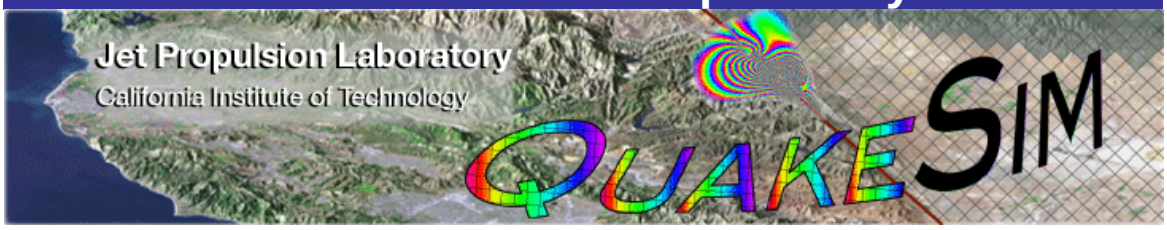
Milestone K - Interoperability



Numerical Simulations For Active Tectonic Processes: Increasing Interoperability And Performance

JPL Task Order: 10650

Milestone K – Interoperability due date: 9/30/2004

Customer delivery - Documented source code made publicly available via the Web.

- Demonstrate integration of one external user application into the framework using the GRID framework wizards
- Issue testable 5 year earthquake forecast for M>5 for S. California
- Publish the availability of the Portal to the Earthquake community in a peer reviewed periodical such as "Concurrency: Practice and Experience", or "EOS" or an AGU journal.

Team

Andrea Donnellan:
Principal Investigator
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
donnellan@jpl.nasa.gov
818-354-4737

Michele Judd:
Technical Task Manager
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
michele.judd@jpl.nasa.gov
818-354-4994

Jay Parker:
Overall Software Engineer
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 238-600
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Jay.W.Parker@jpl.nasa.gov
818-354-6790

Terry Tullis:
Fast Multipole Methods
Brown University
Box 1846, Brown University
Providence, RI 02912-1846
Terry_Tullis@Brown.edu
401-863-3829

Geoffrey Fox:
Information Architect
Community Grid Computing Laboratory
Indiana University
501 N. Morton, Suite 224
Bloomington, IN 47404-3730
gcf@indiana.edu
812-856-7977

Dennis McLeod:
Database Interoperability
Professor
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
mcleod@usc.edu
213-740-4504

John Rundle:
Pattern Recognizers
Center for Computational Science and Engineering
U. C. Davis
Davis, CA 95616
rundle@geology.ucdavis.edu
530-752-6416

Gleb Morein:
Pattern Recognizers
Center for Computational Science and Engineering
U. C. Davis
Davis, CA 95616
gleb@cse.ucdavis.edu

Greg Lyzenga:
Finite Element Models
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 126-347
4800 Oak Grove Drive
Pasadena, CA 91109-8099
greg.lyzenga@jpl.nasa.gov
818-354-6920

Marlon Pierce:
Code Interoperability Software Engineer
Community Grid Computing Lab
Indiana University
501 N. Morton, Suite 224
Bloomington, IN 47404-3730
marpierc@indiana.edu
812-856-1212

Lisa Grant:
Fault Database Architect
University of California, Irvine
Environmental Analysis and Design
Irvine, CA 92697-7070
lgrant@uci.edu
949-824-5491

Robert Granat:
Pattern Recognizers
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 126-347
4800 Oak Grove Drive
Pasadena, CA 91109-8099
robert.granat@jpl.nasa.gov
818-393-5353

Maggi Glasscoe:
GeoFEST Code Verification
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 300-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Margaret.T.Glasscoe@jpl.nasa.gov
818-393-4834

Demonstrate integration of one external user application into the framework using the GRID framework wizards

Overview

This deliverable of Milestone K required the development of Grid framework portal wizards and Web Services that can be used to deploy and manage remote applications that are available to users through the portal. This capability was then demonstrated by adding an external user application to the portal. The following documentation describes the portal wizard framework (which we refer to as a "code wiki") and describes its functionality.

We have provided examples for using the code wizard to deploy applications. For our third party application, we chose an application based on the original Okada method that was provided by Assistant Professor Kristy Tiampo of the University of Western Ontario's Department of Earth Sciences. We also show how to add Disloc and various UNIX commands using the wizard interface.

Code Wiki (Wizard Interface) Administrator Manual

The portal code wiki is intended to simplify the process by which privileged users make applications available through the portal interface. We make the following limiting assumptions in the current version:

1. Only privileged portal users have access to the code wiki interface.
2. All code administrators have the same level of access to applications; we do not provide finer grained access control to specific applications.
3. The desired application is already installed on the host computer.
4. The host computer runs QuakeSim execution Web Services.
5. The execution Web Services for the host computer have been added to the system property file: GEMDSTEST.properties .
6. All applications will run under the same account as the execution Web Service on that particular host.
7. Applications correspond to single UNIX executables or scripts.
8. We do not generate specialized Web forms for authoring application-specific input files.

Several specific examples are given at the end of the document.

We do not currently support remote installation of applications through the browser, although this may be supported in the future. The necessary services to do this have been developed: the Apache Ant engine at the heart of the execution Web Services is an adaptation of build/make tool. While code compilation services are thus possible, there are many uncertainties, such as available compilers and gcc versions. For this reason we have chosen not to address implementation capability in the current version.

How it Works

The application administration services are based around Web Service operations, as discussed in more detail in the QuakeSim User Manual. We store data using simple "in-memory" XML datastores that are implemented using Castor. Data is stored persistently using XML files. This is documented more thoroughly in http://www.servogrid.org/slide/GEM/Interop/AWS.doc.

Information about applications is used to create simple, application specific web pages dynamically from templates. These application templates in turn are used to gather the information from users that is needed to run the remote application. This information is fed to an Apache Ant-based web service on the application's execution hosts. The execution service runs an Ant build script template that is flexible enough to do most single step application executions.

The build script template is shown as follows:

```
<?xml version="1.0"?>
<project name="Template" default="all" basedir=".">

<target name="mkdir">
  <echo message="Making project directory"/>
  <echo message="${workDir.prop}"/>
  <mkdir dir="${workDir.prop}"/>
</target>

<target name="ExecTemplate">
  <echo message="Bindir is ${bindir.prop}"/>
  <echo message="Project Name is ${projectName.prop}"/>
  <echo message="Executable is ${executable.prop}"/>
  <echo message="Standard input is ${stdin.prop}"/>
  <echo message="Standard output is ${stdout.prop}"/>
  <echo message="Command line is ${commandLine.prop}"/>
  <exec executable="${executable.prop}"
      output="${stdout.prop}"
      dir="${workDir.prop}">
```

```
        <arg line="${commandLine.prop}"/>
        <env key="PATH" path="${bindir.prop}:/bin/:$PATH"/>
      </exec>
    </target>
    <target name="RunTemplate" depends="ExecTemplate">
      <echo message="Template run completed"/>
    </target>
    </project>
```

Parameter values (enclosed by ${}) are collected from the template web forms and passed to the script on the remote execution host via web service invocations.

Ant projects can include multiple targets, so we can develop sequential workflow engines that can process jobs with multiple steps. Specialized Ant script templates for specific codes (such as GeoFEST and Virtual California) were developed in the project. The current version does not support code sequencing but this may be developed in future versions.

Privileged Accounts for Code Administration

The code administration portlets allow users to make changes to application metadata that can potentially break portal applications, so these interfaces must be protected. We restrict access to these portlets by using Jetspeed role privileges, described in more detail in the accompanying Appendix A, “Jetspeed Role Configuration.”

For the application wiki, we use the role “code-admin-only” for application administrators. The role definition, given below, is located in the file security.xml, located in $TOMCAT_HOME/webapp/jetspeed/WEB-INF/conf.

```
<security-entry name="code-admin-only">
  <meta-info>
    <title>Code-admin-only</title>
    <description>Access for code administrators.</description>
  </meta-info>
```

```
<access action="*">
  <allow-if role="code-admin" xsi:type="allow-if"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
</access>
```

The admin portlets are associated with this role in their .xreg definition file, as shown in the XML fragment below. This is given in cgl-local-portlets.xreg, also located in the conf directory.

```
<portlet-entry name="SERVO_Admin_Portal" hidden="false" type="ref"
  parent="WebFormPortlet" application="false">
  <security-ref parent="code-admin-only"/>
  <security role="code-admin"/>
  <meta-info>
    <title>SERVO Application Admin Portal</title>
    <description>Application Administration Interface </description>
  </meta-info>
  <classname>commgrids.jetspeed.portlets.WebFormPortlet</classname>
  <parameter name="portal_id" value="awsportal" hidden="true"
    cachedOnName="true" cachedOnValue="true"/>
  <parameter name="base_url"
    value="@HOST_URL@"
    hidden="false" cachedOnName="true" cachedOnValue="true"/>
  <parameter name="first_page"
    value="/jetspeed/GCWS/Admin/Main.jsp" hidden="true"
    cachedOnName="true" cachedOnValue="true"/>
  <url cachedOnURL="true"/>
</portlet-entry>
```

Specific users can be assigned to the code-admin-only role by the portal administrator, as described in “Jetspeed Role Configuration.”

Adding a New Application

First, log in to the portal using an account with "code-admin-only" privileges, as described above. A default account is provided. On the top tab listings, you should see a tab labeled "Code Admin." Click this tab to get the code administration main page.

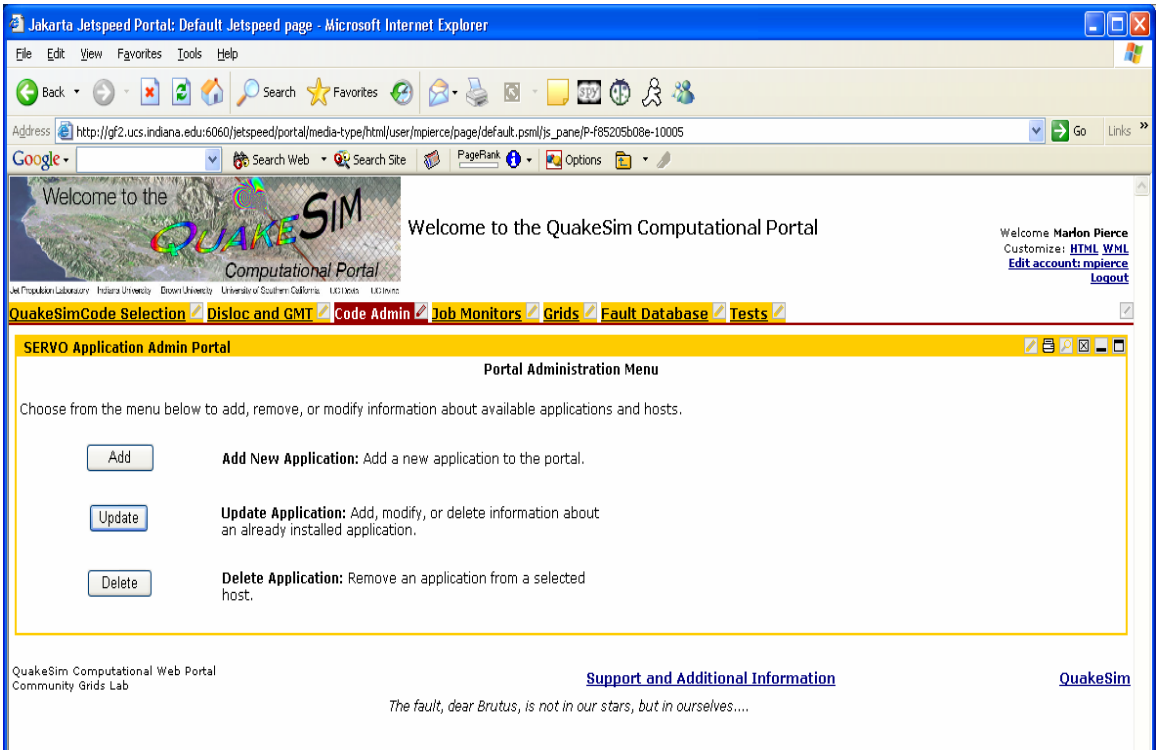


Figure 1. The Code Administration main page.

Click the "Add" button. You should see a screen similar to Figure 2. Fill out the web forms as follows:

1. Provide a useful name for the application.
2. Specify the number of input files and/or parameters (only 1 input file or parameter line is currently supported).
3. Specify the number of output files and/or parameters. There must be at least 1. The first output file is always assumed to hold the results of standard output.
4. Check the box if you wish to opt out of templating (not recommended).
5. Provide a brief description of your application. This can be either plain text or HTML, so you may, for example, include formatted instructions or links to more detailed documentation.

Click “Submit” when you are done.

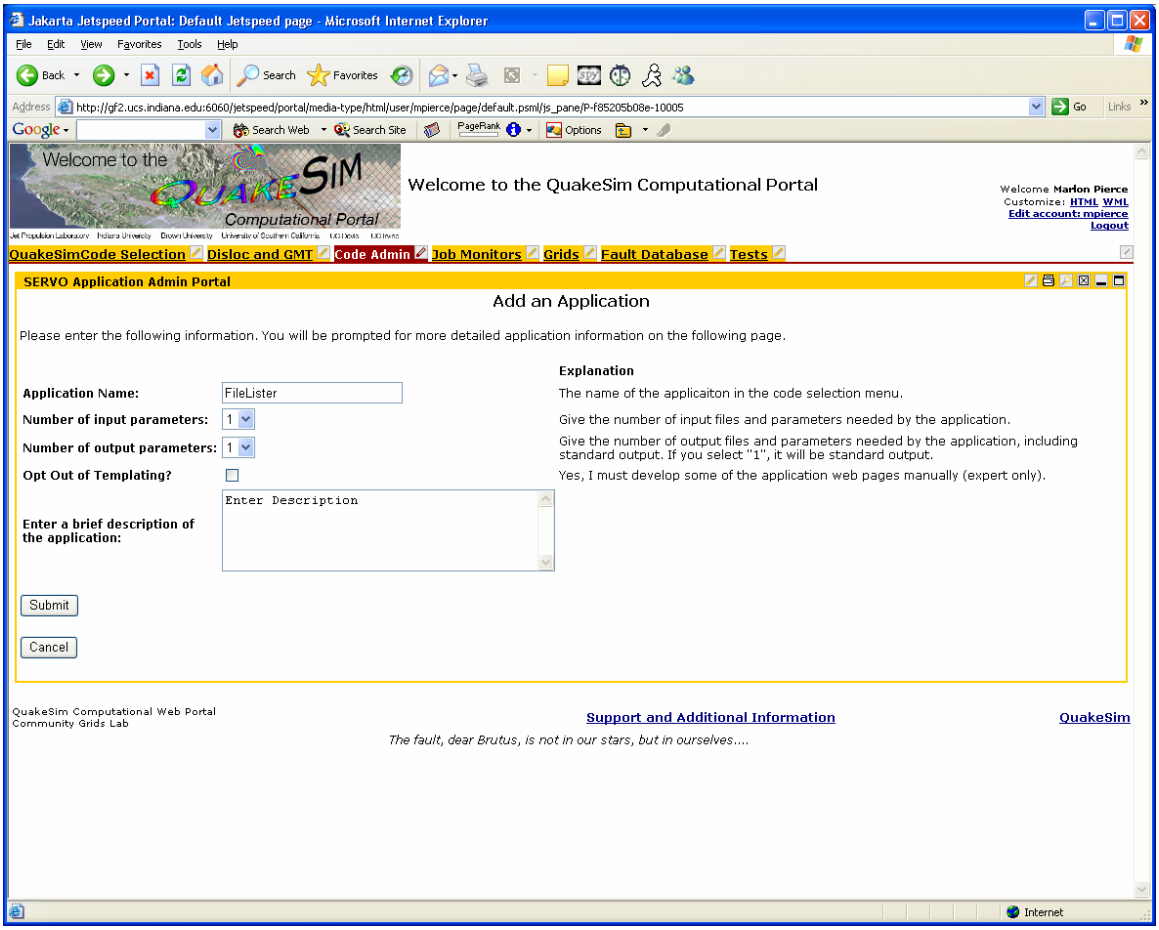


Figure 2. Web forms for adding a new application.

Your choices in the form shown in Figure 2 are used to generate the form shown in Figure 3. In this form, you should provide the following information for each input type:

1. A short name, or handle, for each input file/parameter.
2. The input mechanism (can be either a “c-style” space-separated command line or a UNIX-style standard input).
3. Whether the input is a file or parameter. Files are loaded as file names. Parameters are interpreted literally by the executable. For example, the */bin/cat* requires an input file for execution, so the appropriate input type should be “File” since we must provide disloc with the name of a file to be reach. On the other hand, the executable */bin/echo* takes a parameter that it interprets literally.

4. An input description that will be used in the generated template pages. The description can be plain text or HTML, and can include (for example) links to more detailed documentation.

For each output type, you must provide the following information:

1. Names for output fields. The first output field is always taken to be standard output.
2. An output description for use in the generated forms.

Finally, you must provide a host computer.

1. Select an available host from the drop down menu. This menu is generated from the available execution hosts specified in the GEMDSTEST.properties file.
2. Provide the IP address of the machine (this is not currently used but must be provided).
3. Give the full path of the code executable.

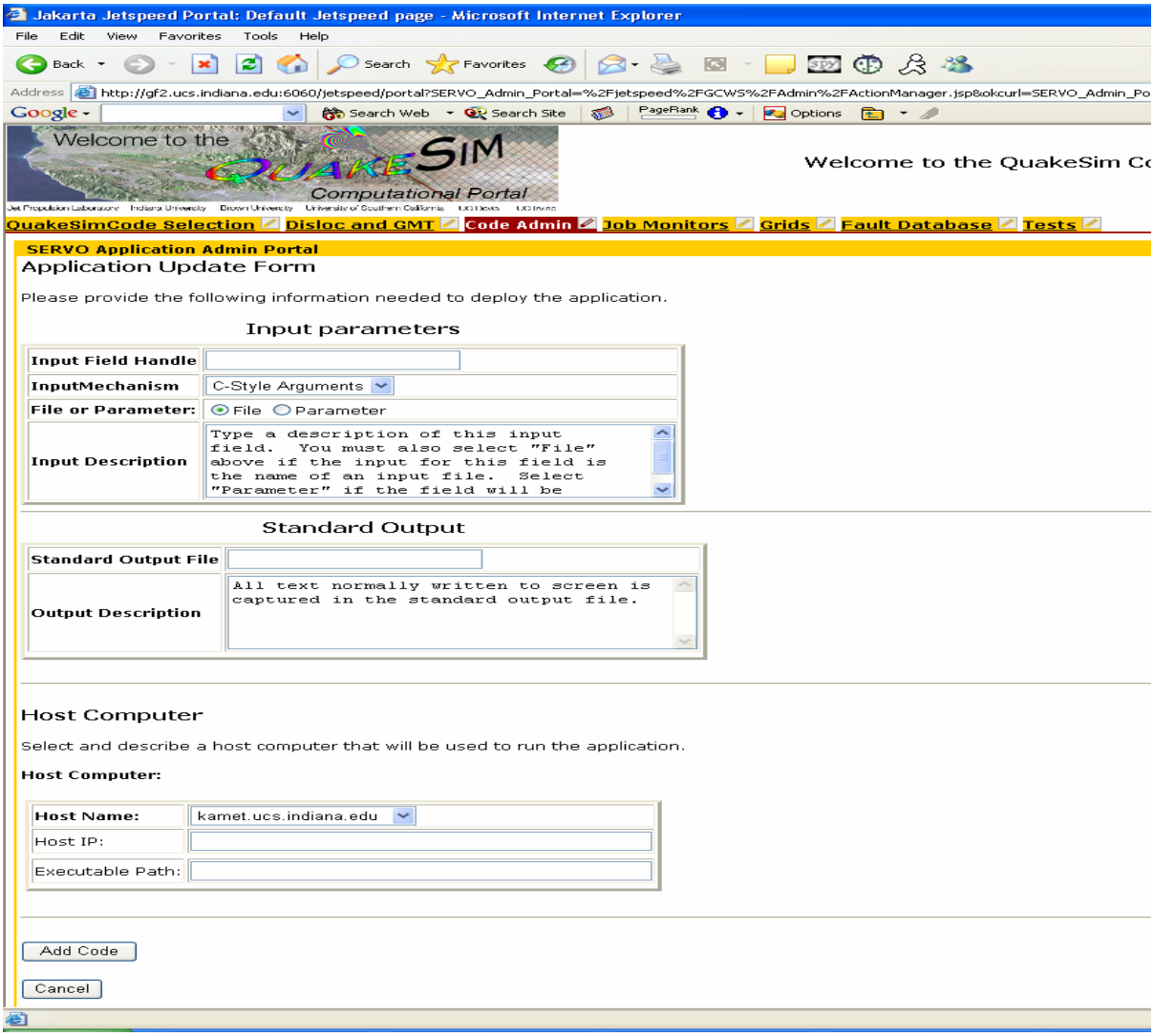


Figure 3. Application forms for specifying the input and output/error fields.

Click “Add Code” when you are done to commit the addition.

You may add hosts and modify setting using the update features described below.

Running the New Application

You can test out your application by clicking the “QuakeSim Code Selection” tab on the top horizontal tab listing. You should see the usual code selection menu with your application added to the list, similar to Figure 4. For example, the application “FileLister” may be added using the above forms. It will appear at the bottom of your list of applications.

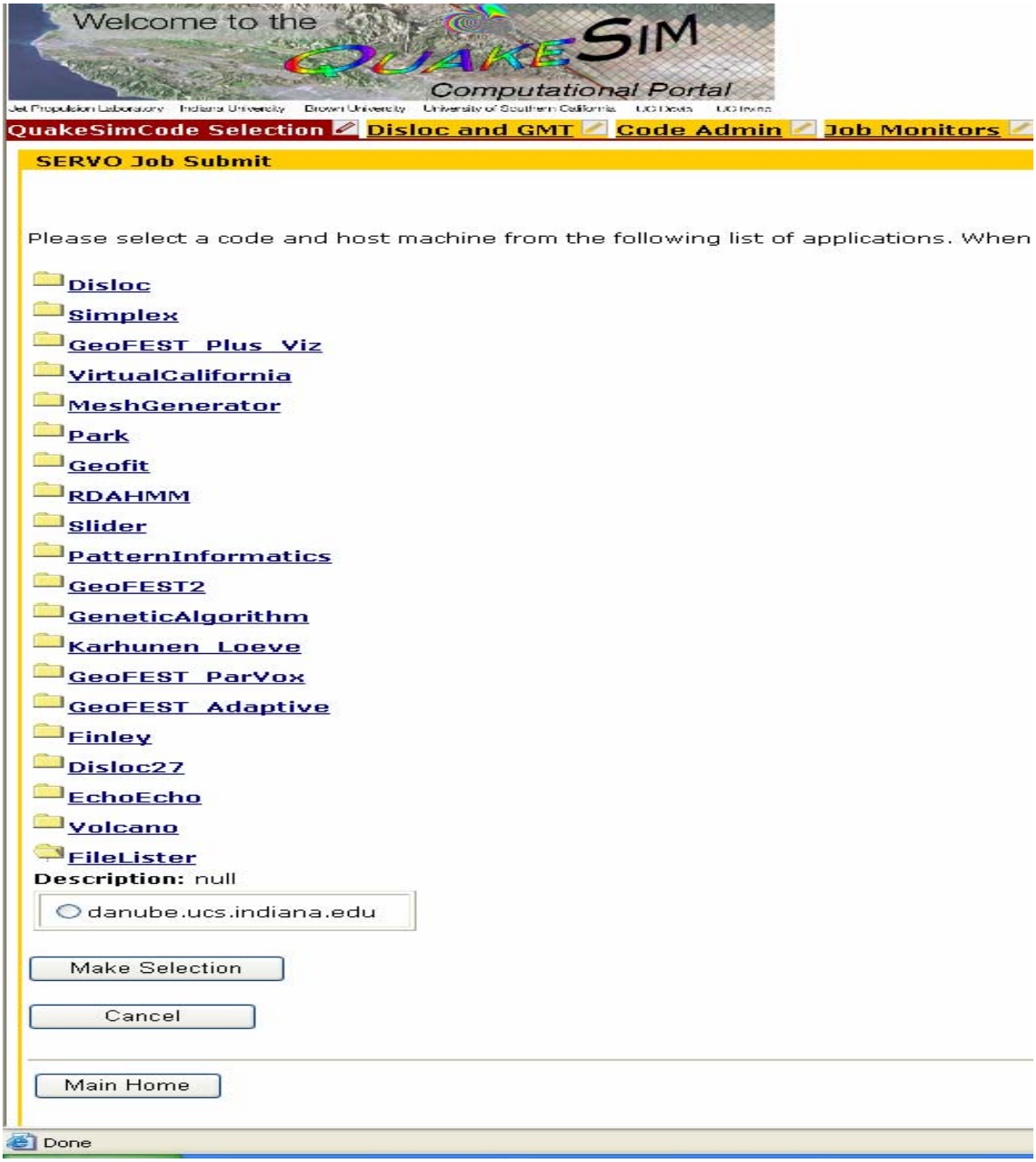


Figure 4. The code selection menu with the new application added to the listing.

Click the application and you will see the host that you selected. Click the round radio button next to the host and then click “Make Selection”. This will forward you to the templated page shown in Figure 5. From here you may perform the following operations:

1. Create New Project: Start with an empty project.
2. Load Project: List old projects, which may be loaded from the archive or deleted.
3. Access Archived Data: Access the output data from previous runs.

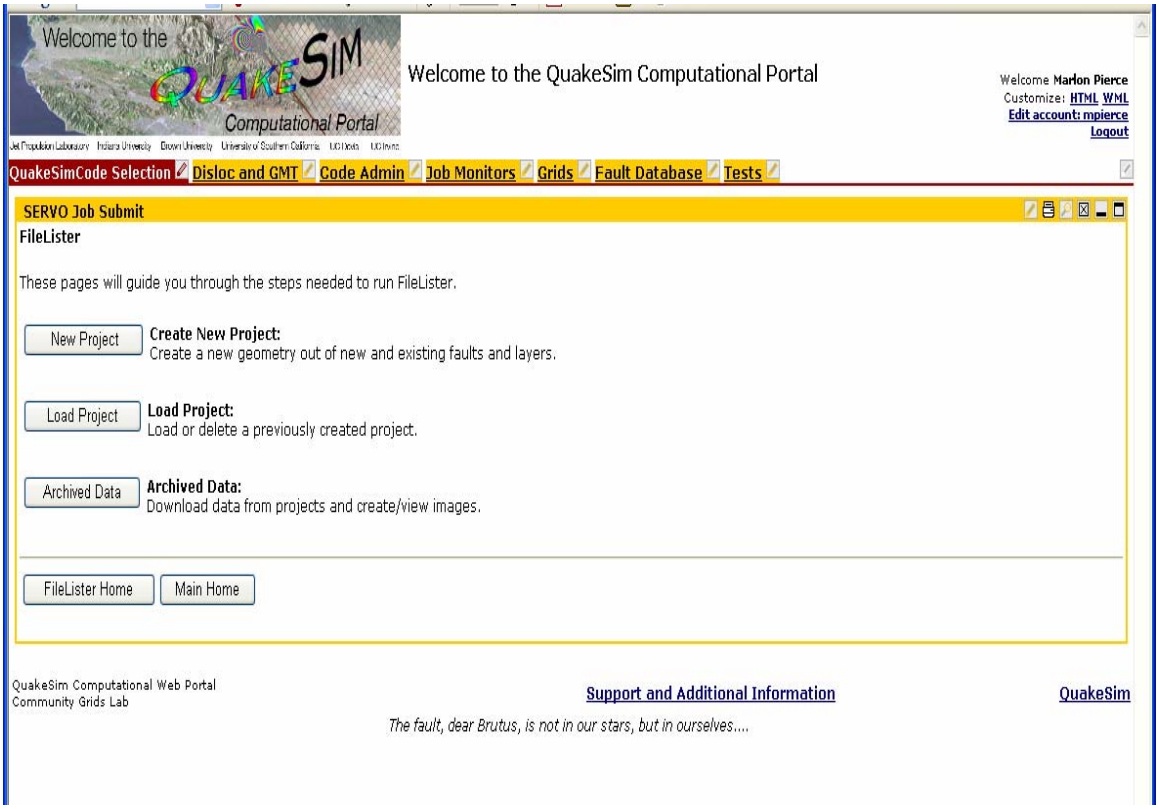


Figure 5. Main menu for newly deployed application.

After choosing a new project or loading an old project, you will be forwarded to a page similar to Figure 6. If you are using a loaded project, the previously provided values for that session will be displayed (and may be edited); for new projects, these forms will be blank. You must provide the name of the project and the input files and/or parameters. For file input, you may optionally upload files from your desktop using the upload button at the bottom of the form.

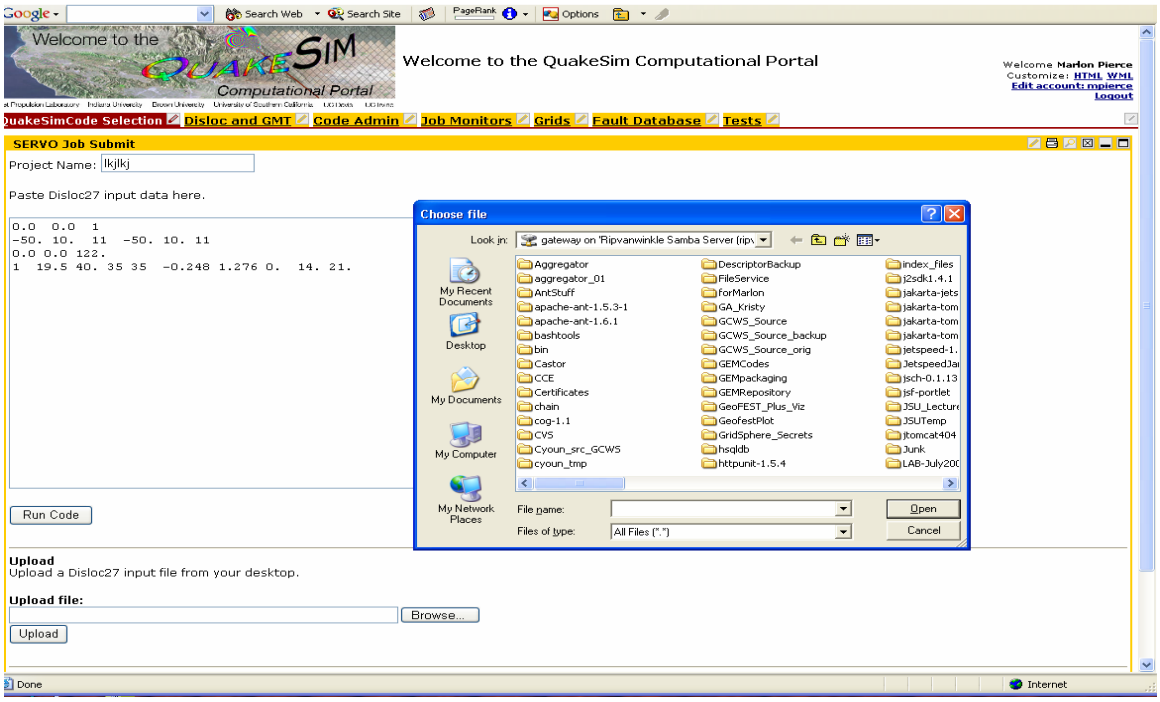


Figure 6. Project page example.

By selecting “Archived Data” for your project (Figure 4) you will get a listing of the output data from previous project runs (these correspond to the project listings seen on the “Load Project” menu.) A sample is shown in Figure 7. For each project, you will get one download link for each output type you selected (see Figure 2). The sample page shows Disloc deployed in this fashion. There are two download links: one for standard output (usually empty but required) and one for the output file. Clicking this link will download the file and open in a browser if the output is text.

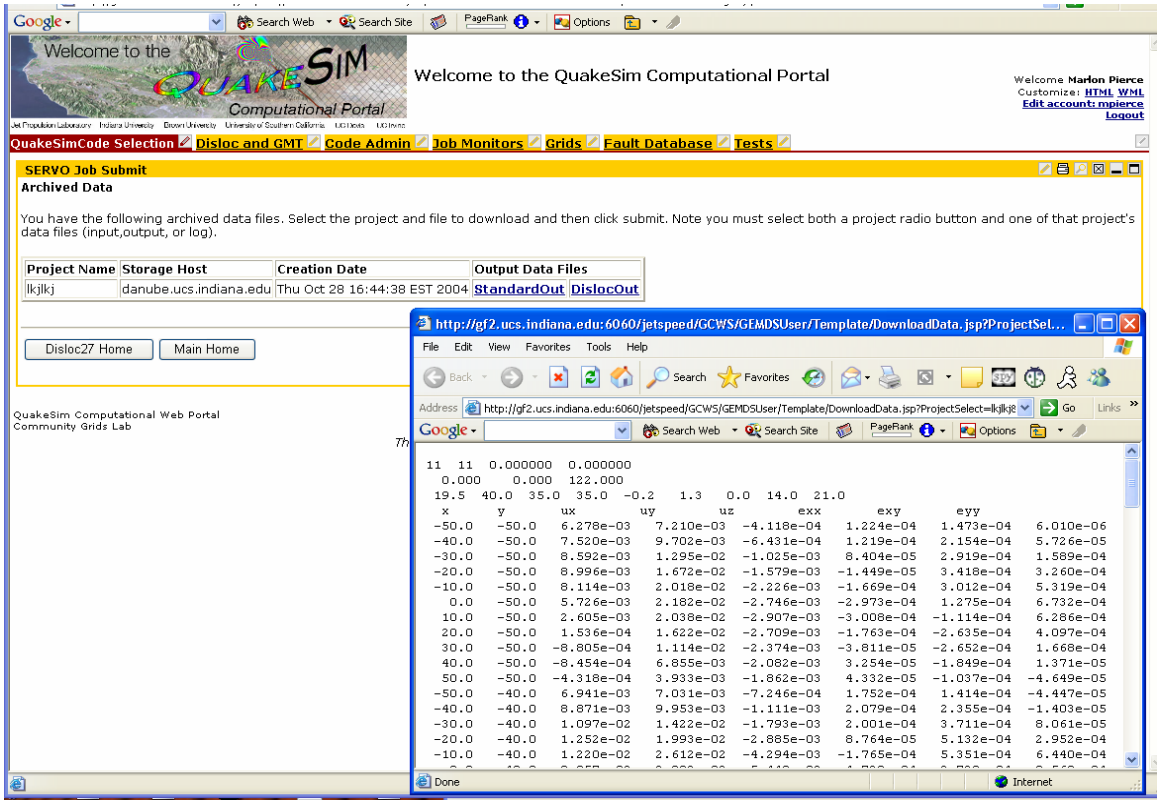


Figure 7. Templated download forms and sample download data.

Updating Applications

From the main administration menu shown in Figure 1, select the “Update” button to modify information about currently deployed applications. A sample display is shown in Figure 8.

The listings correspond to the application listings shown in the standard Code Selection menu (Figure 4). For each application, you may choose to

1. Update the information provided in Figure 3 for both the application and the available hosts.
2. Add a new host. This will add hosts to the list shown in the code selection menu (Figure 4).
3. Delete a host for the application. This will remove hosts from the application listings in the Code Selection menu (Figure 4).

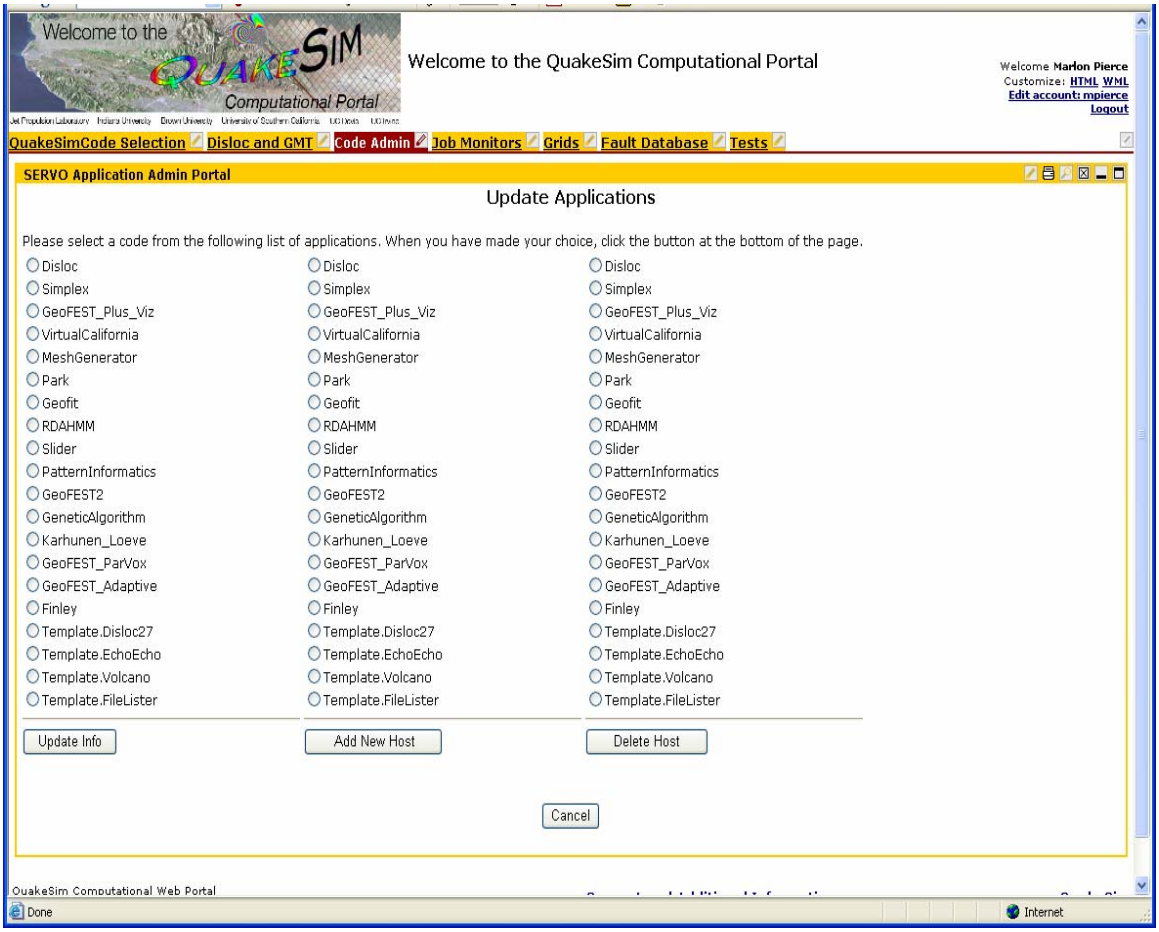


Figure 8. Updatable information selections.

Selecting one of the applications on the left-hand column and clicking the "Update Info" button under the column takes you to a form similar to Figure 9. You can update any of the information in these fields.

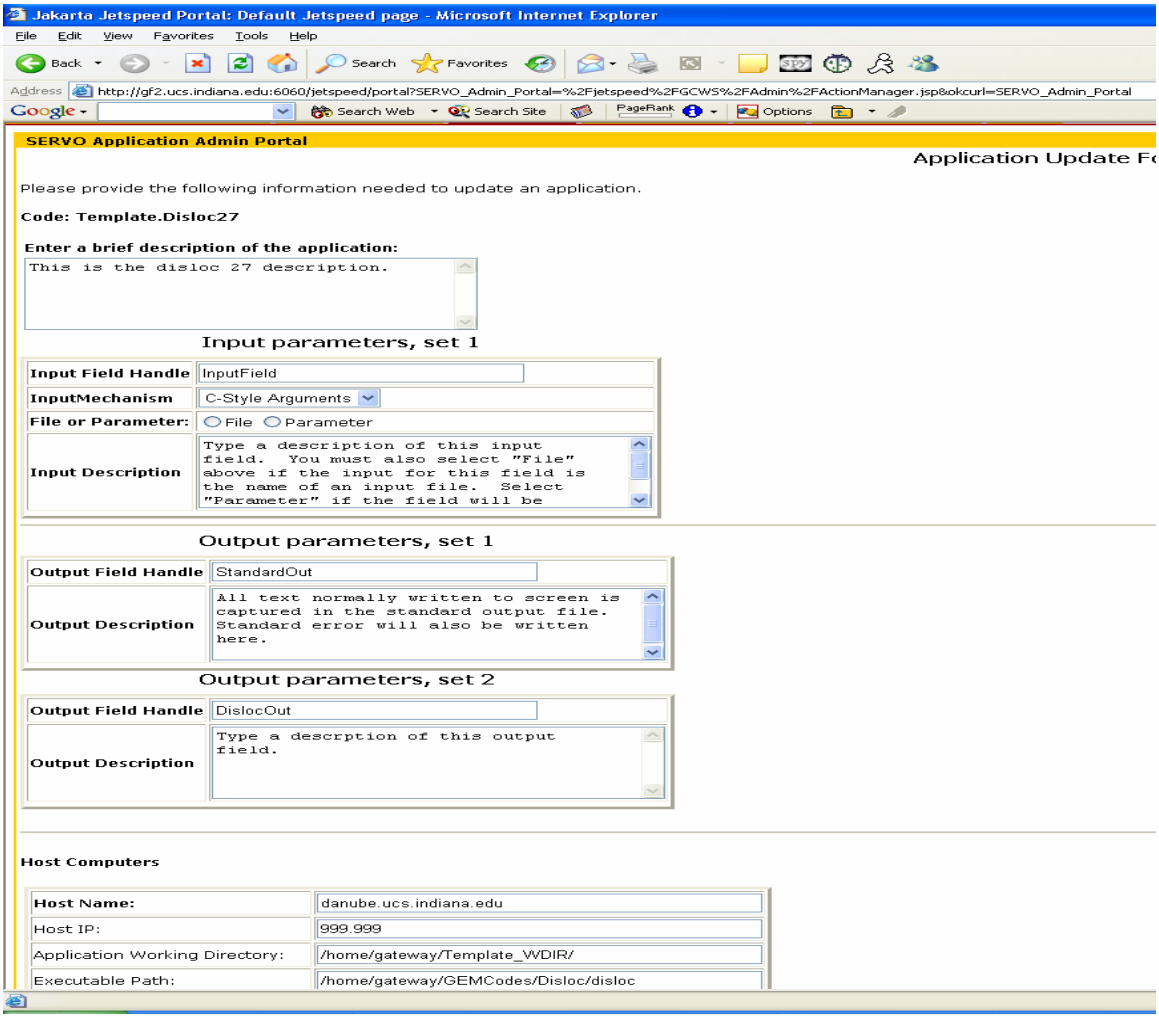


Figure 9. Application update form displays the input, output, and host information.

Selecting an entry from the middle column and clicking “Add New Host” at the bottom takes you to a form similar to Figure 10. This will show the current list of available hosts and a form for adding a new host. Add the required information and click update.

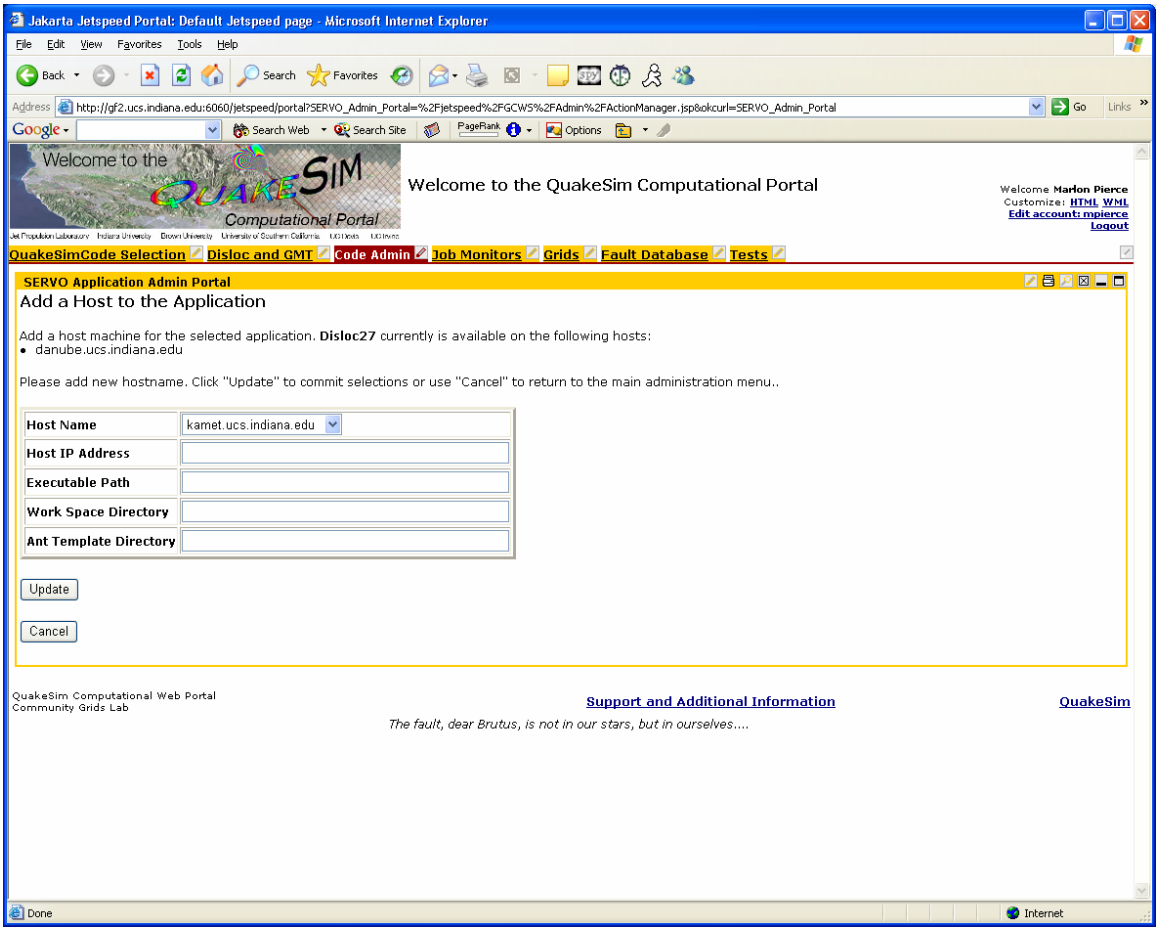


Figure 10. Forms for adding a host.

To delete a host from a selected application, choose it from the right most column (Figure 8) and click the “Delete Host” button underneath the column. This will take you to a page similar to Figure 11, which will show a list of currently deployed hosts.

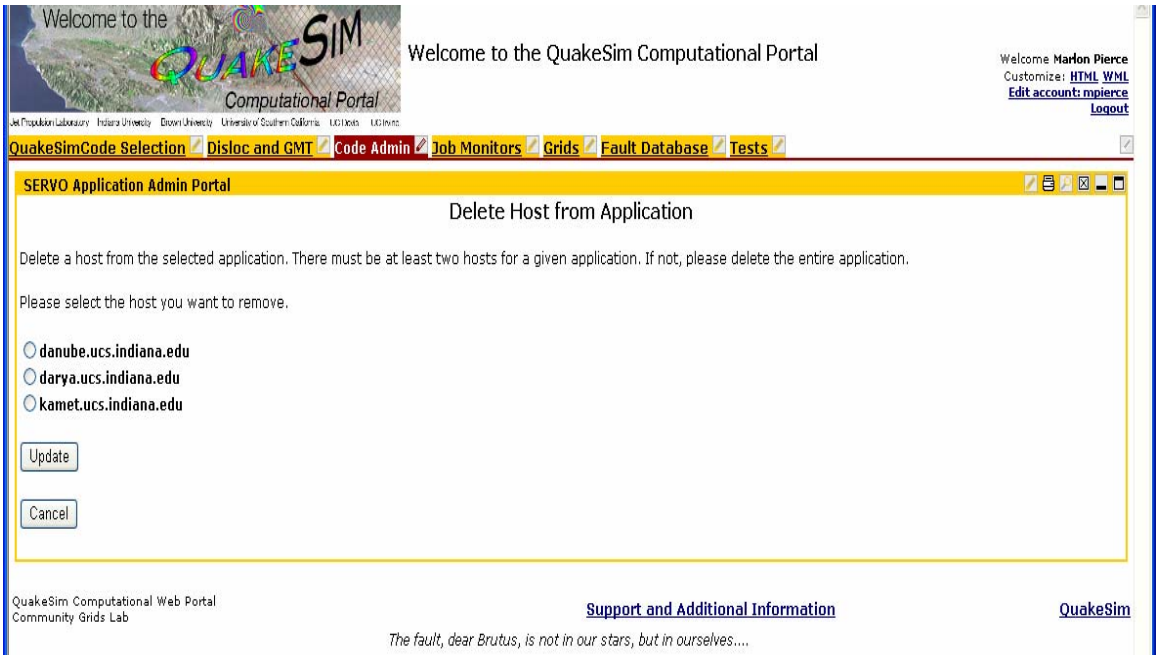


Figure 11. Sample page for deleting hosts for a particular application.

To delete a host, select the host and click “Update.” Applications are required to have at least one host.

Deleting Applications

Entire applications may be deleted using the Delete feature from the main administration menu, Figure 5. Choosing this button will forward you to a form similar to Figure 12. Select an application from the list by clicking it and then click the “Remove” button below the column. This will permanently remove the application from the Code Selection menu, Figure 4. It does not, however, delete the application from the host computer.

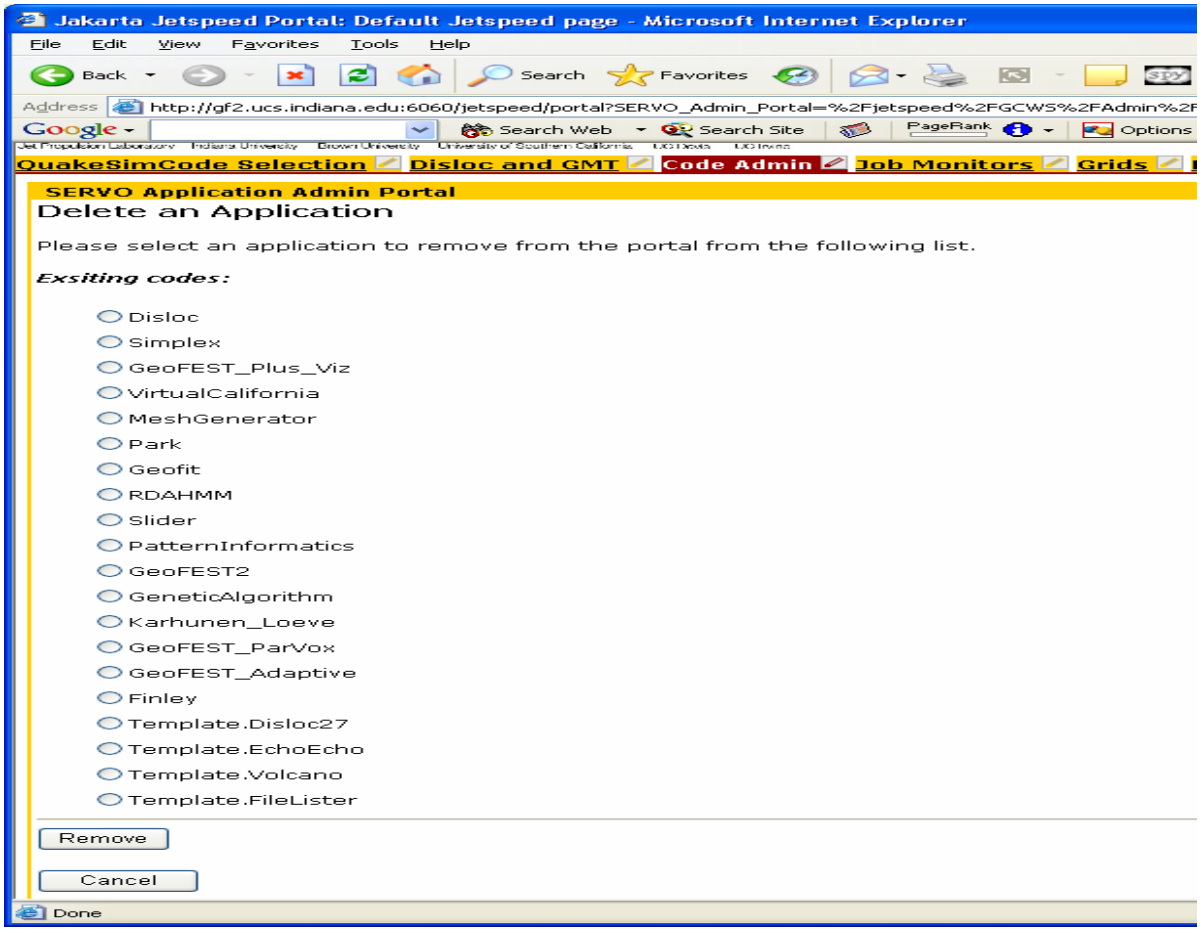


Figure 12. Selection options for deleting an application.

Example: Deploying Disloc

The Disloc application may be deployed on any of the example hosts. The location of the Disloc executable on all hosts in the test bed is

/home/gateway/GEMCodes/Disloc/disloc

Disloc is run from the command line by using the command

[shell>disloc <input.txt> <output.txt>

That is, you must specify an input file and an output file on the command line. Disloc under correct operation will generate no standard output but will create standard error.

To deploy this with the code wiki, login to a privileged account, click the “Code Admin” tab across the top, and fill out the form as described above.

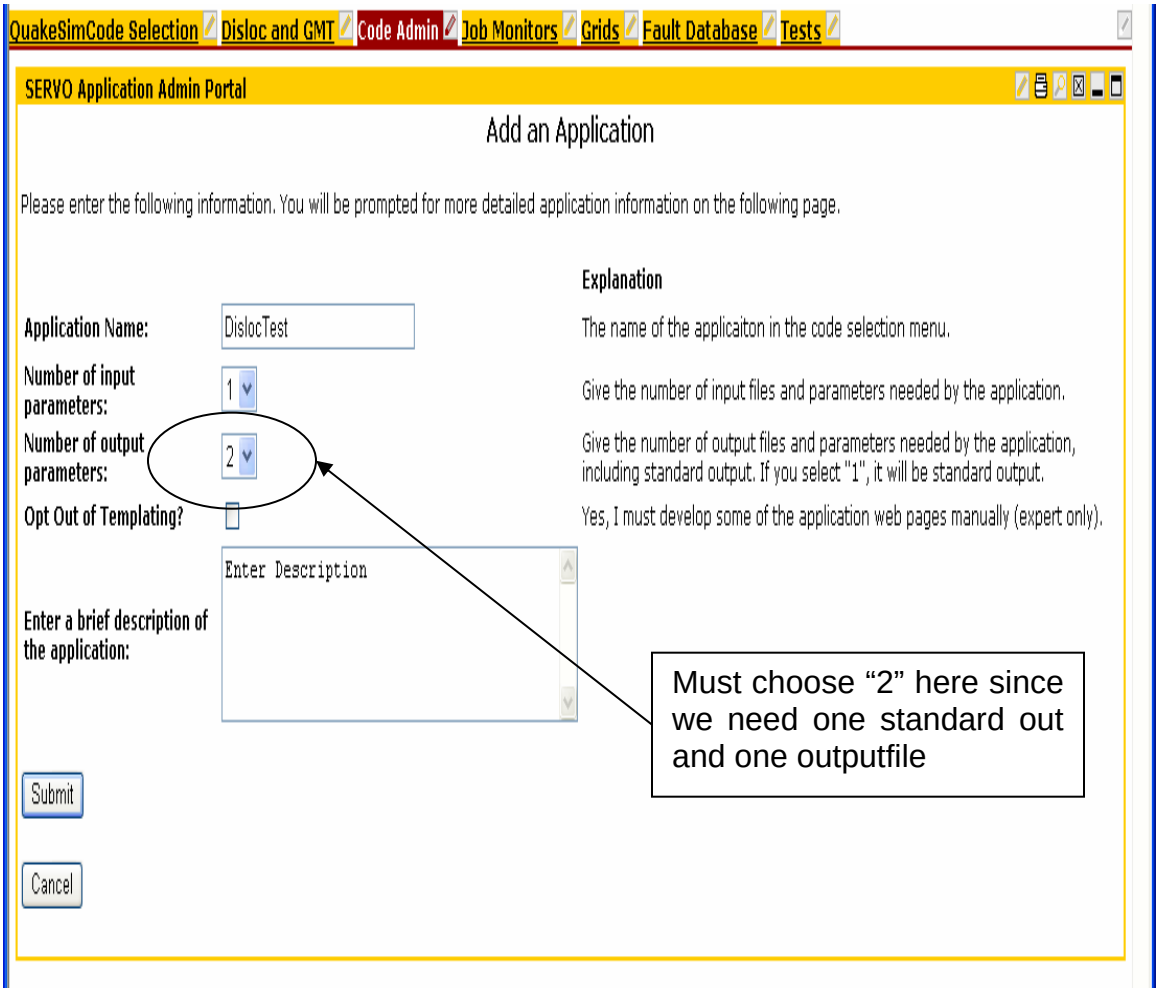


Figure 13. Adding DislocTest as an application

For Disloc, you must choose “2” for the “Number of output parameters” field, because Disloc generates both standard output/error (implicitly) as well as output to a file (explicitly).

Click the submit button on the form. You will then need to specify the input and output parameters. For Disloc to be deployed correctly, you must enter the correct information for “Input Parameters” and give the correct path for the “Executable Path” parameter in the Host form. See Figure 14.

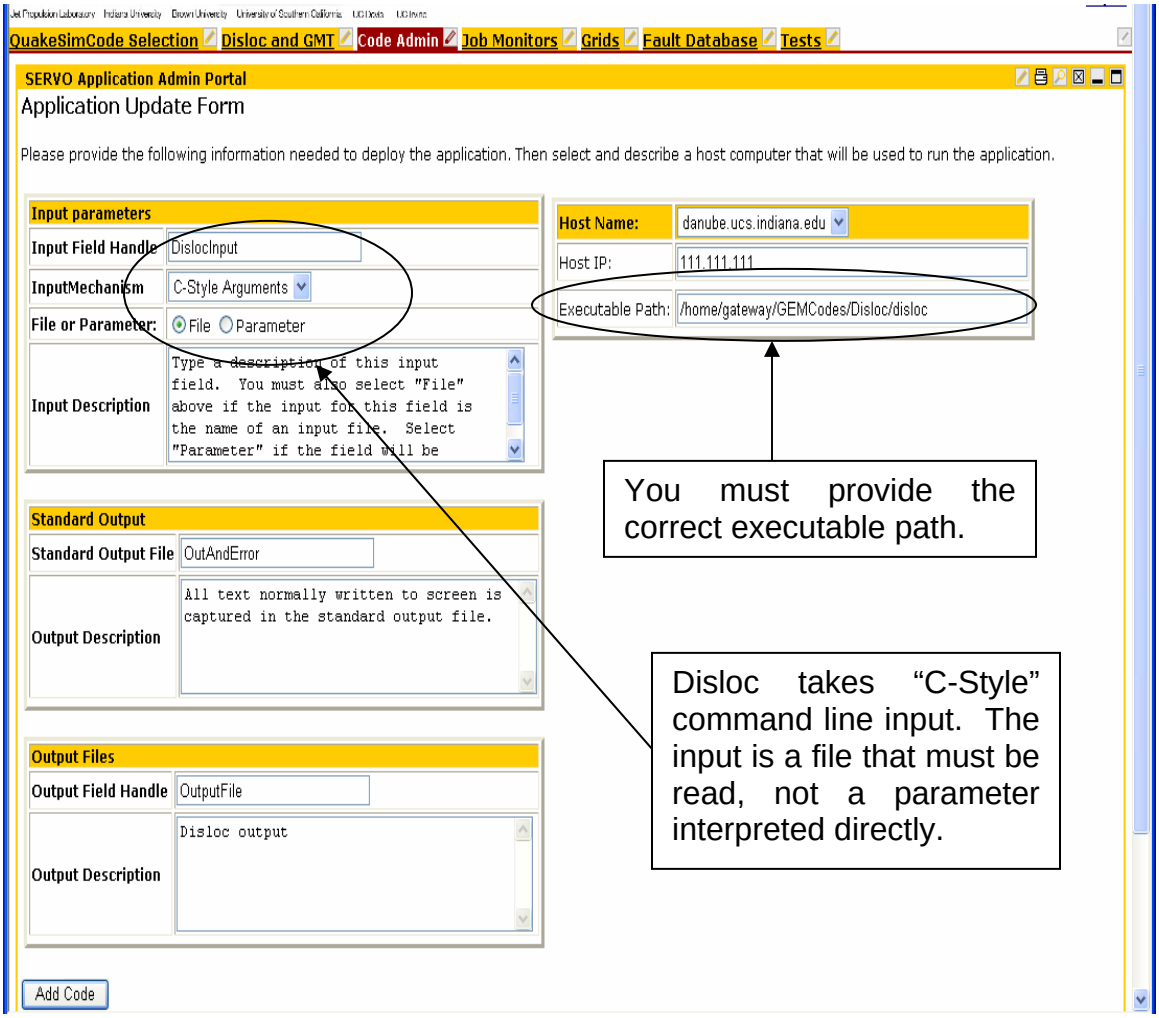


Figure 14. Specifying Disloc input and output.

As described in the introduction, Disloc expects an input file that is read as a "C-Style" command line argument. That is, the input field is not read from standard input and is not interpreted directly as a parameter value, but instead a file that contains parameter values.

You may now run the Disloc application and download the application as described in previous sections.

Example: Deploying Third Party Applications

Disloc is an example application of the Okada method (Ref. 1). Various implementations of this method exist, and it has been applied to a number of different problems. In order to test the application management interfaces, we obtained an alternative Okada application from Prof. Kristy Tiampo (University of Western Ontario, http://www.uwo.ca/earth/Tiampo.html) used to simulate

volcanic stresses. This application makes use of subroutines obtained directly from Prof. Okada.

Tiampo's code was successfully installed on all test bed resources. To run from the command line, use the following command:

[shell> /home/gateway/GEMCodes/Okada/volcanic_stress < input.data > output.data

The file mayon.dat is provided as a sample input file that can be uploaded to run the application.

To add this application, you should log in as a privileged user, select the "Code Admin" tab, and then choose "Add Application" as shown in Figure 1. For this particular application, input comes from Standard Input and output is directed to Standard Output, so you should use the settings shown in Figure 15.

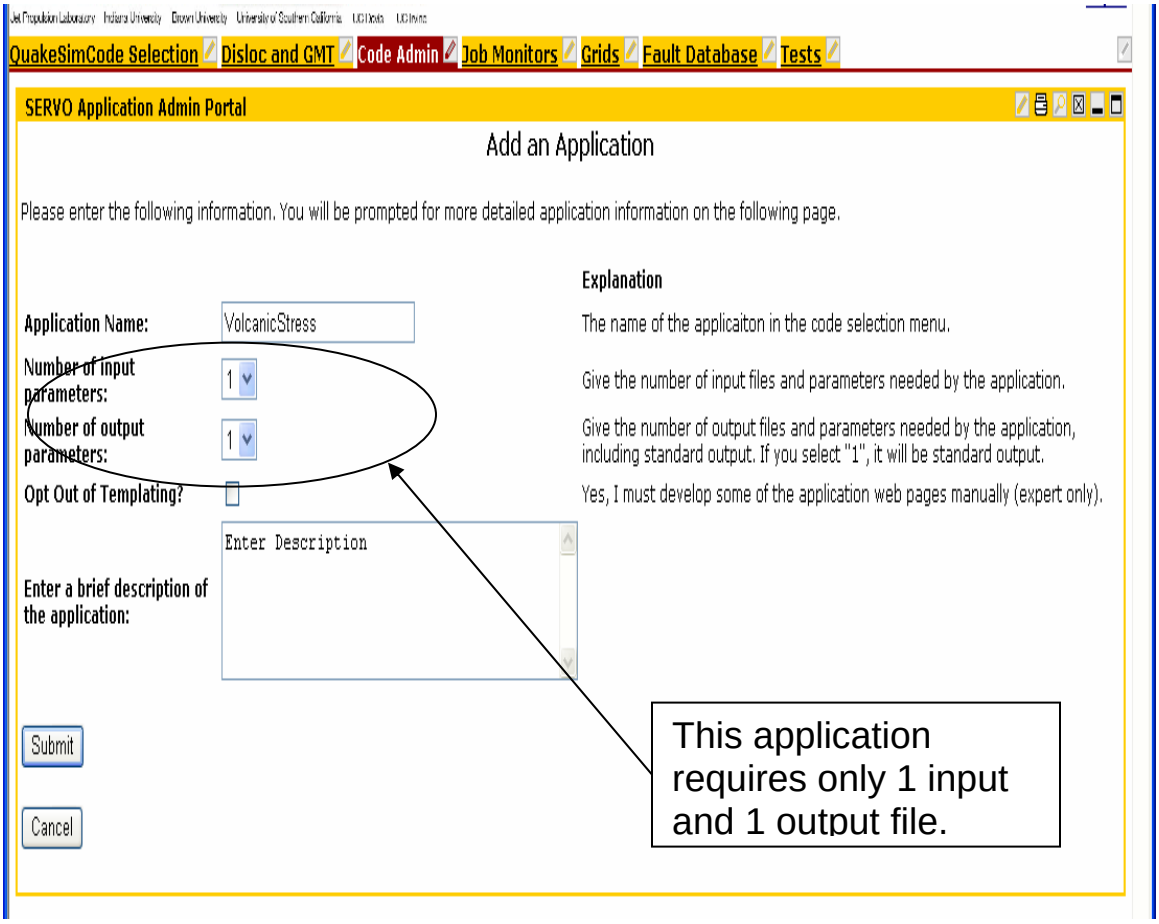


Figure 15. Specifying input for the Volcanic Stress application of the Okada method.

To deploy this correctly, you must specify that the application takes 1 input and 1 output. Note that the one output field is standard output, so code output and/or any error messages will be written to the same file.

Click submit to further specify the input fields, as shown in Figure 16. For the application to be deployed correctly, you must specify the correct input mechanism.

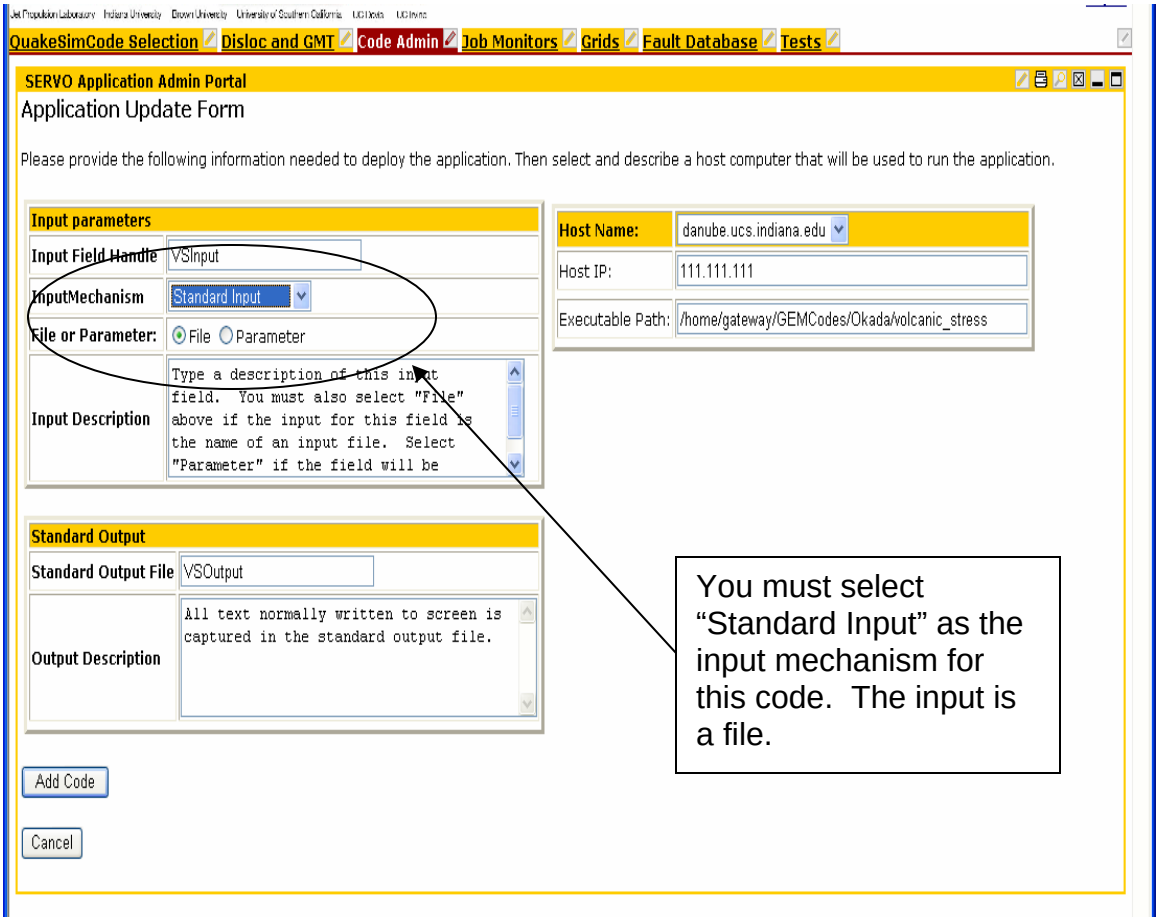


Figure 16. Specify the input fields and host.

The input mechanism for this application is "Standard Input" rather than "C-Style." The input field is a file of parameters rather than a parameter, so select this.

You may now add the code. A sample data file, mayon.dat, is provided to test the upload form, Figure 6.

Example: Running UNIX Commands

You may also deploy UNIX commands as applications. These run with the permission of the "gateway" user on testbed accounts. That is, they have the same permissions and restrictions as the user account that runs the Web Server.

The procedure is the same as the previous examples. We describe the "ls" command here since this illustrates the use of parameter input fields instead of file input files as shown previously.

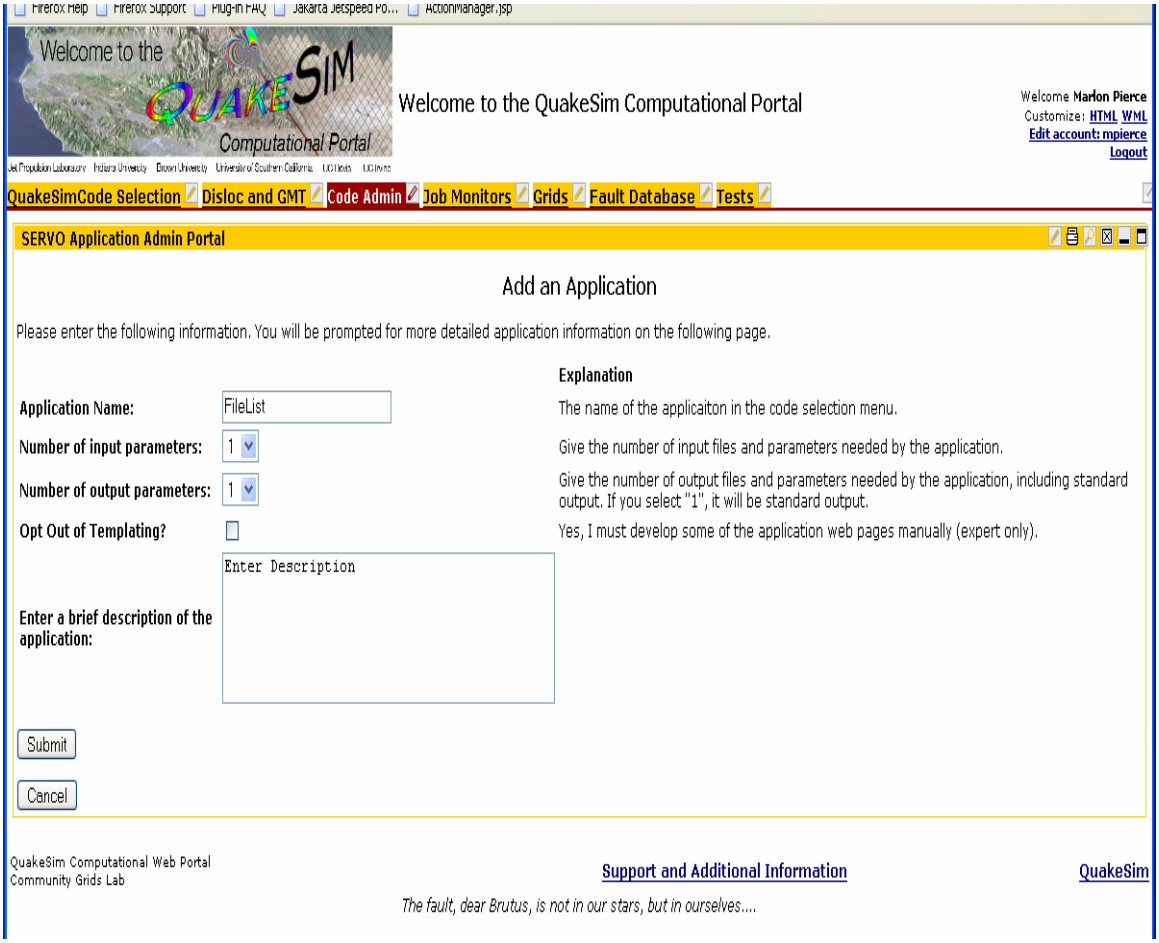


Figure 17. Specifying input and output fields when wrapping the UNIX ls command.

This is as before. The ls command takes one input field and generates one output field (standard out).

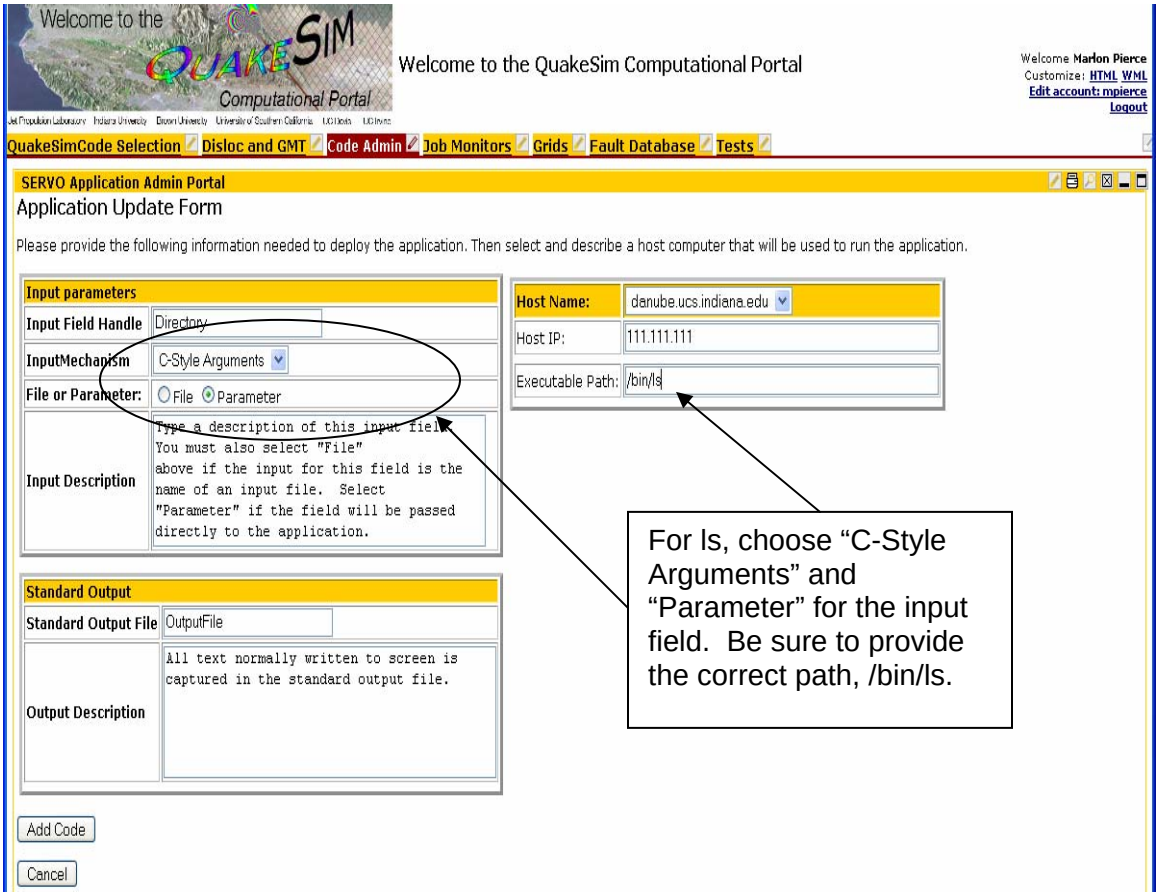


Figure 18. Describe the FileLister input and output fields.

The input field for /bin/ls is the directory or file that is to be listed. This is an input parameter, not a file, since it is interpreted directly.

After deployment, you can run the ls command on different directories and download the content through the download page, Figure 7.

Note that the “input parameter” can include multiple parts: the entire set of input fields is passed to the executable as a single argument. So valid input values for this parameter include “/tmp/” and “-ltr /tmp/”.

References

1. Okada,Y., 1992, Internal deformation due to shear and tensile faults in a half-space, Bull.Seism.Soc.Am., 82, 1018-1040

Issue testable 5-year earthquake forecast for M>5 for Southern California

One of the primary goals of the QuakeSim project was to develop the technology to issue 5-year and longer forecasts for earthquake activity. Furthermore, we developed software technology through the web services component of the QuakeSim portal to allow users to make forecasts using a grid computing approach to the problem. As a result, we optimized the process for producing the forecasts, and issued forecast maps through both the QuakeSim portal and through the web site: http://hirsute.cse.ucdavis.edu/~rundle/EQ_FORECASTS/ . Furthermore, we are presently finalizing two papers on earthquake forecasting, one using Virtual California that has been accepted in Proceedings of the National Academy of Sciences [1], and one on the Pattern Informatics (PI) method that is being finalized for submission to Nonlinear Processes in Geophysics. We have published other papers on these techniques as well.

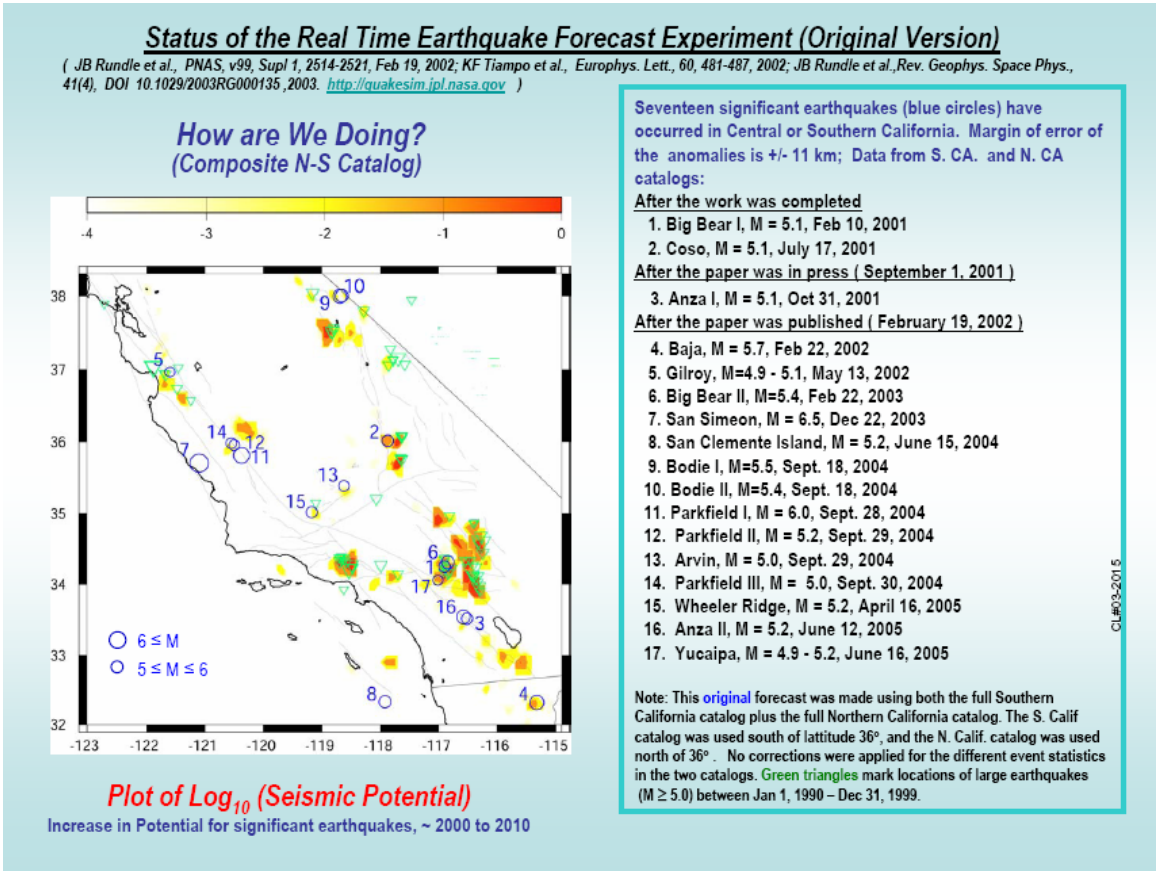


Figure 19. Earthquake forecast (and performance results) issued on the QuakeSim website at http://quakesim.jpl.nasa.gov/scorecard.html.

Specifically, PI method quantifies temporal variations in seismicity patterns. The result is a map of areas in a seismogenic region (hot spots) where earthquakes are likely to occur during a specified time window in the future. The method does not predict earthquakes, but it does forecast the regions (hot spots) where earthquakes are most likely to occur in the relatively near future (typically five to ten years). The objective is to reduce the areas of earthquake risk relative to those given by long-term hazard assessments. A forecast for California was published by our group in 2002 and reproduced on the QuakeSim web site. The period of validity was the time period January 1, 2000 - 2010.

More recently, the Relative Earthquake Likelihood group of the Southern California Earthquake Center has solicited 5-year forecasts of future activity to be posted as of September 1, 2005. The corresponding papers will be published in Seismological Research Letters. We plan to participate in this activity. Furthermore, our current forecast and methods will be reviewed by the California Earthquake Prediction Evaluation Council on September 20, 2005. This group includes representatives from the state and federal governments, as well as private organizations.

References:

[1] JB Rundle, PB Rundle, A Donnellan, D Turcotte, R Shcherbakov, P Li, BD Malamud, LB Grant, GC Fox, D McLeod, G Yakovlev, J Parker, W Klein, KF Tiampo, Statistical hazard analysis for great San Francisco earthquakes: Inferences from numerical simulations, Proc. Nat. Acad. Sci., in press (2005)

Publish the availability of the Portal to the Earthquake community in a peer reviewed periodical such as "Concurrency: Practice and Experience," or "EOS" or an AGU journal.

The availability of the QuakeSim Portal has been published twice in the peer-reviewed periodical *Computing in Science and Engineering*:

> Grant, L.B.; Gould, M.M.; Donnellan, A.; McLeod, D.; Chen, A.Y.-A.; Sang-Soo Sung; Pierce, M.; Fox, G.C.; Rundle, P.; "**A Web Services-Based Universal Approach to Heterogeneous Fault Databases**." *Computing in Science & Engineering* [see also IEEE Computational Science and Engineering] Volume 7, Issue 4, July-Aug. 2005 (51 – 57).

> Donnellan, A.; Rundle, J.; Ries, J.; Fox, G.; Pierce, M.; Parker, J.; Crippen, R.; DeJong, E.; Ben Chao; Weijia Kuang; McLeod, D.; Matu'ura, M.; Bloxham, J.; "**Illuminating Earth's Interior Through Advanced Computing**." *Computing in Science & Engineering* [see also IEEE Computational Science and Engineering] Volume 6, Issue 1, Jan-Feb 2004 (36 – 44).

NOTE: The research described in this report is managed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Appendix A

Jetspeed Access Control Mechanisms

Prepared by Marlon Pierce
Community Grids Lab

Overview

This document provides a step-by-step guide for defining new Jetspeed roles that can be associated with different users. These roles can be used to restrict access to certain portlets (and thus their content).

Jetspeed Users, Groups, and Roles

Jetspeed allows users to posses one or more roles and belong to one or more groups. We'll first review the basic concepts, which are used in Jetspeed default settings to restrict access to administration portlets. We'll then describe the process for adding new roles with restricted access. A bit of Jetspeed's design to support this reviewed in the Appendix.

The following screen shot gives a sample listing of users for a particular Jetspeed portal. This is just the standard User Browser available to the site administrator.

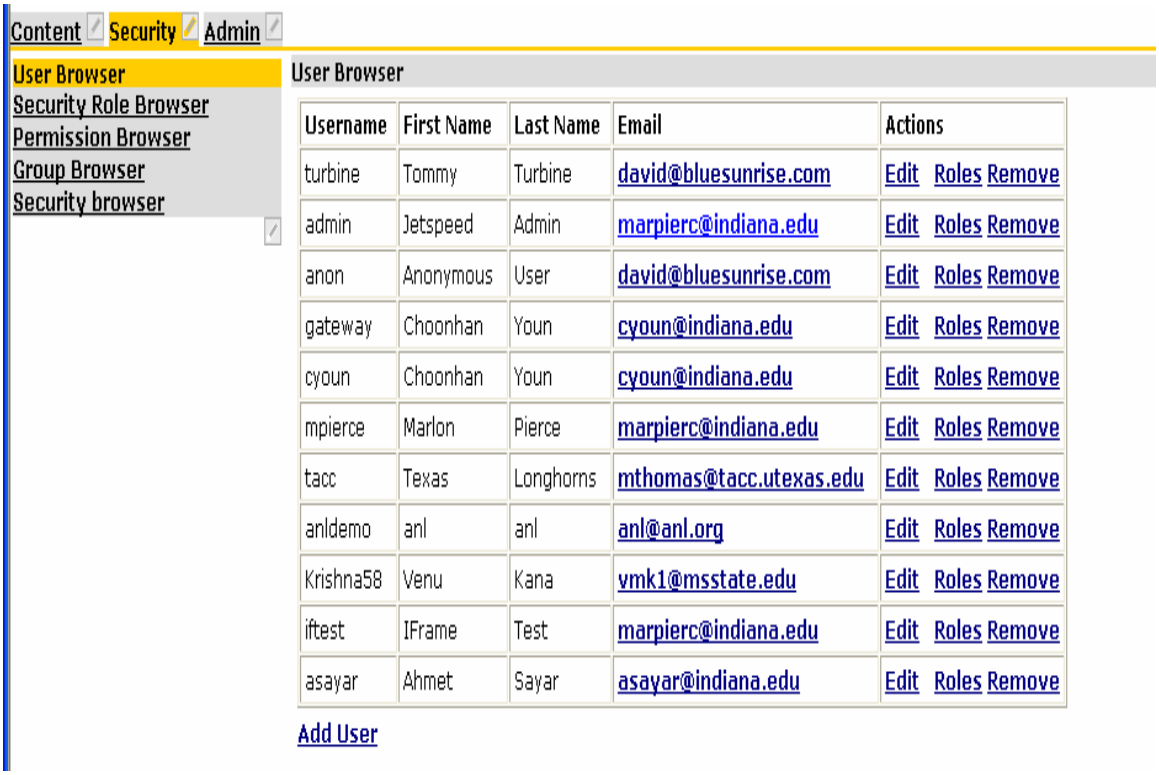


Figure 1. Jetspeed user browser displays a list of accounts.

Clicking the “Roles” link for the user mpierce gives the following screen, indicating that this user participates in the user role only. The user mpierce may be added to the admin role by simply clicking the link.

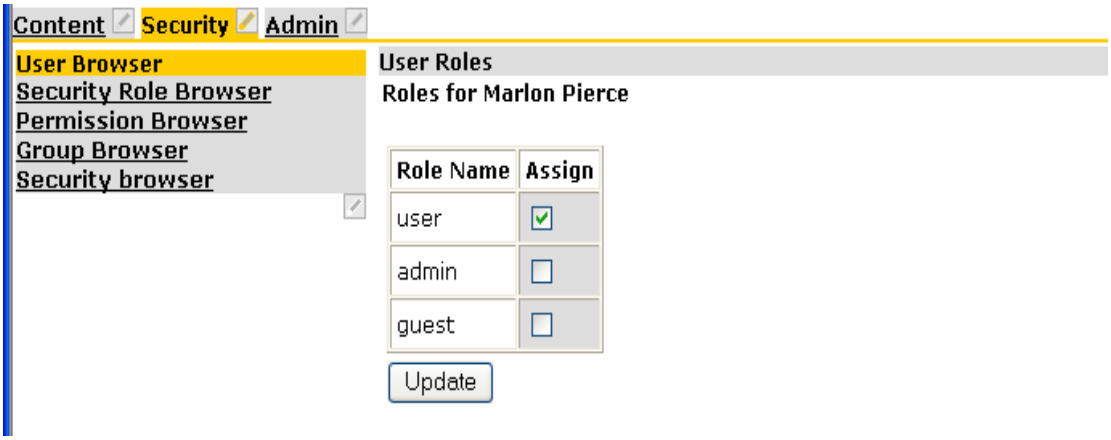


Figure 2. Default user roles.

The administration portlets are specified in the admin.xreg configuration file and may be examined for security examples. The sample entry below illustrates how the JavaRuntimePortlet is associated with the admin role and the admin-only security definition.

```
<portlet-entry name="JavaRuntimePortlet" hidden="true"
      type="instance" application="false">
      <security role="admin"/>
      <security-ref parent="admin-only"/>
      <meta-info>
         <title>Java Runtime Portlet</title>
      </meta-info>
      <classname>org.apache.jetspeed.portal.portlets.admin.JavaRuntimePortlet
      </classname>
      <media-type ref="html"/>
      <url cachedOnURL="true"/>
      <category group="Jetspeed">admin</category>
   </portlet-entry>
```

The important tags for the current discussion are in bold. These define the security role and the security-refs that are allowed to view this portlet. The security-refs are used to associate allowed (or prohibited) actions that are associated with a particular Jetspeed role. The definitions of role access privileges are stored in the configuration file security.xreg. For example, the entry for the admin-only security entry is

```
<security-entry name="admin-only">
      <meta-info>
         <title>Admin-only</title>
         <description>Full access to user with the admin role.</description>
      </meta-info>
      <access action="*">
         <allow-if role="admin"/>
      </access>
   </security-entry>
```

This means that the admin-only entry allows all access actions to the role "admin". The <allow-if role> tag can be repeated. For example, one could define a "monitor-only" security entry that would allow some management portlets to be viewed by both admins and a newly defined "monitor" role.

The full list of options that can be used to express a <security-entry> are given in the appendix. A list of Jetspeed actions is also given in the appendix.

Step-by-Step Guide for Restricted Areas

We will now take a specific example of how to specify restrictions to portlets so that they can only be viewed by a new role, manager. All steps must be done by a user with administrator privileges, so log in as admin first.

1. Create a new role for the manager. Go to the "Security Roles" link under the "Security" tab.

Figure 3 Create a new Jetspeed role through the admin interface.

Select "Add Role" and enter the new role name.

2. Shutdown Jetspeed so that you can edit registry entries. Add a security-entry for the manager role to the security.xml configuration file. This is

located in $JETSPEED_HOME/WEB-INF/conf/ directory. The following XML snippet can be used:

```
<security-entry name="manager-restricted">
   <meta-info>
      <title>Manager Restricted</title>
      <description>Restricted to those with manager roles.</description>
   </meta-info>
   <access action="*">
      <allow-if role="manager"/>
   </access>
</security-entry>
```

Figure 4. Create a new role.

3. Use the security role browser to add permissions for the manager role. You should add at least “View”.

Figure 5. Use the security role browser to view and modify role permissions.

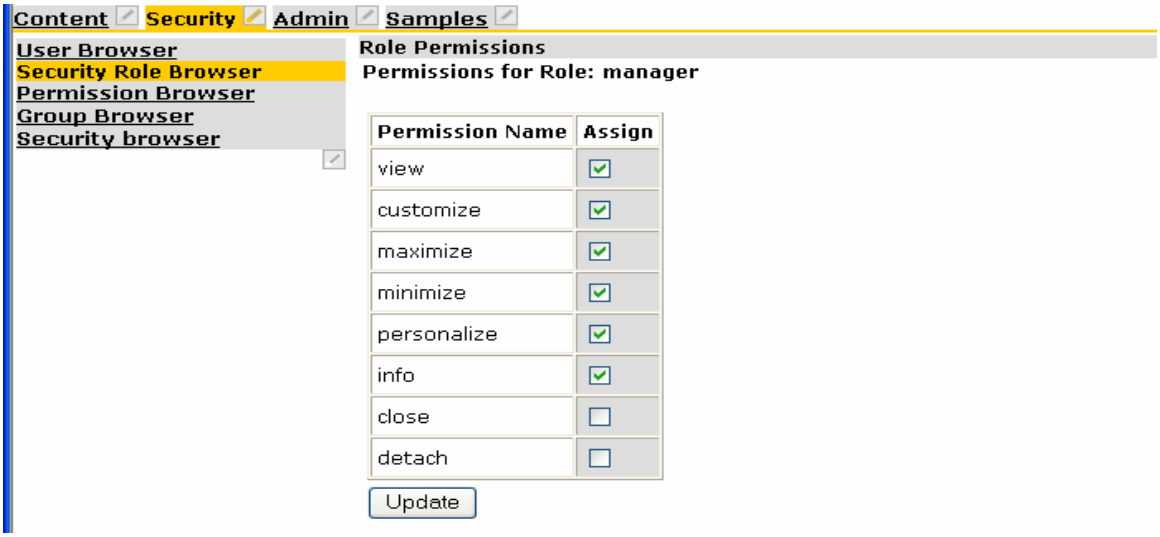
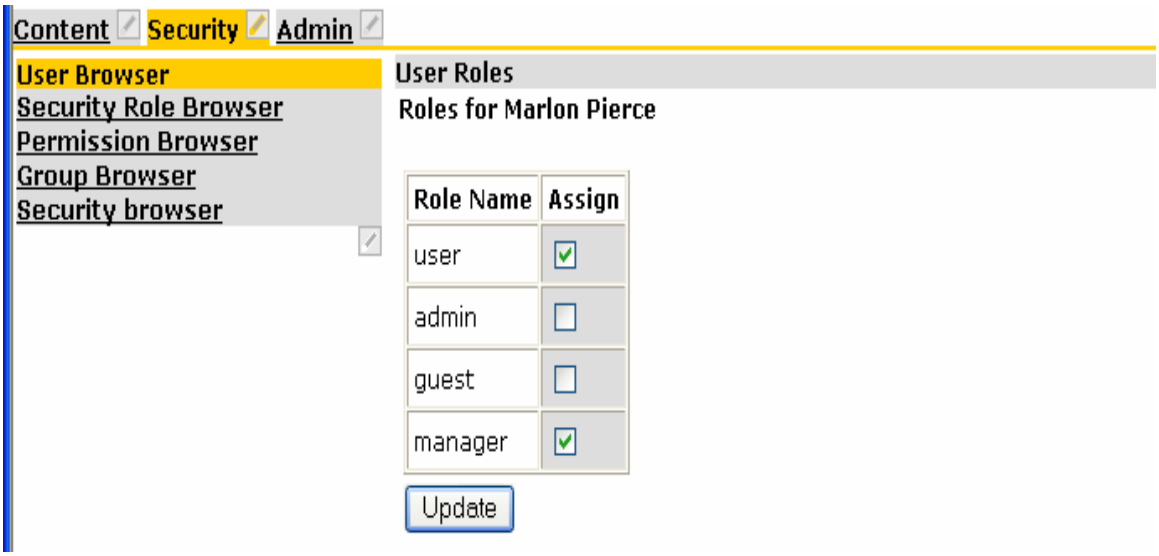


Figure 6. Permissions for the manager role.

4. Assign one or more users to the new role. Go to the User Browser and select the "Roles" link next to the user to be modified. See the screen shot below. Note that you will want to assign both user and manager roles to a regular user.



5. Add or modify an existing portlet instance entry so that it is only accessible by the manager role. For example, you may wish to collect all manager portlets in a single registry entry in $JETSPEED_HOME/WEB-INF/conf called manager-portlets.xreg with the following contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<registry>
   <portlet-entry name="ManagersOnly" hidden="false" type="ref"
      parent="WebPagePortlet" application="false">
```

```
        <security role="manager"/>
        <security-ref parent="manager-restricted”/>
        <meta-info>
           <title>Managers Only</title>
           <description>Manager Only</description>
        </meta-info>

<classname>org.apache.jetspeed.portal.portlets.WebPagePortlet</classname>
        <parameter name="dont_remove_applet" value="yes" hidden="false"
           cachedOnName="true" cachedOnValue="true"/>
        <url cachedOnURL="true">http://sportsillustrated.cnn.com/</url>
    </portlet-entry>
</registry>
```

You should of course do this while the server is stopped and restart it after editing.

6. Restart everything and log in as the new user with the manager role. Customize and add the restricted portlet “Managers Only” to your view. If you do not see the portlet in the portlet list, this has possibly been caused by a typo in one of the registry files that you edited.

Also log in as any other user without manager permissions. You should not see the manager-restricted portlet in the selection choices.

Appendix

Jetspeed Security Overview

Jetspeed follows the interface/implementation pattern for its security framework, allowing custom implementations to be added in a well defined way. These interfaces express generically various security concepts (authentication and access control) and define an interface “contract” that must be used by different security implementations.

The following sets of interfaces describe services that need to be implemented (Jetspeed provides default implementations).

Interface	Description
PortalAuthentication	Defines the contract between the portal and security provider required for authentication a Jetspeed User.
PortalAccessController	Defines the contract between the portal and security provider required for authorized access control for priveleged portal actions.
UserManagement	Defines the contract between the portal and security provider required for managing users.
RoleManagement	Defines the contract between the portal and security provider required for managing roles.
GroupManagement	Defines the contract between the portal and security provider required for managing groups.
PermissionManagement	Defines the contract between the portal and security provider required for managing permissions.
CredentialsManagement	Defines the contract between the portal and security provider required for managing credentials.

These service interfaces interact with the object model classes listed below. The object model classes abstractly define various “data objects” which possess specific parameters.

Interface	Description
JetspeedUser	Defines the minimal attributes of a user in the portal system.
Role	Defines the minimal attributes of a role in the portal system.
Group	Defines the minimal attributes of a group in the portal system.
Permission	Defines the minimal attributes of a permission in the portal system.

All service component and object model interfaces come with standard implementations and can be overridden by adding a new service implementation to the JetspeedSecurity.properties file located in $JETSPEED_HOME/conf.

Security Ref Schema Attributes and Elements

The following tables show attributes and elements that may be used to define a security ref entry (given in the security.xml configuration file).

Security Ref Base Attributes	
Attribute	**Description**
name	Required. The unique name of the security reference.

Security Ref Base Elements	
Element	**Description**

access	0..n access elements may be defined. If no element is specified, then everyone is denied access.

Access Attributes	
Attribute	**Description**
action	Defines the action which we are controlling access to. '*' represents all actions.

Access Elements	
Element	**Description**
allow-if	Defines which role will be granted access for the associated action. If not defined, access is denied to everyone for the associated action.
allow-if-owner	Defines the owner granted access for the associated action.

Allow-if Attributes	
Attribute	**Description**
role	Defines the security role required for this security constraint.

Jetspeed Actions

The following table lists Jetspeed access actions. All except view are associated with visible portlet icons.

Access Actions		
Icon	**Action**	**Description**
N/A	view	Allows to select a portlet in customizer and view its contents
	customize	Allows to customize a portlet once selected in profile
	info	Allows to view any additional information about a portlet
	maximize	Allows to view portlet in full screen mode
	minimize	Allows to minimize portlet (hide its content) and display its caption only
	close	Allows to temporarily close a portlet (hide its caption and content)
	print	Allows to display current portlet in "print friendly format" (without navigation and portlet control). Note that the default screen template/layout used may be overriden by setting `action.print.template` property in jr.props to your custom screen template.