

# Gridbus Portlets

-- USER GUIDE --

[www.gridbus.org/broker](http://www.gridbus.org/broker)

<b>GRIDBUS PORTLETS</b>	<b>1</b>
<b>1. GETTING STARTED</b>	<b>2</b>
1.1. PREREQUISITES:	2
1.2. INSTALLATION:	2
<b>2. AUTHENTICATION</b>	<b>3</b>
<b>3. WORKING WITH PROJECTS</b>	<b>4</b>
3.1. CREATING A NEW PROJECT	4
3.2. USING AN EXISTING PROJECT	4
<b>4. GRID CONFIGURATION</b>	<b>5</b>
4.1. EXECUTION RESOURCE SELECTION	5
4.1.1. CREATING A NEW EXECUTION RESOURCE	5
4.2. DATA RESOURCE SELECTION	5
4.2.1. CREATING A NEW DATA RESOURCE	6
<b>5. RUNNING A PROJECT</b>	<b>7</b>
5.1. QUALITY OF SERVICE	7
5.2. PROJECT EXAMPLES	8
5.2.1. EX 1 (API) - SIMPLEPROJECTRUNAPI	8
5.2.2. EX 2 (API + XPML) – SIMPLEPROJECTRUN	8
5.2.3. EX 3 (XPML) – SIMPLEPROJECTRUNXPML	9
5.3. MONITORING EXECUTION	10
5.4. RESETTING FAILED JOBS	10
5.5. STATISTICS	10
5.6. BROKER OUTPUT FILES (STDOUT AND STDERR)	11

# 1. Getting Started

This section of the user guide will help you to get the portlets installed and running.

## 1.1. Prerequisites:

Before deploying the Gridbus Portlets you should have the following:

- Successful Gridsphere installation (v2.1.5 tested)
- Successful Grid Portlets installation with login authentication via MyProxy
- SQL database server (scripts provided for and tested with MySQL)

The portlets have been developed under Java 1.4

## 1.2. Installation:

Start by unpacking the Gridbus Portlets distribution. In order to successfully deploy you will have to first setup the databases, one for the Gridbus Broker to store Grid job details and another for the Gridbus Portlets themselves. There are two scripts provided to create the database tables – broker.sql and gridbusportlets.sql

1. Create a database called 'broker' and run the broker.sql script
2. Create a database called 'gridbusportlets' and run the gridbusportlets.sql script

Database setup is complete. Test to make sure the tables have been created and the database can be accessed normally.

*Note: some data has been inserted into the gridbusportlets database for the user hussein. You can later create a Gridsphere user 'hussein' so that you can view the sample data, or you can replace 'hussein' with your username before running the script to make the data available for that user.*

Two configuration files in the *GridbusPortlets\webapp\WEB-INF\classes\* directory need to be modified with the correct database settings - Broker.hibernate.cfg.xml and DB.properties. Once these files have been modified the database configuration is complete.

The final step is to deploy the portlets using 'ant deploy', before deploying ensure the build.properties file is correctly configured.

## **2. Authentication**

New users will need to have a Grid Certificate available and uploaded to a MyProxy server (unless an alternative Gridsphere auth module is being used). Once the proxy is obtained during the login, it is held by the portal for later use by the default GridPortlets Credential Management Service. This achieves a single sign-on mechanism and the Grid Proxy will be used to authenticate to the various Grid resources during application execution..

## 3. Working with Projects

Projects are used to represent a single experiment setup. New projects can be created or existing projects can be returned to and resumed, restarted, repeated, and modified and so on. Once a Project is selected you will go on to configure the input files and parameters specific to the selected Project. Changes made to a Project can be saved against that Project at each step, so you can always log out of the Portal and come back to continue work at a later time by selecting the Project you are interested in.

### 3.1. Creating a new Project

Once logged in, you will be presented with a listing of your Projects. If you are conducting a new experiment you will need to create a new Project. The Project is created by specifying a name in the *Project Name* text field and then selecting *create*. When you create a new Project, it is automatically selected as the current working Project.

### 3.2. Using an existing Project

If you have a number of existing Projects you can click the *select* option beside the Project name to activate it as the current working Project.

The screenshot displays a web application interface with a navigation bar at the top containing tabs: Welcome, Projects, Ex 1 (API), Ex 2 (API + XPML), Ex 3 (XPML), Monitoring, and Grid Admin. Below the navigation bar is a 'Project Selection' section. On the left, there is a 'Create New Project' dialog box with a 'Project Name' input field and a 'create' button. Below this is a 'Current projects' table with the following data:

ID	NAME	CREATED (DATE)	ACTION
1	My Project	Thu Mar 22 23:06:23 EST 2007	[select]

On the right, there is a 'Current project details' panel for 'My Project'. It includes links for '[monitor]' and '[results]'. The status is 'Stopped' and it started on 'Sun Mar 25 20:08:55 EST 2007'. It also shows 'Jobs Complete : 100 %' and 'Budget Spent : N/A'. An 'update' button is located at the bottom of the details panel.

In the example the Project *My Project* has been created and selected.

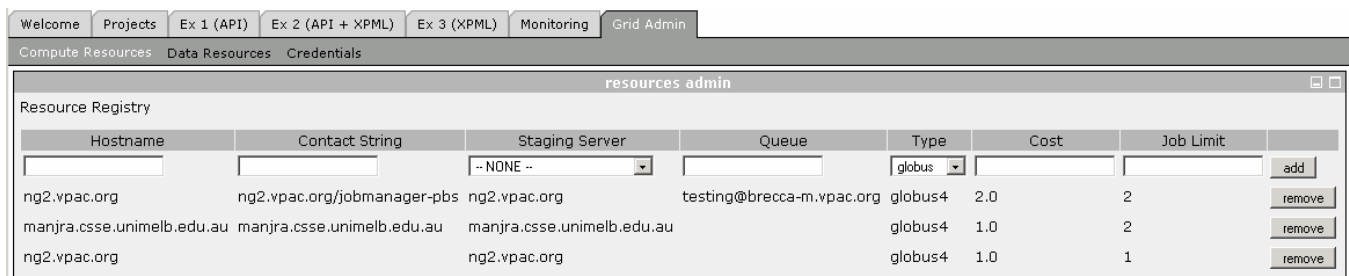
## 4. Grid Configuration

Before running the Project, the Grid environment must be configured. This determines which resources should be used during execution.

### 4.1. Execution Resource Selection

In order to run an experiment on the Grid, the set of Grid Resources needs to be configured.

#### 4.1.1. Creating a new Execution Resource



The screenshot shows the 'resources admin' portlet interface. At the top, there are navigation tabs: 'Welcome', 'Projects', 'Ex 1 (API)', 'Ex 2 (API + XPML)', 'Ex 3 (XPML)', 'Monitoring', and 'Grid Admin'. Below these are sub-tabs: 'Compute Resources', 'Data Resources', and 'Credentials'. The main content area is titled 'Resource Registry' and contains a table with the following columns: Hostname, Contact String, Staging Server, Queue, Type, Cost, Job Limit, and an 'add' button. The table lists three resources:

Hostname	Contact String	Staging Server	Queue	Type	Cost	Job Limit	
<input type="text"/>	<input type="text"/>	-- NONE --	<input type="text"/>	globus	<input type="text"/>	<input type="text"/>	add
ng2.vpac.org	ng2.vpac.org/jobmanager-pbs	ng2.vpac.org	testing@brecca-m.vpac.org	globus4	2.0	2	remove
manjra.csse.unimelb.edu.au	manjra.csse.unimelb.edu.au	manjra.csse.unimelb.edu.au		globus4	1.0	2	remove
ng2.vpac.org		ng2.vpac.org		globus4	1.0	1	remove

To create a new Resource, enter the details of the resource in the *Resources Admin* portlet. This option is designed for admin usage and general users will not need to add their own resources. The properties to specify are as follows:

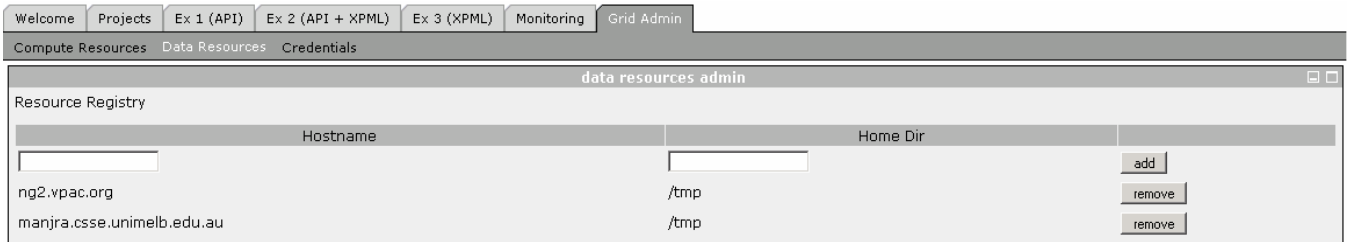
- **Hostname:** Hostname of the Grid Resource
- **Contact String:** Hostname of the Resource plus any additional information which may be required when accessing the resource. In case of Globus, the jobmanager should be specified here (eg. jobmanager-pbs if PBS queuing system is being used).
- **Staging Server:** Resource used as an intermediary for transferring data between web server and execution node.
- **Queue:** Name of the queue to use
- **Type:** Middleware type
- **Cost:** G\$ cost to use the Resource
- **Job Limit:** The maximum number of jobs to submit to the Resource at any one time

### 4.2. Data Resource Selection

In the Gridbus Portlets the Data Resources have two purposes, the first is for the destination of output files and the second is as staging servers for execution resources. A staging server is necessary when certain files (such as a job script or the stdout and stderr for a job) need to be transferred between the webserver and the compute element. The staging server is a place where both the webserver and

compute element can put and get files from. In some cases this maybe the same hostname as the compute resource, but in other cases it may be a secondary storage resource (eg. ngdata).

### 4.2.1. Creating a new Data Resource



To specify a new Data Resource simply enter its hostname and the path to the directory to be used. Note that the user's proxy will need to have appropriate permissions to access that directory during execution.

## 5. Running a Project

Once the Grid environment has been configured we can now execute Jobs over the Grid. The Gridbus Portlets are designed to be generic and used as the basis for other Grid portals. The Project Management and Grid Resource components are not specific to any application. The following section describes 3 different execution modules (portlets) which highlight the ability to make use of these portlets and the Gridbus Broker in various ways.

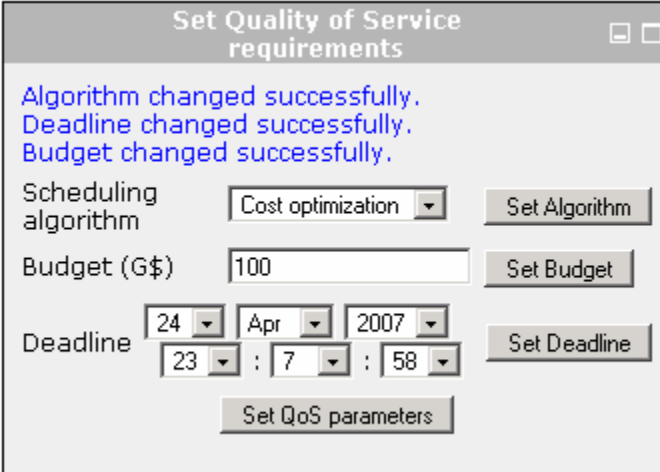
In each of the 3 examples, the Gridbus Broker (<http://www.gridbus.org/broker>) is used to manage the Grid application scheduling, execution, monitoring and data staging.

### 5.1. Quality of Service

Before running a Project it is required that the Quality of Service values are set. Enter the desired values in the Quality of Service dialog and click the *Set QoS parameters* button. Each QoS parameter can also be set independently by clicking the *Set button* for that parameter.

Different Resources within the Grid environment have different costs and different performance values. When executing in this environment the Gridbus Broker will try to schedule the workload optimally over the available Resources. Scheduling optimization can be controlled by specifying values for the Deadline, Budget and Algorithm to be used.

If you require the experiment to complete by the time specified by your *Deadline*, without consideration of budget, then select *Time optimization*, otherwise, if you prefer to execute at the cheapest cost, based on your *Budget*, select *Cost optimization*.



Set Quality of Service requirements

Algorithm changed successfully.  
Deadline changed successfully.  
Budget changed successfully.

Scheduling algorithm: Cost optimization [Set Algorithm]

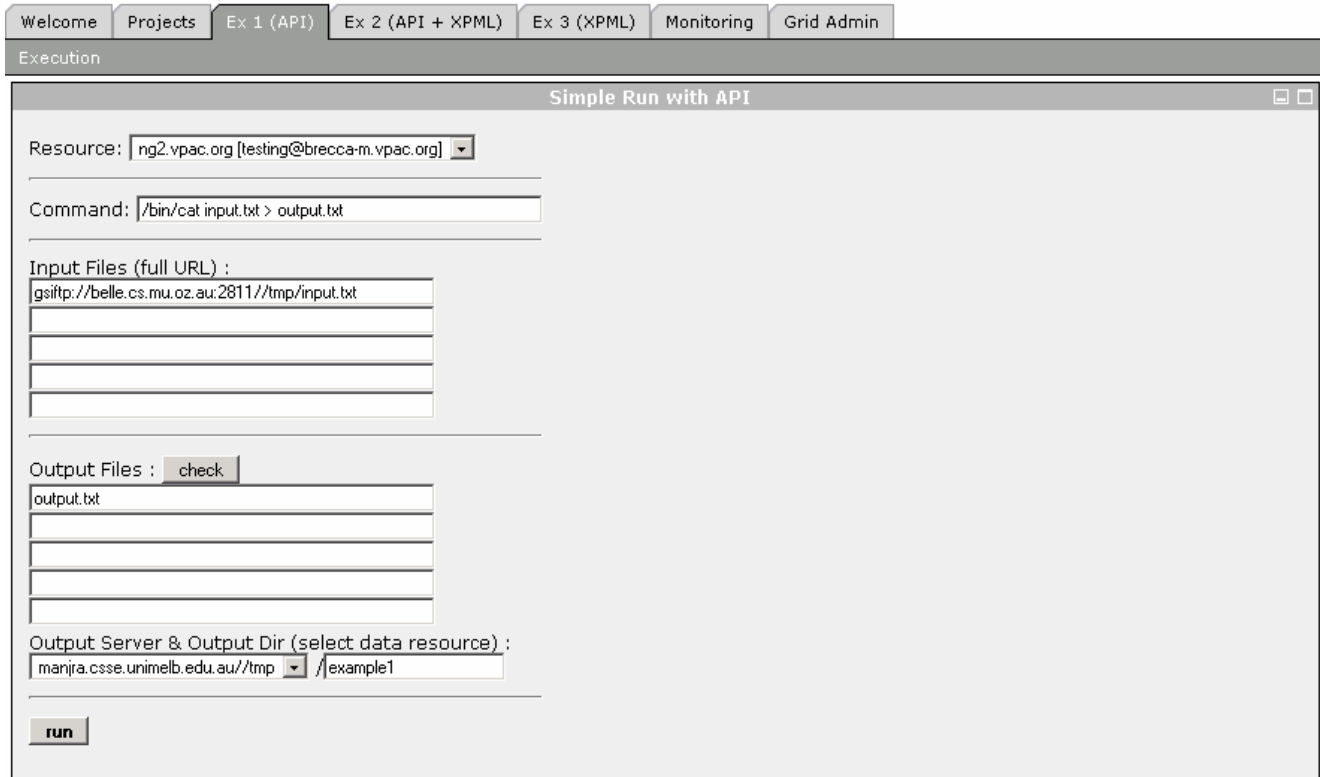
Budget (G\$): 100 [Set Budget]

Deadline: 24 Apr 2007 23:07:58 [Set Deadline]

[Set QoS parameters]

## 5.2. Project Examples

### 5.2.1. Ex 1 (API) - SimpleProjectRunAPI



The screenshot shows a web application interface titled "Simple Run with API". At the top, there is a navigation bar with tabs: "Welcome", "Projects", "Ex 1 (API)", "Ex 2 (API + XPML)", "Ex 3 (XPML)", "Monitoring", and "Grid Admin". Below the navigation bar, the main content area is titled "Execution". The interface contains several form fields and a "run" button:

- Resource:** A dropdown menu showing "ng2.vpac.org [testing@brecca-m.vpac.org]".
- Command:** A text input field containing "/bin/cat input.txt > output.txt".
- Input Files (full URL):** A list of input files, with the first one being "gsiftp://belle.cs.mu.oz.au:2811/tmp/input.txt".
- Output Files:** A section with a "check" button and a text input field containing "output.txt".
- Output Server & Output Dir (select data resource):** A dropdown menu showing "manjra.csse.unimelb.edu.au/tmp" and a text input field containing "/example1".
- run:** A button at the bottom left of the form.

This example allows the user to run a command over the Grid. Here the command will be run on ng2.vpac.org using the queue [testing@brecca-m.vpac.org](mailto:testing@brecca-m.vpac.org). An input file 'input.txt' will be downloaded from a GridFTP server on belle.cs.mu.oz.au and its contents copied into a file called output.txt and stored on the GridFTP server located at manjra.csse.unimelb.edu.au. Once the job has executed successfully and the file is available on manjra, a link will appear beside to output filename enabling it to be downloaded from the GridFTP server directly from within the portlet. If the 'Any' option is selected for the execution resource the Broker will manage resource selection on the users behalf.

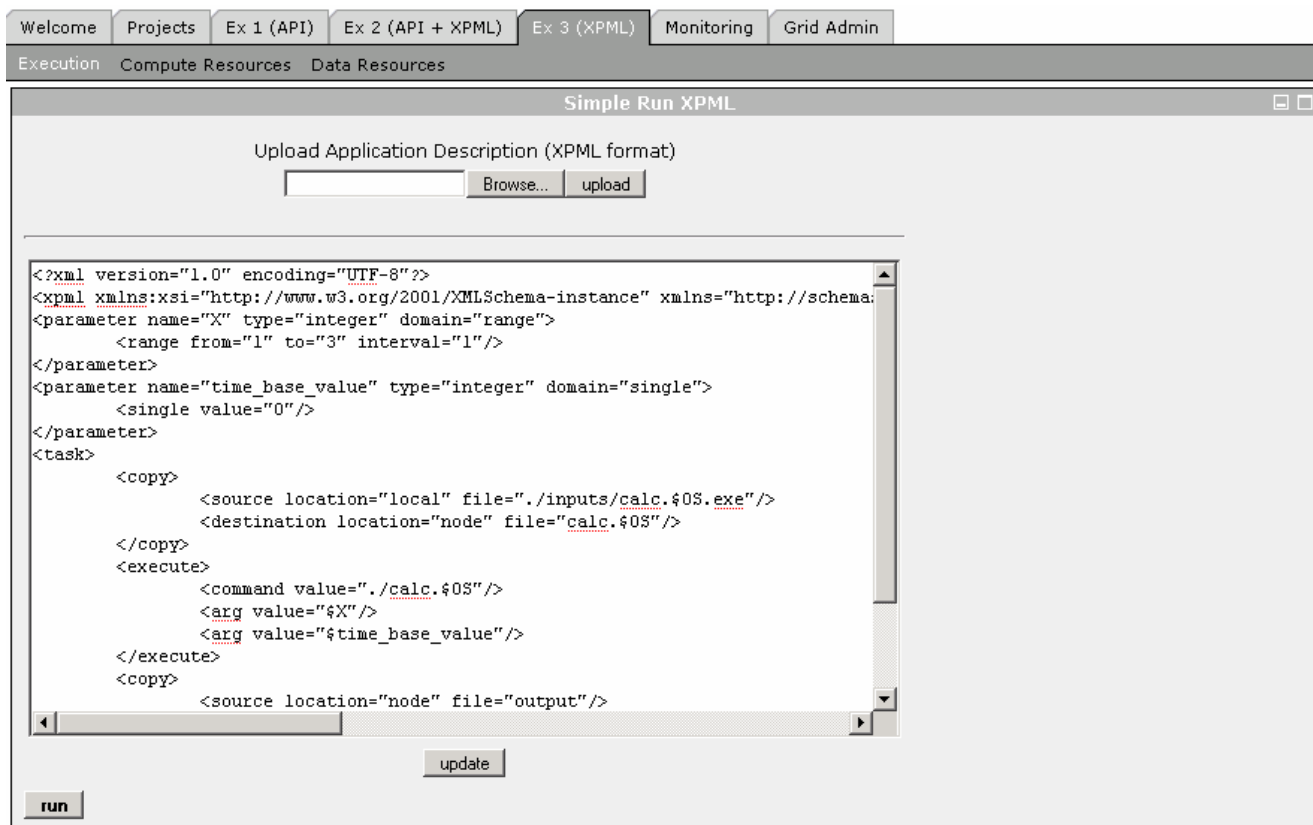
In the implementation of this portlet the Gridbus Broker API is used directly to create Grid Jobs objects which are executed and managed on the Grid via the Broker.

### 5.2.2. Ex 2 (API + XPML) – SimpleProjectRun

This example is the same as Ex 1 (API), however the implementation is different. It uses classes of the Broker to generate an XML representation of the Grid application - XPML (see Gridbus Broker manual <http://www.gridbus.org/broker/3.0/manual.v3.0.pdf> for details on XPML).



### 5.2.3. Ex 3 (XPML) – SimpleProjectRunXPML



This example allows the user to upload or edit an XPML description of a Grid application to be run over the Grid. In this case an executable called 'calc' will be transferred from the webserver to the various Grid execution resources and executed using varying parameters (parameter sweep). The outputs will be returned to the webserver. GUI Tools exist for the Gridbus Broker to assist in creating XPML input files in a user-friendly manner (see Gridbus Broker Workbench).

Before running the Project the user should access the 'Compute Resources' sub-menu to select the resources that should be used during this execution. The user can select one or many resources.

### 5.3. Monitoring Execution

Once the project is running, the status of executing jobs can be monitored in the *Execution status monitoring* dialog. Jobs will transition between states as they are scheduled, submitted, execute and so on. Clicking on the name of a Job in the *Execution status monitoring* dialog, will reveal its details such as execution time, resource it was submitted to, and its handle. Resource information can also be viewed by clicking on the resource hostname.

Once Jobs have completed they will enter the *Done* state indicated by a green box.

The screenshot shows the 'Execution status monitoring' window with the following data:

**Broker Status:** Started  
**Start Time:** Mar 25, 2007 11:56:10 PM  
**Time taken:** 00:03:29.250  
**Time remaining:** 719:08:17.993  
**Budget spent (G\$):** 0.0 of 100.0 (0.0%)  
**Jobs completed:** 0 of 20 (0.0%)

**Job Status Legend:**

W	Ready/Waiting	16
I	Stage In	2
O	Stage Out	0
S	Scheduled	0
R	Running	2
D	Done	0
F	Failed	0
?	Unknown	0
All	All	20

**Job details for Job ID j1:**

Status:

Submitted to Server: ng2.vpac.org  
Submitted date/time: Mar 25, 2007 11:57:12 PM.  
Completed date/time:  
Time taken:  
Job handle: https://ng2.vpac.org:8443/wsrf/servic

**Resource details for manjra.csse.unimelb.edu.au:**

System

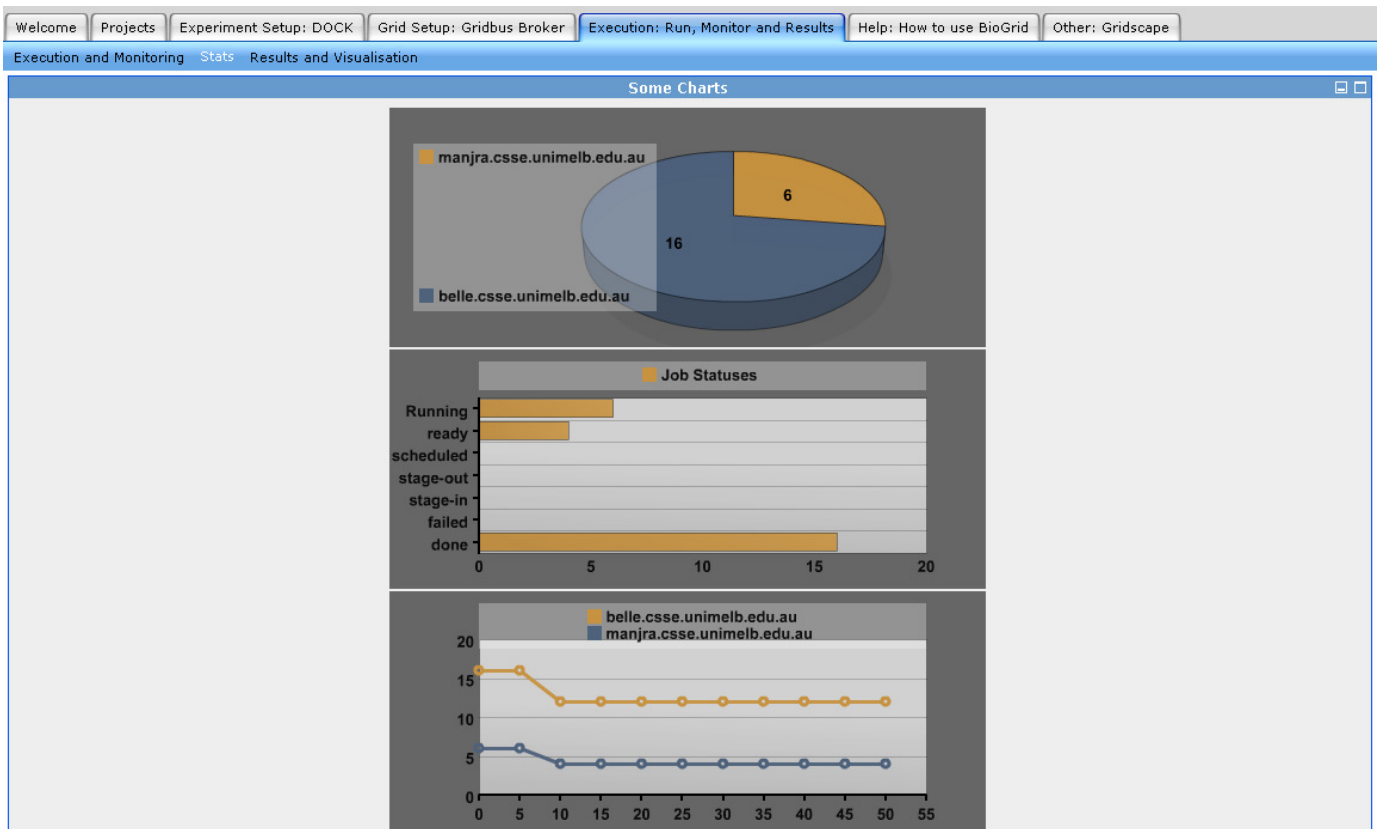
Type	806
Architecture	x86
Memory	0
CPUs	1

### 5.4. Resetting failed Jobs

If any of the Jobs fail during execution, they will appear with a Red failed box. By clicking the Job's name and displaying the Job details, the Reset Job button can be accessed allowing the Job to be rescheduled onto the Grid environment.

### 5.5. Statistics

A set of performance graphs are available to give a simple visual representation of the execution progress at a glance.



## 5.6. Broker Output Files (*stdout* and *stderr*)

During execution the Gridbus Broker maintains a directory for each job in a project. In each directory a number of files are stored for the job, including job *stdout* and *stderr*. For a Globus Job the RSL will be output. Any output files specified to be stored locally in the XPML will also appear in the Job's directory. A portlet has been developed to make it easy for the user to browse these directories and download files. This can also be useful in determining why certain jobs have failed by checking the *stdout* and *stderr*.

