



**USER MANUAL**  
**APPLICATION PORTAL**

**WP3.1**

---

Document Filename:	<b>CG3.1-v1.0-ALGO-ApplicationPortalUserManual.doc</b>
Work package:	<b>WP3.1</b>
Partner(s):	<b>PSNC, DATAMAT, ALGO</b>
Lead Partner:	<b>PSNC</b>
Config ID:	<b>CG3.1-v1.0-ALGO-ApplicationPortalUserManual</b>
Document classification:	<b>PUBLIC</b>

---

## Document Log

Version	Date	Summary of changes	Author
1-0	5/01/2005	Draft version	Miltos Kokkosoulis

---

## CONTENTS

<b>COPYRIGHT NOTICE .....</b>	<b>4</b>
<b>1. INTRODUCTION.....</b>	<b>5</b>
<b>2. PRODUCT USAGE .....</b>	<b>6</b>
2.1. GENERIC PORTLETS .....	6
2.1.1. <i>Proxy Manager</i> .....	6
2.1.2. <i>Job Submission</i> .....	7
2.1.3. <i>Job Get Output</i> .....	9
2.1.4. <i>Job List Match</i> .....	10
2.1.5. <i>Job Log Info</i> .....	11
2.2. PORTLETS FOR THE CROSSGRID APPLICATIONS.....	11
2.2.1. <i>Flood Application portlet</i> .....	11
2.2.2. <i>WAM Application portlet</i> .....	12
2.2.3. <i>Neural Network Graphics and Neural Network Application portlets</i> .....	13
2.2.4. <i>Air Pollution application portlet</i> .....	15
<b>3. CONTACT INFORMATION AND CREDITS.....</b>	<b>16</b>
<b>4. THE CROSSGRID LICENSE AGREEMENT .....</b>	<b>17</b>

## **COPYRIGHT NOTICE**

Copyright (c) 2005 by *CrossGrid Consortium*. All rights reserved.

Use of this product is subject to the terms and licenses stated in the EDG license agreement. Please refer to Chapter 4 for details.

*JAVA* is a registered trademark of *SUN Microsystems Inc.*. All rights reserved.

This research is partly funded by the European Commission IST-2001-32243 Project “CrossGrid”.

## 1. INTRODUCTION

Portals are web sites that provide centralized access to sets of resources. All portals provide personalization, security/authentication/authorization mechanisms and navigation choices. They constitute gateways for accessing distributed resources and information in a consistent and intuitive manner.

In that sense, the CrossGrid Application Portal is developed in order to provide users that have no knowledge of the underlying Grid technology (e.g. physicists, chemists, etc.) with access to the CrossGrid testbed and the Grid services that it offers.

## 2. PRODUCT USAGE

The Application Portal uses the RAS (Roaming Access Server) and JSS (Job Submission Services) web services.

There is a number of portlets that comprise the CrossGrid Portal Toolkit. A portlet is a region on a page that contains, manipulates and displays data almost independently, as if it was a web page on its own. Of course, there may be some interconnections and dependencies among portlets that exist on the same web portal. The user has the ability to minimize, maximize, restore or close any of the portlets that he has access to do so.

### 2.1. GENERIC PORTLETS

#### 2.1.1. Proxy Manager

This is the portlet that plays the most important role in the whole structure of the portal. It provides the authentication mechanism for the user to get access to the rest of the portlets. When the user is authenticated once at the beginning, he becomes authorized to use the facilities of the whole portal. With the use of this portlet, the user is authenticated once and he is then authorized to have access to the portlets of the CrossGrid Application Portal. The authentication and authorization takes place as soon as the user has valid credentials delegated in a MyProxy server in the testbed. The valid proxy certificate will be retrieved by the Proxy Manager portlet, and the user will automatically gain access to all portlets.

xportlets : ProxyManager

The following GSI proxy credentials are loaded into your account:

(default proxy) /C=GR/O=HellasGrid/OU=rtd.algo.com.gr/CN=Miltos Kokkosoulis

Click the button below to add another GSI proxy credential to your account:

**Figure 1.** A snapshot of the Proxy Manager portlet, showing the credentials that have been loaded for the specific user.

xportlets : ProxyManager

Fill out the following parameters and click 'Get Proxy' to retrieve your proxy from the MyProxy server specified below:

Hostname:

Port:

Username:

Password:

Lifetime:  (hours)

Store for duration of user session only?

Note: the lifetime of your GSI proxy credential will depend on the maximum lifetime specified for delegated credentials when you stored your long-term GSI proxy credential in the MyProxy server (by default 2 hours).

**Figure 2.** The figure above shows the screen that is presented to the user who wants to load a valid proxy certificate for himself from a MyProxy server into the Proxy Manager portlet.

xportlets : ProxyManager

**Subject:** C=GR,O=HellasGrid,OU=rtd.algo.com.gr,CN=Miltos Kokkosoulis,CN=proxy,CN=proxy,CN=proxy

**Issuer:** C=GR,O=HellasGrid,OU=rtd.algo.com.gr,CN=Miltos Kokkosoulis,CN=proxy,CN=proxy

**Bits:** 512

**Time Left:** 1 hour(s) 59 minute(s) 48 second(s)

**Limited:** no

**Figure 3.** A user can see the details of his/her certificate by clicking the “View” button.

### 2.1.2. Job Submission

The main feature of this portlet is that the potential user is able to submit any job to the CrossGrid testbed by following the available instructions.

At the top of the portlet, the user has the ability to give a name to the job (it's not obligatory), and after that the name of the executable or the full path to it together with any numerical parameters,

if they exist, must be written in a text box. The full path must be included if the directory in which the application executable resides is not included in the user's path.






If the job is an MPI job, then MPICH must be written in the text box under the executable. The number of processes follows, which should also be added in that case. If the job is not an MPI job, these two text boxes should be left blank.

The user can give a name in the StdOutput and StdError text fields for the standard output and error files that will be produced.

In case that the application executable is not installed in the testbed, the user has to write the location of the executable. This location is the full URL (starting with gsiftp://). Then, it will be transferred to the Computing Element via globus-url-copy.

The user can take advantage of the dynamic row creation that can happen for both the input and output files. These will too be transferred to the CE via globus-url-copy. In the case of the input files, the user can add any files that are needed for the specific application to run by pressing the "add" button. While in the case of the output files, the user can specify just the names of the files that are going to be produced after the execution of the application, if there are such files created at the end. In both cases, the user can remove the rows that he added dynamically by selecting any of them and pressing the "remove" button.

**Job Submission Portlet**

**Builds the JDL and submits the job to the CrossGrid testbed**

Job name

Executable  **The name or path to the executable**

Job type  **MPICH, if its an MPI job**

Number of processes

StdOutput

StdError

Executable location  **If the application executable is not installed in the testbed (in all CEs) you have to specify its URL and it will be transferred to the CE via globus-url-copy.**

**Below, you have an opportunity to specify the URL of each input file needed by the application - if there are any -. They are going to be transferred to the CE via globus-url-copy.**  
**Specify also the filenames (Just names, not URLs) of the output files that your application produces - if there are any -.**

Input files

Output files

Virtual Organisation

For those applications that will only run at a unique Computing Element, there is the ability to specify that CE in the appropriate text field (named ce\_Id). Then, there are two drop-down boxes from which one can choose the RB and LB hosts (all the available hosts in the testbed are included). The user can also insert their respective ports, in case some changes are made and the default ones are not used.

### 2.1.3. Job Get Output

The Job Get Output portlet provides now the feature of showing the status of the submitted job inside its area. And whenever there is a “Done (finished)” flag for the job, the output files are presented in the form of HTML links. The user can click on these links and another browser window will open for each of the links, with the contents of the respective output file.

#### Job GetOutput Portlet

Current files related to <https://rb01.lip.pt:9000/OHP5UUwNOnQaFXRp1xxjqg>:

[StdOutput](#)

[StdError](#)

Back

**Figure 5.** A snapshot of the “job get output” portlet, showing an id of a job that has been executed successfully

If the flag is one of the following: Done (Failed), Running, Aborted, Scheduled, Waiting, then the portlet screen is similar to the one below.

#### Job GetOutput Portlet

Status for job [https://rb01.lip.pt:9000/UrzftO\\_Y9Dmb9ziZ30GtQ](https://rb01.lip.pt:9000/UrzftO_Y9Dmb9ziZ30GtQ): Done(Failed)

Current Jobs

Select job id:

Submit

**Figure 6.** A snapshot of the “job get output” portlet, showing an id of a job that failed to be executed successfully

If the flag is Waiting or Scheduled or Running, the user is then able to try many times to retrieve the output files, until this is successfully done. Eventually the status of the submitted job will reach the Done (Finished) flag, and the links to the output files will appear.

Sometimes the output files that were specified by the user may not all of them be produced at the CE at the end of the application’s execution. In that case, the HTML links of the output files that

were produced will appear inside the portlet, while on the other hand there will be a warning about the output files that were not produced.

#### 2.1.4. Job List Match

A user can find out through this portlet the available Computing Elements in the testbed that can execute the particular job that he/she has specified. In other words, this portlet is useful when someone wants to determine whether there are any CEs available (how many, where they are physically located) that can meet the requirements of a job.

**Job List Match Portlet**

Job List Match Service	
Executable	<input type="text"/>
Arguments	<input type="text"/>
StdOutput	<input type="text"/>
StdError	<input type="text"/>
InputSandbox	<input type="text"/>
OutputSandbox	<input type="text"/>
Requirements	<input type="text"/>
Resource Broker	rb01.lip.pt ▾
Port	7772
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

Resource Broker:	rb01.lip.pt ▾
Port:	7772
JDL file:	<input type="text"/> <input type="button" value="Browse..."/>
<input type="button" value="Submit"/>	

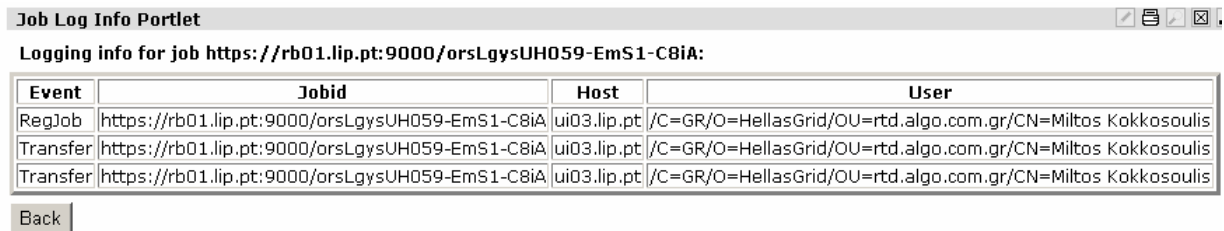
**Figure 7.** A snapshot of the “Job List Match” portlet

### 2.1.5. Job Log Info

This portlet can be used in order to obtain information about previously submitted jobs and their status. Also the user can find out what exactly was the course of a particular job from the start of the submission until the exit outcome.

The route that the submitted job followed from the submission until its termination is shown analytically in the table that the portlet creates and displays on the screen. There are 4 columns in that table:

- Event (the name of the event that took place)
- Job Id (the specific “stamp” that the job has)
- Host (the actual computer machine on which the event took place)
- User (the certificate of the user under which the particular action occurred)



The screenshot shows a web browser window titled "Job Log Info Portlet". Below the title bar, there is a text label: "Logging info for job https://rb01.lip.pt:9000/orsLgysUH059-EmS1-C8iA:". Below this is a table with four columns: "Event", "Jobid", "Host", and "User". The table contains three rows of data. The first row is "RegJob", the second is "Transfer", and the third is "Transfer". All rows have the same Jobid, Host, and User values. Below the table is a "Back" button.

Event	Jobid	Host	User
RegJob	https://rb01.lip.pt:9000/orsLgysUH059-EmS1-C8iA	ui03.lip.pt	/C=GR/O=HellasGrid/OU=rtd.algo.com.gr/CN=Miltos Kokkosoulis
Transfer	https://rb01.lip.pt:9000/orsLgysUH059-EmS1-C8iA	ui03.lip.pt	/C=GR/O=HellasGrid/OU=rtd.algo.com.gr/CN=Miltos Kokkosoulis
Transfer	https://rb01.lip.pt:9000/orsLgysUH059-EmS1-C8iA	ui03.lip.pt	/C=GR/O=HellasGrid/OU=rtd.algo.com.gr/CN=Miltos Kokkosoulis

Figure 8. A snapshot of the “Job Log Info” portlet

## 2.2. PORTLETS FOR THE CROSSGRID APPLICATIONS

### 2.2.1. Flood Application portlet

This portlet is designed for the job submission and simulation for the CrossGrid Flood application. So, the user can put a name for the job he is submitting and then provide all the other parameters that are necessary for the particular job to be executed:

Start date

Hydrograph file

Simulation duration in days

Number of CPUs

Storage element

Scripts directory

Output directory

Mesh file

Other parameters

Then, as soon as the user submits the job onto the Grid, the execution of the job takes place. And when it finishes executing, it returns a text file, with some numerical values, as an output. After that, the user

has the ability to submit a visualization job with the use of the numerical values that were returned by the previously submitted job.

The screenshot shows the 'Flood Application Portlet' interface. It features a table of parameters with text input fields. Below the table is a 'Submit job' button. At the bottom, there are two rows of controls: the first row has a 'Submitted jobs' dropdown menu, a 'Get results' button, and a 'Submit visualisation job' button; the second row has a 'Submitted visualization jobs' dropdown menu and a 'Get visualization results' button.

Parameter	Value
Job name	mkok1
Start date	last
Hydrograph file	/data/flood-vo/hydraulics/DaveF/input
Simulation duration in days	1
Number of CPUs	4
Storage element	flood-vo.ui.sav.sk
Scripts directory	/data/flood-vo/hydraulics/DaveF/scrip
Output directory	/data/flood-vo/hydraulics/DaveF/outpi
Mesh file	/data/flood-vo/hydraulics/DaveF/input
Other parameters	

Submit job

Submitted jobs

Submitted visualization jobs

**Figure 9.** A snapshot of the Flood application portlet with all the parameter text fields.

### 2.2.2. WAM Application portlet

This is the sea wave model application portlet. A user submits a job with the use of all the parameters that follow and the wind files that are located in a machine on the testbed. Of course, the user can choose which particular area he/she wants to study and proceed with the simulation.

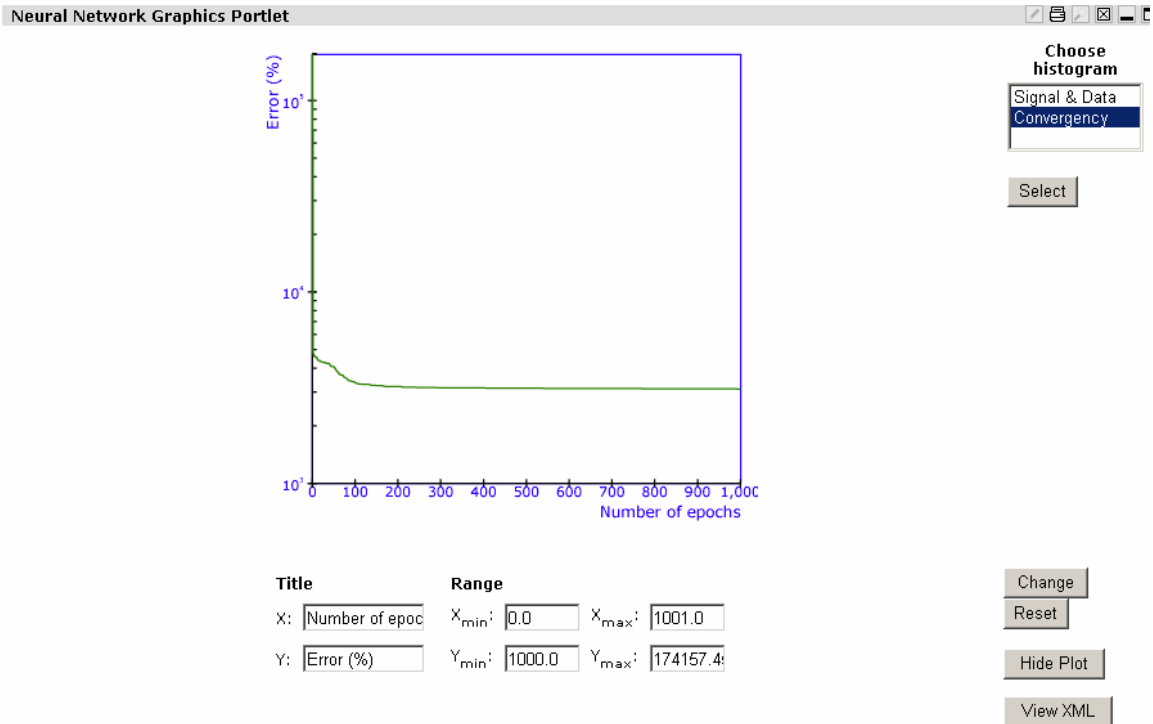
**WAM Application Portlet**

Start	<input type="text" value="2002030100"/>
Length	<input type="text" value="3"/>
Area	<input type="text" value="baltic_sea"/>
Wind files	<input type="text" value="gsiftp://xgridse.icm.edu.pl/flatfiles/SE00/flatfiles/meteo/WV"/>
Restart files	<input type="text"/>
Output frequency	<input type="text" value="1"/>
	<input checked="" type="checkbox"/> significant_wave_height
	<input type="checkbox"/> mean_wave_direction
	<input type="checkbox"/> wave_mean_frequency
	<input type="checkbox"/> friction_velocity
	<input type="checkbox"/> wind_direction
	<input type="checkbox"/> wave_peak_frequency
	<input type="checkbox"/> drag_coefficient
	<input type="checkbox"/> normalized_wave_stress
	<input type="checkbox"/> swell_wave_height
	<input type="checkbox"/> swell_direction
	<input type="checkbox"/> wind_wave_direction
	<input type="checkbox"/> swell_mean_frequency

**Figure 10.** A snapshot of the WAM portlet. All parameters' check boxes are shown.

### 2.2.3. Neural Network Graphics and Neural Network Application portlets

The following is an example of the neural network graphics portlet.



The user can choose the type of histogram he/she wants, and it will be displayed inside the portlet area. The histogram can be modified by the user and also the XML code can be displayed, as well.

**Neural Network Application Portlet**

Input layer:

Output layer:

**Training**

Method:

Num epochs:

The basic portlet for the neural network application is the figure shown above. The user picks the numbers for the input and output layers, as well as the values for the other parameters, and clicks the “submit” button.

#### 2.2.4. Air Pollution application portlet

##### Air Pollution Application Portlet

<b>Date</b>	<input type="text"/>
<b>Start hour</b>	<input type="text"/>
<b>Interval of hours</b>	<input type="text"/>
<input type="button" value="Submit job"/>	

These are the three parameters that are needed for the submission of a job in the air pollution modelling application. As soon as the user inputs these parameters and submits the job, the job is transferred to any particular Computing Element that has the air pollution modelling application installed. After it has been executed, the resulting Scalable Vector Graphics (SVG) file is created and it is transferred to the portlet area, where it is displayed to the user.

### 3. CONTACT INFORMATION AND CREDITS

## 4. THE CROSSGRID LICENSE AGREEMENT

Copyright (c) 2005 CrossGrid. All rights reserved.

This software includes voluntary contributions made to the CrossGrid Project. For more information on CrossGrid, please see <http://www.eu-crossgrid.org>.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted. Any exercise of rights under this license by you or your sub-licensees is subject to the following conditions:

1. Redistributions of this software, with or without modification, must reproduce the above copyright notice and the above license statement as well as this list of conditions, in the software, the user documentation and any other materials provided with the software.

2. The user documentation, if any, included with a redistribution, must include the following notice: “This product includes software developed by the CrossGrid Project (<http://www.eu-crossgrid.org>).”

Alternatively, if that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the software itself.

3. The names “CrossGrid” and “CG” may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by [cgooffice@cyfronet.krakow.pl](mailto:cgooffice@cyfronet.krakow.pl).

4. You are under no obligation to provide anyone with any bug fixes, patches, upgrades or other modifications, enhancements or derivatives of the features, functionality or performance of this software that you may develop. However, if you publish or distribute your modifications, enhancements or derivative works without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted participants in the CrossGrid Project a worldwide, non-exclusive, royalty-free, perpetual license to install, use, reproduce, display, modify, redistribute and sub-license your modifications, enhancements or derivative works, whether in binary or source code form, under the license conditions stated in this list of conditions.

## 5. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE CROSSGRID PROJECT AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE CROSSGRID PROJECT AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE, MODIFICATIONS, ENHANCEMENTS OR DERIVATIVE WORKS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

## 6. LIMITATION OF LIABILITY

THE CROSSGRID PROJECT AND CONTRIBUTORS SHALL HAVE NO LIABILITY TO LICENSEE OR OTHER PERSONS FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.