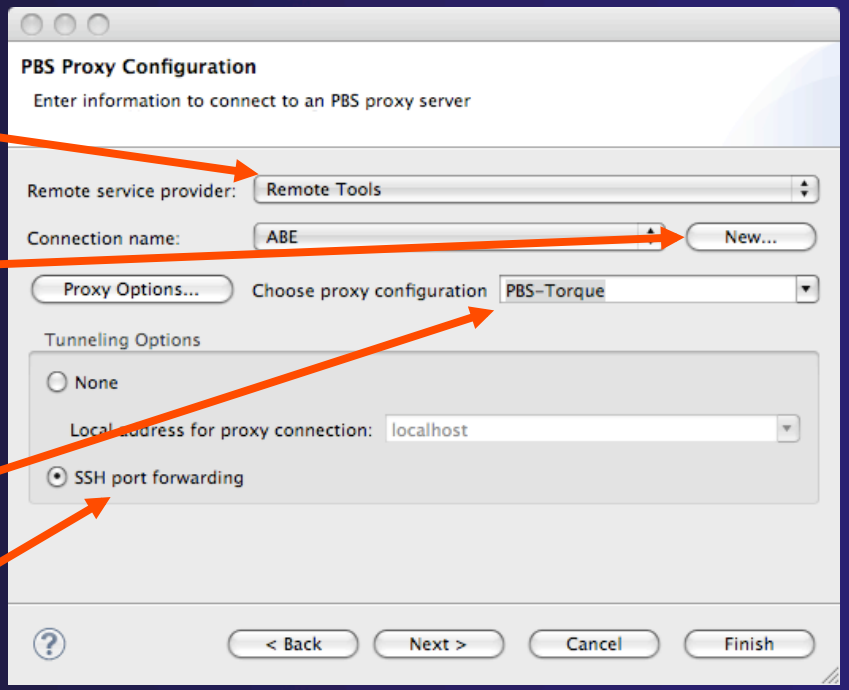
1. In the Resource Managers view, right click, choose "Add Resource Manager".
2. Under Resource Manager Types, choose PBS, then click "Next>".



Adding a PBS Resource Manager

PBS Proxy Configuration

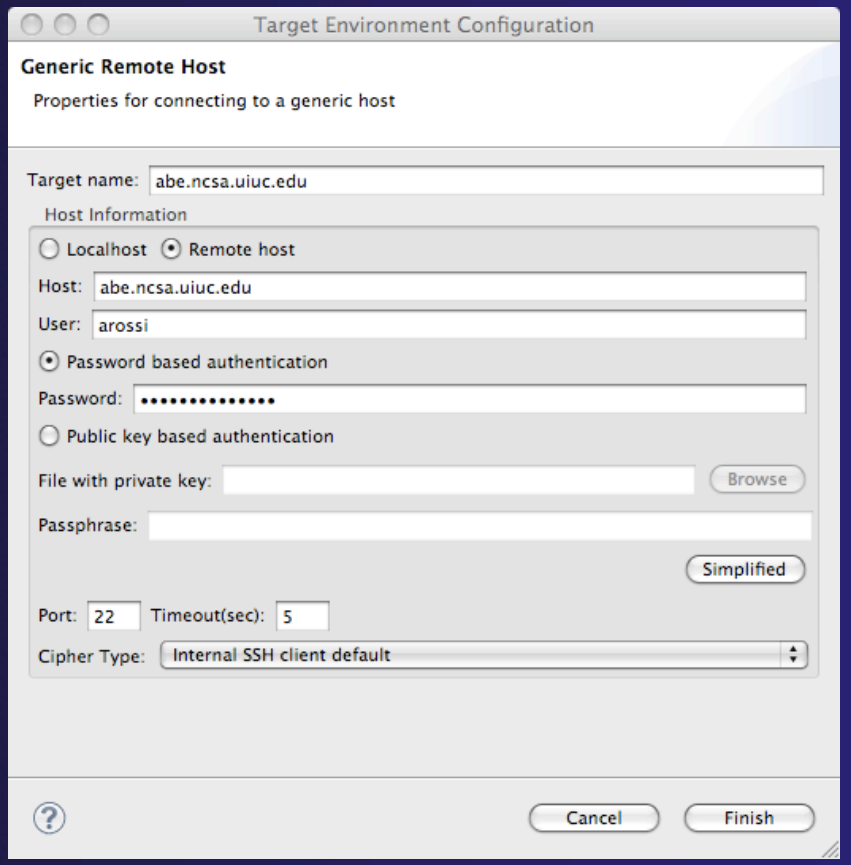
1. For Remote service provider, choose "Remote Tools".
2. Click "New…" for new remote connection; or, choose connection from combo.
3. Choose proxy configuration (currently does nothing).
4. SSH port forwarding.



Adding a PBS Resource Manager

Generic Remote Host

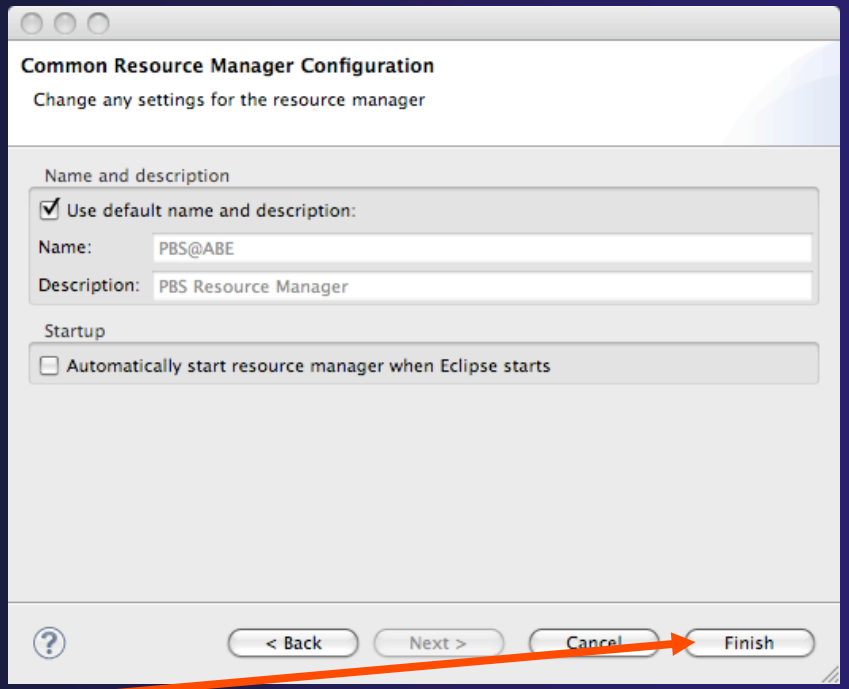
- Activated by "New..." on previous wizard page
- Choice of password or public key
- "Advanced" or "Simplified" button (default settings usually sufficient)
- Finish returns to proxy config page; click "Next>" ...

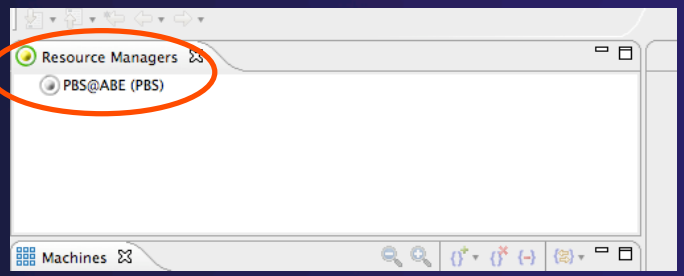


Adding a PBS Resource Manager

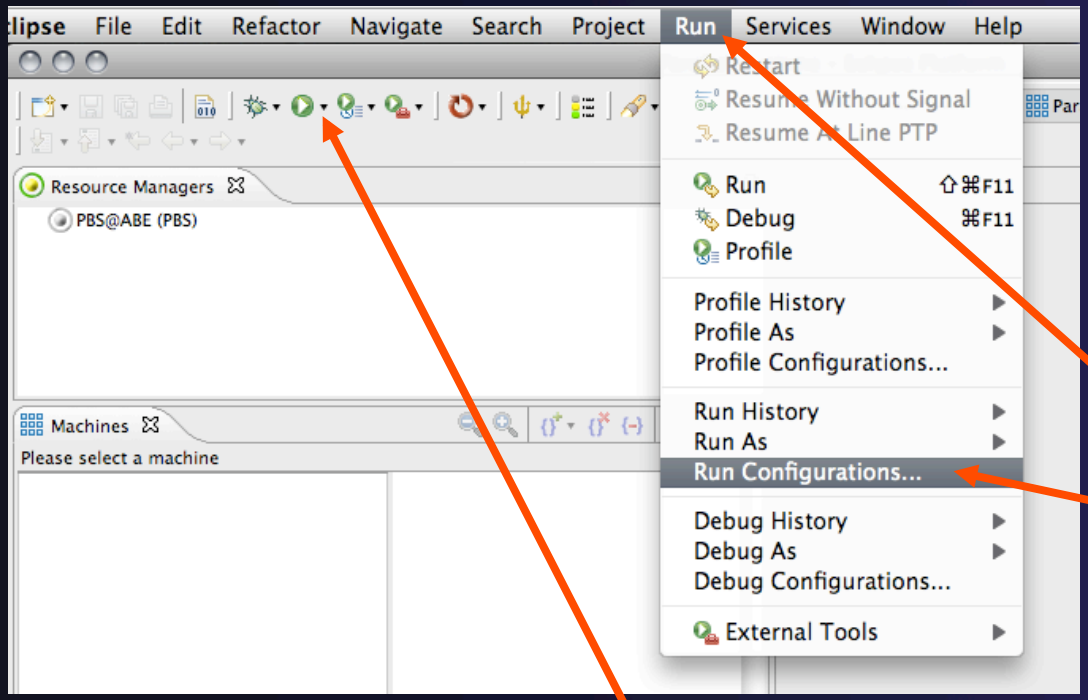
Common RM Configuration

- Change name
- Auto start
- Default usually sufficient
- Click Finish, and the new RM will appear in the Resource Managers view.





Creating a PBS Run Configuration

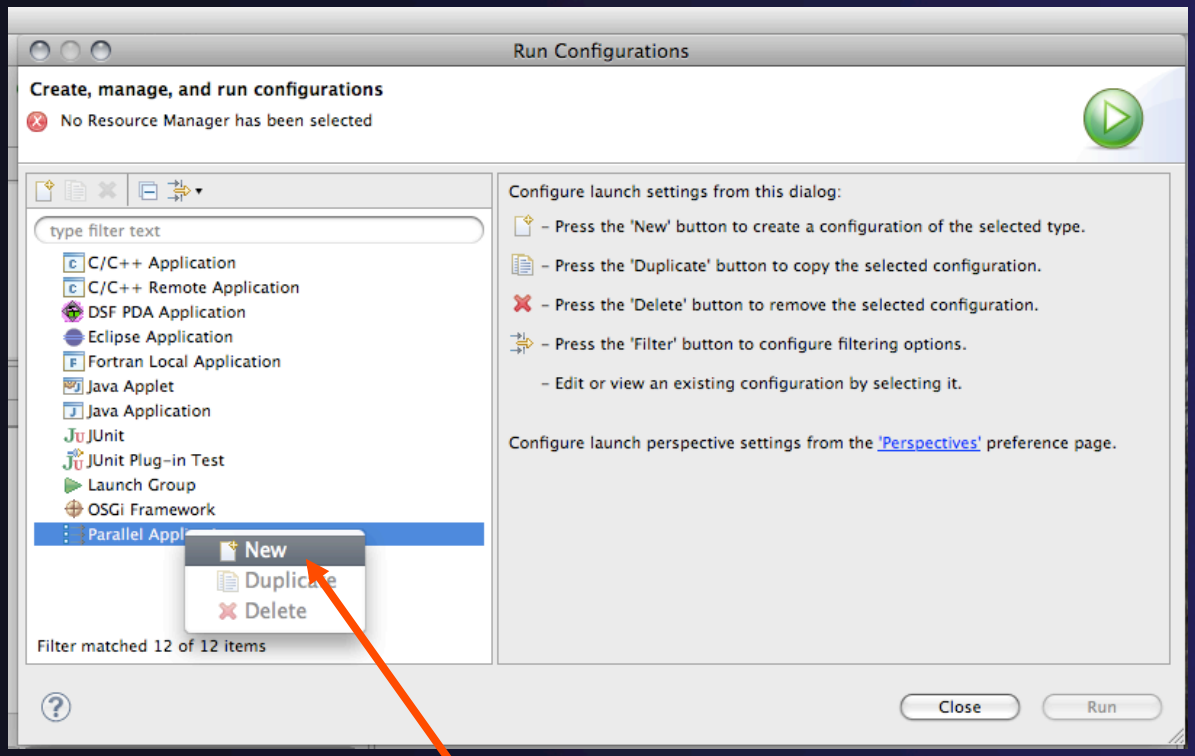


Two Ways

From Run Menu, select "Run Configurations …".

Click small black arrow next to green Run icon, then select "Run Configurations …".

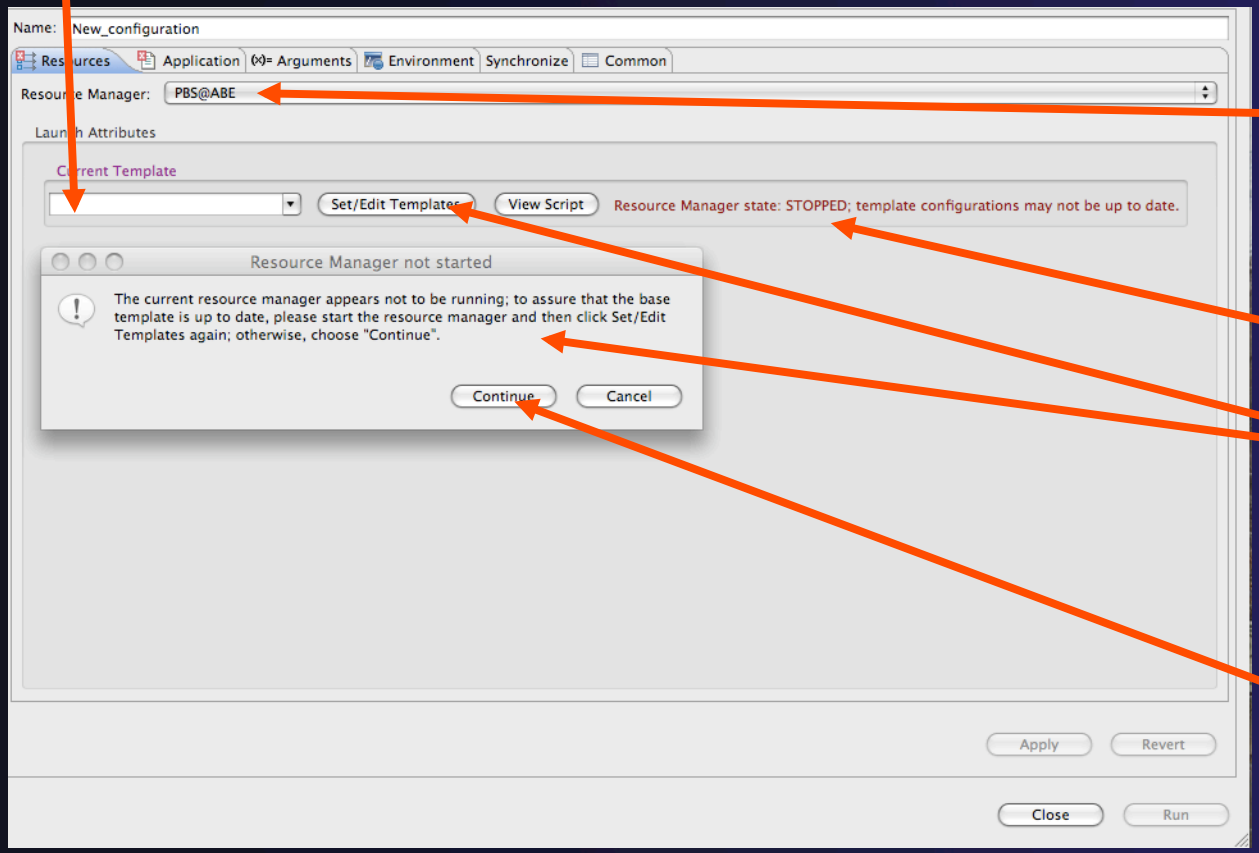
Creating a PBS Run Configuration



In the list of types, right click on "Parallel Application" and choose "New".

Configuring the PBS Run Configuration

No templates yet for this RM



Since there is only one Resource Manager, it is automatically selected.

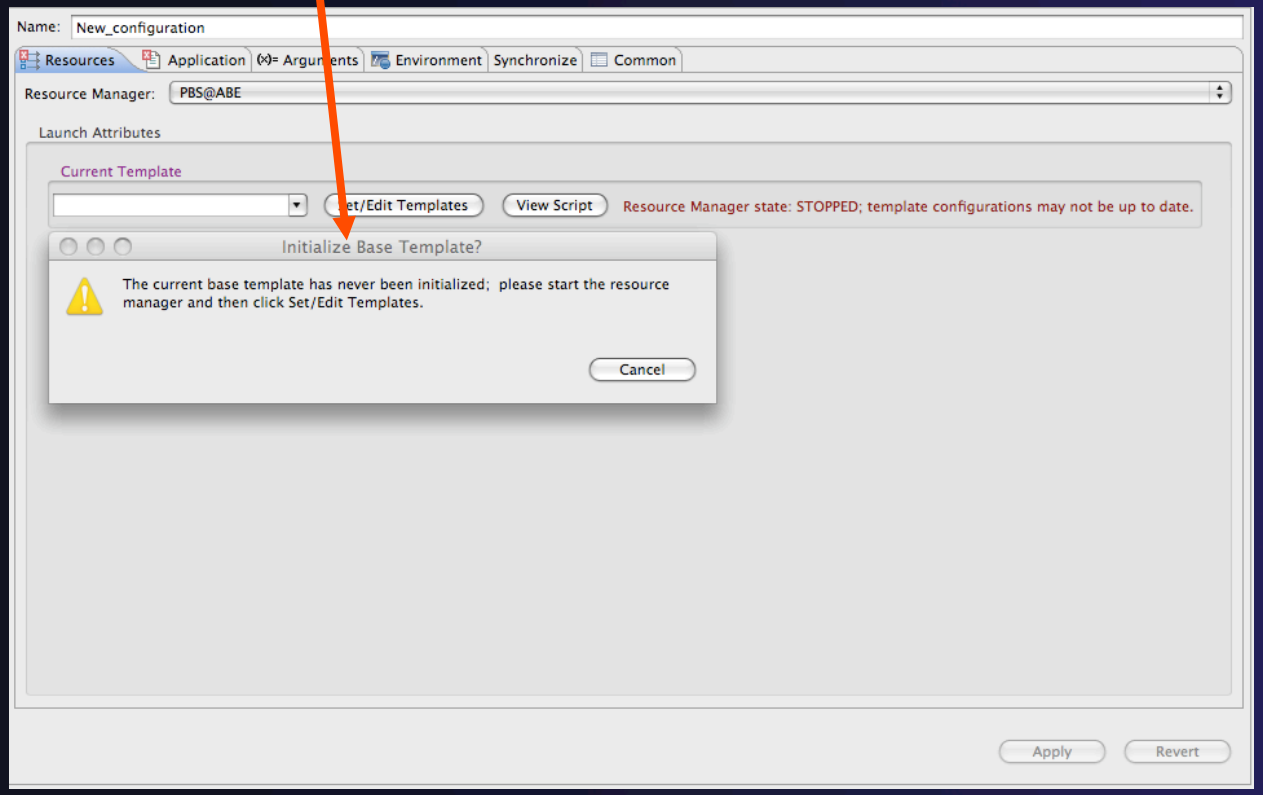
Warning: RM not started

Clicking "Set/Edit Templates" brings up this message.

Let's choose "Continue"…

Configuring the PBS Run Configuration

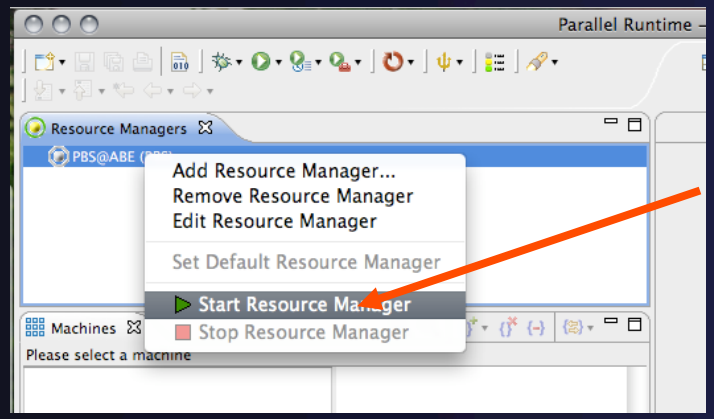
Warning: "No base template"



A PBS Resource Manager Configuration requires a "base template", but this is not generated until the RM is launched for the first time.

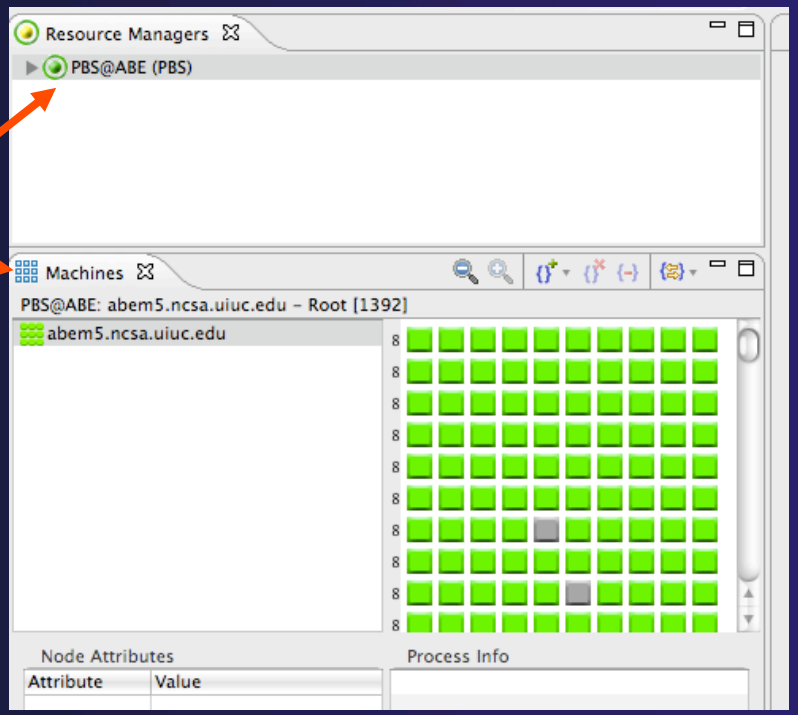
Subsequent to this initial run, you will still get the warning when the RM is not running, but "Continue" will allow you to edit the existing templates.

Starting the Resource Manager

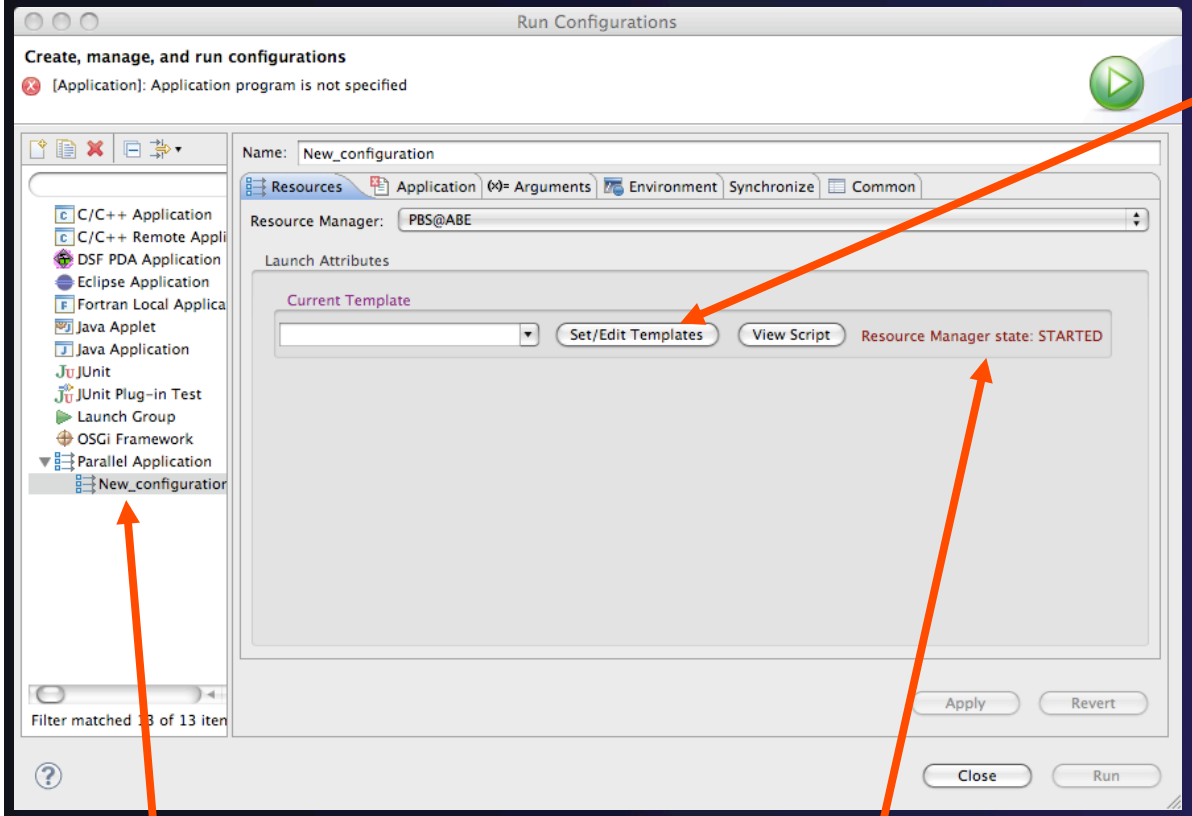


1. Close the Run configuration.
2. In the Resource Managers view, right click on the RM and select "Start Resource Manager".

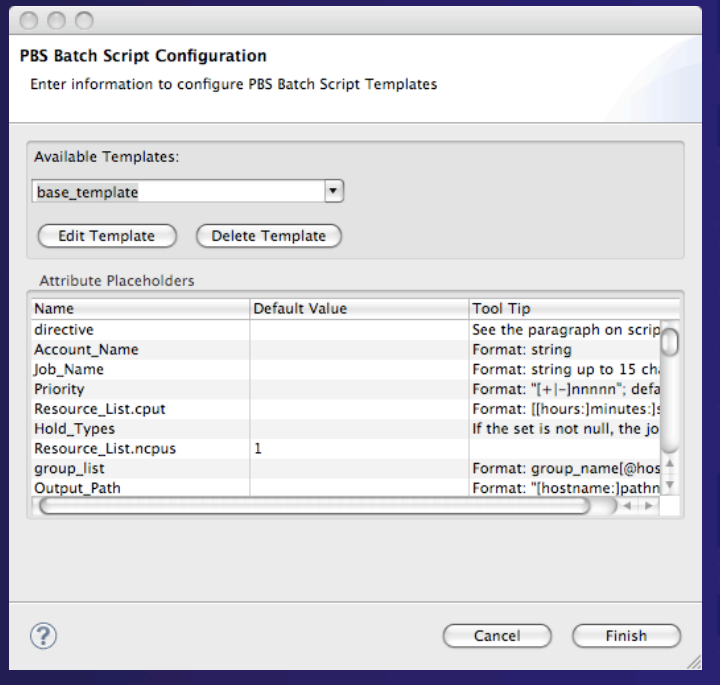
If all is well, the icon should turn green and the Machines view should populate. (Problems are indicated by a light blue or red RM icon, and possibly an error message.)



Editing PBS Batch Templates



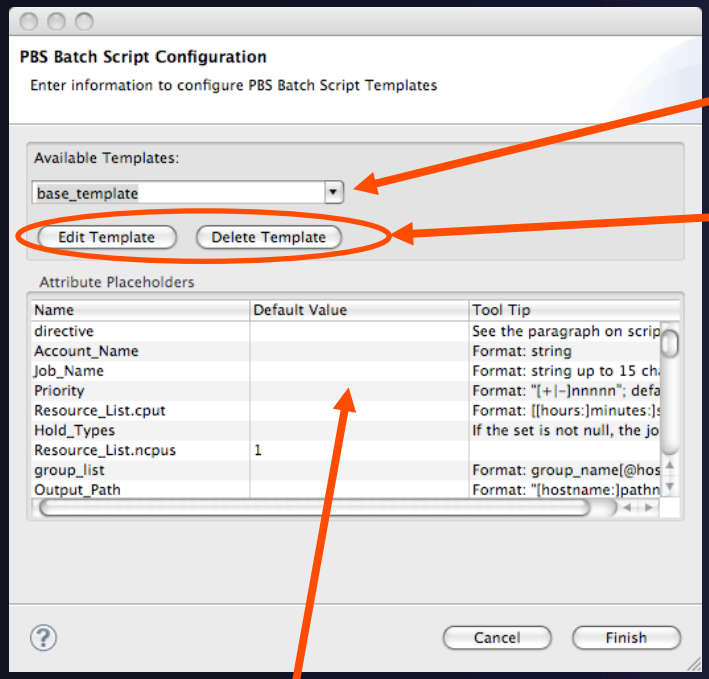
Clicking "Set/Edit Templates" now generates and displays the base template; this contains all the valid attributes this proxy server knows about.



Select to re-open

Message: STARTED

Editing PBS Batch Templates

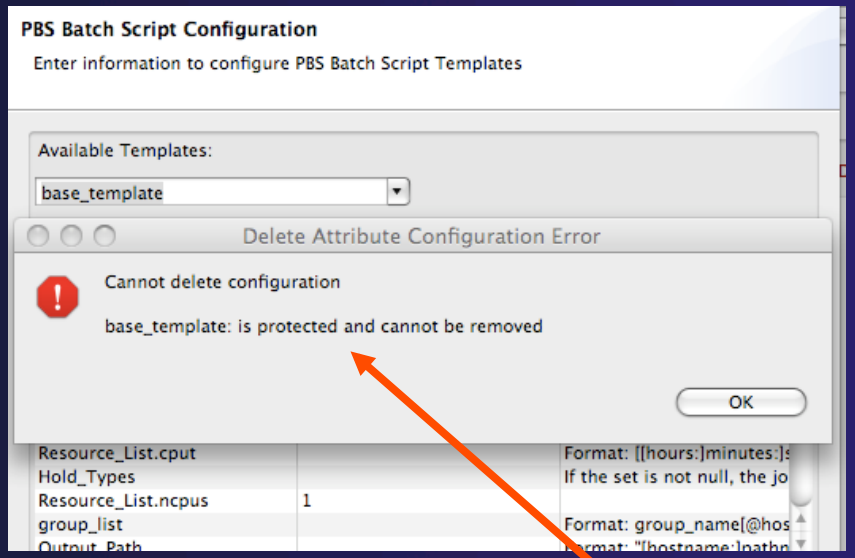


List of the templates available for this Resource Manager

Edit or Delete the selected template

So let's try to delete the base template …

Table view of configurable attributes, default values and tool tip text.



Not possible.

Editing PBS Batch Templates

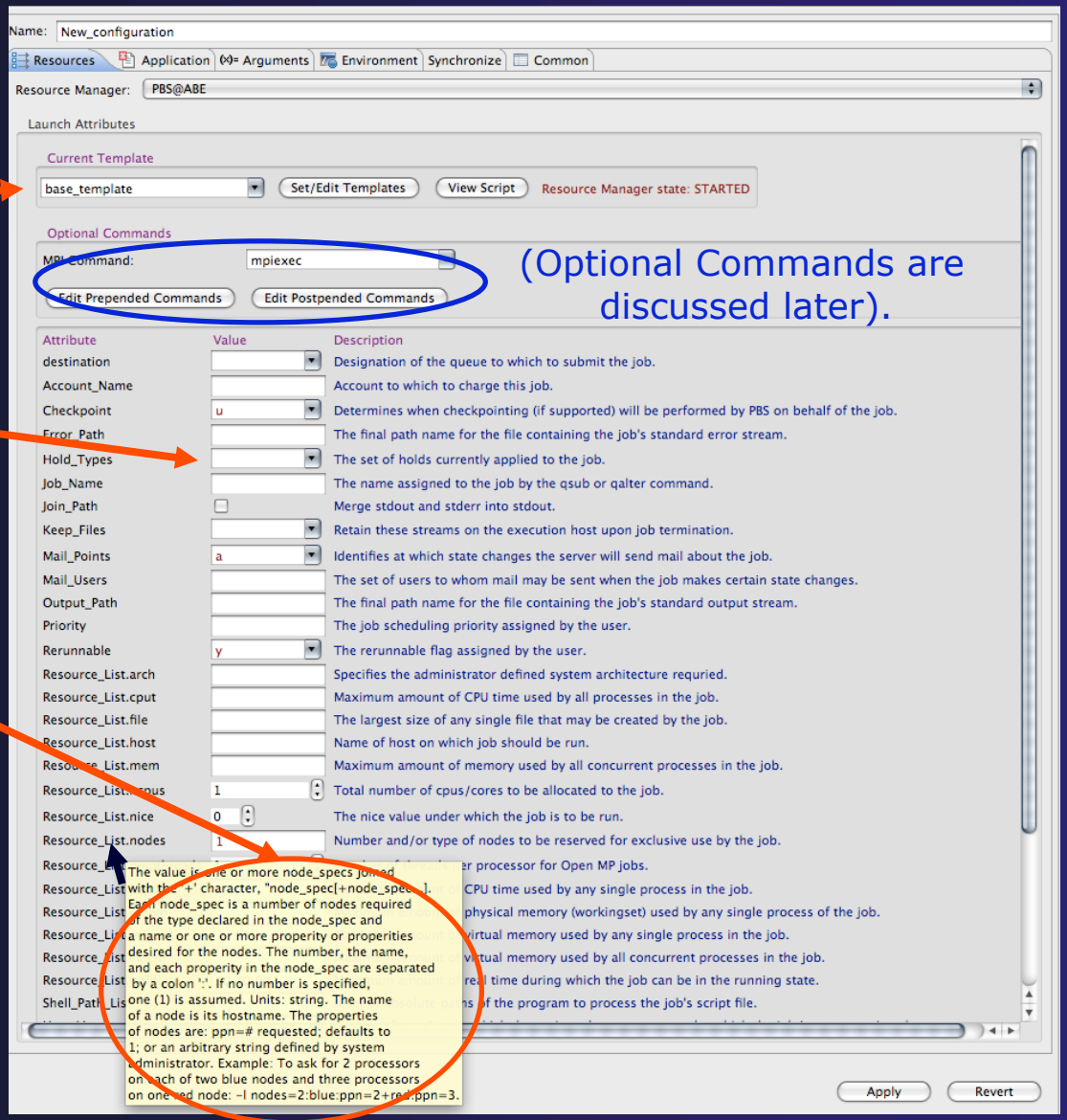
Clicking "Finish" on the previous wizard returns us to the Run Configuration.

Choose the base template: the "Launch Attributes" area now populates.

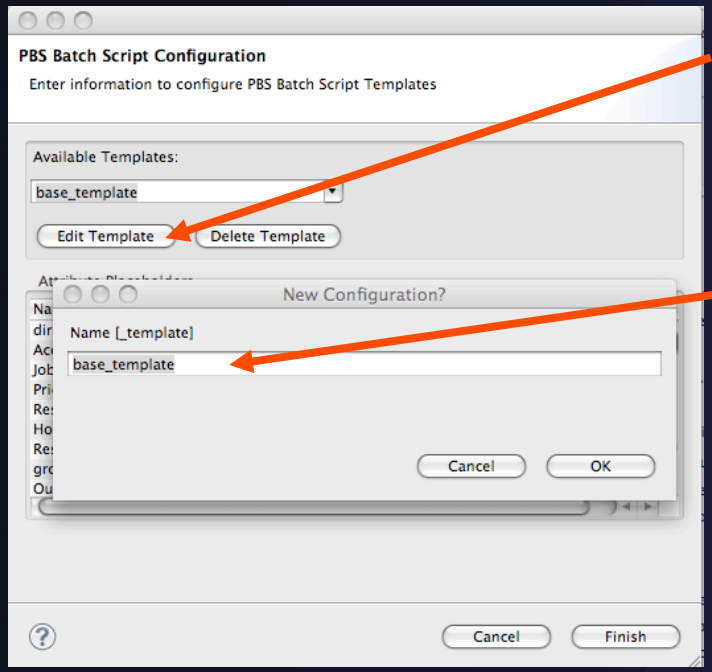
Each configurable attribute appears mapped to widget by type, along with a short definition.

Hovering over the attribute name will display a tool tip, if one is available.

We probably don't need every attribute, so let's try to customize. Click "Set/Edit Templates" again …



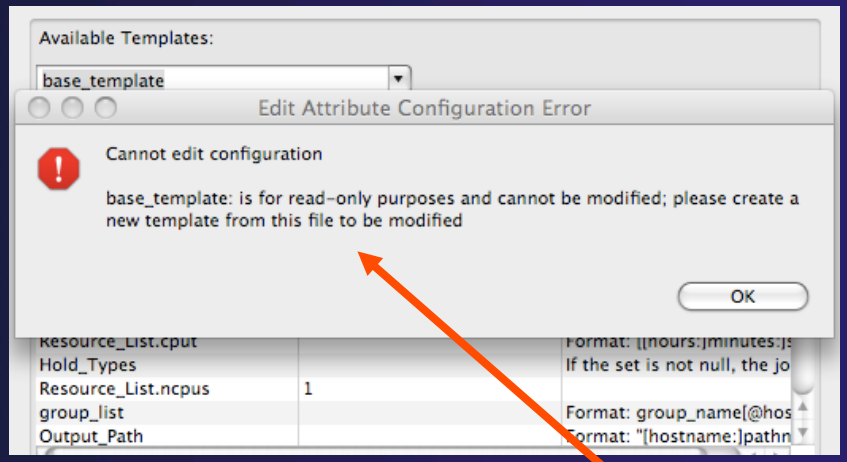
Editing PBS Batch Templates



Choose "Edit Template".

If you provide a new name (with or without the "_template" suffix), a new template is created; otherwise, the current template will be modified.

Leaving "base_template" as the name …

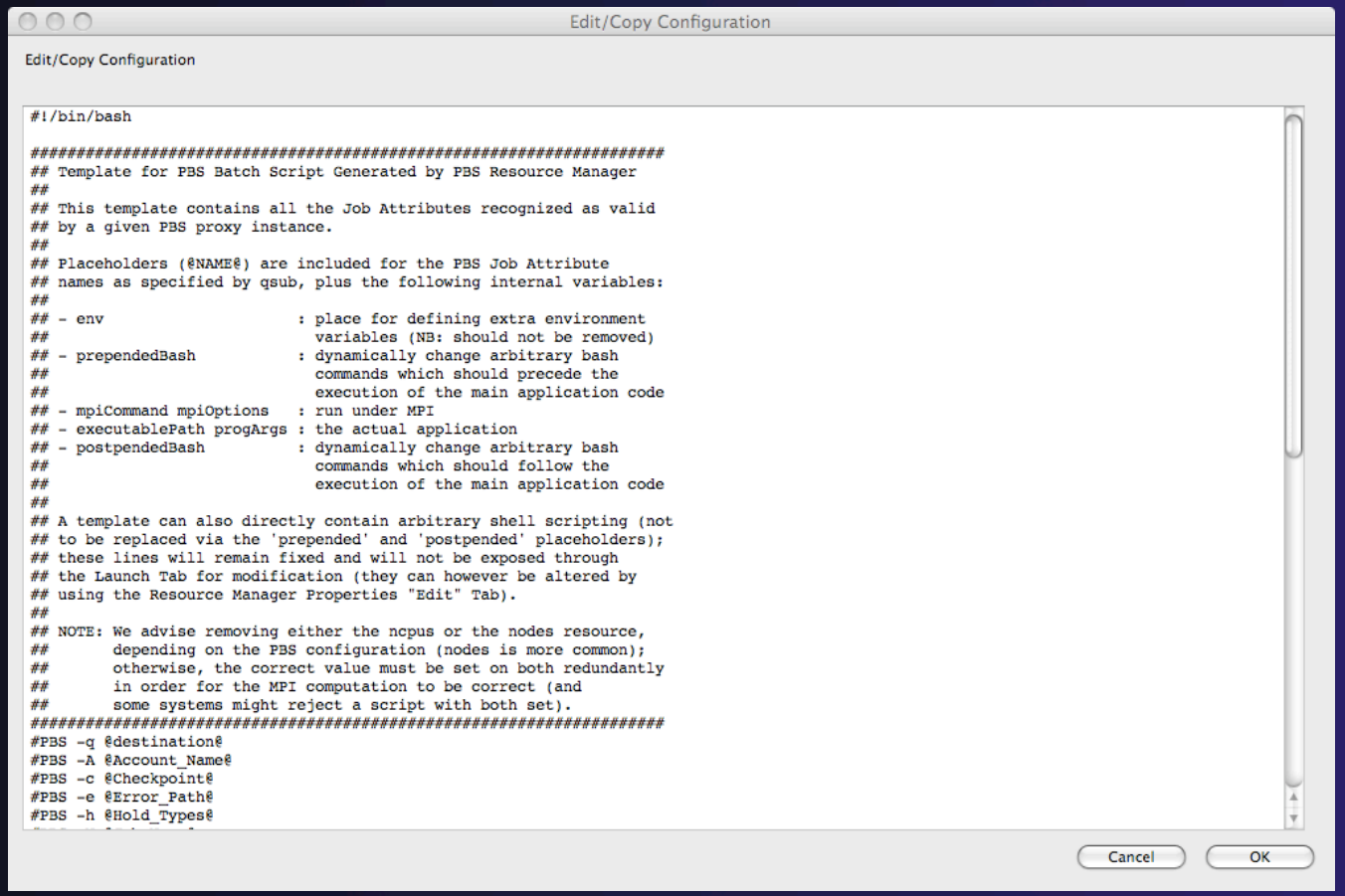


Not possible.

The base template can be used "as is", or it can serve to create custom templates, but it cannot be modified.

Editing PBS Batch Templates

Change name to "abe" and click "OK". This displays the base template in the editor view:



Editing PBS Batch Templates

```
#PBS -A @Account_Name@
#PBS -c @Checkpoint@
#PBS -C @directive@
#PBS -e @Error_Path@
#PBS -h @Hold_Types@
#PBS -j @Join_Path@
#PBS -k @Keep_Files@
#PBS -l arch=@Resource_List.arch@
#PBS -l cput=@Resource_List.cput@
#PBS -l file=@Resource_List.file@
#PBS -l host=@Resource_List.host@
#PBS -l mem=@Resource_List.mem@
#PBS -l ncpus=@Resource_List.ncpus@
#PBS -l nice=@Resource_List.nice@
#PBS -l nodes=@Resource_List.nodes@
#PBS -l ompthreads=@Resource_List.ompthreads@
#PBS -l pcput=@Resource_List.pcput@
#PBS -l pmem=@Resource_List.pmem@
#PBS -l pvmem=@Resource_List.pvmem@
#PBS -l vmem=@Resource_List.vmem@
#PBS -l walltime=@Resource_List.walltime@
#PBS -m @Mail_Points@
#PBS -M @Mail_Users@
#PBS -N @Job_Name@
#PBS -o @Output_Path@
#PBS -p @Priority@
#PBS -q @destination@
#PBS -r @Rerunnable@
#PBS -S @Shell_Path_List@
#PBS -u @User_List@
#PBS -v @Variable_List@
#PBS -V @export_all@
#PBS -W depend=@depend@
#PBS -W group_list=@group_list@
#PBS -W stagein=@stagein@
#PBS -W stageout=@stageout@
@env@
@prependedBash@
cd @directory@
@mpiCommand@ @mpiOptions@ @executablePath@ @progArgs@
@postpendedBash@
```

The template uses placeholder syntax to generate the widget display and the actual batch script; in substituting the actual values, a placeholder marker (or, in the case of the PBS directives, the entire line) is eliminated if it is unused or would be an empty string. The following placeholders are valid:

@qsubAttribute@

These are the attributes associated with valid *qsub flags. These are attributes generally accepted by all systems; future releases will provide specific versions (such as PBS Pro or Torque). If the attribute name is currently not recognized, an error will be reported.*

@env@

If the user defines environment variables via the Run Configuration Environment tab, these will be captured and inserted as bash "export" commands at this location in the batch script.

@prependedBash@, @postpendedBash@

Any arbitrary bash commands added via the "Edit ... Commands" button/dialog are inserted here. Note: the template itself can be modified to carry arbitrary commands; these placeholders allow you to vary the additional commands without creating an entirely new template.

@directory@

Not all PBS systems accept the -d *qsub option, so any explicit directory change associated with the execution of the main application is taken care of here.*

Editing PBS Batch Templates

```
#PBS -A @Account_Name@
#PBS -c @Checkpoint@
#PBS -C @directive@
#PBS -e @Error_Path@
#PBS -h @Hold_Types@
#PBS -j @Join_Path@
#PBS -k @Keep_Files@
#PBS -l arch=@Resource_List.arch@
#PBS -l cput=@Resource_List.cput@
#PBS -l file=@Resource_List.file@
#PBS -l host=@Resource_List.host@
#PBS -l mem=@Resource_List.mem@
#PBS -l ncpus=@Resource_List.ncpus@
#PBS -l nice=@Resource_List.nice@
#PBS -l nodes=@Resource_List.nodes@
#PBS -l ompthreads=@Resource_List.ompthreads@
#PBS -l pcput=@Resource_List.pcput@
#PBS -l pmem=@Resource_List.pmem@
#PBS -l pvmem=@Resource_List.pvmem@
#PBS -l vmem=@Resource_List.vmem@
#PBS -l walltime=@Resource_List.walltime@
#PBS -m @Mail_Points@
#PBS -M @Mail_Users@
#PBS -N @Job_Name@
#PBS -o @Output_Path@
#PBS -p @Priority@
#PBS -q @destination@
#PBS -r @Rerunnable@
#PBS -S @Shell_Path_List@
#PBS -u @User_List@
#PBS -v @Variable_List@
#PBS -V @export_all@
#PBS -W depend=@depend@
#PBS -W group_list=@group_list@
#PBS -W stagein=@stagein@
#PBS -W stageout=@stageout@
@env@
@prependedBash@
cd @directory@
@mpiCommand@ @mpiOptions@ @executablePath@ @progArgs@
@postpendedBash@
```

@mpiCommand@

Placeholder for the MPI command; if there is no MPI command, the mpiOptions placeholder is also eliminated.

@mpiOptions@

Currently, this consists of the "-n [mpicores]" argument. This is automatically computed from the appropriate *qsub attribute: if ncpus is defined, this is used; else the nodes value is parsed for nodes X ppn.*

@executablePath@

Replaced on the basis of the Run Configuration Application tab's executable path.

@progArgs@

Replaced on the basis of the Run Configuration Arguments tab.

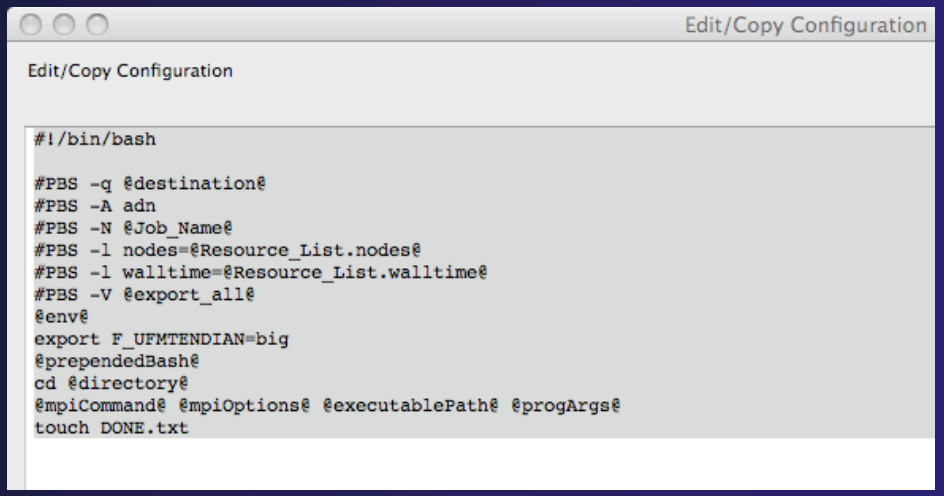
Basically, this can be any PBS script, but only those syntagmata marked by @...@ will translate into a widget accepting a value from the user. The entire script, with those markers replaced by the provided values, is shipped to the PBS proxy when "Run" is clicked.

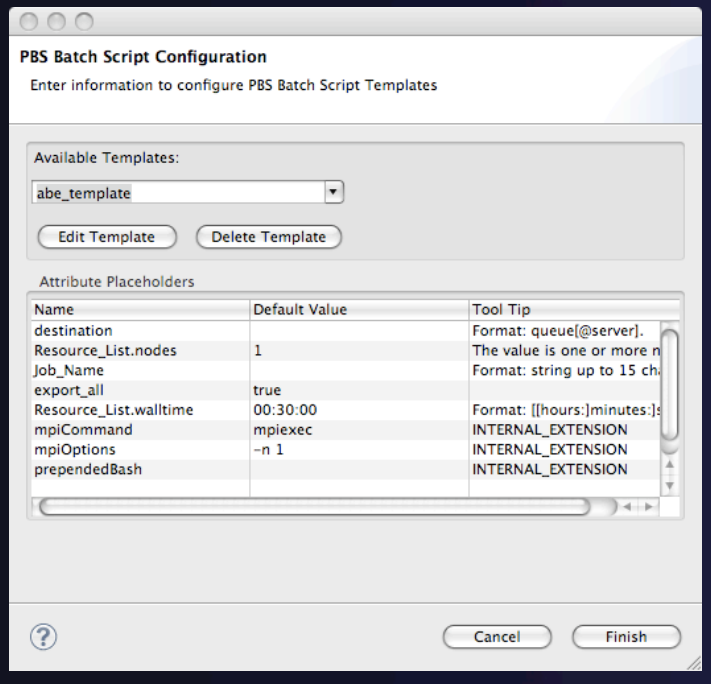
Editing PBS Batch Templates

Let's try to eliminate and change attributes.

In the editor, we

1. Remove the info header and all but a few basic attributes.
2. Hardcode *account* to "adn".
3. Add hardcoded env variable F_UFMTENDIAN=big.
4. Remove the @postpendedBash@ placeholder.
5. Add command "touch DONE.txt".



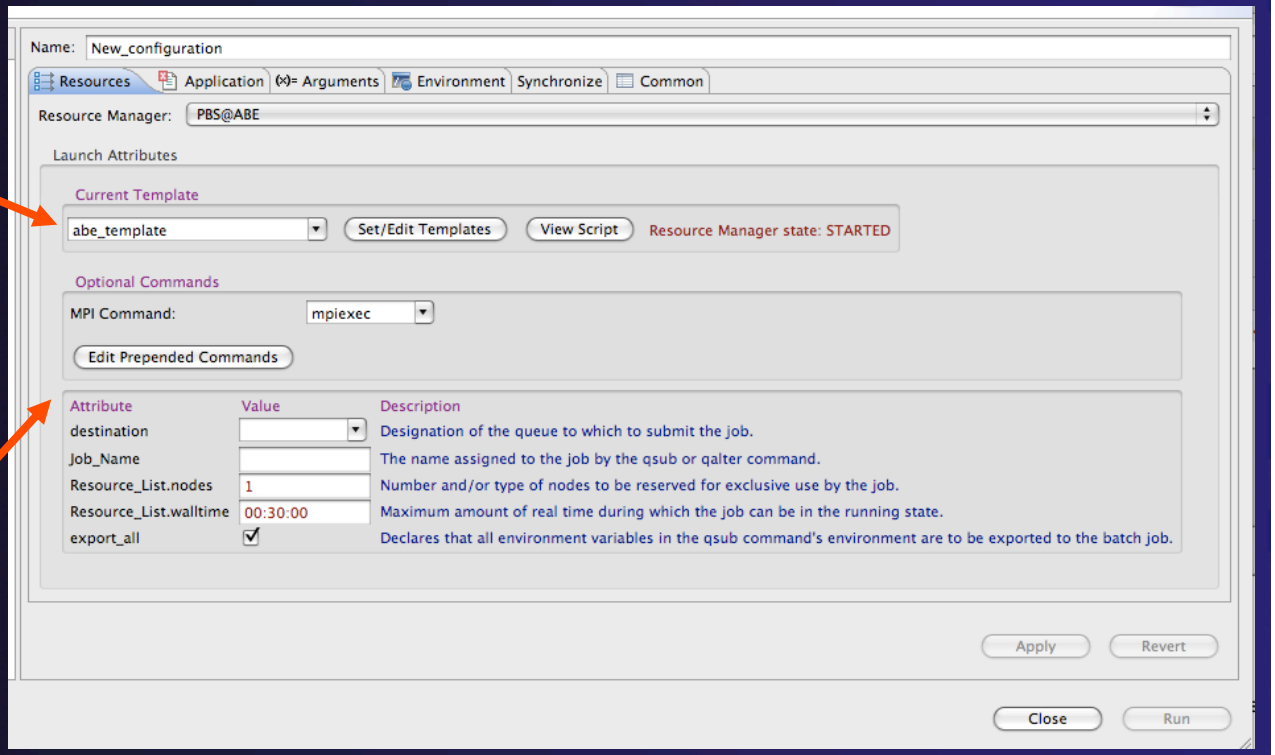


Click "OK" to close editor. We now see the summary view of the new configuration. Click "Finish".

Editing PBS Batch Templates

On the Resources Tab of the Run Configuration, now choose "abe_template" from the Current Template combo.

The dynamic panes now repopulate from the new template. (Note how the "Edit Postpended Commands" button has been eliminated as well.)



Configuring a PBS Job

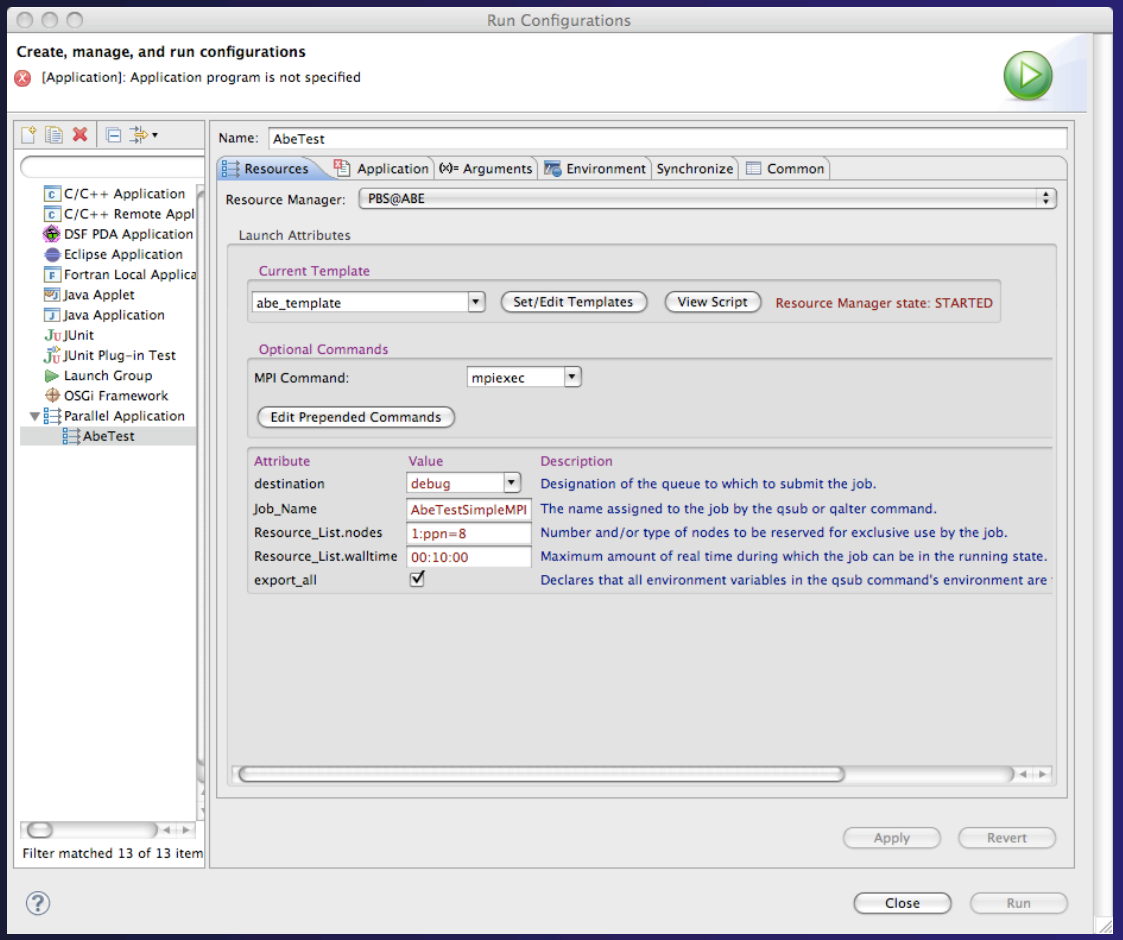
Now let's fill in some values for this particular job.

In the Resources Tab, we

1. Leave "mpiexec" as the command for MPI.
2. Choose destination (queue): "debug".*
3. Set JobName: "AbeTestSimpleMPI"
4. Specify the full "nodes" line: "1:ppn=8".**
5. Set walltime to "00:10:00".

* The destination list is populated automatically from the queues the proxy tells the UI about (there is a slight delay at start-up for this event to arrive).

** The MPI –n option is automatically computed from the nodes or ncpus attribute; in this case, 1 X 8.

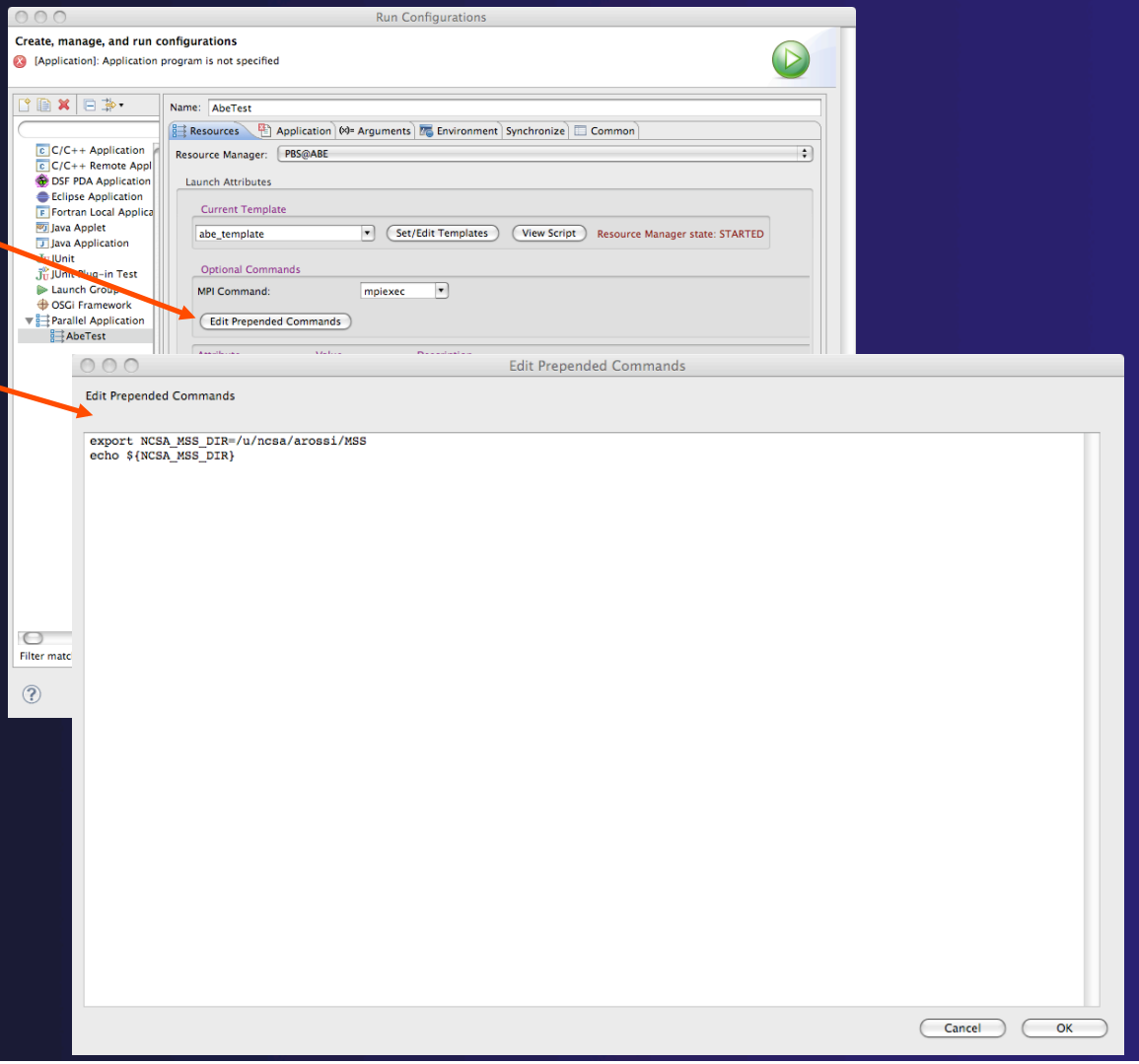


Configuring a PBS Job

Let's also prepend some commands.

In the Resources Tab, we

1. Click "Edit Prepended Commands". This brings up an editor.
2. Add a variable definition and echo statement.
3. Click "OK".

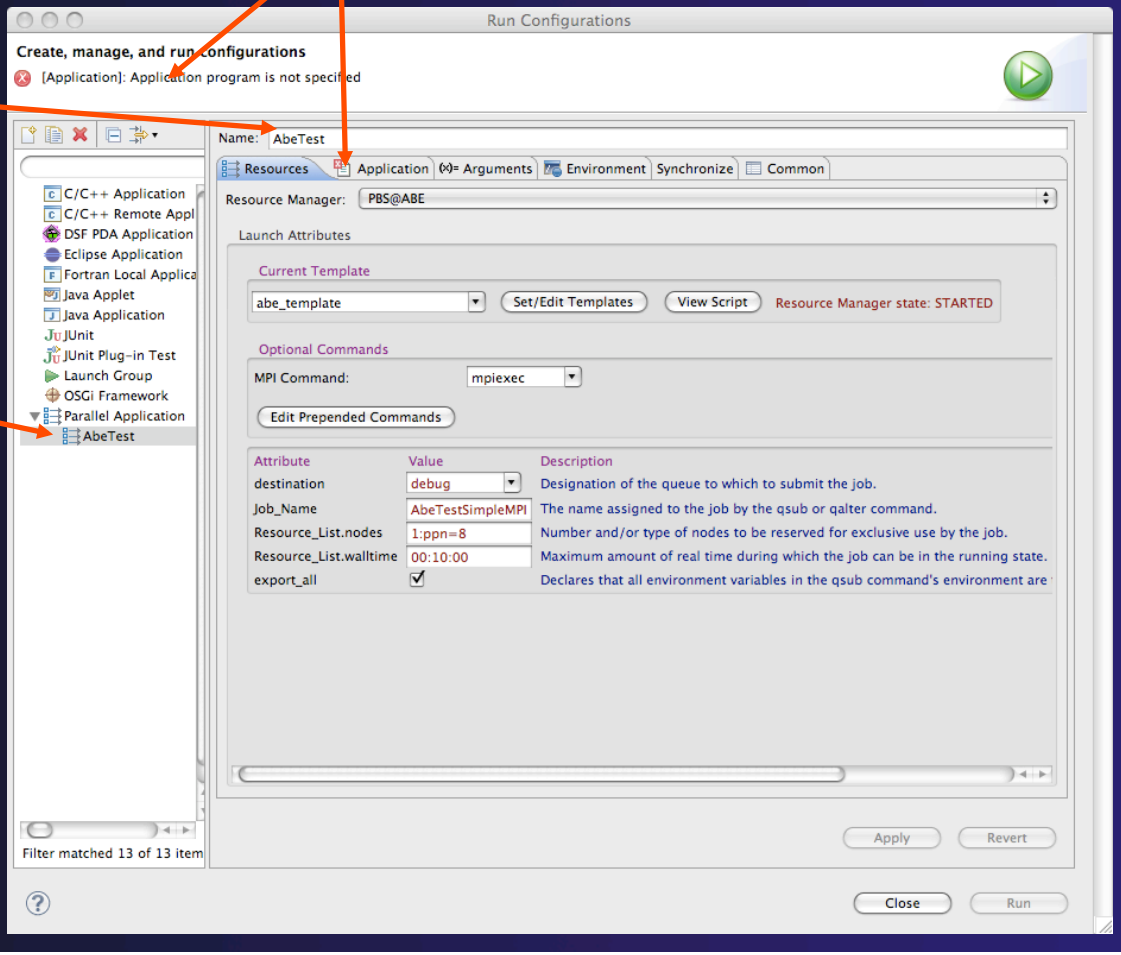


These commands apply to this instance of the template. What you add is not hardcoded into the template, but is like the values for the attributes.

Configuring a PBS Job

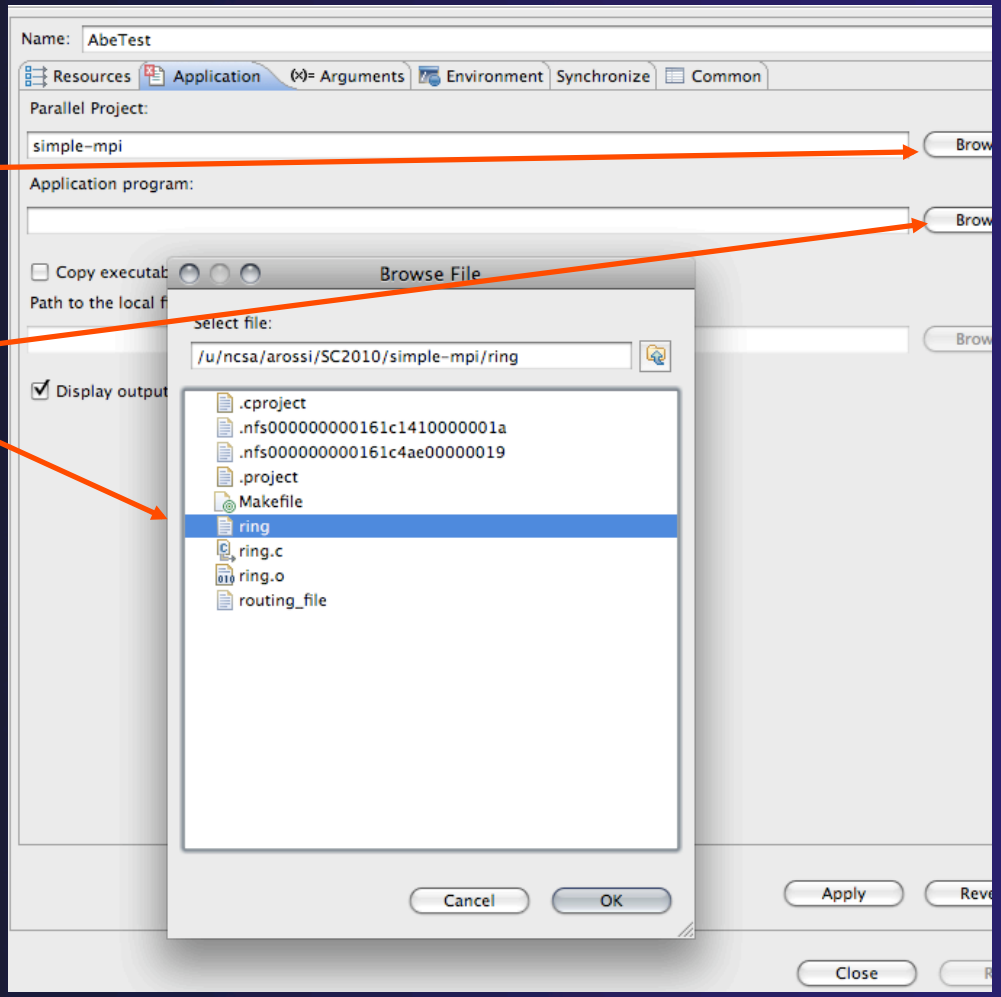
Note, however, that you still cannot click "Run". There is still a red 'X' on the Application Tab. We need to set the application for this job.

Fill in the name for this configuration ("AbeTest") and click "Apply". These values are saved and the name now appears in the list at the left.



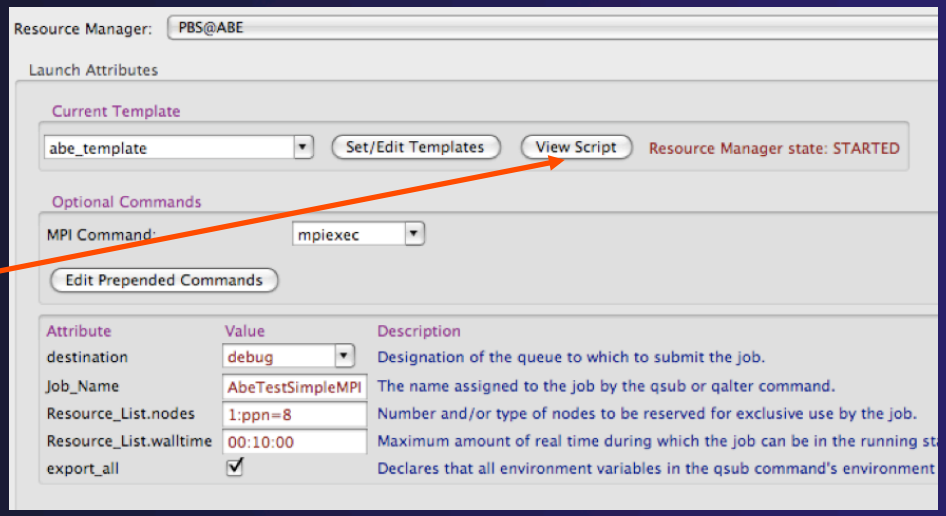
Configuring a PBS Job

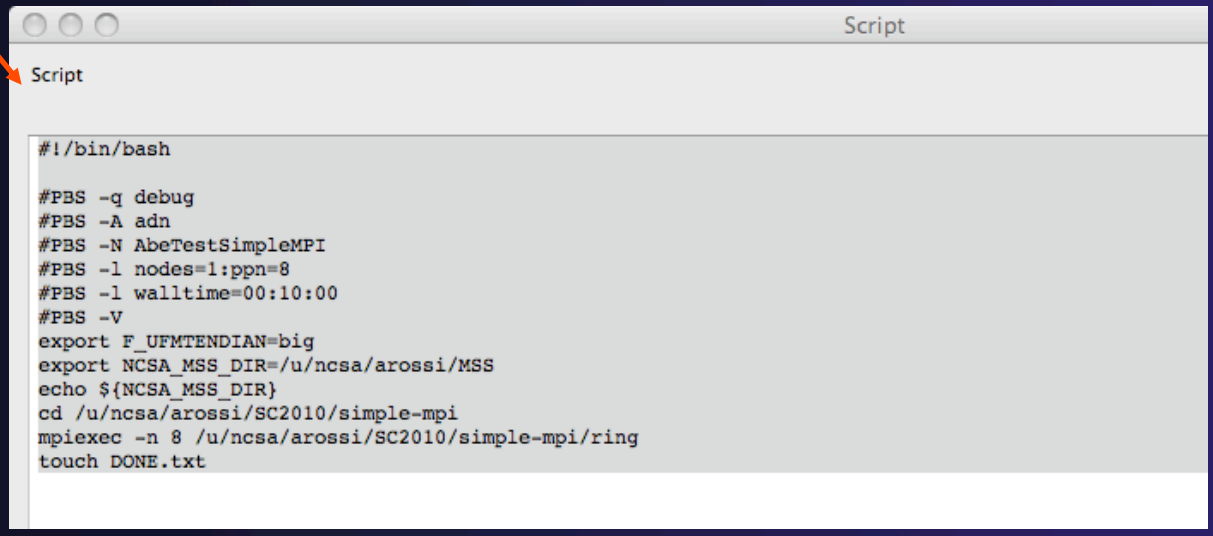
1. Click on the Application Tab.
2. Use "Browse" button to find remote project (here, "simple-mpi").
3. Use "Browse" button to find the built executable to run. (Select and click "OK".)
4. On the Application Tab, click "Apply". The red mark should disappear.
5. Click on the Resources Tab. The "Run" button should now be activated.



Configuring a PBS Job

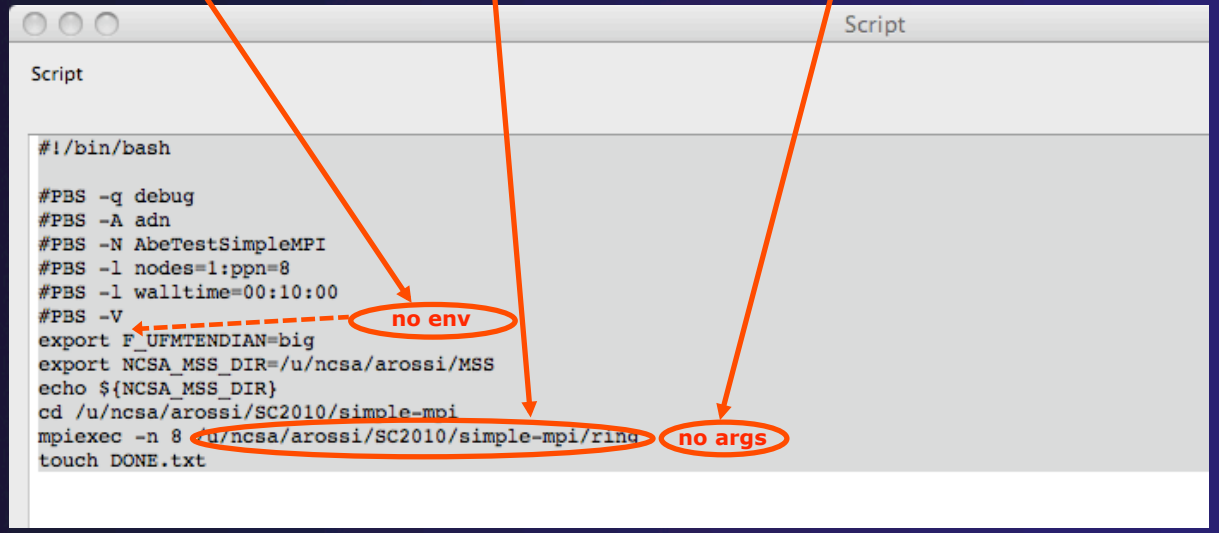
To inspect the actual batch script (with values filled in) that will be used with this launch, click "View Script"





Configuring a PBS Job

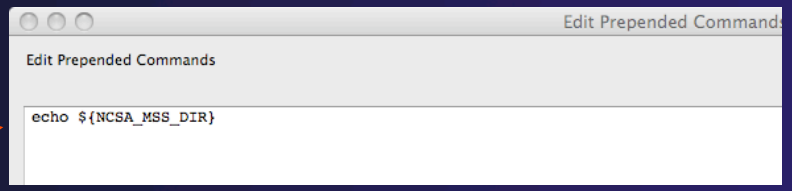
Recall that the values for the @*env*@, @*executablePath*@, and @*progArgs*@ placeholders are all determined on the basis of what is set in the Application, Arguments and Environment tabs.

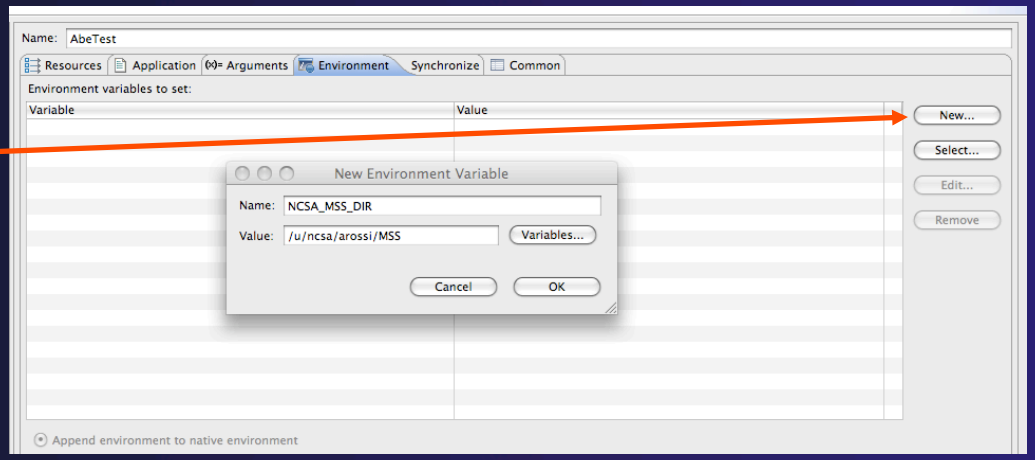


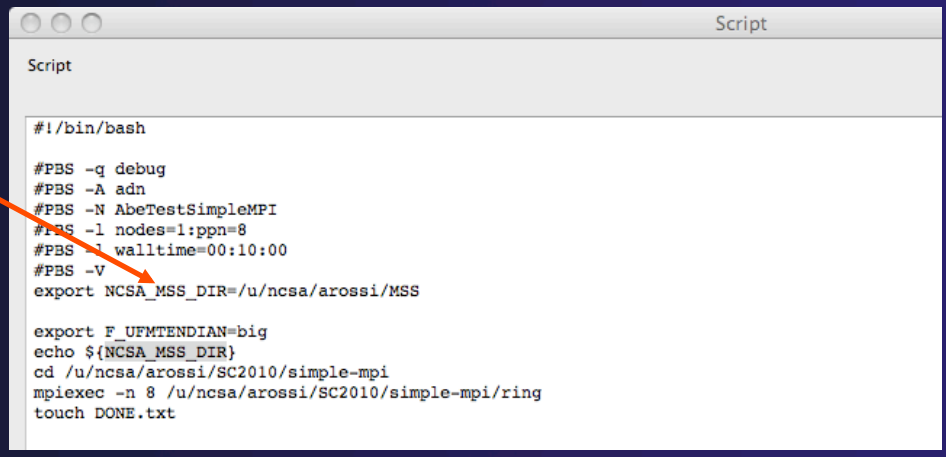
For example, we could modify the prepended command value to contain only the "echo" statement, and set the actual variable using the Environment Tab.

Configuring a PBS Job

1. Edit Prepended Commands: eliminate first line.
2. Click on Environment Tab; click "New", and fill in with the name and value we just removed; click "OK". Click "Apply".
3. Click on Resources Tab.
4. Click on "View Script".
5. Notice NCSA_MSS_DIR now appears where @*env*@ was in the template.





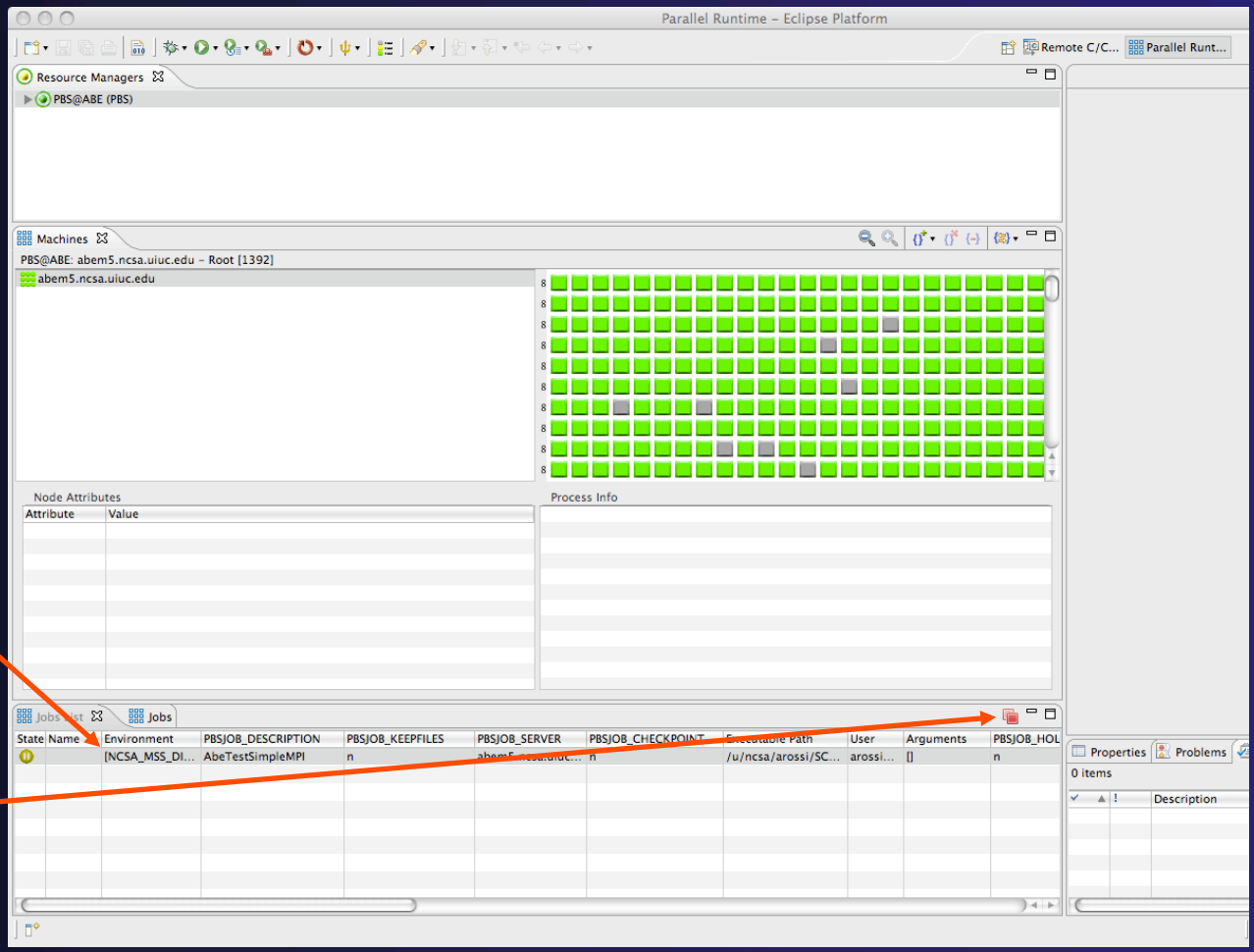


Launching a PBS Job

On the Run Configuration, click "Run" …

Job Info appears in the Jobs List view.

When you select the line, a red box appears allowing you to cancel the job if desired.



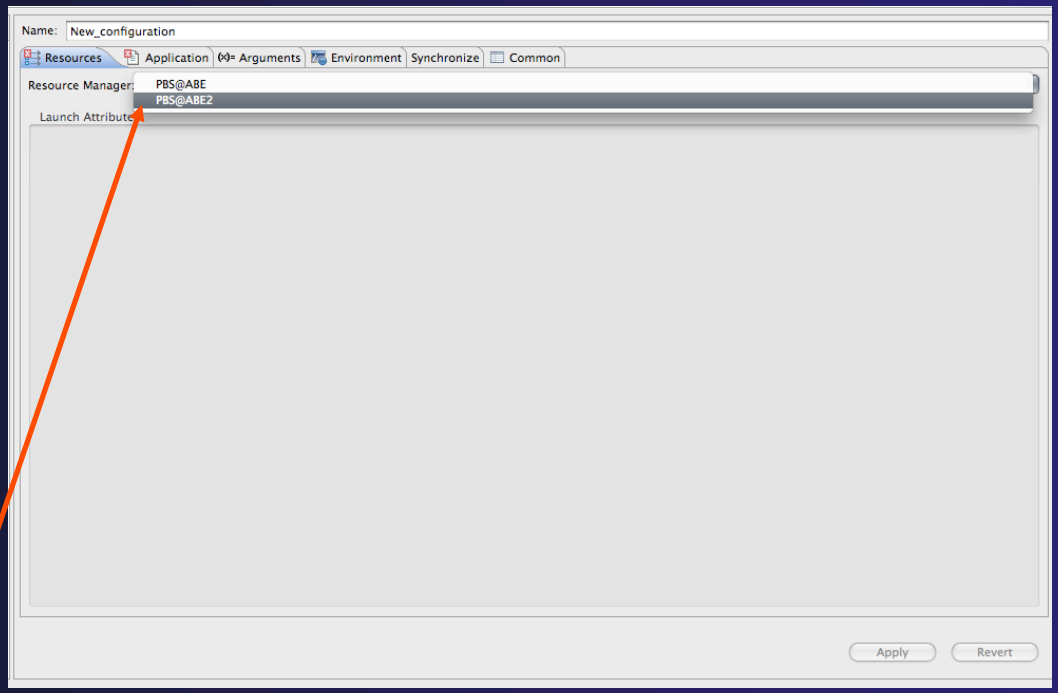
Sharing RMs and Attribute Values across Run Configurations

- A Resource Manager has an underlying configuration; this stores:
  - Connection
  - Valid attributes for the resource
  - Batch Templates
- A Launch (Run/Debug) Tab has an underlying configuration; this stores:
  - Attribute & other widget values (Resources Tab)
  - Application, Arguments, Environment, etc. values

Let’s see how this works ...

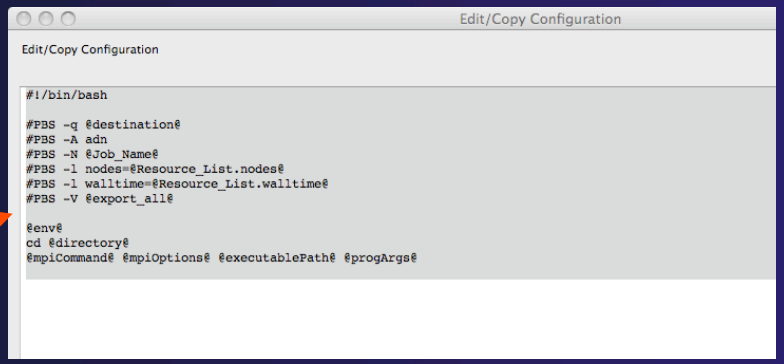
Sharing RMs and Attribute Values across Run Configurations

1. Create a new PBS Resource Manager (call it PBS@ABE2) as we did in slides 1-4.
   - ✓ *Note: PBS RMs cannot share live/active connections, so we need to create a new Remote Connection, "ABE2" (this can have the same URL), before associating the new RM with it.*
2. Start PBS@ABE2.
3. Open a new Run Configuration as we did in slides 5-6; call it *Abe2Test*.
   - ✓ *Note: now that there is more than one RM, we must choose which one to use*; choose PBS@ABE2.

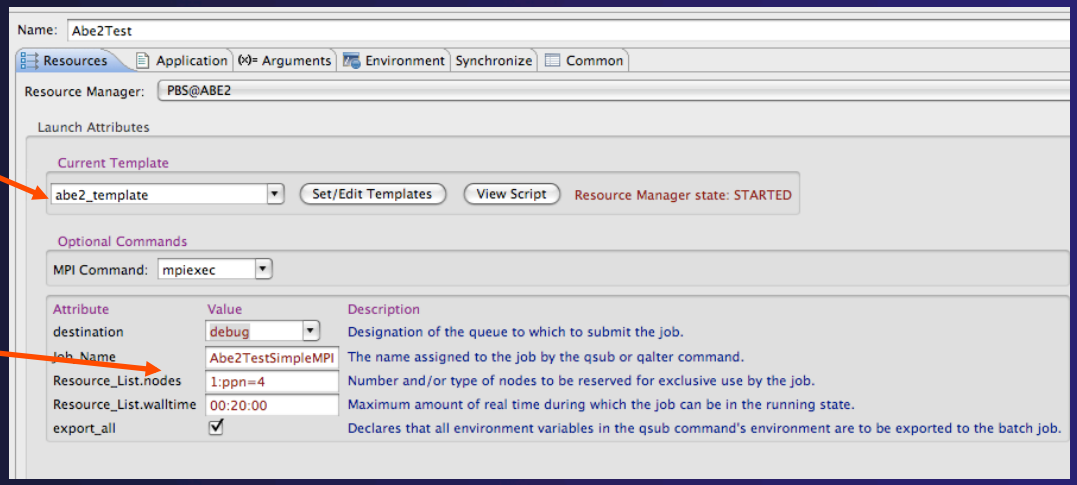


Sharing RMs and Attribute Values across Run Configurations

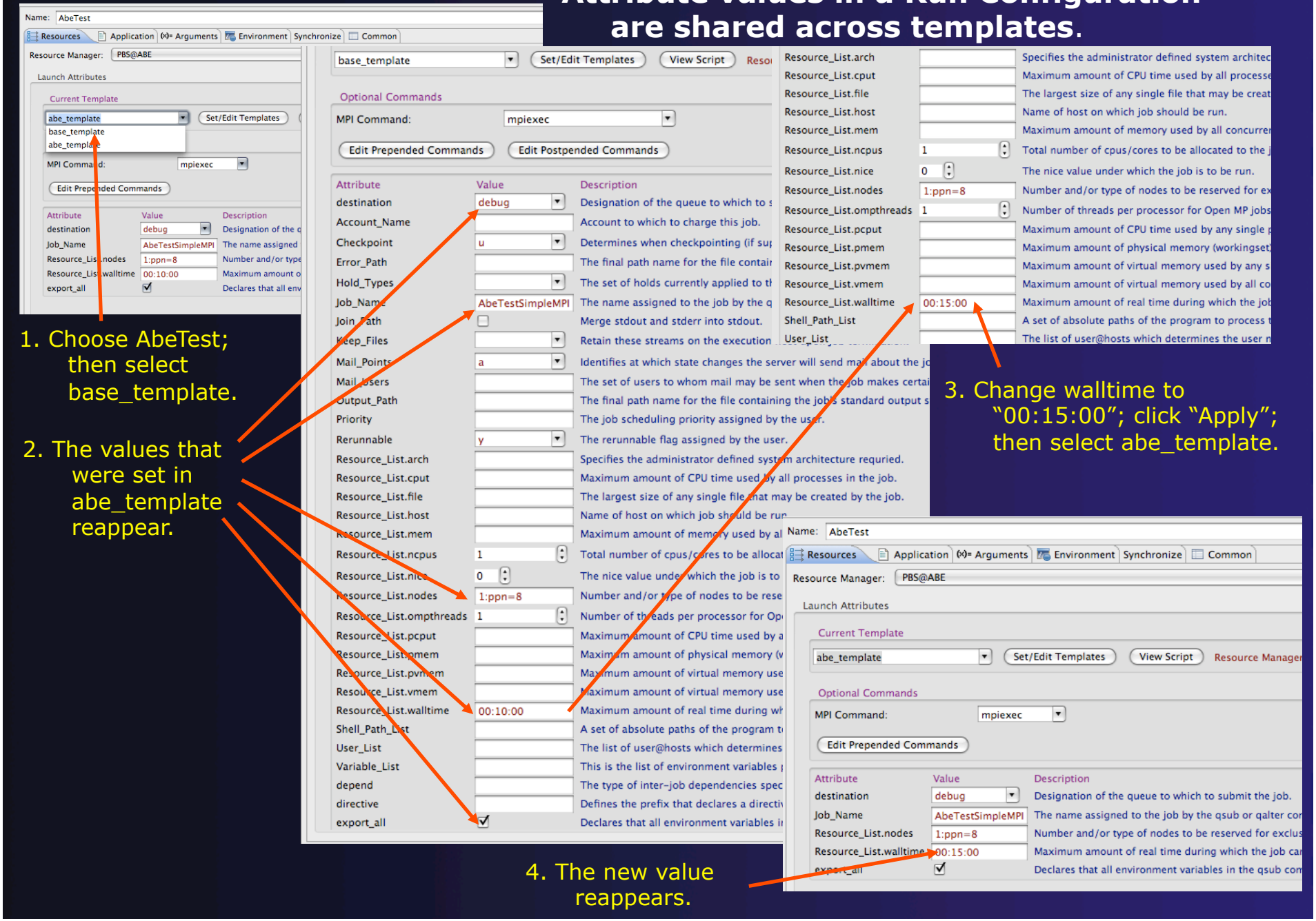
1. Create a new custom template using PBS@ABE2's base template (refer to slides 12-18). Give it a slightly different configuration (e.g., eliminate both optional buttons) so as to distinguish it; call it "abe2".



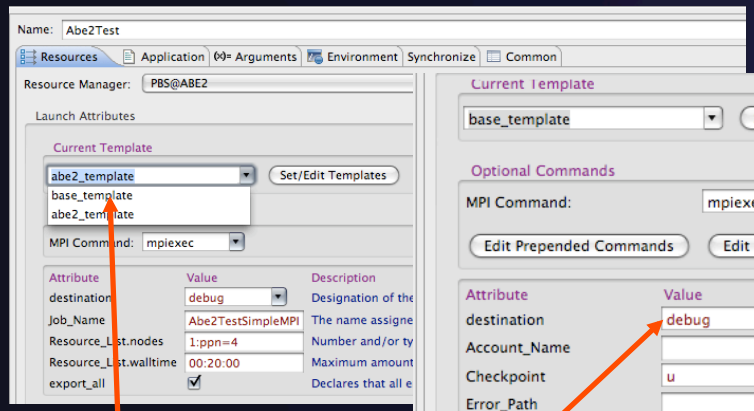
2. Choose abe2_template from Current Templates.
3. Fill in values:
   - Job_Name: Abe2TestSimpleMPI.
   - nodes: 1:ppn=4.
   - walltime: 00:20:00.



Attribute values in a Run Configuration are shared across templates.

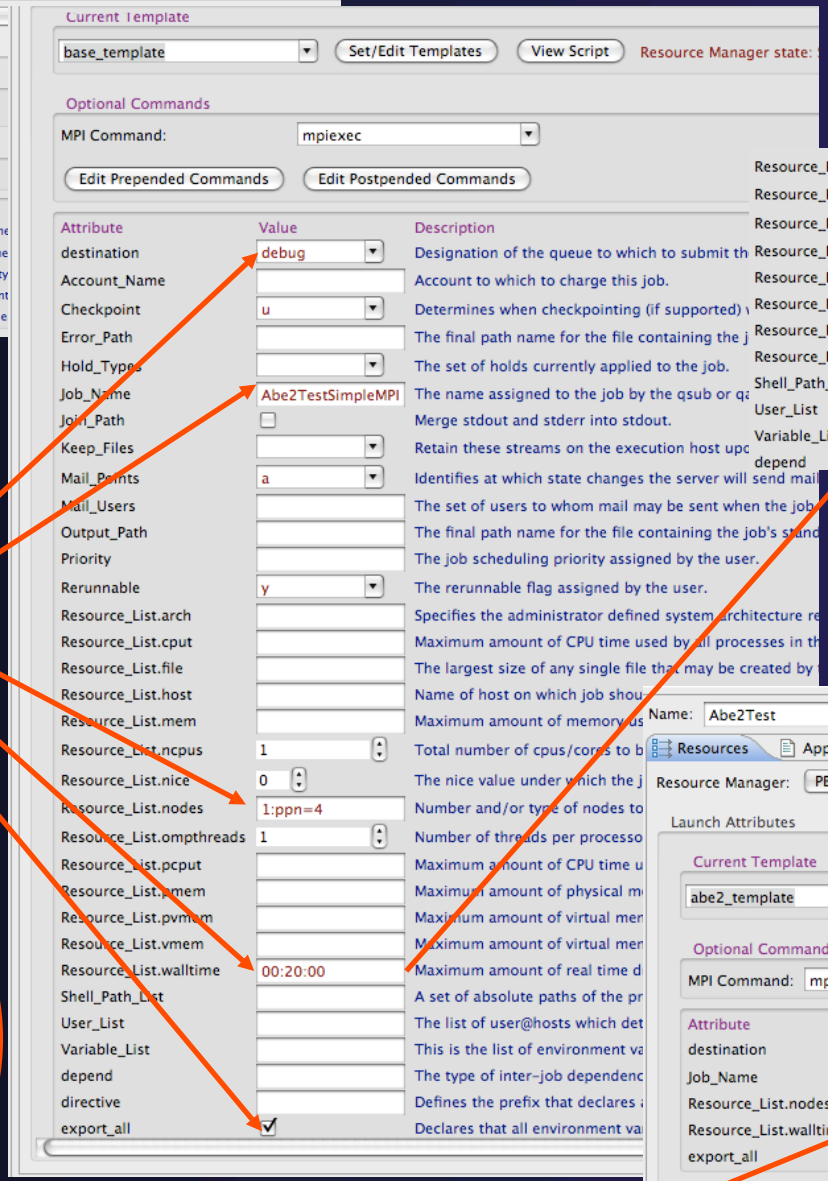


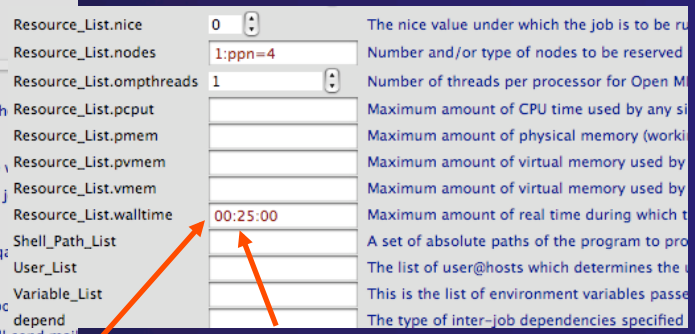
Attribute values remain bound to their Run Configuration.



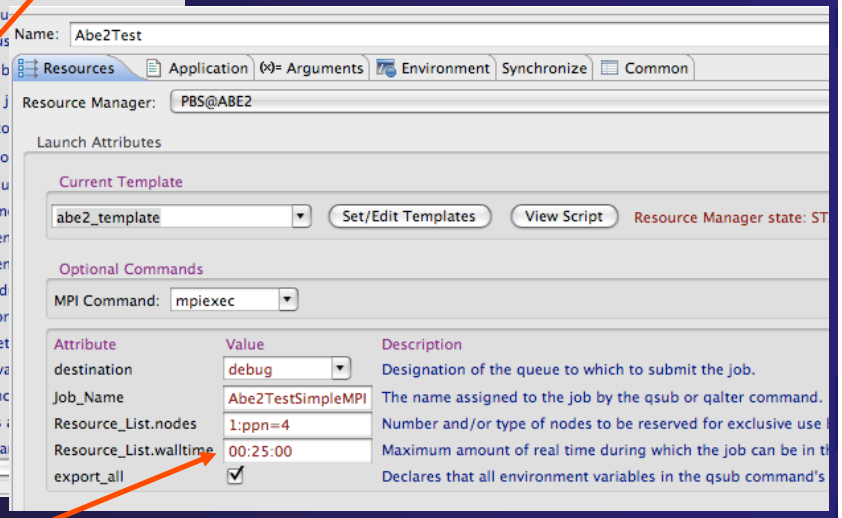
1. Choose Abe2Test; then select base_template.

2. The values that were set in abe2_template reappear.





3. Change walltime to "00:25:00"; click "Apply"; then select abe2_template.

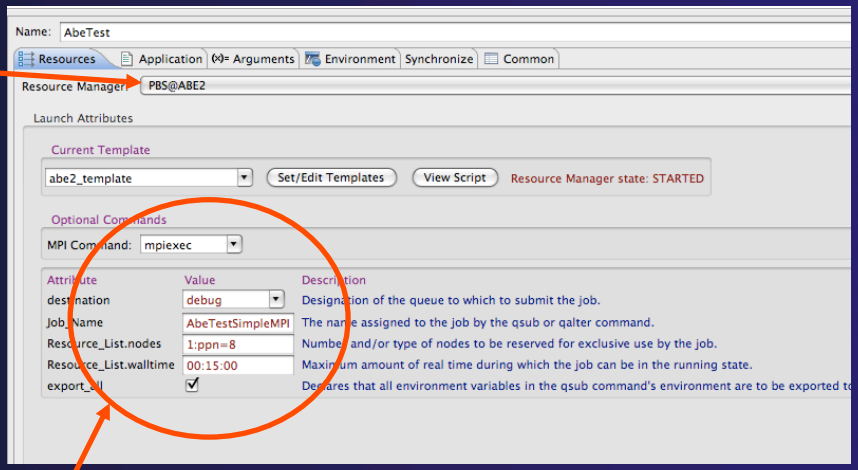


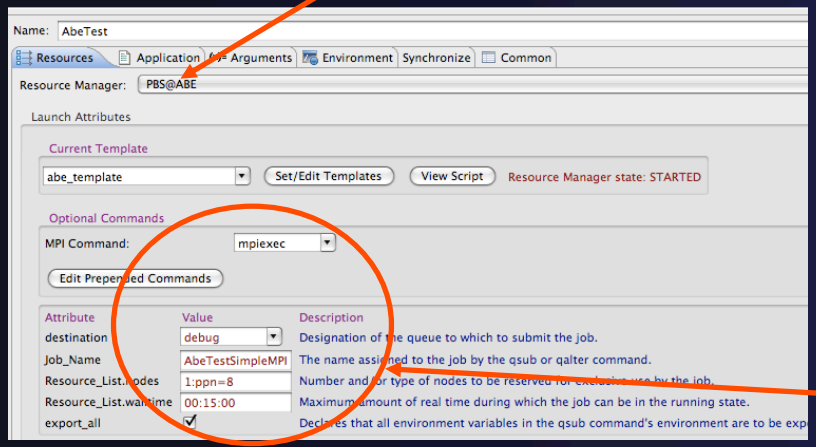
4. The new value reappears.

Inside Abe2Test, we can do the same thing without being affected by or affecting AbeTest values.

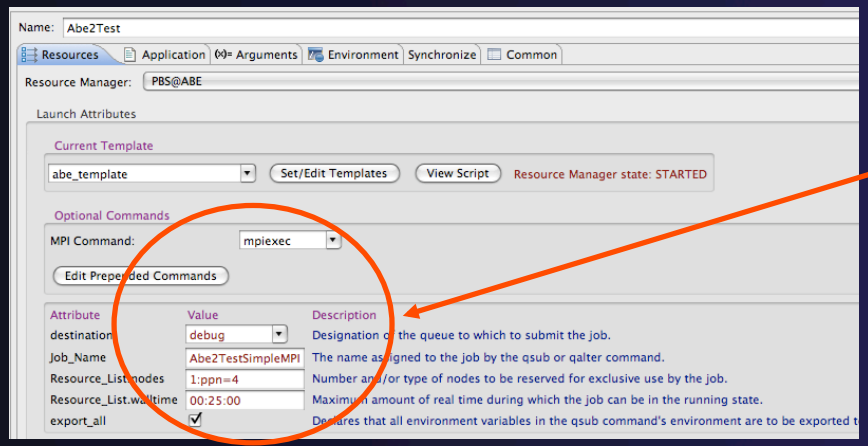
Swapping Resource Managers inside a Run Configuration

In AbeTest, switch to PBS@ABE2





abe2_template differs from abe_template, but values are same.



Similarly, switching to PBS@ABE in Abe2Test fills in abe_template with Abe2Test values.

Features to be Added (Soon)

1. Provide a “minimal” default template along with the base template.

2. Provide for export and import of templates.