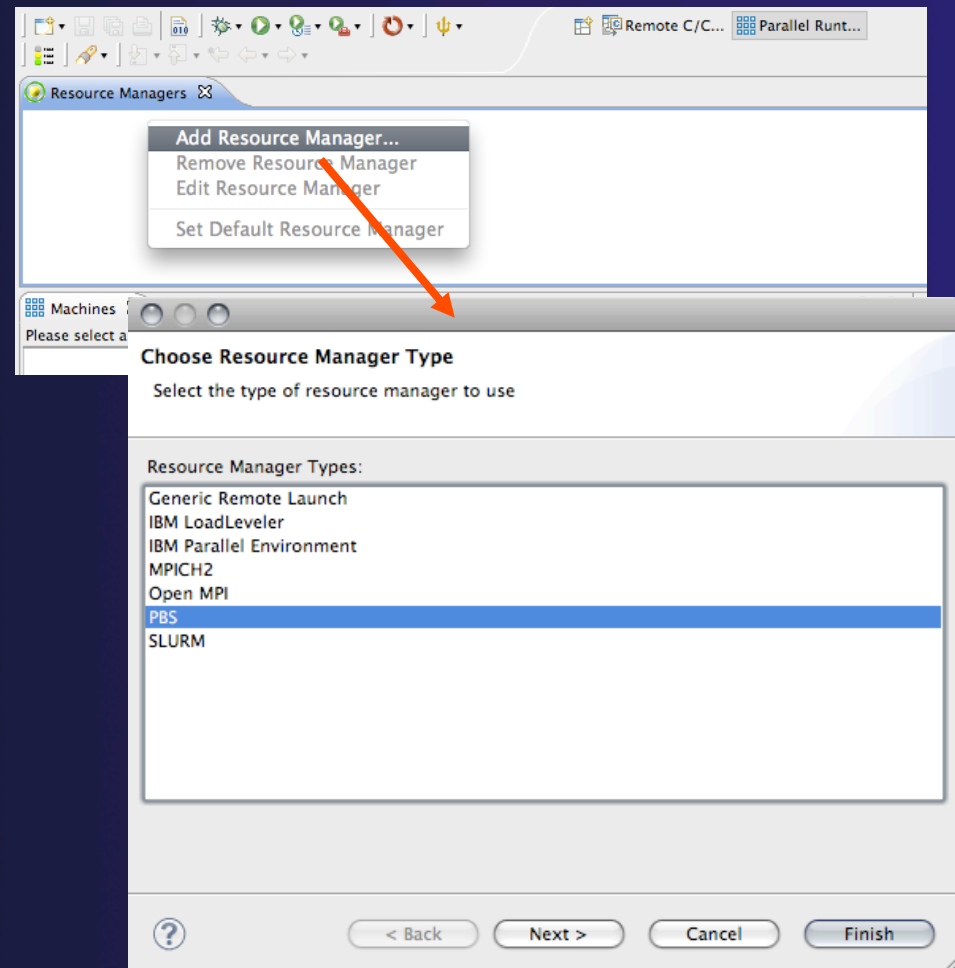


Adding a PBS Resource Manager



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

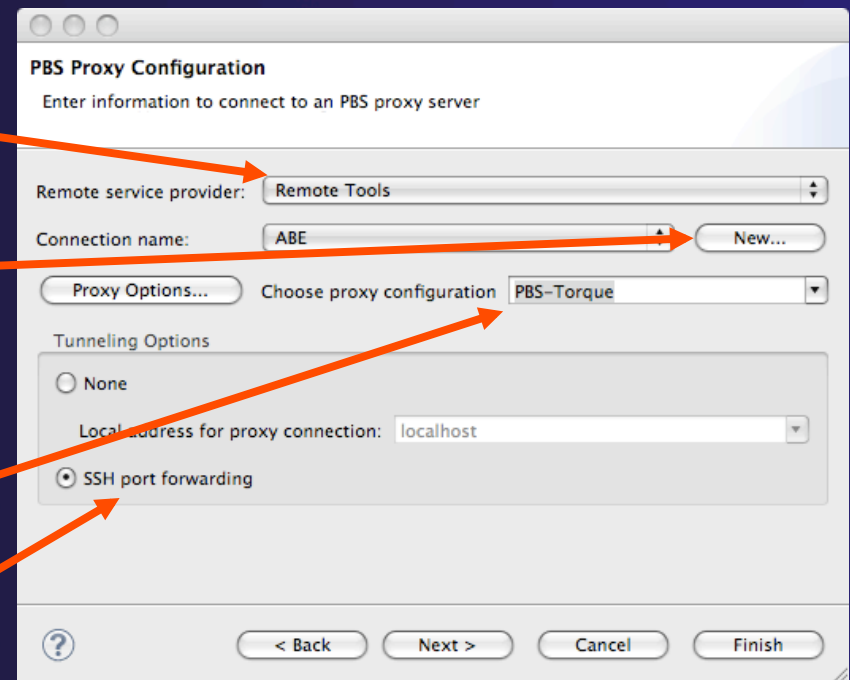


Adding a PBS Resource Manager

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.



PBS Proxy Configuration

Enter information to connect to an PBS proxy server

Remote service provider: Remote Tools

Connection name: ABE New...

Proxy Options... Choose proxy configuration PBS-Torque

Tunneling Options

None

Local address for proxy connection: localhost

SSH port forwarding

< Back Next > Cancel Finish

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>” ...

The screenshot shows a window titled "Target Environment Configuration" with a sub-header "Generic Remote Host". Below the sub-header is the text "Properties for connecting to a generic host". The form contains the following fields and options:

- Target name:
- Host Information section with two radio buttons: Localhost and Remote host
- Host:
- User:
- Authentication options: Password based authentication and Public key based authentication
- Password:
- File with private key: with a "Browse" button
- Passphrase:
- A "Simplified" button is located at the bottom right of the authentication section.
- Port: Timeout(sec):
- Cipher Type:

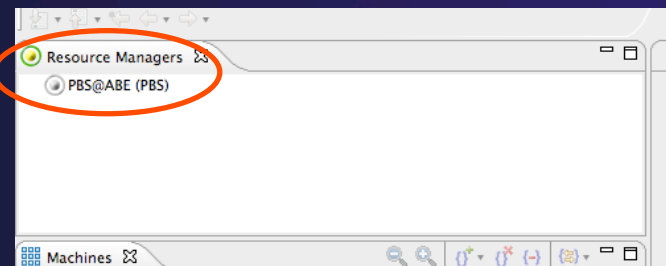
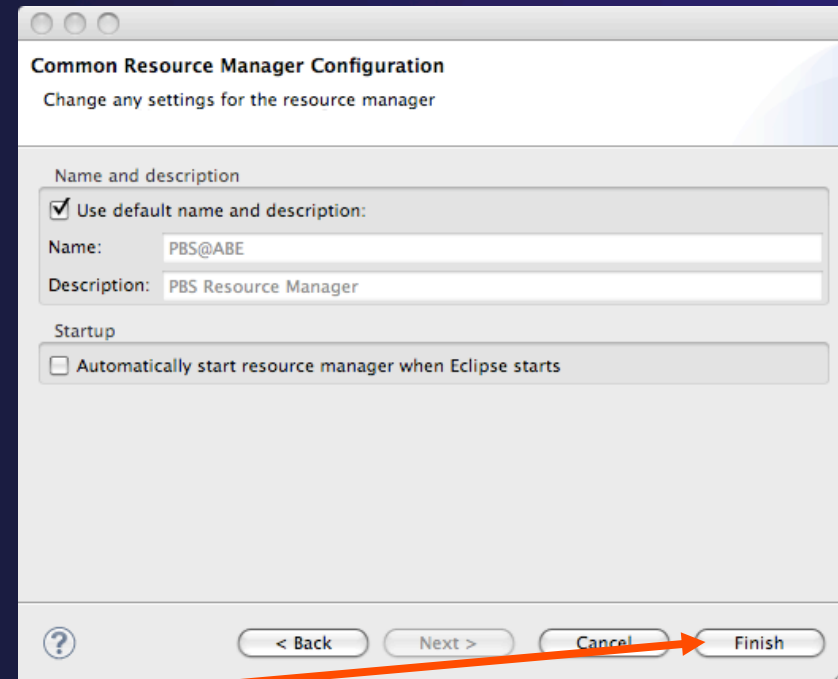
At the bottom of the window, there is a help icon (question mark in a circle), a "Cancel" button, and a "Finish" button.



Adding a PBS Resource Manager

Common RM Configuration

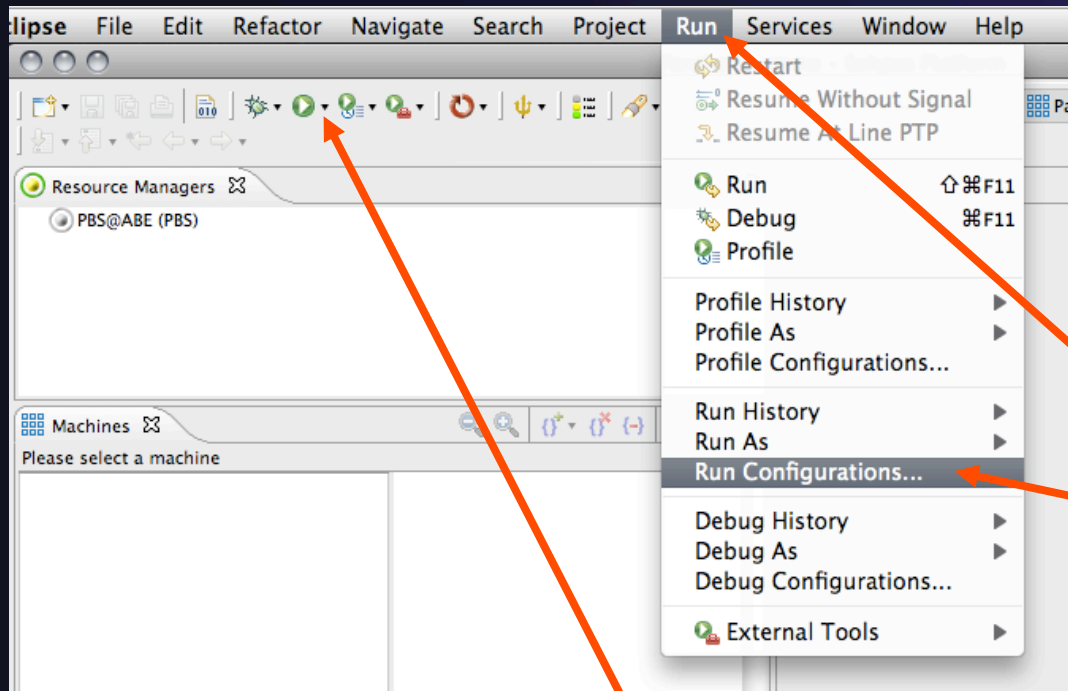
- ★ Change name
- ★ Auto start
- ★ Default usually sufficient
- ★ Click Finish, and the new RM will appear in the Resource Managers view.



Creating a PBS Run Configuration



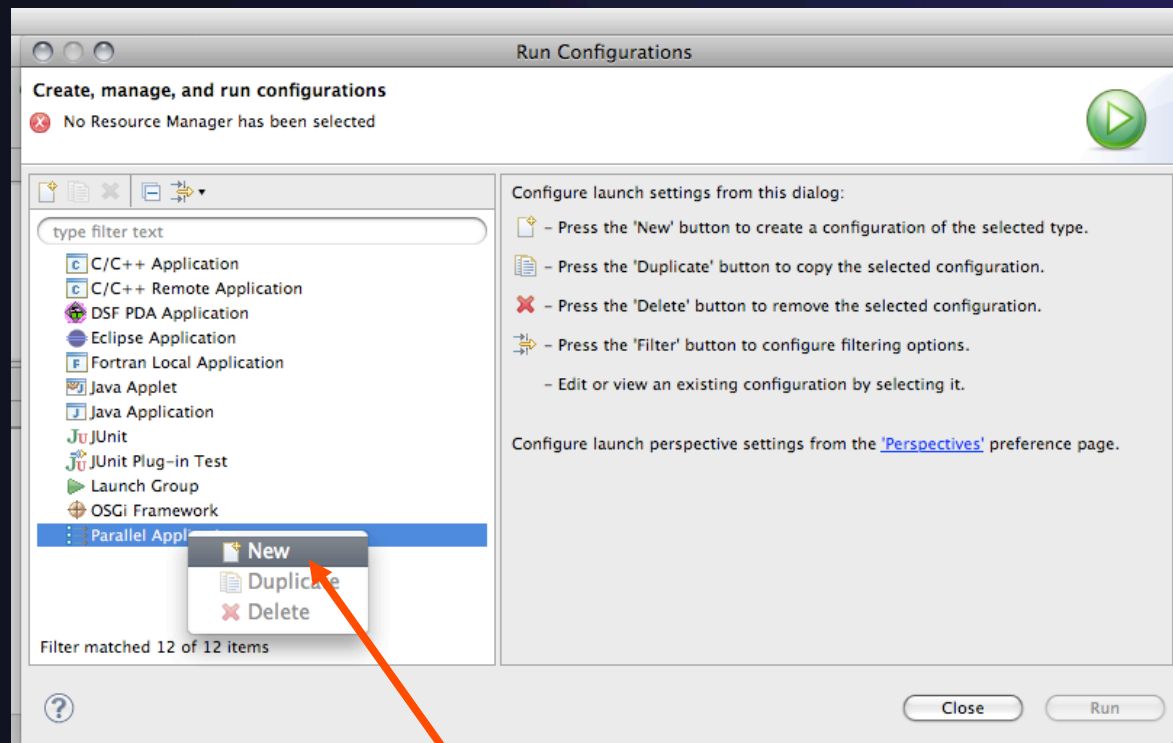
Two Ways



From Run Menu, select "Run Configurations ...".

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

Creating a PBS Run Configuration

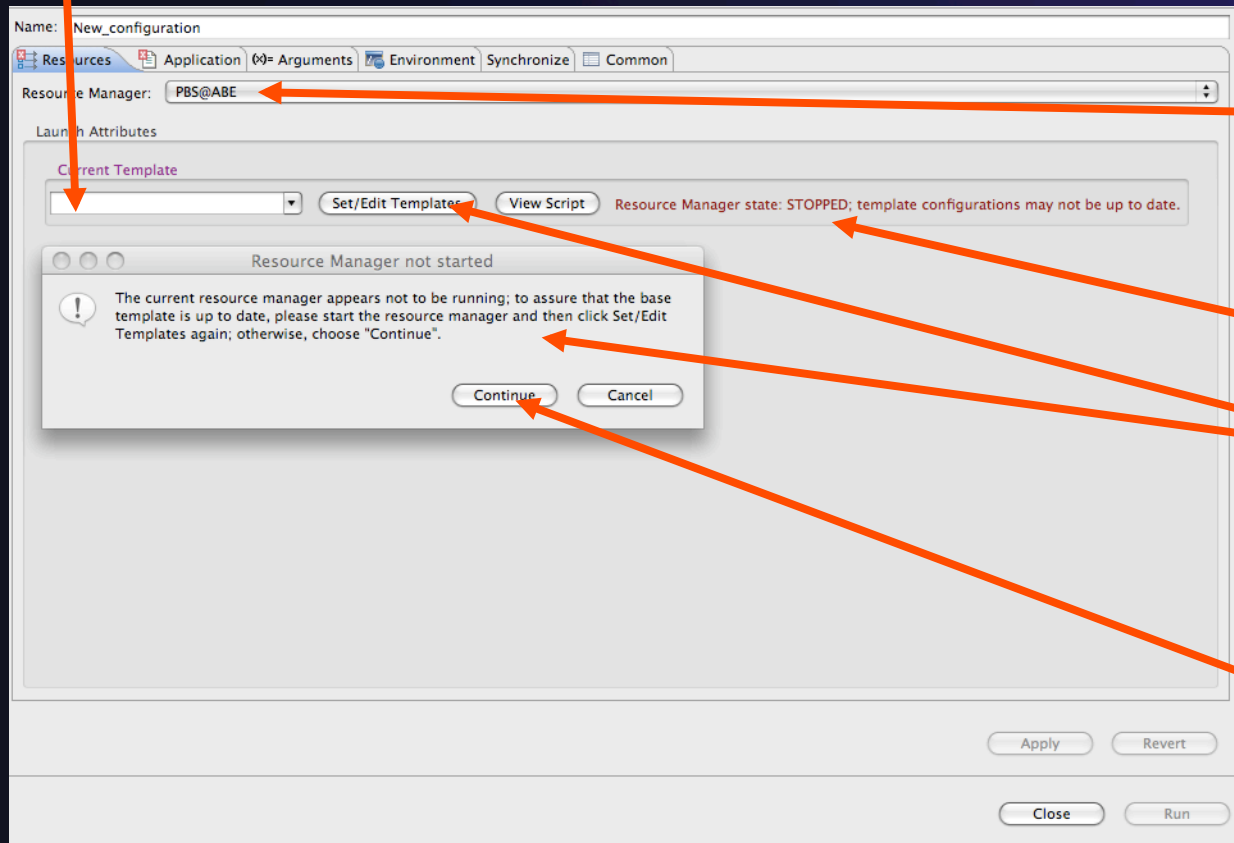


In the list of types, right click on "Parallel Application" and choose "New".

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"...

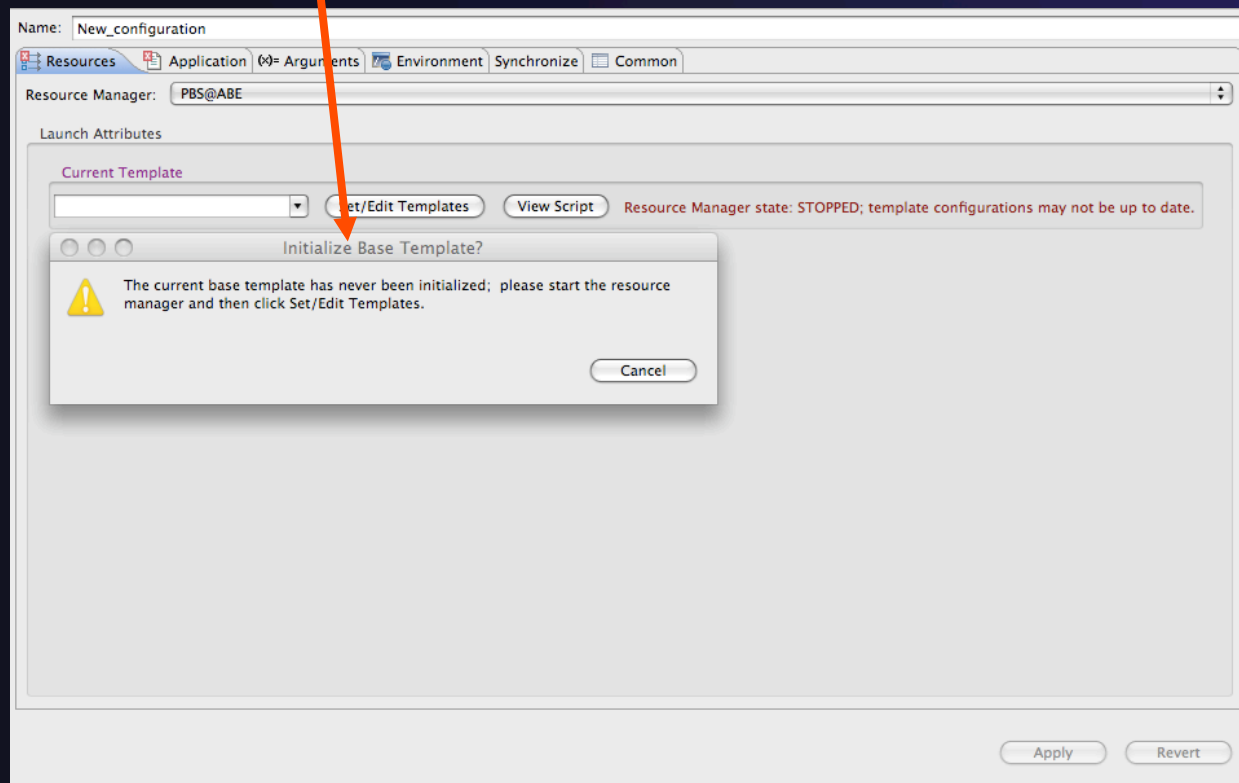
Configuring the PBS Run Configuration



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.

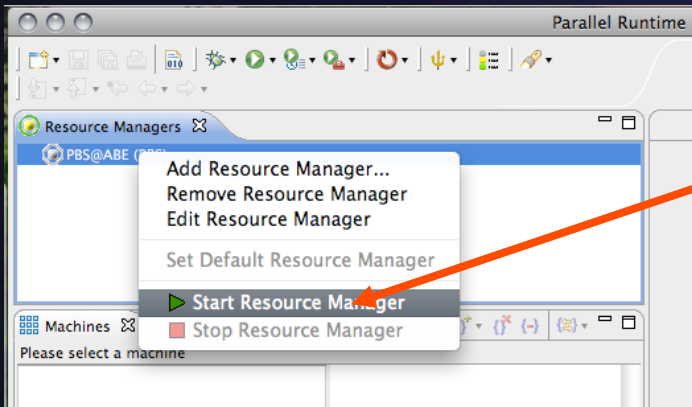
Subsequent to this initial run, you will still get the warning when the RM is not running, but "Continue" will allow you to edit the existing templates.



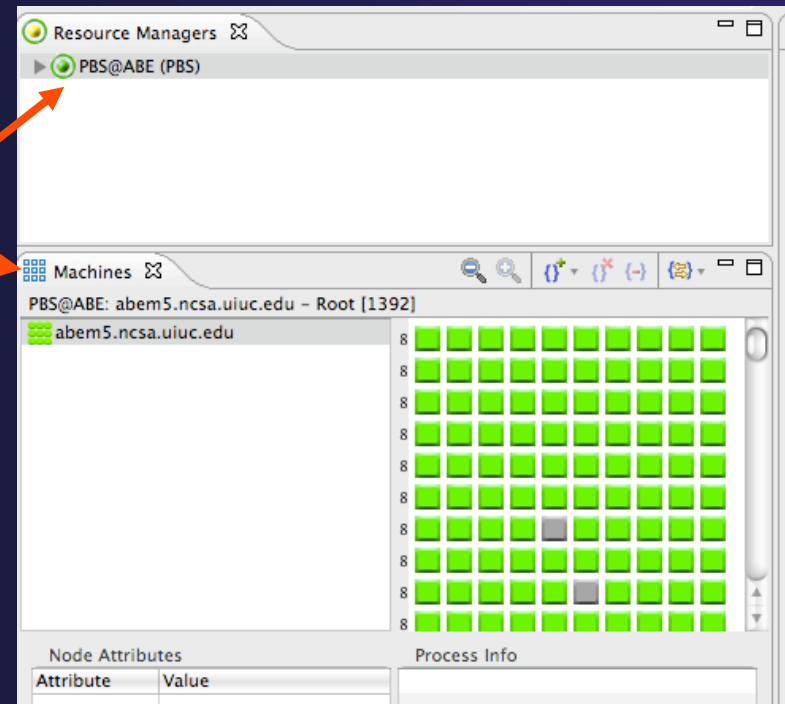


Starting the Resource Manager

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

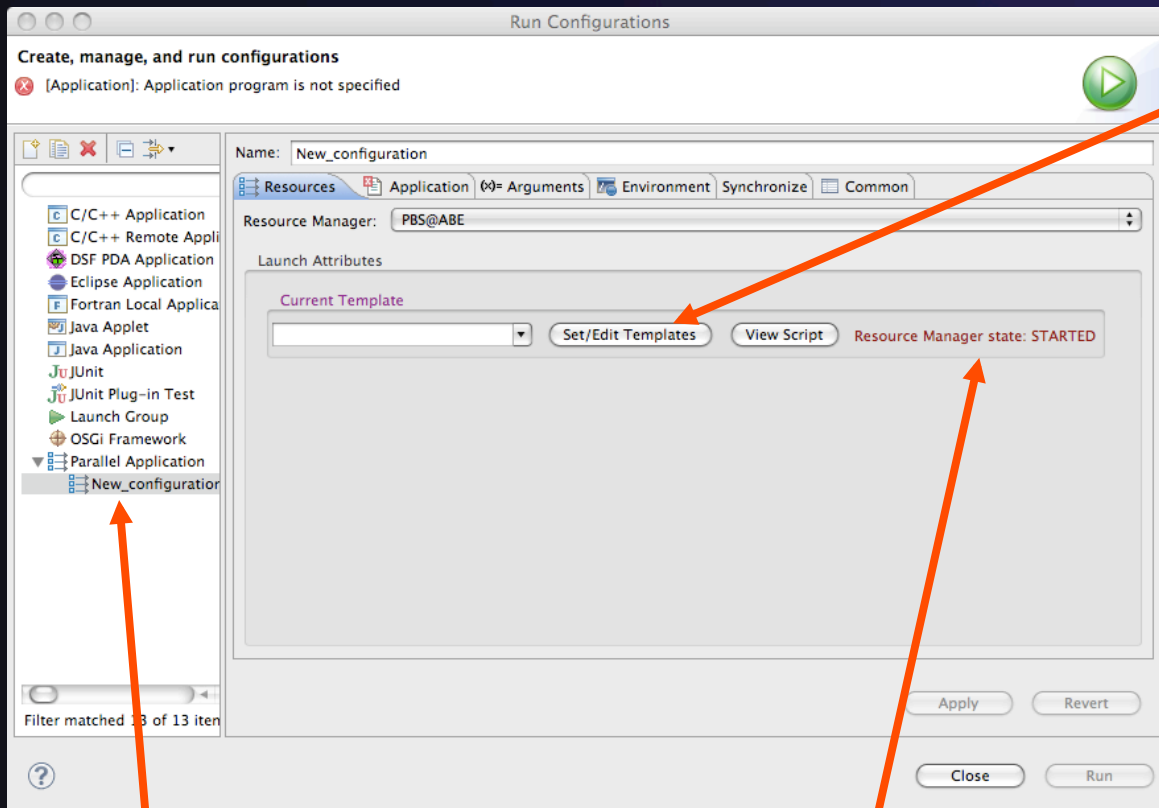


If all is well, the icon should turn green and the Machines view should populate. (Problems are indicated by a light blue or red RM icon, and possibly an error message.)

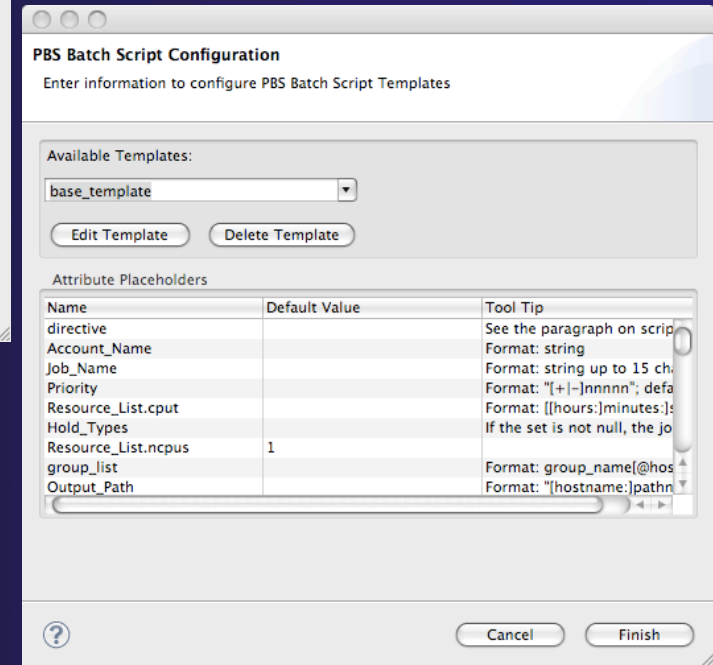




Editing PBS Batch Templates



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

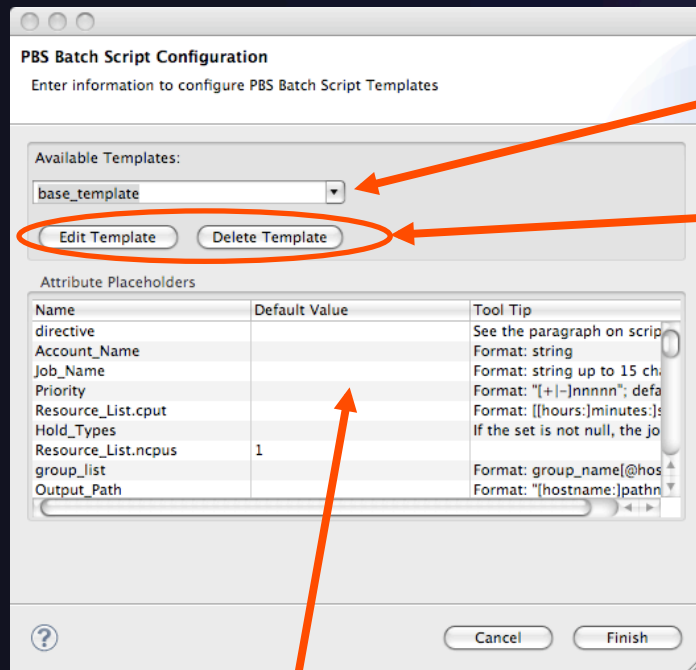


Select to re-open

Message: STARTED



Editing PBS Batch Templates

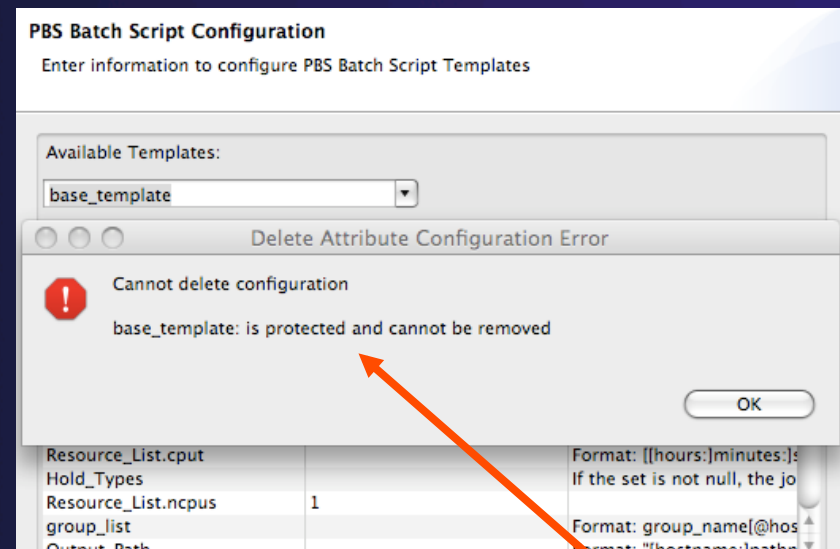


List of the templates available for this Resource Manager

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 tooltip, if one is available.

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

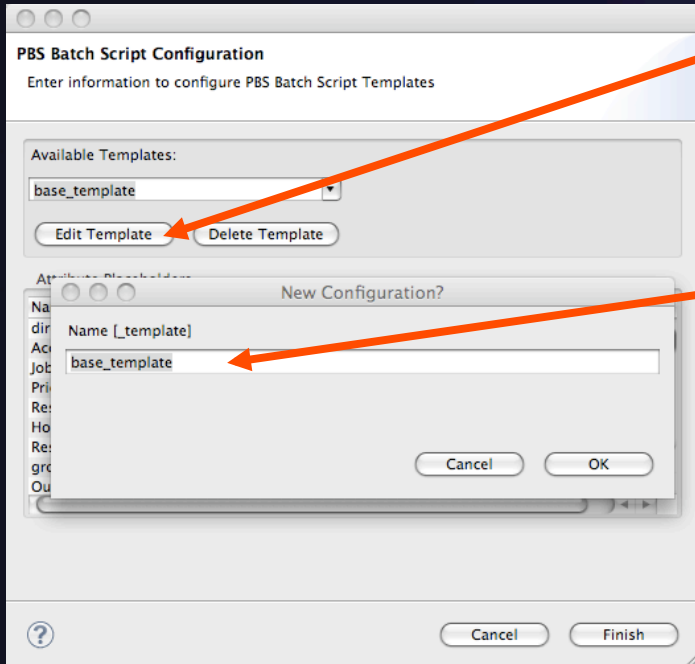
Current Template: base_template

Optional Commands: mpirexec

Attribute	Value	Description
destination		Designation of the queue to which to submit the job.
Account_Name		Account to which to charge this job.
Checkpoint	u	Determines when checkpointing (if supported) will be performed by PBS on behalf of the job.
Error_Path		The final path name for the file containing the job's standard error stream.
Hold_Types		The set of holds currently applied to the job.
Job_Name		The name assigned to the job by the qsub or qalter command.
Join_Path	<input type="checkbox"/>	Merge stdout and stderr into stdout.
Keep_Files		Retain these streams on the execution host upon job termination.
Mail_Points	a	Identifies at which state changes the server will send mail about the job.
Mail_Users		The set of users to whom mail may be sent when the job makes certain state changes.
Output_Path		The final path name for the file containing the job's standard output stream.
Priority		The job scheduling priority assigned by the user.
Rerunnable	y	The rerunnable flag assigned by the user.
Resource_List.arch		Specifies the administrator defined system architecture required.
Resource_List.cput		Maximum amount of CPU time used by all processes in the job.
Resource_List.file		The largest size of any single file that may be created by the job.
Resource_List.host		Name of host on which job should be run.
Resource_List.mem		Maximum amount of memory used by all concurrent processes in the job.
Resource_List.ncpus	1	Total number of cpus/cores to be allocated to the job.
Resource_List.nice	0	The nice value under which the job is to be run.
Resource_List.nodes	1	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.ppn		Number of processors per node for Open MP jobs.
Resource_List.time		CPU time used by any single process in the job.
Resource_List.vmem		physical memory (workingset) used by any single process of the job.
Resource_List.vmem_max		virtual memory used by any single process in the job.
Resource_List.vmem_per_node		virtual memory used by all concurrent processes in the job.
Resource_List.walltime		real time during which the job can be in the running state.
Shell_Path_List		paths of the program to process the job's script file.

Resource_List.nodes tooltip: The value is one or more node_specs joined with the '+' character, 'node_spec(+node_spec)'. Each node_spec is a number of nodes required of the type declared in the node_spec and a name or one or more property or properties desired for the nodes. The number, the name, and each property in the node_spec are separated by a colon ':'. If no number is specified, one (1) is assumed. Units: string. The name of a node is its hostname. The properties of nodes are: ppn=# requested; defaults to 1; or an arbitrary string defined by system administrator. Example: To ask for 2 processors on each of two blue nodes and three processors on one red node: -l nodes=2:blue:ppn=2+red:ppn=3.

Editing PBS Batch Templates

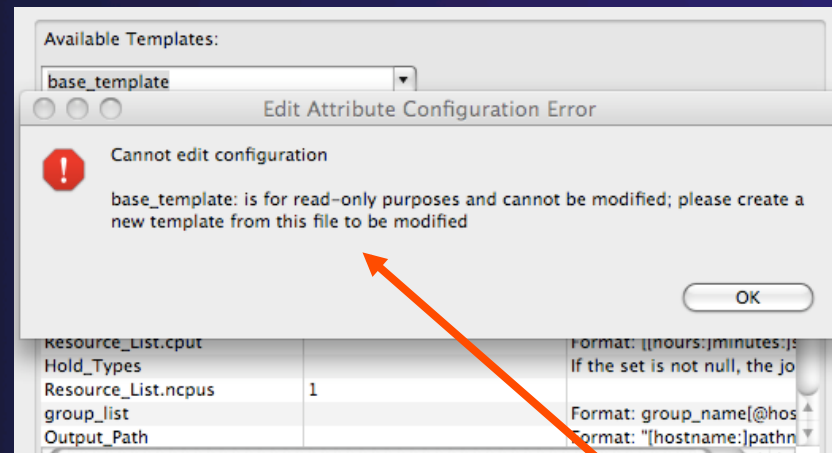


Choose "Edit Template".

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 ...

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



Not possible.

Editing PBS Batch Templates



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

```

Edit/Copy Configuration

Edit/Copy Configuration

#!/bin/bash

#####
## Template for PBS Batch Script Generated by PBS Resource Manager
##
## This template contains all the Job Attributes recognized as valid
## by a given PBS proxy instance.
##
## Placeholders (@NAME@) are included for the PBS Job Attribute
## names as specified by qsub, plus the following internal variables:
##
## - env                : place for defining extra environment
##                       : variables (NB: should not be removed)
## - prependedBash     : dynamically change arbitrary bash
##                       : commands which should precede the
##                       : execution of the main application code
## - mpiCommand mpiOptions : run under MPI
## - executablePath progArgs : the actual application
## - postpendBash      : dynamically change arbitrary bash
##                       : commands which should follow the
##                       : execution of the main application code
##
## A template can also directly contain arbitrary shell scripting (not
## to be replaced via the 'prepended' and 'postpendBash' placeholders);
## these lines will remain fixed and will not be exposed through
## the Launch Tab for modification (they can however be altered by
## using the Resource Manager Properties "Edit" Tab).
##
## NOTE: We advise removing either the ncpus or the nodes resource,
##       depending on the PBS configuration (nodes is more common);
##       otherwise, the correct value must be set on both redundantly
##       in order for the MPI computation to be correct (and
##       some systems might reject a script with both set).
#####
#PBS -q @destination@
#PBS -A @Account_Name@
#PBS -c @Checkpoint@
#PBS -e @Error_Path@
#PBS -h @Hold_Types@

Cancel OK
```

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 -x @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 pcpus=@Resource_List.pcpus@
#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@
@prependingBash@
cd @directory@
@mpiCommand@ @mpiOptions@ @executablePath@ @progArgs@
@postpendingBash@
```

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.

@prependingBash@, @postpendingBash@

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 -x @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 pcpus=@Resource_List.pcpus@
#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 @dir@tory@
@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.

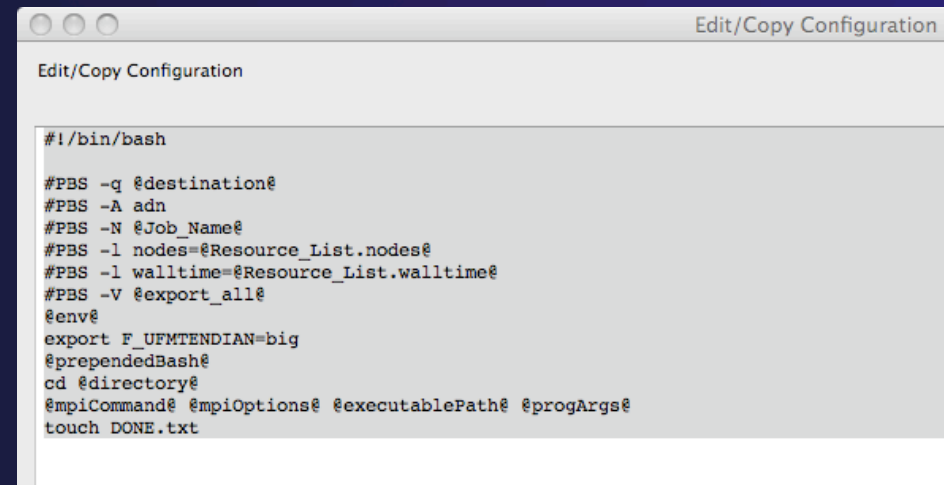
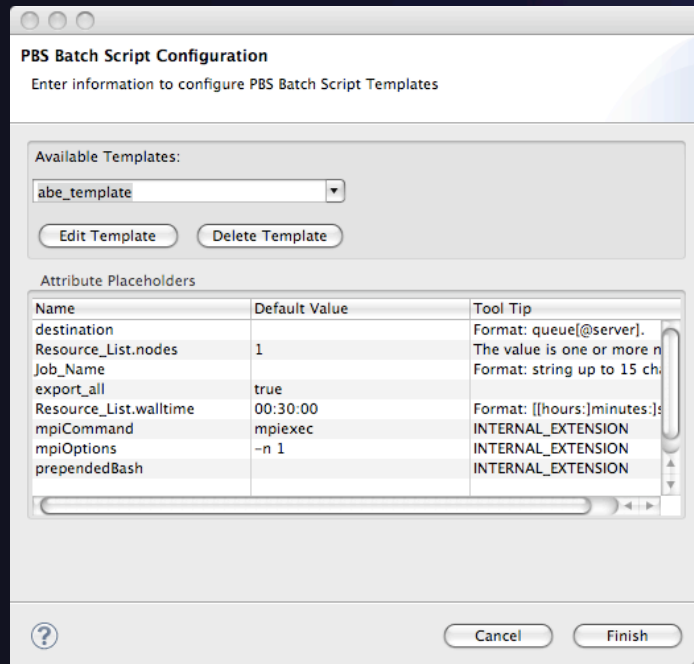
Editing PBS Batch Templates



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.

Name: New_configuration

Resource Manager: PBS@ABE

Launch Attributes

Current Template: abe_template (Set/Edit Templates View Script Resource Manager state: STARTED)

Optional Commands

MPI Command: mpiexec (Edit Prepended Commands)

Attribute	Value	Description
destination		Designation of the queue to which to submit the job.
Job_Name		The name assigned to the job by the qsub or qalter command.
Resource_List.nodes	1	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.walltime	00:30:00	Maximum amount of real time during which the job can be in the running state.
export_all	<input checked="" type="checkbox"/>	Declares that all environment variables in the qsub command's environment are to be exported to the batch job.

Apply Revert

Close Run

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

Configuring a PBS Job



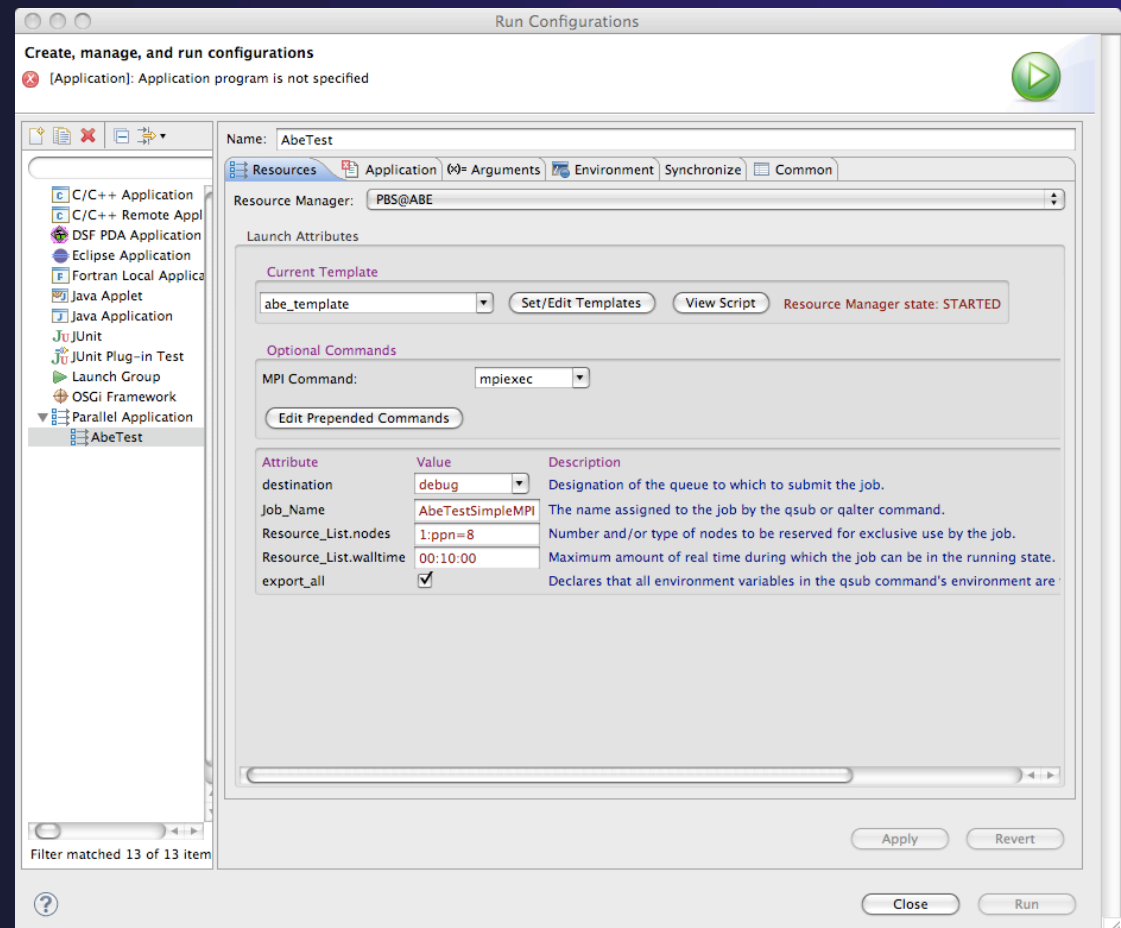
Now let's fill in some values for this particular job.

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

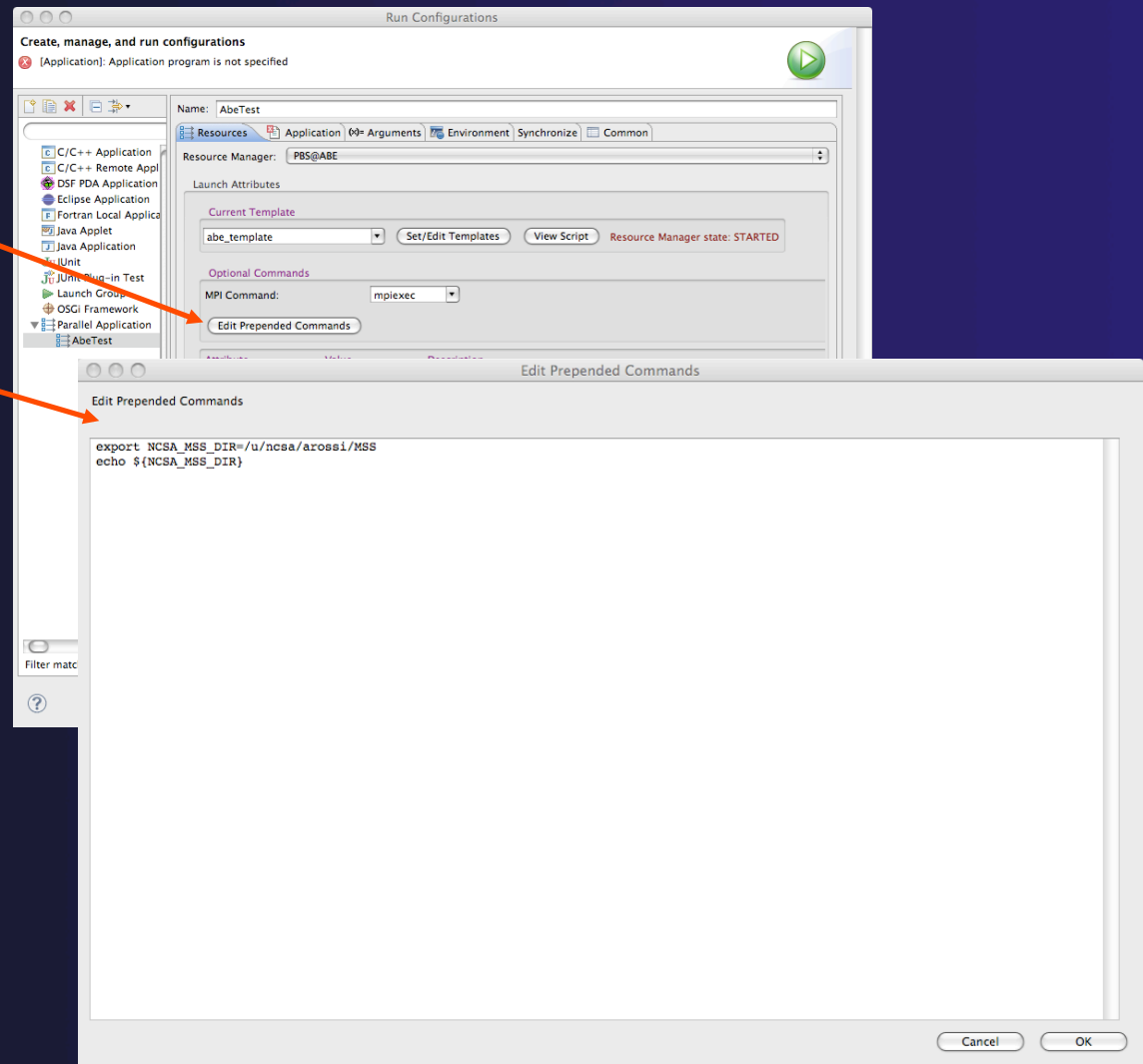


Let's also prepend some commands.

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.

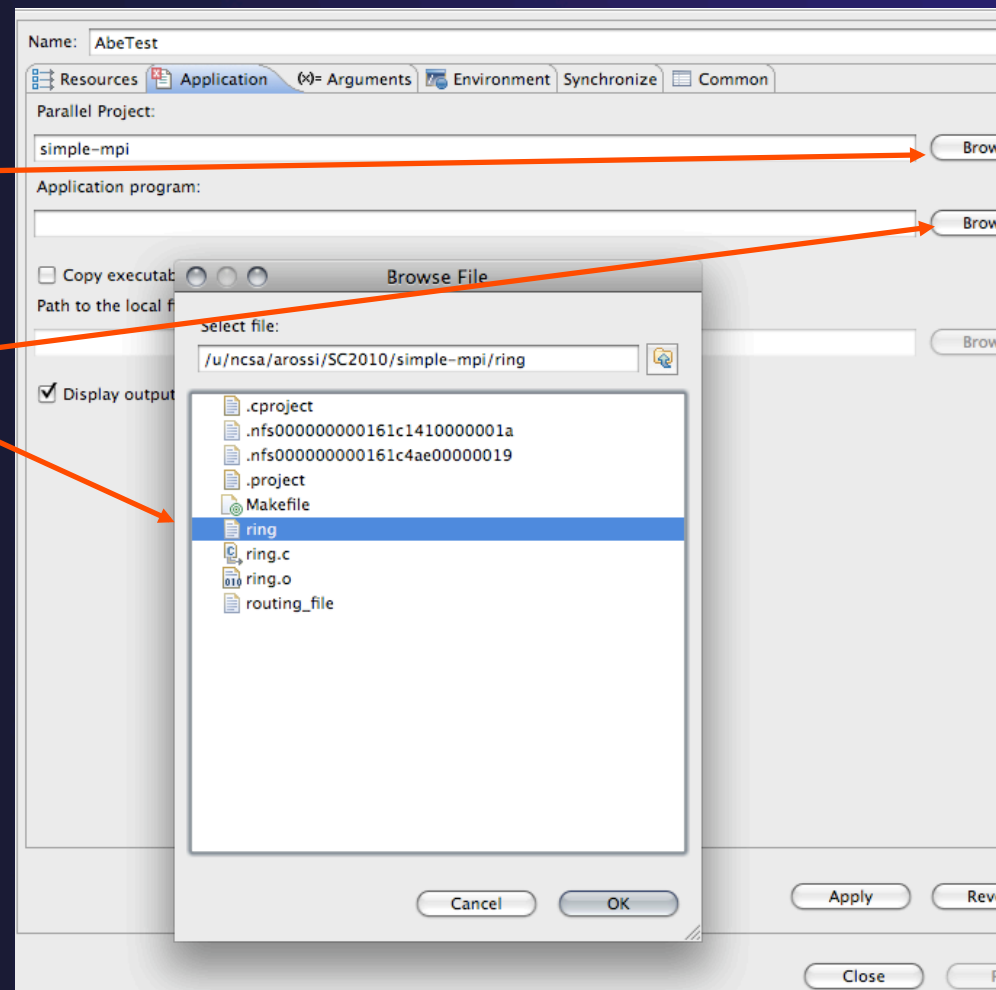
Fill in the name for this configuration ("AbeTest") and click "Apply". These values are saved and the name now appears in the list at the left.

Attribute	Value	Description
destination	debug	Designation of the queue to which to submit the job.
Job_Name	AbeTestSimpleMPI	The name assigned to the job by the qsub or qalter command.
Resource_List.nodes	1:ppn=8	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.walltime	00:10:00	Maximum amount of real time during which the job can be in the running state.
export_all	<input checked="" type="checkbox"/>	Declares that all environment variables in the qsub command's environment are

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"

Resource Manager: PBS@ABE

Launch Attributes

Current Template: Resource Manager state: STARTED

Optional Commands

MPI Command:

Attribute	Value	Description
destination	<input type="text" value="debug"/>	Designation of the queue to which to submit the job.
Job_Name	<input type="text" value="AbeTestSimpleMPI"/>	The name assigned to the job by the qsub or qalter command.
Resource_List.nodes	<input type="text" value="1:ppn=8"/>	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.walltime	<input type="text" value="00:10:00"/>	Maximum amount of real time during which the job can be in the running state.
export_all	<input checked="" type="checkbox"/>	Declares that all environment variables in the qsub command's environment will be exported to the job.

Script

```
#!/bin/bash

#PBS -q debug
#PBS -A adn
#PBS -N AbeTestSimpleMPI
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:10:00
#PBS -v
export F_UFMTENDIAN=big
export NCSA_MSS_DIR=/u/ncsa/arossi/MSS
echo ${NCSA_MSS_DIR}
cd /u/ncsa/arossi/SC2010/simple-mpi
mpiexec -n 8 /u/ncsa/arossi/SC2010/simple-mpi/ring
touch DONE.txt
```

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.

```
Script
Script
#!/bin/bash
#PBS -q debug
#PBS -A adn
#PBS -N AbeTestSimpleMPI
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:10:00
#PBS -V
export F_UFMTENDIAN=big
export NCSA_MSS_DIR=/u/ncsa/arossi/MSS
echo ${NCSA_MSS_DIR}
cd /u/ncsa/arossi/SC2010/simple-mpi
mpirexec -n 8 /u/ncsa/arossi/SC2010/simple-mpi/ring
touch DONE.txt
```

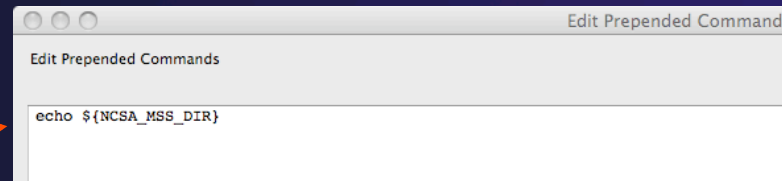
The image shows a terminal window titled "Script" containing a PBS job script. Three red annotations are present: a dashed arrow points from the text "no env" to the `#PBS -V` line; a solid arrow points from the text "no env" to the `mpirexec` command; and a solid arrow points from the text "no args" to the path `/u/ncsa/arossi/SC2010/simple-mpi/ring` in the `mpirexec` command.

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

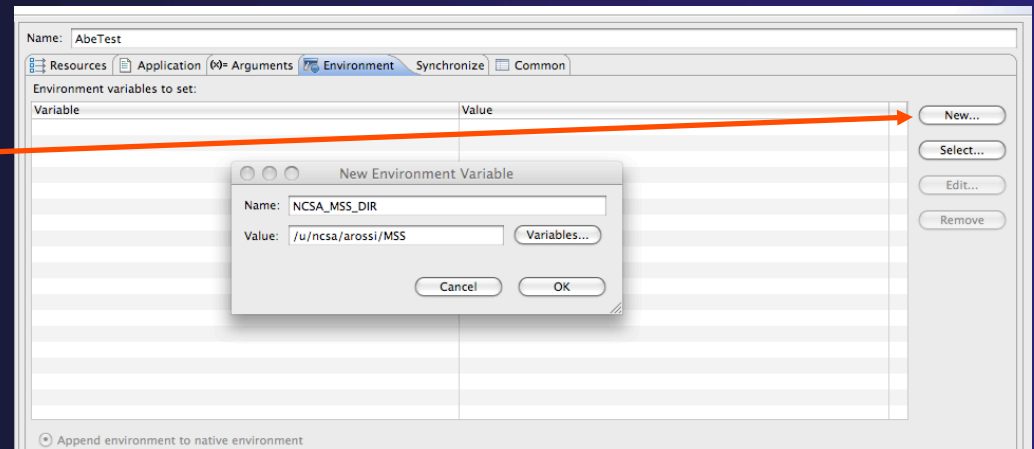
Configuring a PBS Job



1. Edit Prepended Commands: eliminate first line.



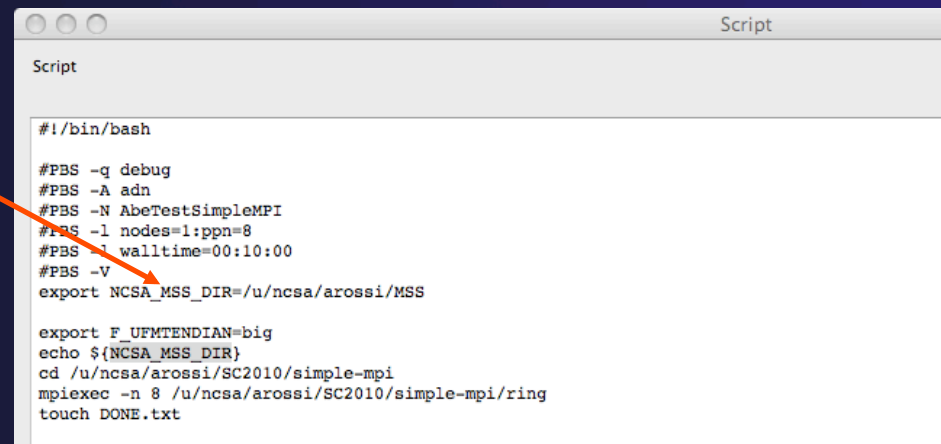
2. Click on Environment Tab; click "New", and fill in with the name and value we just removed; click "OK". Click "Apply".



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.

The screenshot shows the Eclipse Parallel Runtime interface. The top toolbar contains various icons for job management. The 'Resource Managers' view shows 'PBS@ABE (PBS)'. The 'Machines' view shows 'PBS@ABE: abem5.ncsa.uiuc.edu - Root [1392]' and a grid of 8x8 green squares representing a job configuration. Below this are 'Node Attributes' and 'Process Info' sections. The 'Jobs List' view at the bottom contains a table with the following data:

State	Name	Environment	PBSJOB_DESCRIPTION	PBSJOB_KEEFILES	PBSJOB_SERVER	PBSJOB_CHECKPOINT	Executable Path	User	Arguments	PBSJOB_HOL
🟡		[NCSA_MSS_DI...	AbeTestSimpleMPI	n	abem5.ncsa.uiuc.edu	n	/u/ncsa/arossi/SC...	arossi...	[]	n

An orange arrow points from the text 'When you select the line, a red box appears allowing you to cancel the job if desired.' to a red square icon in the 'Jobs List' table.

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



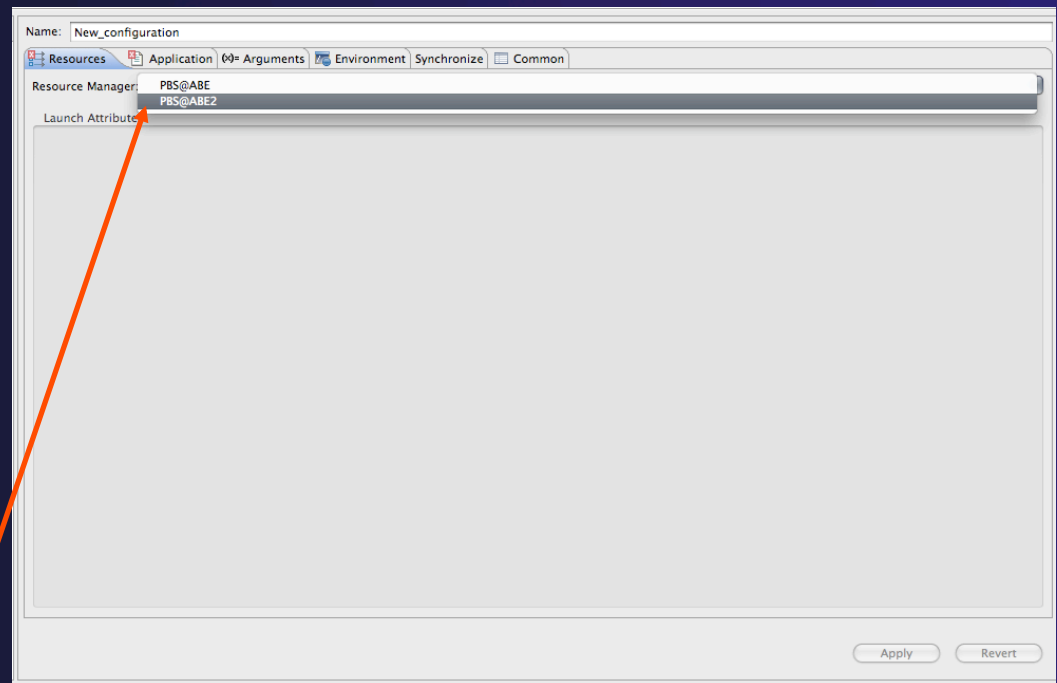
1. Create a new PBS Resource Manager (call it PBS@ABE2) as we did in slides 1-4.

✓ *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".

```
#!/bin/bash

#PBS -q @destination@
#PBS -A adm
#PBS -N @Job_Name@
#PBS -l nodes=@Resource_List.nodes@
#PBS -l walltime=@Resource_List.walltime@
#PBS -V @export_all@

env@
cd @directory@
@mpiCommand@ @mpiOptions@ @executablePath@ @progArgs@
```

2. Choose abe2_template from Current Templates.

3. Fill in values:

- Job_Name: Abe2TestSimpleMPI.
- nodes: 1:ppn=4.
- walltime: 00:20:00.

The screenshot shows the 'Edit/Copy Configuration' window for a job named 'Abe2Test'. The 'Current Template' is set to 'abe2_template'. The 'Optional Commands' section shows 'MPI Command' set to 'mpibexec'. The 'Attribute' table is as follows:

Attribute	Value	Description
destination	debug	Designation of the queue to which to submit the job.
Job_Name	Abe2TestSimpleMPI	The name assigned to the job by the qsub or qalter command.
Resource_List.nodes	1:ppn=4	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.walltime	00:20:00	Maximum amount of real time during which the job can be in the running state.
export_all	<input checked="" type="checkbox"/>	Declares that all environment variables in the qsub command's environment are to be exported to the batch job.

parallel tools platform

Attribute values in a Run Configuration are shared across templates.

The screenshot shows the PBS configuration tool interface. On the left, a 'Launch Attributes' panel shows the 'Current Template' as 'abe_template'. The main window displays a list of attributes with their values and descriptions. The 'Resource_List.walltime' attribute is highlighted with a red arrow pointing to the value '00:15:00'. A second screenshot on the right shows the 'Current Template' changed to 'base_template', with the 'Resource_List.walltime' attribute still showing '00:15:00'. Red arrows connect the text instructions to the corresponding UI elements.

Attribute	Value	Description
destination	debug	Designation of the queue to which to submit the job.
Account_Name		Account to which to charge this job.
Checkpoint	u	Determines when checkpointing (if supported) should occur.
Error_Path		The final path name for the file containing the job's error output.
Hold_Types		The set of holds currently applied to the job.
Job_Name	AbeTestSimpleMPI	The name assigned to the job by the qsub or qalter command.
Join_Path	<input type="checkbox"/>	Merge stdout and stderr into stdout.
Keep_Files		Retain these streams on the execution of the job.
Mail_Points	a	Identifies at which state changes the server will send mail about the job.
Mail_Users		The set of users to whom mail may be sent when the job makes certain state changes.
Output_Path		The final path name for the file containing the job's standard output.
Priority		The job scheduling priority assigned by the user.
Rerunnable	y	The rerunnable flag assigned by the user.
Resource_List.arch		Specifies the administrator defined system architecture required.
Resource_List.cput		Maximum amount of CPU time used by all processes in the job.
Resource_List.file		The largest size of any single file that may be created by the job.
Resource_List.host		Name of host on which job should be run.
Resource_List.mem		Maximum amount of memory used by all processes in the job.
Resource_List.ncpus	1	Total number of cpus/cores to be allocated to the job.
Resource_List.nice	0	The nice value under which the job is to be run.
Resource_List.nodes	1:ppn=8	Number and/or type of nodes to be reserved for exclusive use.
Resource_List.ompthreads	1	Number of threads per processor for Open MP jobs.
Resource_List.pcpout		Maximum amount of CPU time used by any single process.
Resource_List.pmem		Maximum amount of physical memory (working set) used by any process.
Resource_List.pvmem		Maximum amount of virtual memory used by any process.
Resource_List.vmem		Maximum amount of virtual memory used by all processes in the job.
Resource_List.walltime	00:15:00	Maximum amount of real time during which the job can run.
Shell_Path_List		A set of absolute paths of the program to process the job.
User_List		The list of user@hosts which determines the user name and host.
Variable_List		This is the list of environment variables to be passed to the job.
depend		The type of inter-job dependencies specified.
directive		Defines the prefix that declares a directive.
export_all	<input checked="" type="checkbox"/>	Declares that all environment variables in the qsub command are to be passed to the job.

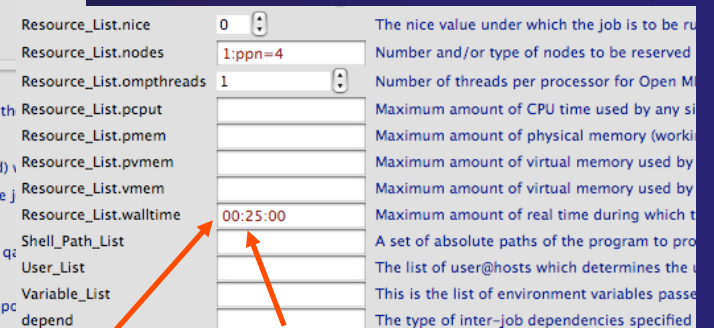
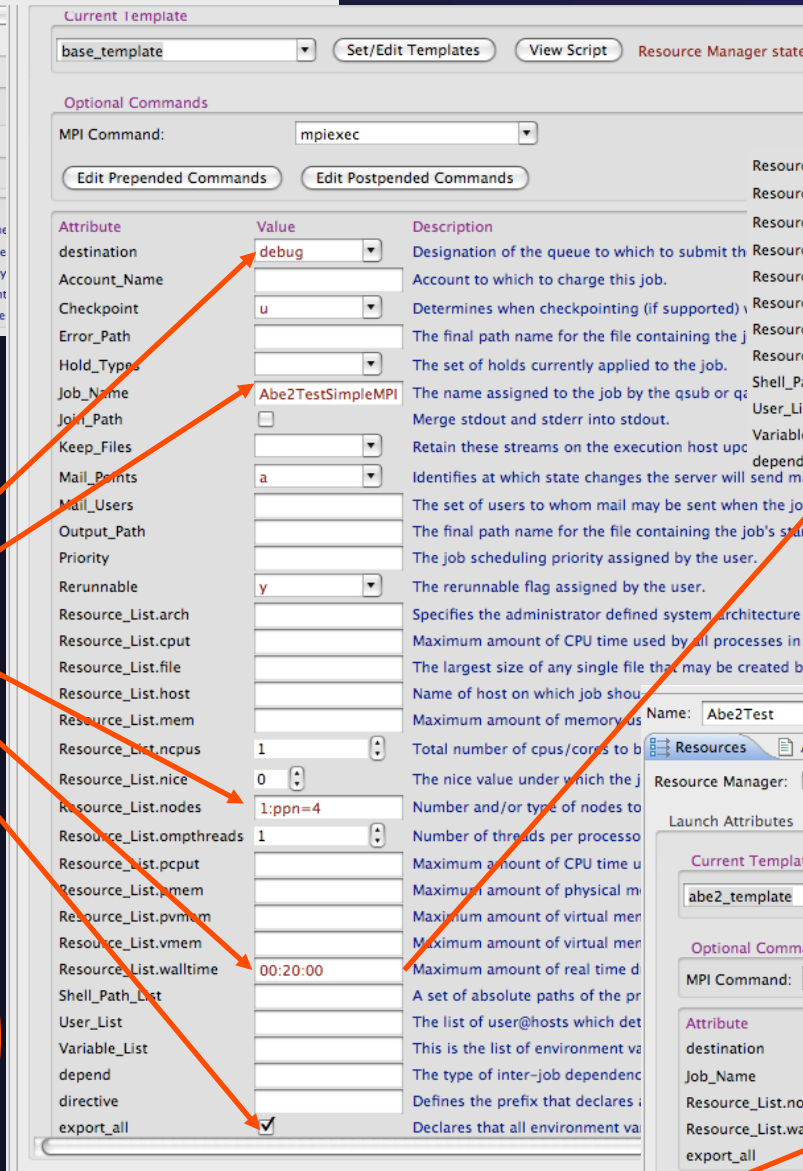
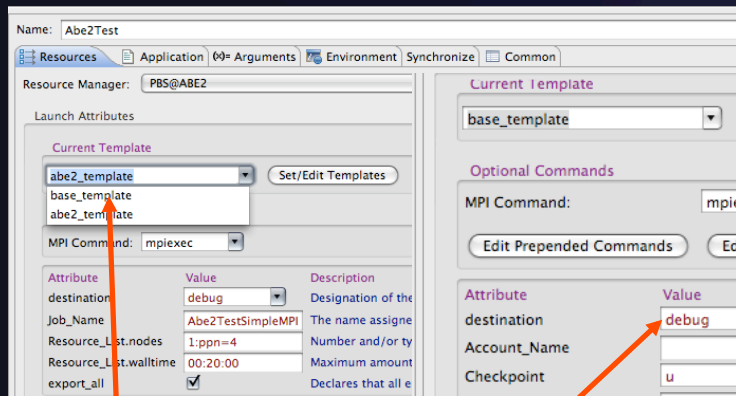
1. Choose AbeTest; then select base_template.

2. The values that were set in abe_template reappear.

3. Change walltime to "00:15:00"; click "Apply"; then select abe_template.

4. The new value reappears.

Attribute values remain bound to their Run Configuration.



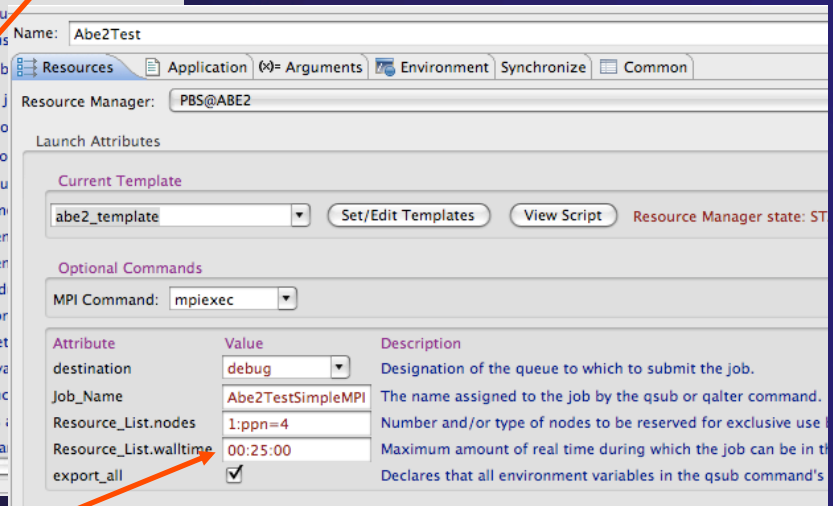
1. Choose Abe2Test; then select base_template.

2. The values that were set in abe2_template reappear.

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

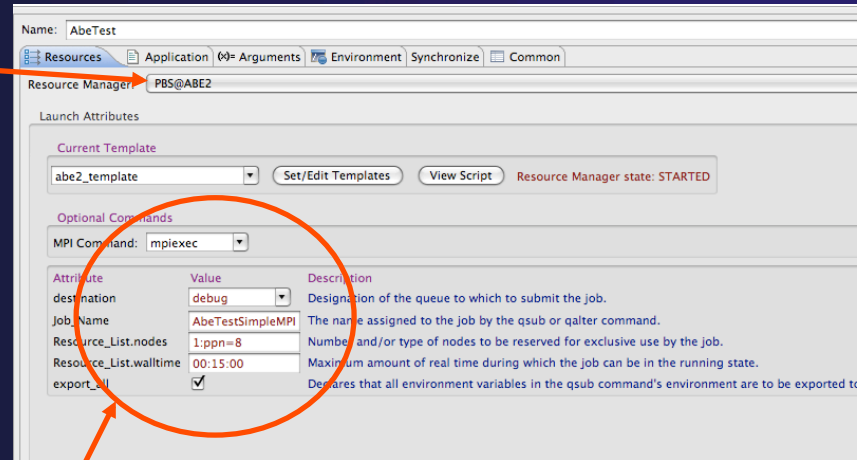
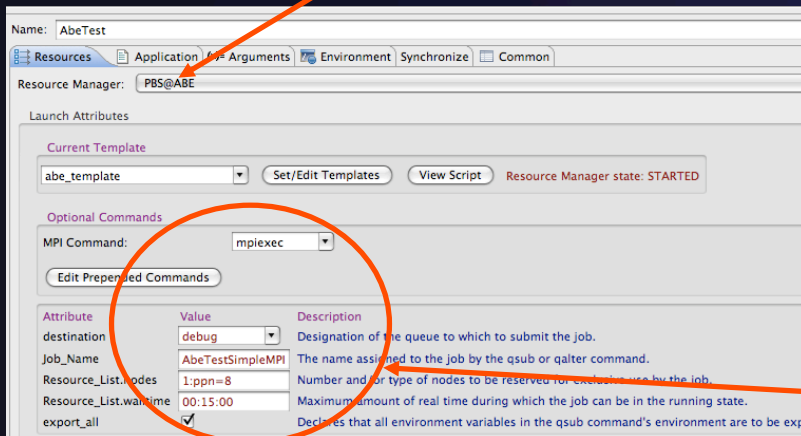
Inside Abe2Test, we can do the same thing without being affected by or affecting AbeTest values.

4. The new value reappears.

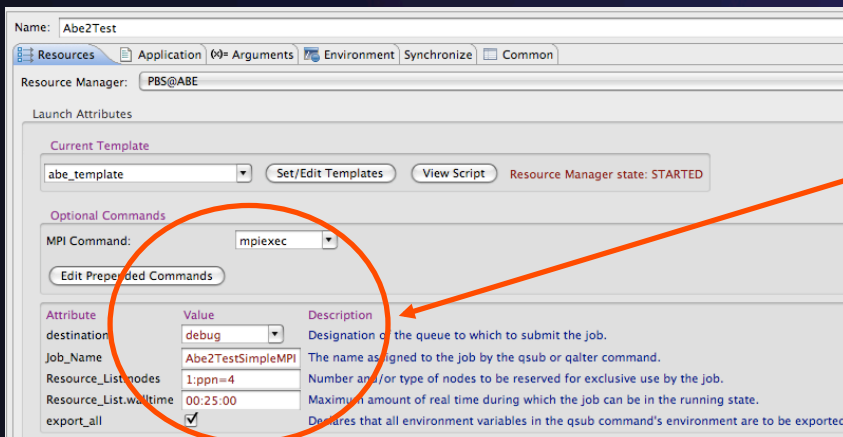


Swapping Resource Managers inside a Run Configuration

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.