



The NGS Grid Portal

David Meredith
NGS + Grid Technology Group, e-Science Centre,
Daresbury Laboratory, UK

d.j.meredith@dl.ac.uk

NGS Portal

The NGS portal can be used to access and interact with the HPC and Data resources available on the Grid via SSO (Certificates + myproxy):

- **Browse for different applications** available on a Grid this includes your own personal applications and pre-configured applications available on a particular Grid (e.g. the NGS is currently publishing applications within the NGS portal to be made easily available for its users).
- **Submit/monitor** compute jobs/applications.
- **Access and move data** around the Compute and Data Grid (Gridftp, srb).

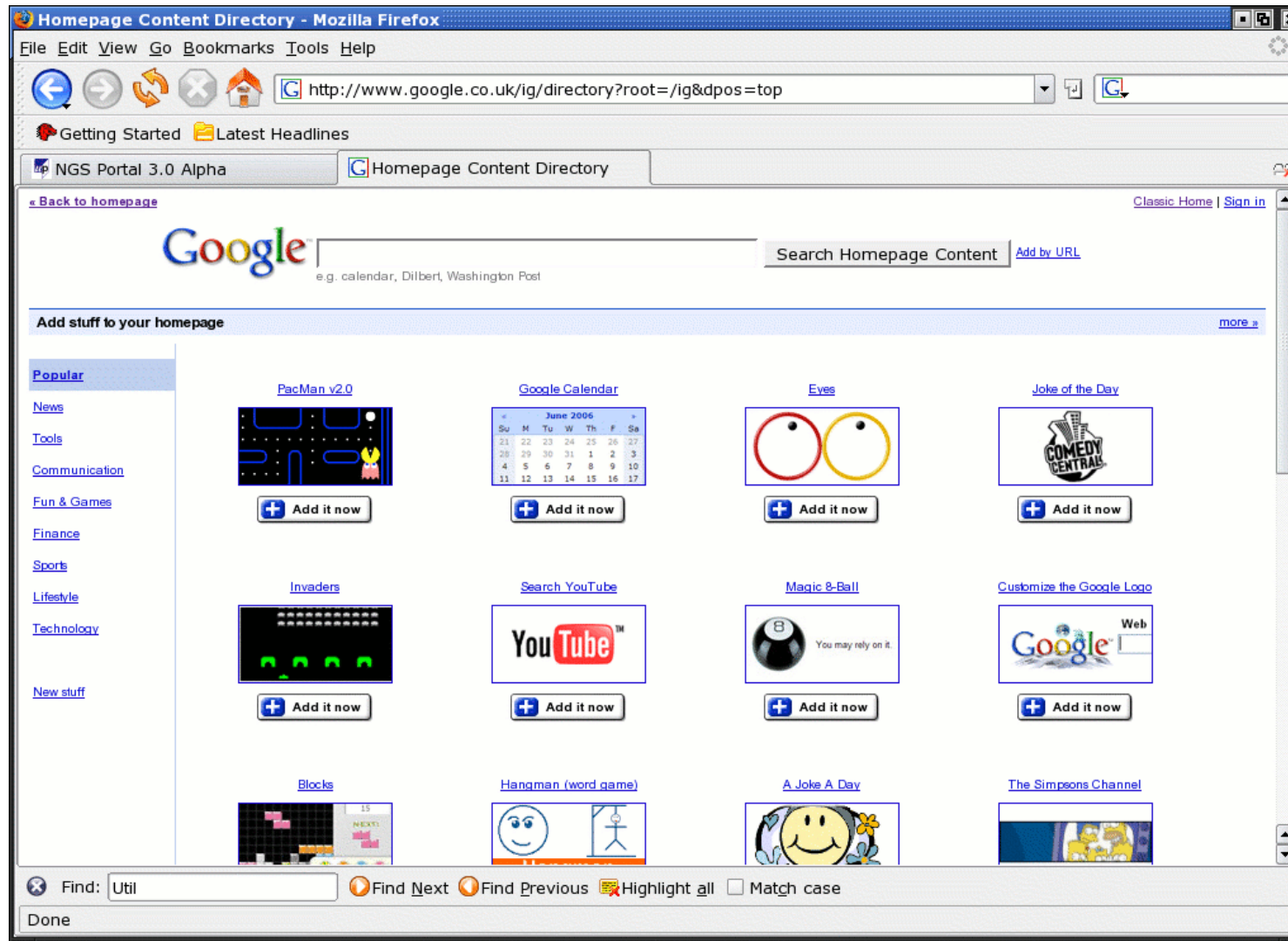
Portal and Portlets

- The NGS Grid portal extends a JSR-168 compliant portal container which hosts a selection of ‘portlets.’
- Portlets are online-accessible applications that are hosted and managed within the portal container.
- The list of portlets that are deployed to the portal make up the portal’s overall functionality (users may be interested only in a selection of portlets)
- Main Benefit: Portlets facilitate the sharing and re-use of applications (168-compliant 3rd party portlets can be used within a portal as required).
- NGS and CCLRC e-Science Centre are developing a collection of portlets designed for the Computational and Data Grid.
- The NGS portal is a current implementation of these Grid portlets.

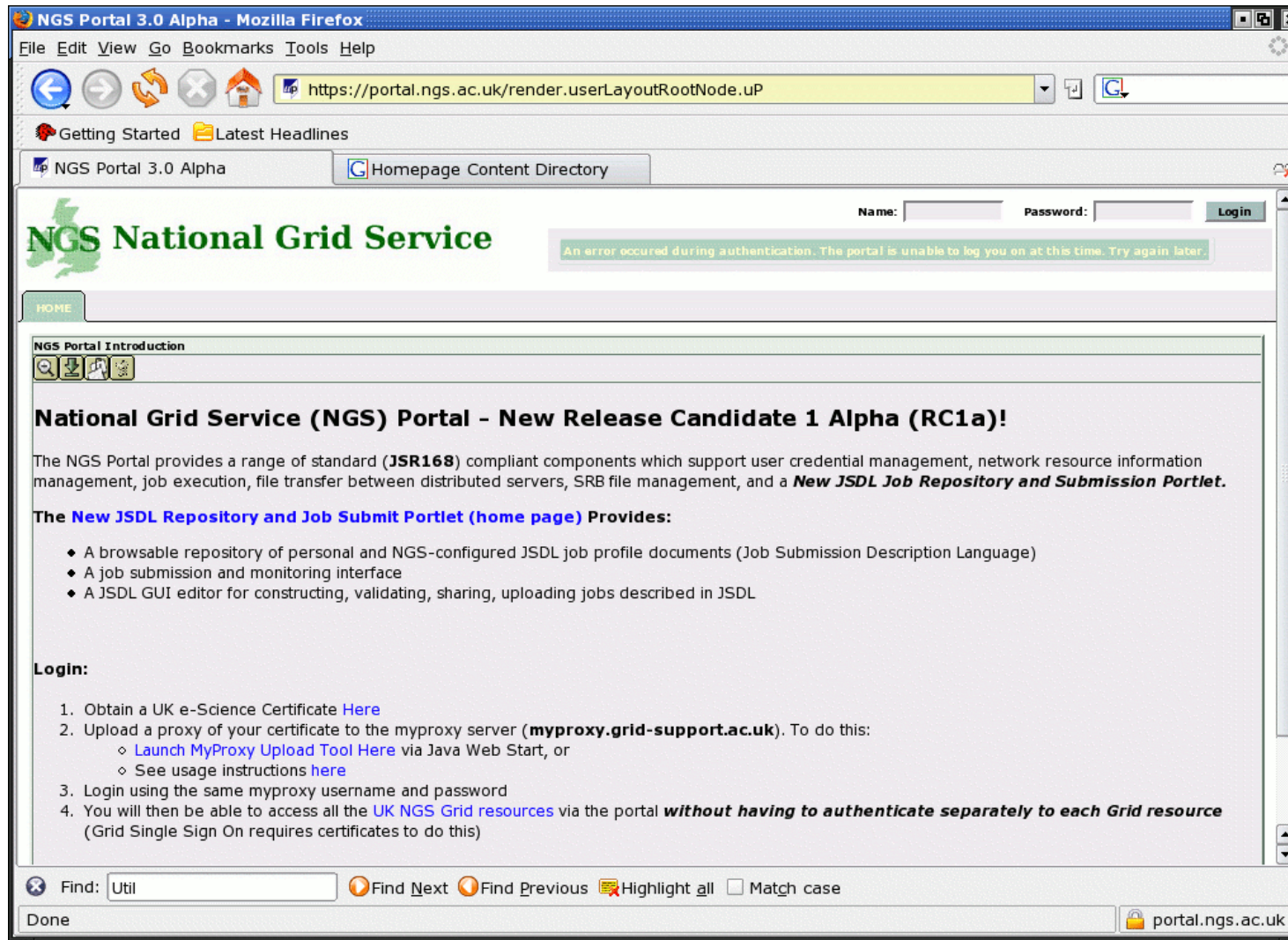
A Famous Portal + Portlets



Select Portlets of Interest (customisation)



NGS portal



NGS Portal 3.0 Alpha - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://portal.ngs.ac.uk/render.userLayoutRootNode.uP

Getting Started Latest Headlines

NGS Portal 3.0 Alpha Homepage Content Directory

Name: Password: Login

An error occurred during authentication. The portal is unable to log you on at this time. Try again later.

HOME

NGS Portal Introduction

National Grid Service (NGS) Portal - New Release Candidate 1 Alpha (RC1a)!

The NGS Portal provides a range of standard (**JSR168**) compliant components which support user credential management, network resource information management, job execution, file transfer between distributed servers, SRB file management, and a **New JSDL Job Repository and Submission Portlet**.

The New JSDL Repository and Job Submit Portlet (home page) Provides:

- A browsable repository of personal and NGS-configured JSDL job profile documents (Job Submission Description Language)
- A job submission and monitoring interface
- A JSDL GUI editor for constructing, validating, sharing, uploading jobs described in JSDL

Login:

1. Obtain a UK e-Science Certificate [Here](#)
2. Upload a proxy of your certificate to the myproxy server (**myproxy.grid-support.ac.uk**). To do this:
 - [Launch MyProxy Upload Tool Here](#) via Java Web Start, or
 - See usage instructions [here](#)
3. Login using the same myproxy username and password
4. You will then be able to access all the [UK NGS Grid resources](#) via the portal **without having to authenticate separately to each Grid resource** (Grid Single Sign On requires certificates to do this)

Find: Util Find Next Find Previous Highlight all Match case

Done portal.ngs.ac.uk

After Login – Grid portlets

The screenshot shows the NGS National Grid Service web portal. The browser title is "NGS Portal 3.0 Alpha - Mozilla Firefox". The address bar shows the URL: https://portal.ngs.ac.uk/tag.da985b5b91e3be5f.render.userLayoutRootNode.uP?uP_sp. The page features a navigation menu with items: Proxy Manager, Grid Information, SRB, Job Submit & File Transfer, and JSDL Repository & Job Submit. The main content area is titled "JSDL Repository" and includes a sub-menu with items: Start, Credentials, Browse + Load Jobs, Upload/ Download Tool, Job Categories, Browse Host, Admin, and Info/ Todo. Below the sub-menu, there are links for ActiveJob: Detail, Candidate Hosts, Description, Args, Env, File Systems, Stage File/Dir, JSDL, and Submit. The user is authenticated as "AUTHENTICATED_USER" and has an active job named "JobProfile name". The "Job Submission Portlet (Release Candidate 1 - RC1)" displays a message: "Loaded Grid proxy credentials ok". Below this, there are instructions: "For Help/Information place your cursor over the info icons", "Blue menu items relate to generic Grid operations", and "Orange menu items relate directly to the currently loaded 'Active Job'". A link "Browse the NGS Application Registry Start Here" is also present. The search bar at the bottom shows "Find: Util" and options for "Find Next", "Find Previous", "Highlight all", and "Match case". The status bar at the bottom indicates "Done" and the URL "portal.ngs.ac.uk".

JSDL Repository / Job Submit Portlet

1. A **browsable JSDL database** of personal and shared JSDL job profile documents (Job Submission Description Language, i.e. ‘job recipes/templates’).
 - JSDL can be **browsed for, selected and loaded** in order to run applications on the Grid (loaded either ‘out-of-the-box’ or, more usually loaded and modified/tweaked as required).
 - JSDL can be **searched for** by category of interest in the portal (e.g bioinformatics, chemistry, tutorials/examples).
 - JSDL documents can be **pre-configured and published** by the portlet administrator(s) to be made available to all other users.
2. A **JSDL GUI editor** for constructing, validating, sharing, uploading jobs described in JSDL.
3. A **Grid job submission** and monitoring application (currently, only Globus but more Grid middleware providers are being added, e.g. GridSam/WSRF).

JSDL – Job Submission Description Language

```

<jSDL:Application>
  <jSDL:ApplicationName>gnuplot</jSDL:ApplicationName>
  <jSDL-posix:POSIXApplication>
    <jSDL-posix:Executable>
      /usr/local/bin/gnuplot
    </jSDL-posix:Executable>
    <jSDL-posix:Argument>control.txt</jSDL-posix:Argument>
  <jSDL-posix:Argument>DavesControlFile.txt</jSDL-posix:Argument>
    <jSDL-posix:Input>input.dat</jSDL-posix:Input>
    <jSDL-posix:Output>output1.png</jSDL-posix:Output>
  </jSDL-posix:POSIXApplication>
</jSDL:Application>
<jSDL:Resources>
  <jSDL:IndividualPhysicalMemory>
<jSDL:LowerBoundedRange>2097152.0</jSDL:LowerBoundedRange>
  </jSDL:IndividualPhysicalMemory>
  <jSDL:TotalCPUCount>
    <jSDL:Exact>1.0</jSDL:Exact>
  </jSDL:TotalCPUCount>
</jSDL:Resources>
....
  
```

1. XML Schema language for describing compute jobs in a platform independent language (XML).
2. Is agnostic of middleware - no dependencies on Globus, WSRF, gLite (portal that is generic and not tied to any particular set of Grid technologies).
3. GGF / OGF Standard.
4. JSDL documents can be validated against the JSDL and JSDL POSIX XSD Schema to ensure its correctness

JSDL – Job Submission Description Language XSD Schema

```

<!--=====-->
<xsd:complexType name="Environment_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="name" type="xsd:NCName" use="required"/>
      <xsd:attribute name="filesystemName" type="xsd:NCName" use="optional"/>
      <xsd:anyAttribute namespace="##other" processContents="lax"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--=====-->
<xsd:complexType name="Argument_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:normalizedString">
      <xsd:attribute name="filesystemName" type="xsd:NCName" use="optional"/>
      <xsd:anyAttribute namespace="##other" processContents="lax"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
  
```

JSDL Repository / Database

Select category of interest – e.g. ‘RAL Bioinformatics.’

Browsing the JSDL database for public and personal job profiles.

List jobs, read job descriptions and load a job to initialise the ‘Active Job.’

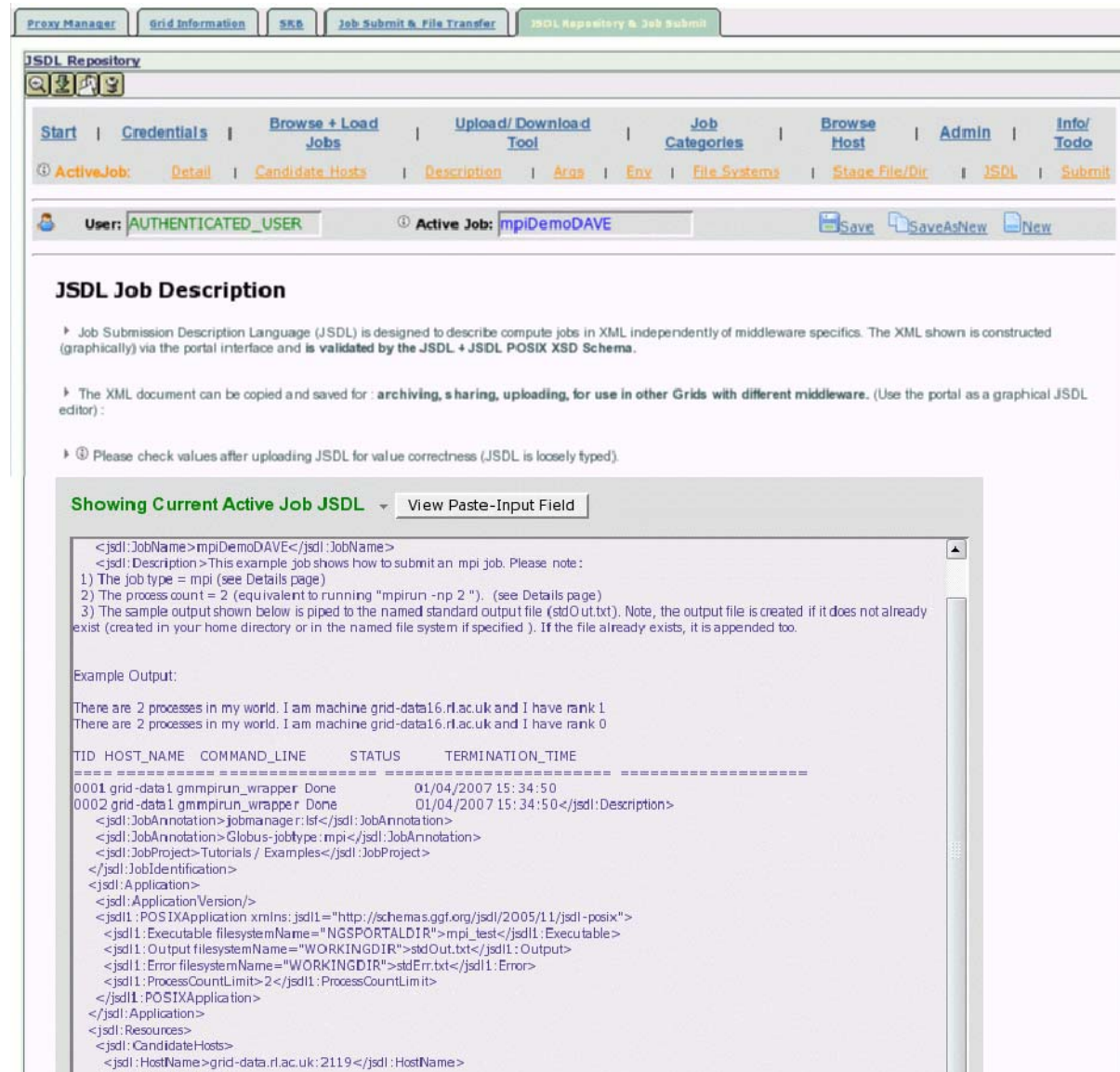
The screenshot shows the JSDL Repository web interface. At the top, there are navigation tabs: Proxy Manager, Grid Information, SRS, Job Submit & File Transfer, and JSDL Repository & Job Submit. Below the tabs, there are several menu items: Start, Credentials, Browse + Load Jobs, Upload/Download Tool, Job Categories, Browse Host, Admin, and Info/ Todo. A secondary menu includes ActiveJob, Detail, Candidate Hosts, Description, Args, Env, File Systems, Stage File/Dir, JSDL, and Submit. The user is logged in as AUTHENTICATED_USER and has an active job named JobProfile name. The main section is titled 'Browse and Load the 'Active Job'' and contains instructions: 1) Select a job category (e.g. Tutorials/Examples) and list the jobs in that category (NGS or personal jobs); 2) Select and load the ActiveJob from the list (click on the load link). Below the instructions are search filters: 'In Job Category' set to 'RAL Bioinformatics', 'With Status' set to 'all', and 'Search / List Job Profiles' with buttons for 'NGS' and 'Personal'. The results show 9 jobs found, listed in a table with columns for Description, Name, Exe, Modified, Status (Check), and Load. A 'Reset' button is at the bottom right of the table.

Description	Name	Exe	Modified	Status (Check)	Load
View SIESTA (paralell example)	SIESTA (paralell example)	/apps/siesta-mpi/siesta-2	Jan 17, 2007	<input type="checkbox"/>	load
View FASTA (serial example)	FASTA (serial example)	/usr/local/applications/bio	Jan 17, 2007	<input type="checkbox"/>	load
View NAMD (paralell example)	NAMD (paralell example)	/usr/local/applications/bio	Jan 17, 2007	<input type="checkbox"/>	load
View GROMACS (grompp exarr)	GROMACS (grompp exarr)	/usr/local/applications/bio	Jan 17, 2007	<input type="checkbox"/>	load
View GROMACS (MD example)	GROMACS (MD example)	/usr/local/applications/bio	Jan 17, 2007	<input type="checkbox"/>	load
View mpiBLAST (blastn example)	mpiBLAST (blastn example)	/usr/local/applications/bio	Jan 18, 2007	<input type="checkbox"/>	load
View EMBOSS (seqret example)	EMBOSS (seqret example)	/usr/local/applications/bio	Jan 18, 2007	<input type="checkbox"/>	load
View EMBOSS (wosname exa)	EMBOSS (wosname exa)	/usr/local/applications/bio	Jan 18, 2007	<input type="checkbox"/>	load
View BLAST (serial application)	BLAST (serial application)	/usr/local/applications/bio	Jan 19, 2007	<input type="checkbox"/>	load

Active Job's JSDL

The Active Job JSDL is automatically created, updated and validated by the portal by changing parameters in the portal GUI.

The portlet acts as a JSDL GUI editor



JSDL Job Description

- Job Submission Description Language (JSDL) is designed to describe compute jobs in XML independently of middleware specifics. The XML shown is constructed (graphically) via the portal interface and is validated by the JSDL + JSDL POSIX XSD Schema.
- The XML document can be copied and saved for : **archiving, sharing, uploading, for use in other Grids with different middleware.** (Use the portal as a graphical JSDL editor) :
- Please check values after uploading JSDL for value correctness (JSDL is loosely typed).

Showing Current Active Job JSDL

```

<jSDL:JobName>mpiDemoDAVE</jSDL:JobName>
<jSDL:Description>This example job shows how to submit an mpi job. Please note:
1) The job type = mpi (see Details page)
2) The process count = 2 (equivalent to running "mpirun -np 2 "). (see Details page)
3) The sample output shown below is piped to the named standard output file (stdOut.txt). Note, the output file is created if it does not already exist (created in your home directory or in the named file system if specified). If the file already exists, it is appended too.

Example Output:

There are 2 processes in my world. I am machine grid-data16.rl.ac.uk and I have rank 1
There are 2 processes in my world. I am machine grid-data16.rl.ac.uk and I have rank 0

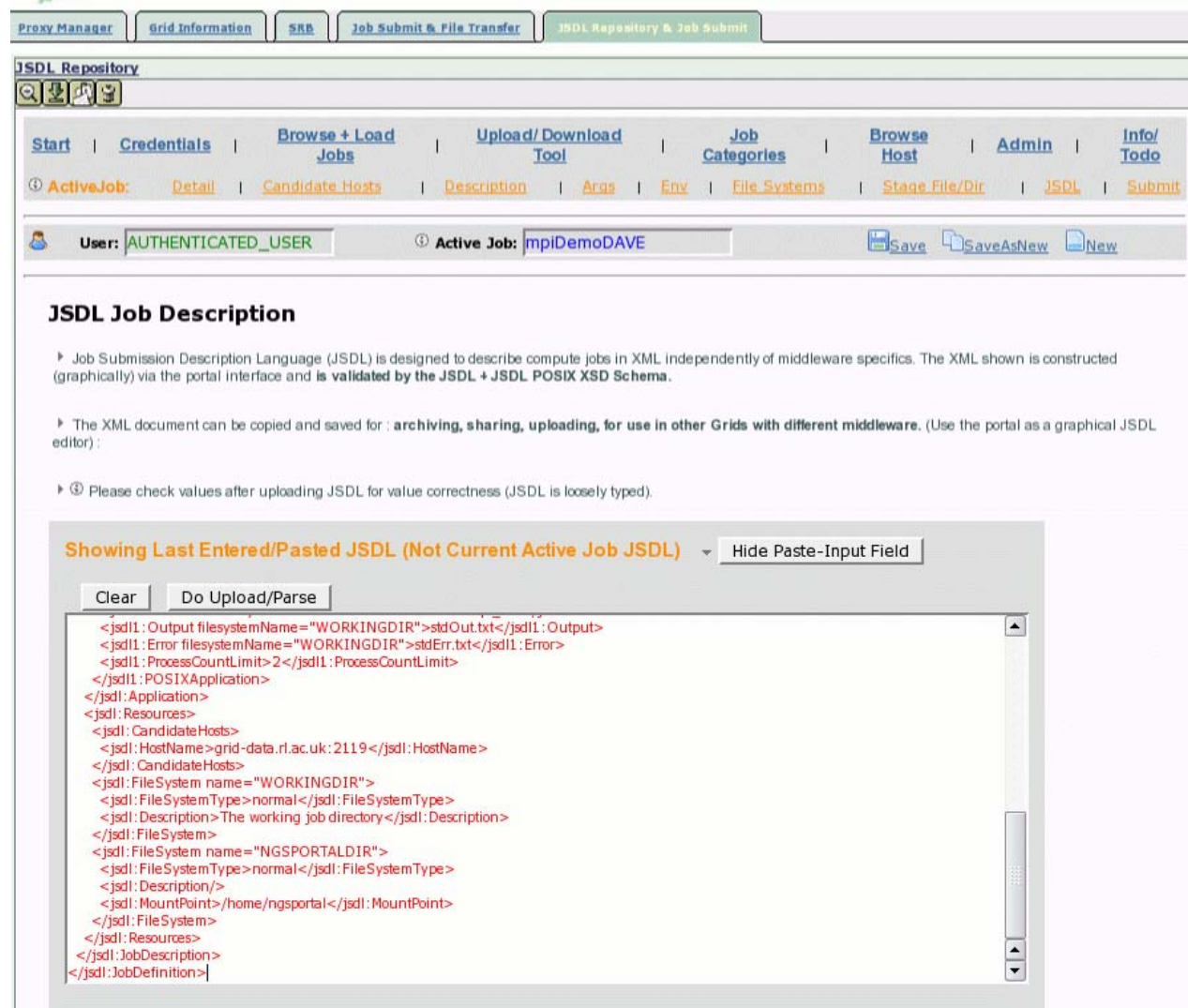
TID  HOST_NAME  COMMAND_LINE  STATUS  TERMINATION_TIME
-----
0001  grid-data16  gmpirun_wrapper  Done    01/04/2007 15:34:50
0002  grid-data16  gmpirun_wrapper  Done    01/04/2007 15:34:50</jSDL:Description>
<jSDL:JobAnnotation>jobmanager:isf</jSDL:JobAnnotation>
<jSDL:JobAnnotation>Globus-jobtype:mpi</jSDL:JobAnnotation>
<jSDL:JobProject>Tutorials / Examples</jSDL:JobProject>
</jSDL:JobIdentification>
<jSDL:Application>
<jSDL:ApplicationVersion>
<jSDL:POSIXApplication xmlns:jSDL="http://schemas.ggf.org/jSDL/2005/11/jSDL-posix">
<jSDL:Executable filesystemName="NGSPORTALDIR">mpi_test</jSDL:Executable>
<jSDL:Output filesystemName="WORKINGDIR">stdOut.txt</jSDL:Output>
<jSDL:Error filesystemName="WORKINGDIR">stdErr.txt</jSDL:Error>
<jSDL:ProcessCountLimit>2</jSDL:ProcessCountLimit>
</jSDL:POSIXApplication>
</jSDL:Application>
<jSDL:Resources>
<jSDL:CandidateHosts>
<jSDL:HostName>grid-data16.rl.ac.uk:2119</jSDL:HostName>
  
```

Upload / Share JSDL

JSDL documents can be uploaded to the portal in order to initialise the Active Job.

Validation errors and messages are displayed in the interface when uploading JSDL.

The portlet allows the sharing of job profiles and JSDL between users and user communities.



JSDL Repository

Start | Credentials | Browse + Load Jobs | Upload/Download Tool | Job Categories | Browse Host | Admin | Info/ToDo

Active Job: Detail | Candidate Hosts | Description | Args | Env | File Systems | Stage File/Dir | JSDL | Submit

User: AUTHENTICATED_USER | Active Job: mpiDemoDAVE | Save | SaveAsNew | New

JSDL Job Description

- Job Submission Description Language (JSDL) is designed to describe compute jobs in XML independently of middleware specifics. The XML shown is constructed (graphically) via the portal interface and is validated by the JSDL + JSDL POSIX XSD Schema.
- The XML document can be copied and saved for: archiving, sharing, uploading, for use in other Grids with different middleware. (Use the portal as a graphical JSDL editor):
- Please check values after uploading JSDL for value correctness (JSDL is loosely typed).

Showing Last Entered/Pasted JSDL (Not Current Active Job JSDL) | Hide Paste-Input Field

Clear | Do Upload/Parse

```

<jSDL:Output filesystemName="WORKINGDIR">stdOut.txt</jSDL:Output>
<jSDL:Error filesystemName="WORKINGDIR">stdErr.txt</jSDL:Error>
<jSDL:ProcessCountLimit>2</jSDL:ProcessCountLimit>
</jSDL:POSIXApplication>
</jSDL:Application>
<jSDL:Resources>
<jSDL:CandidateHosts>
<jSDL:HostName>grid-data.rl.ac.uk:2119</jSDL:HostName>
</jSDL:CandidateHosts>
<jSDL:FileSystem name="WORKINGDIR">
<jSDL:FileSystemType>normal</jSDL:FileSystemType>
<jSDL:Description>The working job directory</jSDL:Description>
</jSDL:FileSystem>
<jSDL:FileSystem name="NGSPORTALDIR">
<jSDL:FileSystemType>normal</jSDL:FileSystemType>
<jSDL:Description/>
<jSDL:MountPoint>/home/ngsportal</jSDL:MountPoint>
</jSDL:FileSystem>
</jSDL:Resources>
</jSDL:JobDescription>
</jSDL:JobDefinition>
  
```

Active Job Detail

Input fields are filled out for pre-configured applications.

Changes to the parameters in the GUI will update the generated JSDL automatically.

Input fields are taken from the JSDL and JSDL-POSIX extension schemas.

③ JSDL Job Identification:

① Job Name:

① Job Category: [Select](#)

① Job Description: [Job Description](#)

③ JSDL Application:

① App Name:

① App Version:

③ JSDL Job Resources:

① Execute Host+Port: [\[grid-data.rl.ac.uk:2119 \]](#)

① Job Manager:

① Resource Count:

① Min + Max Mem (MB):

① File Systems + ① Stage Data: [File Systems](#) [Stage Data](#)

③ JSDL POSIX Extensions:

① WORKINGDIR (Mount Point Path): [<Clear](#) ③

① Create New (Use Optional Name): [<Create WORKINGDIR](#) ③

(Specify path Relative to the [File System Mount Point](#) OR Full path starting with '/' - both produce valid JSDL executable element) ▾

① Executable Or Script:

(Specify paths Relative to [File System Mount Points](#) - Recommended)
(Can also specify a Full path starting with '/') ▾

① File Paths: Std In/Out/Error Files:

① Input File (Must Exist):

① Output File (Created if not exist):

① Error File (Created if not exist):

① JobType + ProcessCount:

① Wall Time:

① Arguments + Environment: [Environment](#) [Arguments](#)

[Update Active Job Profile](#)

Environment Variables / Arguments

Environment Variables

Setup environment variables required to run the job e.g. ["NGSMODULES" \(how to\)](#)

Edit	Name	Value	
▶ Edit	TMP	/tmp	<input type="checkbox"/>
▶ Edit	NGSMODULES	envVarValue1	<input type="checkbox"/>

[Close Detail](#)

Add required env vars, e.g. 'NGSMODULES' – used to configure application environment

Add Environment Variable:

① Name:

② Value:

Add

Add NGSMODULES

Command Line Arguments

- ▶ Enter or paste arguments below
- ▶ Use Space and/or Line separated values
- ▶ Use double quotes if space is required in a single argument, e.g. "a and b"

```
/usr/local/applications/bioinformatics/EBI/data/rel_tpa_inv_01_r89.dat seqr.out -osformat fasta
```

Update

Clear

Configured Arguments

```
/usr/local/applications/bioinformatics/EBI/data/rel_tpa_inv_01_r89.dat
seqr.out
-osformat
fasta
```

Paste and parse command line arguments (space and/or line separated values)

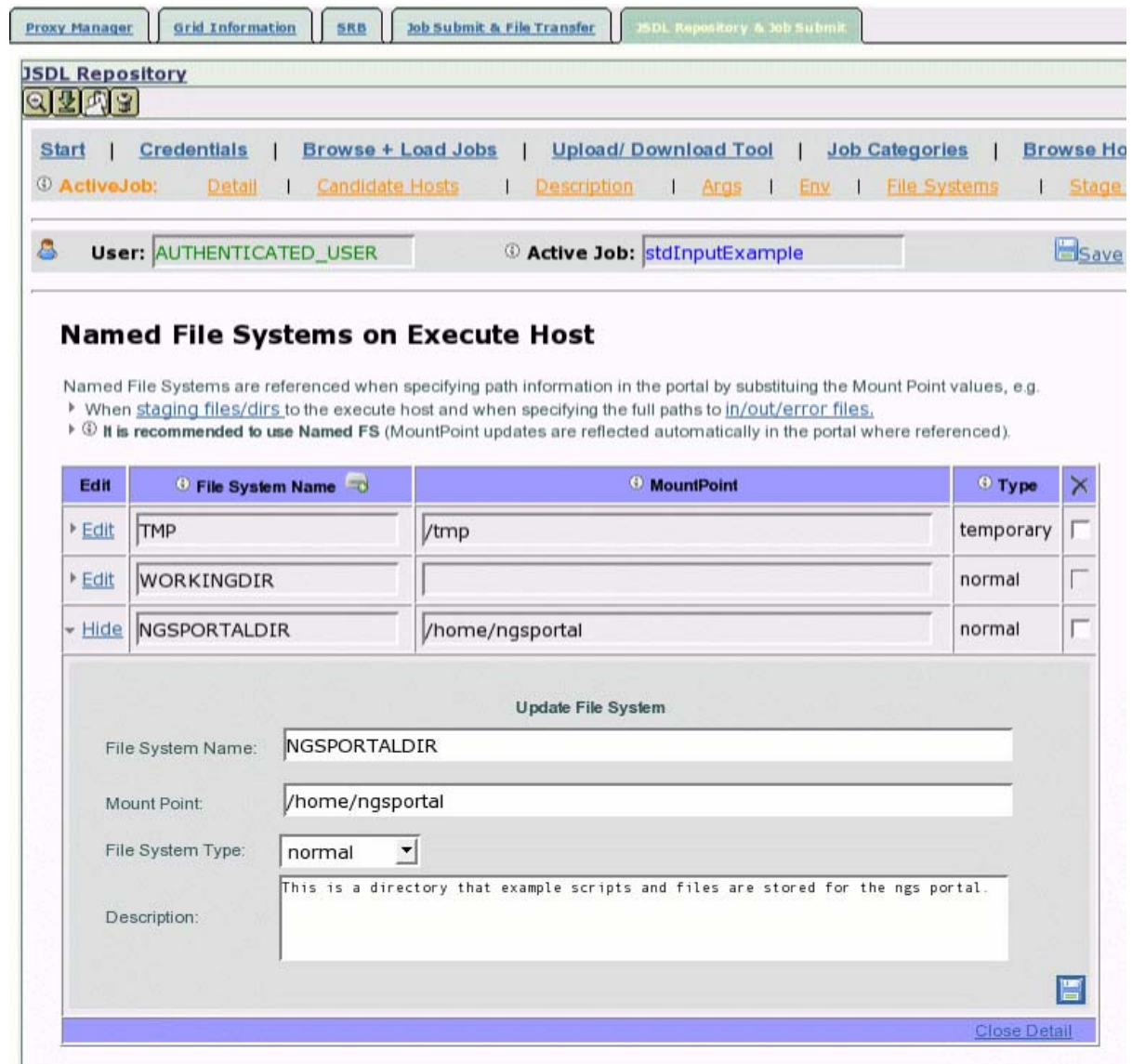
Named File Systems

Named file systems used to declare mount points that are required on the consuming system.

File system names are referenced throughout the portlet (and JSDL doc) for substituting mount points where required.

Changes to a FS mount point will be updated automatically throughout the portal/JSDL.

Used when specifying path info e.g., locations to files/dirs, stage data locations etc.



The screenshot shows the 'JSDL Repository' interface with a navigation bar at the top containing 'Proxy Manager', 'Grid Information', 'SRB', 'Job Submit & File Transfer', and 'JSDL Repository & Job Submit'. Below the navigation bar, there are several tabs: 'Start', 'Credentials', 'Browse + Load Jobs', 'Upload/ Download Tool', 'Job Categories', and 'Browse Ho'. A secondary set of tabs includes 'ActiveJob: Detail', 'Candidate Hosts', 'Description', 'Args', 'Env', 'File Systems', and 'Stage'. The user is identified as 'AUTHENTICATED_USER' and the active job is 'stdInputExample'. The main content area is titled 'Named File Systems on Execute Host' and contains explanatory text and a table of file systems.

Named File Systems are referenced when specifying path information in the portal by substituting the Mount Point values, e.g.

- ▶ When [staging files/dirs](#) to the execute host and when specifying the full paths to [in/out/error files](#).
- ▶ ⓘ It is recommended to use Named FS (MountPoint updates are reflected automatically in the portal where referenced).

Edit	File System Name	MountPoint	Type	
▶ Edit	TMP	/tmp	temporary	<input type="checkbox"/>
▶ Edit	WORKINGDIR		normal	<input type="checkbox"/>
▼ Hide	NGSPORTALDIR	/home/ngsportal	normal	<input type="checkbox"/>

Update File System

File System Name:

Mount Point:

File System Type:

Description:

Stage Data

Compile a list of required data (i.e. data that should be copied to the consuming system from remote locations across the Grid prior to job execution).

Data is staged relative to named file systems.

The source URI can be either specified manually or, more normally, browsed for in the 'Browse Host' page.



The screenshot shows the 'JSDL Repository' web interface. At the top, there are navigation tabs: Proxy Manager, Grid Information, SRB, Job Submit & File Transfer, and JSDL Repository & Job Submit. Below these are utility icons and a menu with options like Start, Credentials, Browse + Load Jobs, Upload/Download Tool, Job Categories, and Browse Host. A secondary menu includes ActiveJob, Detail, Candidate Hosts, Description, Args, Env, File Systems, and Stage File/Dir. The user is identified as 'AUTHENTICATED_USER' and the active job is 'stdInputExample'. The main section is titled 'Stage Data To The Execute Host' and contains instructions on building a list of files/dirs to be copied. Below the instructions are two options: 'Browse Host For Stage Data (Recommended)' and 'Manually Add Stage Data (Show Data Entry Panel)'. The 'Manually Add Stage Data' option is selected, showing a table with columns for Edit, File System Name, File/Dir Name, Creation, and Source URI. The table lists three entries for the 'WORKINGDIR' file system: 'dave.pl', 'dave.txt', and 'dummy.scala'. Below the table is a form for configuring the data source and destination. The 'Data Source' section has a 'Source URI' field with the value 'gsiftp://grid-data.rl.ac.uk:2811/home/ngs0153/dummy.scala'. The 'Destination on Execute Host' section has a 'File/Dir Path' field with a dropdown for 'WORKINGDIR' and a text field for 'dummy.scala'. There is also a 'Description' field and a 'Creation Option' dropdown set to 'overwrite'. At the bottom, there are buttons for 'View StageLog', 'Validate SourceURIs', 'Validate MountPoints', 'Do Staging', 'Delete Selected', and 'Close Detail'.

Edit	File System Name (Add)	File/Dir Name	Creation	Source URI (Browse Host)
Edit	WORKINGDIR	dave.pl	<input type="checkbox"/> overwrite	gsiftp://grid-data.rl.ac.uk:2
Edit	WORKINGDIR	dave.txt	<input type="checkbox"/> overwrite	gsiftp://grid-data.rl.ac.uk:2
Hide	WORKINGDIR	dummy.scala	<input type="checkbox"/> overwrite	gsiftp://grid-data.rl.ac.uk:2

Data Source:

Source URI:

Destination on Execute Host:

File/Dir Path: File System: Name (can be a relative path from FS):

Description:

Creation Option:

Browse Hosts

Browse remote Grid hosts for stage data.

Select files and directories that should be copied to the consuming system via GsiFtp (more protocols to be supported inc srb, ftp, webdav).

The screenshot shows the 'Browse Host' section of the JSDL Repository. At the top, there are navigation tabs: Proxy Manager, Grid Information, SRB, Job Submit & File Transfer, and JSDL Repository & Job Submit. Below these are sub-tabs: Start, Credentials, Browse + Load Jobs, Upload/Download Tool, Job Categories, and Browse Host. A user information bar shows 'User: AUTHENTICATED_USER' and 'Active Job: JobProfile name'. The main content area is titled 'Browse Host' and contains several form fields: 'Browse Host/Port' (set to 'grid-data.rl.ac.uk' and '2811'), 'ExecuteHost' (set to '()'), 'Selected URI' (set to 'grid-data.rl.ac.uk:2811/home/ngs0153'), 'Selected File', 'Selected Dir' (set to '/home/ngs0153'), 'Pwd' (set to '/home/ngs0153'), 'Make Dir In Pwd' (with a 'Create' button), and 'Staging Options' (set to 'WORKINGDIR' and 'overwrite'). Below these fields is a table listing files and directories.

Name	Stage	Mode	Size (bytes)	Modified
Baccup	<input type="checkbox"/>	755	3864	Aug 13
d	<input type="checkbox"/>	755	3864	Dec 17
daff.txt	<input type="checkbox"/>	644	21	Dec 17
daffTest1	<input type="checkbox"/>	755	3864	Jul 25
dave	<input type="checkbox"/>	755	3864	Nov 20
daveName	<input type="checkbox"/>	755	3864	Jan 8
dave.pl	<input type="checkbox"/>	755	3319	Aug 22
daveTestLSF_13-Dec-2006_13_56_28	<input type="checkbox"/>	755	3864	Dec 13
dave.txt	<input type="checkbox"/>	664	10	Dec 19
delGramLogs.sh	<input type="checkbox"/>	644	22	Nov 12
dummy.scala	<input type="checkbox"/>	644	5813	Nov 26
ftpGUI.jar	<input type="checkbox"/>	644	13028780	Feb 14
gram_job_mgr_9260.log	<input type="checkbox"/>	644	8340	Jan 4
hello.java	<input type="checkbox"/>	644	124	Jan 4
lca	<input type="checkbox"/>	775	3864	Dec 14

Candidate Hosts

Candidate hosts are consuming systems that can be nominated to run the Active Job.

The candidate host list can be compiled from a personal host list and from a default host list (available to all users).

The screenshot shows the JSDL Repository interface for configuring candidate hosts for an active job. The page title is "Candidate Hosts".

Navigation: Proxy Manager | Grid Information | SRB | Job Submit & File Transfer | JSDL Repository & Job Submit

Breadcrumb: Start | Credentials | Browse + Load Jobs | Upload/ Download Tool | Job Categories | Browse Host | ActiveJob: Detail | Candidate Hosts | Description | Args | Env | File Systems | Stage File/DI

User: AUTHENTICATED_USER | **Active Job:** stdInputExample

Candidate Hosts

Add hosts from NGS and personal lists to this job's candidate host list + select the execute host and port

- Select the Execute Host by clicking on a host name in one of the tables.
- Candidate hosts are associated with the active job and are those which may be selected to run the job.

Selected Execute Host: [grid-compute.leeds.ac.uk:2119](#) | **JobManager Port:** 2119 | Update Port

Add NGS + Personal Hosts To Candidate List

NGS Grid Hosts:	
grid-data.rl.ac.uk	Add ▾
grid-data.man.ac.uk	Add ▾
grid-compute.leeds.ac.uk	Add ▾
grid-compute.oesc.ox.ac.uk	Add ▾
monster2.phy.bris.ac.uk	Add ▾
wesc9-comsc.grid.cf.ac.uk	Add ▾
grid.lancs.ac.uk	Add ▾
grid-compute.cpc.wmin.ac.uk	Add ▾

Personal Grid Hosts: List ▾	
clyde.dl.ac.uk:8080	<input type="checkbox"/> Add ▾

Active JobProfile Candidate Host List

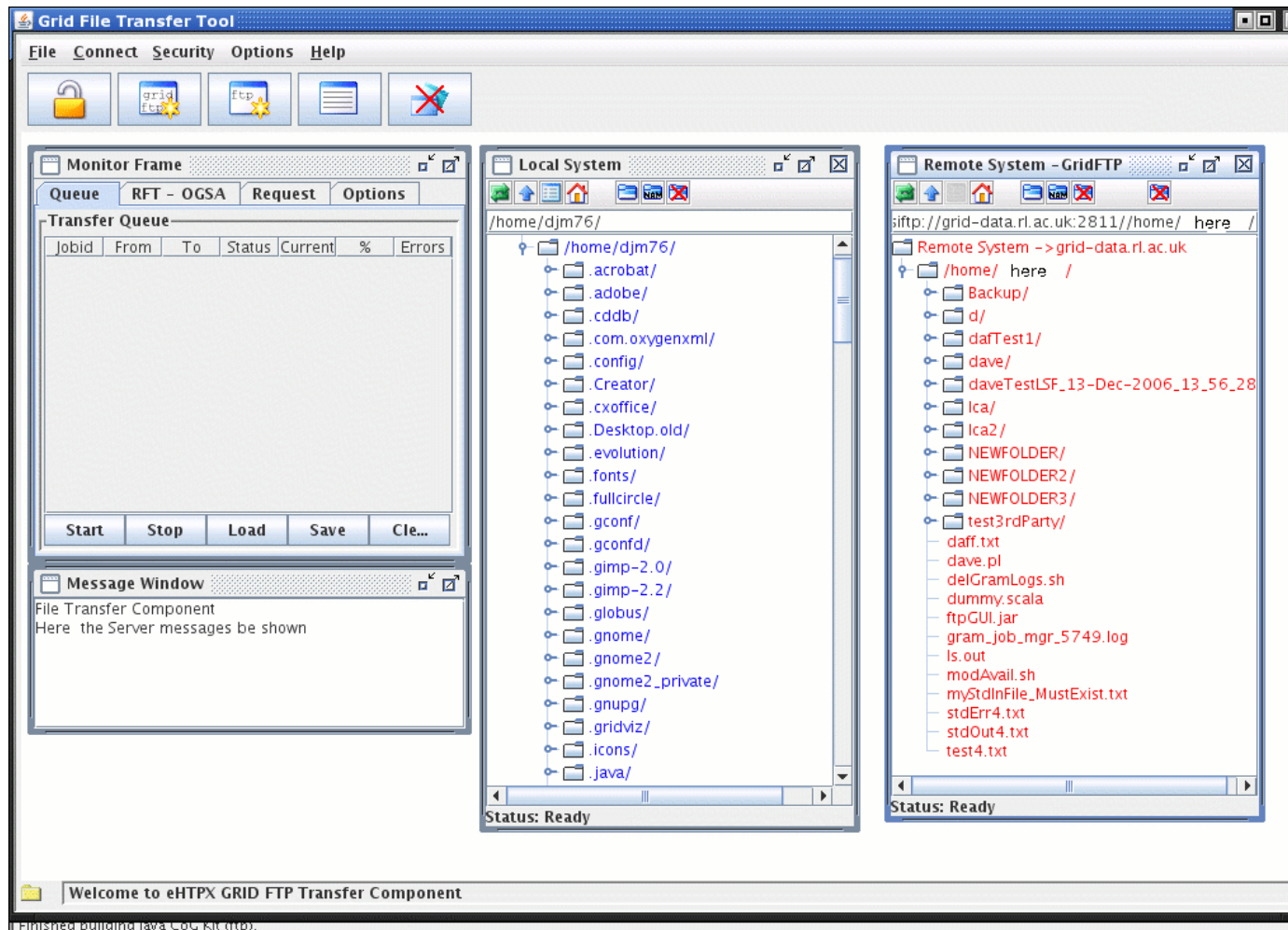
Active JobProfile Candidate Hosts:	
grid-data.rl.ac.uk	<input type="checkbox"/>
grid-compute.leeds.ac.uk	<input type="checkbox"/>

Add Candidate Host: <Add

Add Personal Host: <Add

GridFtp Upload / Download / File Transfer Tool

Transfer files to/from your desktop and a remote Grid ftp server via 'drag-n-drop' – akin to Windows explorer for the Grid



Summary

1. Core of the application is designed to be **generic and not tied to any particular set of Grid technologies** (facilitated using JSDL). Middleware dependencies emerge at **job-submit/monitor time when the specifics of the middleware have to be accommodated** (e.g. parsing the JSDL into RSL, adding mw-specific parameters, e.g. RSL JobType).
2. Application can be **deployed as a JSR-168 portlet or as a standalone Web application**. This helps deployment – e.g. openPortal that was designed to show users what is available before having to ‘log-in’ (encourage users).
3. Currently, application only supports Globus, but GridSAM will be added shortly (more on GridSAM next few days).

TODO / Future

1. Extend application to support more Data Grid + Web protocols for data staging (SRB, WebDav). This will involve browsing / interacting with different data protocols in the interface (e.g. browse SRB), but deciding who actually performs the staging is currently being decided (manage in application, or leave to job submission service). Crossing protocols adds some complexity.
2. Growing list of improvements / suggestions to refine interface – HCI (Human computer interaction).
3. Extend the interface for **Data visualisation** via the portal interface.
4. Release the portlets for use in other projects / Grids.
5. Longer term - Extend the portlet to support the registration of new interfaces, i.e. register **application specific interfaces** designed for specific applications. (Note, new interfaces will build JSDL in same way – will use underlying portlet functionality in same way).
6. A portal is only as good as the underlying deployed infrastructure.... portal development often involves debugging the underlying consuming systems and middleware

Software Stack

JSF (Java Server Faces) interface + MVC control layer (Http session and request scope data)

Spring v2.0 managed business objects (singleton + prototype injected object graphs, declarative transaction demarcation, data source management).

C3p0 db connection pooling

ORM (object relational mapping) - JPA (Java persistence API) + Hibernate 3.2 for domain model.

Java CogKit for Globus API

Apache XMLBeans for JSDL xml-object data binding framework