ARC Job Submission Tool Documentation

Release 0.3

Jonas Lindemann

June 15, 2010

CONTENTS

1 Introduction 3

2 User interface 5
2.1 Main Window 5
2.2 Changing working directory 5
2.3 Client settings 5
2.4 Default services 7

3 Certificate Management 11
3.1 Creating a proxy certificate 11
3.2 Displaying proxy information 12

4 Job definitions 13
4.1 Creating a new job definition 13
4.2 Modifying an existing job definition 14
4.3 Submitting a job definition 15
4.4 Viewing result files 15
4.5 Cleaning result files 15
4.6 Removing a job definition 15
4.7 Viewing XRSL of a job definition 16

5 Managing running jobs 17
5.1 Selecting active jobs 17
5.2 Refreshing job status 18
5.3 Download selected jobs 18
5.4 Download all finished jobs 18
5.5 Killing jobs 18
5.6 Cleaning jobs 18
5.7 Removing unknown jobs 18

6 Script job plugin 19
6.1 Input files 19
6.2 Parameter sweep settings 20

7 Numpy job plugin 23
7.1 Input files 23
7.2 Parameter sweep settings 24

8 Octave job plugin 27
8.1 Input files 27

8.2 Parameter sweep settings 28

9 Xrsl job plugin **31**

9.1 Describing the job in XRSL 31
9.2 The %(taskdir)s directive 32
9.3 Input files 32
9.4 Parameter sweep settings 33

10 Indices and tables **35**

Contents:

CHAPTER

ONE

INTRODUCTION

The ARC Job Submission Tool (arcjobtool) is a standalone graphical user interface for simple job submission and job management. The user interface uses the python{wxPython} graphical user interface library, which is a Python binding for the wxPython library. The benefits of using wxPython is that the finished application can be run on all available platforms, such as Linux, Mac OS X and Microsoft Windows. In addition to being platform independent, wxPython also adapts the appearance of the user interface to the target platform, so that when the application is run on Windows it will look like a native Windows application.

CHAPTER

TWO

USER INTERFACE

The design of the arcjobtool user interface is based on a tabbed window design. Currently there are tabs for **Job Definition**, **Active Jobs** and **Downloaded Jobs**. In the job definition tab the job definitions can be created and managed. The **Active Jobs** tab shows a view of the currently running grid jobs and enables the users to manage and download finished jobs.

The arcjobtool application attempts to implement a fully non-blocking user interface, no operation will lock the user interface. This is especially important when dealing with Grid operations, that can take a while to process.

2.1 Main Window

The main window is divided into 4 parts, menu, workarea, log output and status area. From the menu all the functions in the user interface are available. In the work area the user can switch between the main operating modes of arjobtool using the tabs. The log output displays log messages mainly from the ARC library, but also messages from arcjobtool. In the status area, status information is displayed from long running tasks. The sizes of the the work area and log output area can be changed by dragging the splitter bar up and down.

2.2 Changing working directory

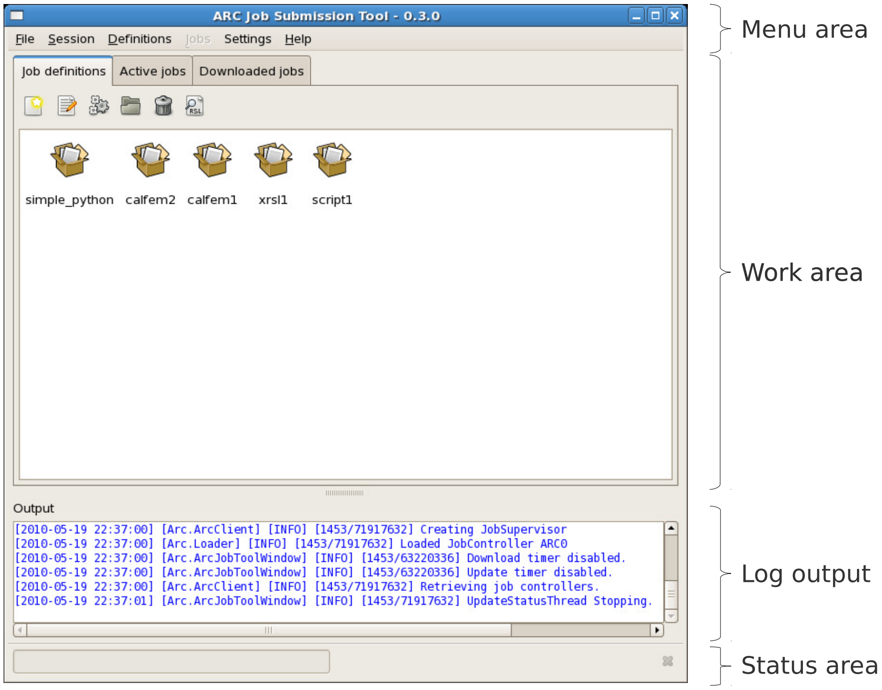
The arcjobtool stores and reads job definitions from the directory it is started. To change the arcjobtool's working directory select *File* → *Change working directory...* from the *File* menu. This will re-read and show any job definitions in this directory.

2.3 Client settings

Client settinsg controls the user specific options in arcjobtool. The settings dialog can be show by selecting *Settings* → *Client...*. The dialog is divied in 3 sections, **General**, **Processing** and **Certificates**. The tabs are described in more detail in the following sections.

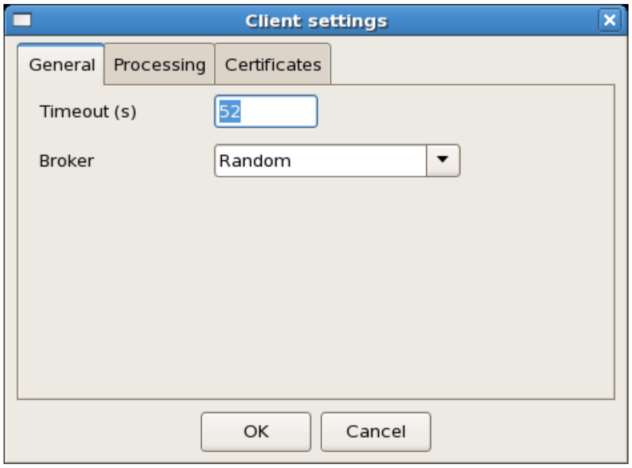
2.3.1 General

In the **General** tab the client timeout and default broker can be specified. The client timeout value determines the amount of time the ARC client should wait on any given resource before giving up. To shorten the submission time the timeout value can be reduced.



```
[2010-05-19 22:37:00] [Arc.ArcClient] [INFO] [1453/71917632] Creating JobSupervisor
[2010-05-19 22:37:00] [Arc.Loader] [INFO] [1453/71917632] Loaded JobController ARC0
[2010-05-19 22:37:00] [Arc.ArcJobToolWindow] [INFO] [1453/63220336] Download timer disabled.
[2010-05-19 22:37:00] [Arc.ArcJobToolWindow] [INFO] [1453/63220336] Update timer disabled.
[2010-05-19 22:37:00] [Arc.ArcClient] [INFO] [1453/71917632] Retrieving job controllers.
[2010-05-19 22:37:01] [Arc.ArcJobToolWindow] [INFO] [1453/71917632] UpdateStatusThread Stopping.
```

Figure 2.1: ARC Job Submission Tool main window



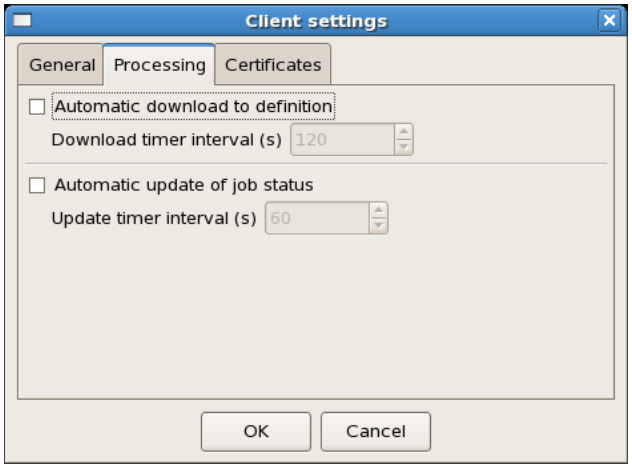
Timeout (s) 52

Broker Random

Figure 2.2: General client settings

The broker setting determines which brokering algorithm that should be used when submitting jobs. The “Random” broker is set as the default broker for the arcjobtool. Random chooses randomly between available submission targets. For more options see the ARC 1.1.x documentation.

2.3.2 Processing



□ Automatic download to definition

Download timer interval (s) 120

□ Automatic update of job status

Update timer interval (s) 60

Figure 2.3: Client processing settings

The settings in the **Processing** tab controls the automatic processing functions in arcjobtool, automatic download and automatic update of job status.

The checkbox **Automatic download to definition** determines if arcjobtool should check for finished jobs at given intervals and initiate download threads for these. The **Download timer interval** control determines at what intervals the check should be performed.

The checkbox **Automatic update of job status** determines if arjobtool should automatically update the status of active jobs. The **Update timer interval** control determines at what intervals the job status should be updates.

> **Warning:** Don’t set too short intervals in the processing settings as this can produce a lot of network traffic.

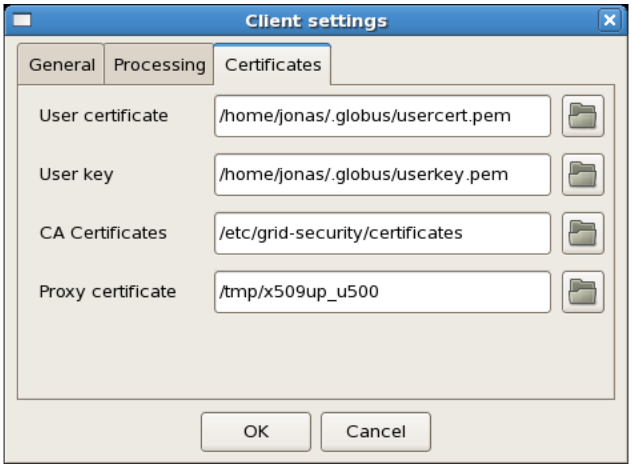
2.3.3 Certificates

The **Certificates** tab configures the different directories that are used by the ARC client library. To change a settings enter the directory manually in the text box or use the directory selection box next to the text box.

Note: The certificate directories shouldn’t be changed unless it is really needed.

2.4 Default services

The default services dialog controls what services should be used when querying for resources on the grid. The dialog is divided in 2 parts **Selected services**, for the services that will be used and **Rejected services** for those services that should _not_ be includes in the resource discovery.



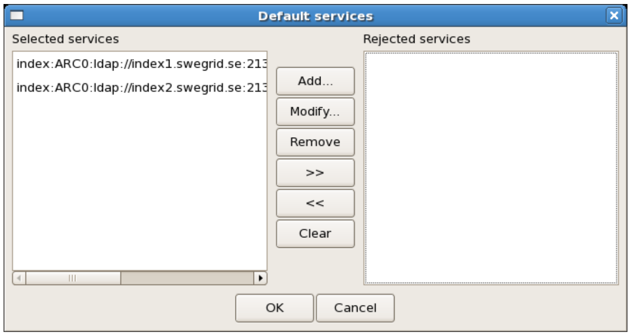
User certificate /home/jonas/.globus/usercert.pem

User key /home/jonas/.globus/userkey.pem

CA Certificates /etc/grid-security/certificates

Proxy certificate /tmp/x509up_u500

Figure 2.4: General certificate settings



Selected services

index:ARC0:ldap://index1.swegrid.se:213

index:ARC0:ldap://index2.swegrid.se:213

Rejected services

Figure 2.5: Default services dialog

2.4.1 Adding a service

To add a selected service click on the **Add...** button. This asks for the url for the service to be added. A service url should be given with the following syntax:

```
<service_type>:<flavour>:<service_url>
```

where *service_type* should be either *index* or *computing*. The flavour specifies what middleware plugin to use when contacting the service (e.g. ARC0, ARC1, CREAM, UNICORE, etc.). *service_url* is the url used when contacting the service.

2.4.2 Modifying an existing service

An existing service can be modified by selecting it in either lists and clicking on **Modify...**. Modify the url and click **OK** or **Cancel** when done.

2.4.3 Removing a service

A service can be removed by selecting in it in either lists and clicking on **Remove**.

Note: If a service was removed accidentally, cancel the entire dialog by clicking on **Cancel**.

CHAPTER

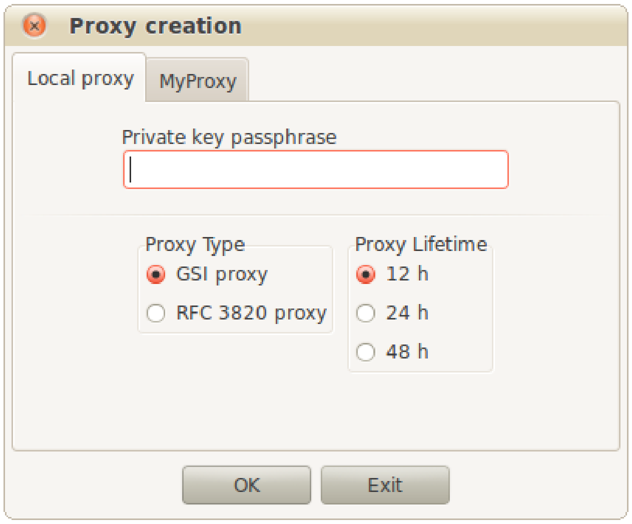
THREE

CERTIFICATE MANAGEMENT

The arcjobtool contains basic proxy certificate tools for viewing certificate and proxy information as well as the ability to create proxy certificate. The following sections describe these functions.

3.1 Creating a proxy certificate

A proxy certificate can be created in arcjobtool by selecting *Session* → *Create proxy certificate...*. This will bring up the following dialog:



Proxy creation

Local proxy | MyProxy

Private key passphrase

Proxy Type
● GSI proxy
○ RFC 3820 proxy

Proxy Lifetime
● 12 h
○ 24 h
○ 48 h

Figure 3.1: Creating a proxy certificate

arcjobtool supports the creation of standard GSI or RFC 3820 proxies. The standard GSI proxies must be used when submitting jobs to ARC0 resources.

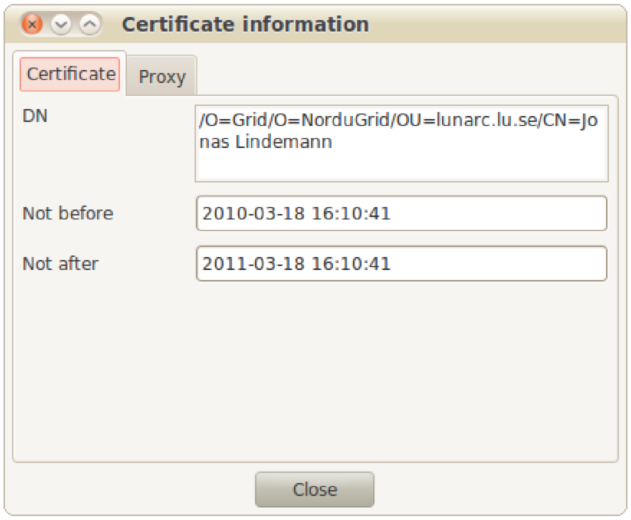
Note: arcjobtools does not currently support the creation of VOMS proxies, but this will be added in future versions of arcjobtools. However, arcjobtool can use existing proxies created with other tools.

The lifetime of the proxy can be selected in the **Proxy lifetime** radio buttons.

To generate a proxy, enter the user certificate passphrase in the **Private key passphrase** text box and click **OK**.

3.2 Displaying proxy information

To show information on used certificate and proxy, click *Session* → *Cert/proxy infor...* in the menu. The dialog has 2 tabs, the first, **Certificate**, displaying certificate information and the second, **Proxy**, displaying information on the proxy certificate.

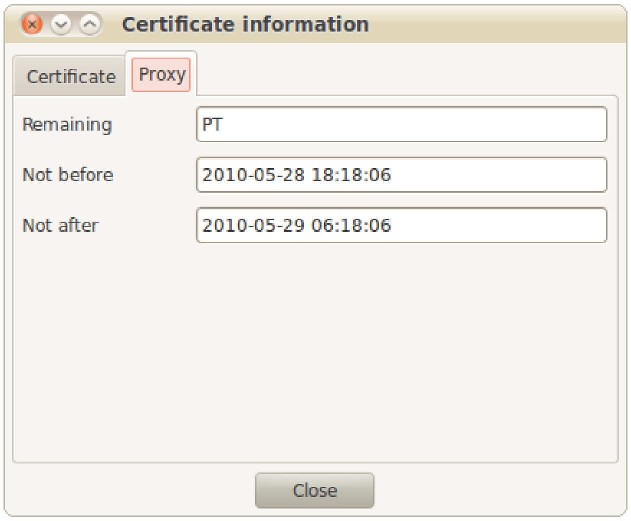


DN /O=Grid/O=NorduGrid/OU=lunarc.lu.se/CN=Jonas Lindemann

Not before 2010-03-18 16:10:41

Not after 2011-03-18 16:10:41

Figure 3.2: Information dialog displaying certificate information.



Remaining PT

Not before 2010-05-28 18:18:06

Not after 2010-05-29 06:18:06

Figure 3.3: Information dialog displaying proxy information.

CHAPTER

FOUR

JOB DEFINITIONS

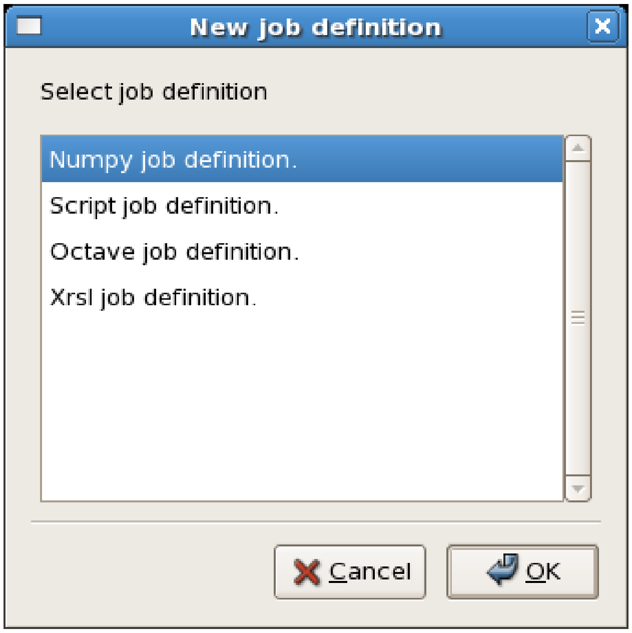
arcjobtool uses the concept of job definitions for job management. A job definition is a directory with the extension `.arcdef` containing the necessary scripts and settings to submit a job. Job definitions are maintained by special plugins installed with the arcjobtool application. The plugin provides both the logic and user interface to manage the job definition. The correct plugin for a given job definition will be automatically loaded by arjobtool.

Results produces by downloaded jobs are downloaded in the job definition directory, so that an entire job definition can be tarred and moved to a different including the result directories.

When arcjobtool is started it scans the current working directory for job definition directories and show these in the **Job definitions** tab. Each job definition is shown with a package icon.

4.1 Creating a new job definition

A new job definition is created by selecting *Definition* → *New...* or using the keyboard shortcut `Control-N`. This will bring up a dialog asking for the type of job definition to create.



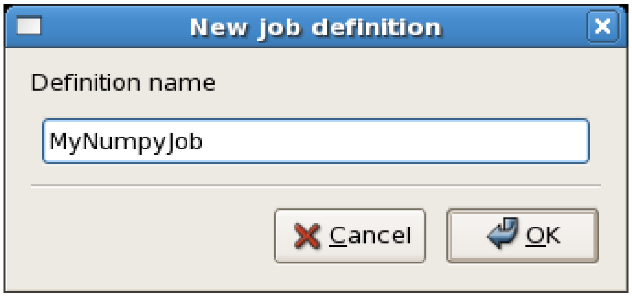
New job definition

Select job definition

- Numpy job definition.
- Script job definition.
- Octave job definition.
- Xrsl job definition.

Figure 4.1: Creating a new job definition

Next, a message box will ask for the name of the job definition to create. Enter a unique name and click **OK**.



New job definition

Definition name

MyNumpyJob

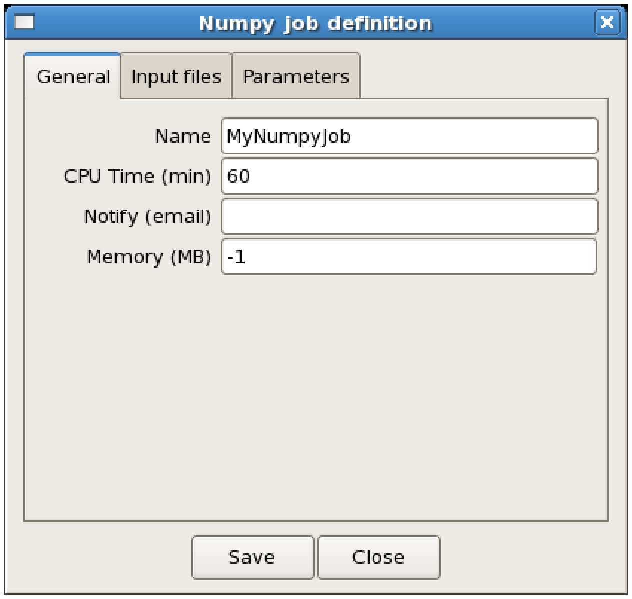
Figure 4.2: Setting the name of the new definition

In the **Job definition** tab an icon representing the newly created definition is shown.

4.2 Modifying an existing job definition

The settings of a job definition can be edited by double clicking on the icon, selecting *Definition* → *Modify...* or using the shortcut key `Control-E`. This will display user interface provided by the plugin maintaining the selected job definition.

The user interface of arcjobtool plugins often consist of one or more tabs. In the first tab, **General**, general job settings such as job name, CPU-time, notification email and memory requirements can be set.



Numpy job definition

Name MyNumpyJob
CPU Time (min) 60
Notify (email)
Memory (MB) -1

Figure 4.3: General job definition settings.

Changed settings are stored in the job definition when the user interface is closed and the **OK** button is clicked.

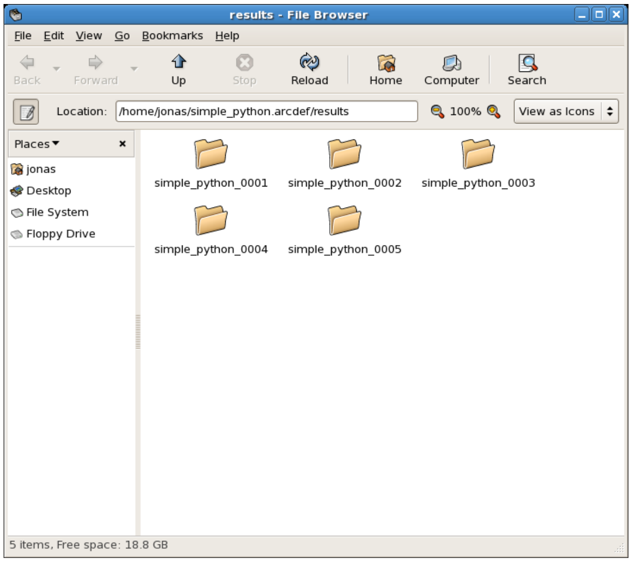
For more details on the job definition user interfaces please see the documentation for the provided plugin.

4.3 Submitting a job definition

A job definition is submitted by selecting it in the **Job definition** tab and selecting *Definition* → *Submit...* or using the shortcut `F7`.

4.4 Viewing result files

To view downloaded results file for a given job definition select *Definition* → *View result files*. This will open a file browser in the correct directory of the job definition.



Location: /home/jonas/simple_python.arcdef/results

100%

View as Icons

5 items, Free space: 18.8 GB

Figure 4.4: Job output files displayed in a file browser.

4.5 Cleaning result files

To clean all files in the result directory, select *Definition* → *Clean results*. This will as for confirmation an then delete all result directories.

Warning: There is no undo for this operation. All result directories will be permanently deleted.

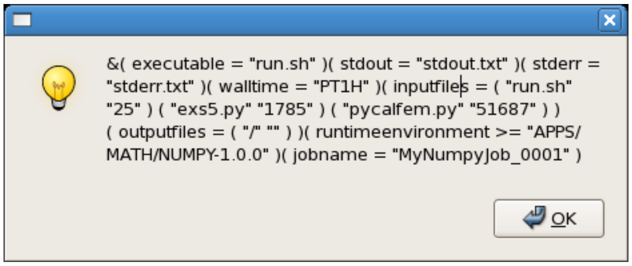
4.6 Removing a job definition

An existing job definition can be remove by selecting the icon in the **Job definition** tab and selecting *Definition* → *Delete* or using the shortcut `Delete`. A confirmation dialog is shown and the job definition is deleted when the **Yes** has been chosen.

Warning: There is no undo for this operation. Job definitions will be permanently deleted.

4.7 Viewing XRSL of a job definition

When debugging job submission problems it can be useful to display the XRSL code that would be used in the job submission process. This can be done by selecting *View XRLS...*. The dialog below shows an example of typical output from this function.



```
&( executable = "run.sh" )( stdout = "stdout.txt" )( stderr =
"stderr.txt" )( walltime = "PT1H" )( inputfiles = ( "run.sh"
"25" ) ( "exs5.py" "1785" ) ( "pycalfem.py" "51687" ) )
( outputfiles = ( "/" "" ) )( runtimeenvironment >= "APPS/
MATH/NUMPY-1.0.0" )( jobname = "MyNumpyJob_0001" )
```

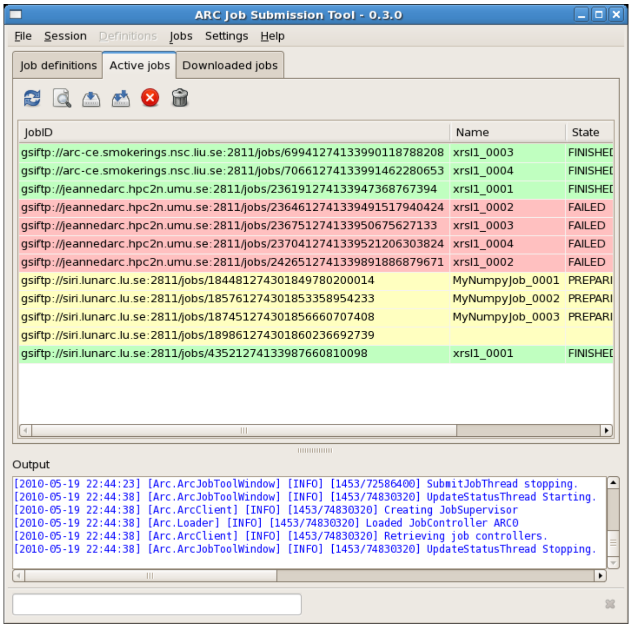
Figure 4.5: Job definition XRSL.

CHAPTER

FIVE

MANAGING RUNNING JOBS

When a job definition has been submitted to a grid resource, the job(s) are managed from the **Active jobs** tab. This tab shows a table with all known running jobs owned by the user. The table displays the **JobID**, **name**, **State**, and **Error**. The individual jobs are also colored according to their state, to make it easier to identify the state of many active jobs. The table can also be sorted by clicking on the table column headers. An example **Active jobs** tab is shown in the following figure:



JobID	Name	State
gsiftp://arc-ce.smokerings.nsc.liu.se:2811/jobs/69941274133990118788208	xrsl1_0003	FINISHED
gsiftp://arc-ce.smokerings.nsc.liu.se:2811/jobs/70661274133991462280653	xrsl1_0004	FINISHED
gsiftp://jeannedarc.hpc2n.umu.se:2811/jobs/236191274133947368767394	xrsl1_0001	FINISHED
gsiftp://jeannedarc.hpc2n.umu.se:2811/jobs/2364612741339491517940424	xrsl1_0002	FAILED
gsiftp://jeannedarc.hpc2n.umu.se:2811/jobs/236751274133950675627133	xrsl1_0003	FAILED
gsiftp://jeannedarc.hpc2n.umu.se:2811/jobs/2370412741339521206303824	xrsl1_0004	FAILED
gsiftp://jeannedarc.hpc2n.umu.se:2811/jobs/2426512741339891886879671	xrsl1_0002	FAILED
gsiftp://siri.lunarc.lu.se:2811/jobs/184481274301849780200014	MyNumpyJob_0001	PREPARI
gsiftp://siri.lunarc.lu.se:2811/jobs/185761274301853358954233	MyNumpyJob_0002	PREPARI
gsiftp://siri.lunarc.lu.se:2811/jobs/187451274301856660707408	MyNumpyJob_0003	PREPARI
gsiftp://siri.lunarc.lu.se:2811/jobs/189861274301860236692739		
gsiftp://siri.lunarc.lu.se:2811/jobs/43521274133987660810098	xrsl1_0001	FINISHED

```
[2010-05-19 22:44:23] [Arc.ArcJobToolWindow] [INFO] [1453/72586400] SubmitJobThread stopping.
[2010-05-19 22:44:38] [Arc.ArcJobToolWindow] [INFO] [1453/74830320] UpdateStatusThread Starting.
[2010-05-19 22:44:38] [Arc.ArcClient] [INFO] [1453/74830320] Creating JobSupervisor
[2010-05-19 22:44:38] [Arc.Loader] [INFO] [1453/74830320] Loaded JobController ARC0
[2010-05-19 22:44:38] [Arc.ArcClient] [INFO] [1453/74830320] Retrieving job controllers.
[2010-05-19 22:44:38] [Arc.ArcJobToolWindow] [INFO] [1453/74830320] UpdateStatusThread Stopping.
```

Figure 5.1: Active jobs in the **Active jobs** tab.

5.1 Selecting active jobs

Jobs can be selected in the active job table by clicking on the row. Multiple job can be selected by using the `Control` or `Shift` modifier keys.

5.2 Refreshing job status

The current status of the active jobs can be refreshed by selecting *Jobs → Refresh status* or using the shortcut `Control-R`. This will query the used resources for the status of the user jobs.

Note: Automatic status update can be enabled by changing the client settings.

5.3 Download selected jobs

To download the selected jobs click *Jobs → Download selected...*. This will donwload results to the corresponding job definition.

Note: arcjobtool can only download results files if a job definition with the same name as the active job exists in the current working directory.

5.4 Download all finished jobs

To download all finished jobas click *Jobs → Download all*. This will donwload _all_ active jobs to corresponding job definitions.

Note: arcjobtool can only download results files if a job definition with the same name as the active job exists in the current working directory.

Note: Automatic download of jobs can be enabled by changing the client settings.

5.5 Killing jobs

Selected jobs that are running can be killed before they are finished by selection *Jobs → Kill selected*. Killing a job will instruct the resource running the job to terminate it.

Note: Killing finished or failed jobs will not remove them. Please use clean instead.

5.6 Cleaning jobs

If a job is failed or finished, but the results are of no interest it can be removed from the selected resources by issuing a clean command. Selected jobs can be cleaned by selecting *Jobs → Clean selected*.

5.7 Removing unknown jobs

If a you is unknown to a resource due to factors out of arcjobtool:s control it is not allways possible to clean or download it. To remove selected jobs forcibly from the job list select *Jobs → Remove from job list*. The jobs are then removed from the active job list and will never be queried for on resources.

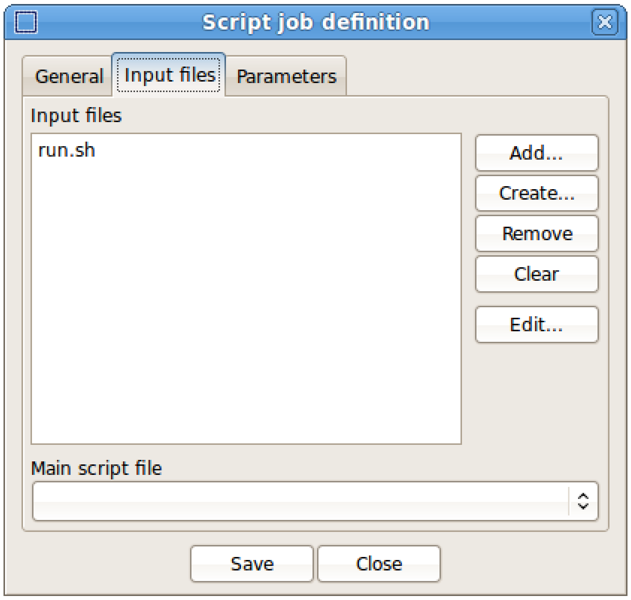
CHAPTER

SIX

SCRIPT JOB PLUGIN

The script plugin support sending generic script-based jobs to resources. The plugin supports submission of parameter sweeps with options of replacing parameters in the scripts with information on, task id, job name and sweep size. Each specific job will be created in a folder inside the job definition directory.

6.1 Input files



Input files

run.sh

Figure 6.1: Input file tab in the script plugin.

Input files will be transferred to the selected resource by arcjobtool and can consist of executables, binary files and textfiles. The **Input files** tab always contains an inputfile called `run.sh` which is the man script that is executed by the job. `run.sh` is maintained by the plugin and should not be modified by the user.

6.1.1 Adding input files

Files are added by clicking on the **Add** button. This brings up a file selection dialog. Select the input file to be included with the job definition. When the file has been selected it is copied into the job definition by arcjobtool. This approach enables the user to move the entire job definition without any dependencies on local files.

6.1.2 Creating new input files

arcjobtool can also create new input files using the **Create...** button. Clicking on this button shows a dialog box asking for a filename. When the filename has been selected an system editor is shown and the file can be edited. arcjobtool will suspend its execution until the editor is closed.

6.1.3 Removing input files

An input file can be removed from the input file list by selecting it in the **Input files** listbox and clicking on **Remove**. This removes the input file from the job definition directory.

6.1.4 Clearing input files

All input files in the job definition can be cleared by clicking on the **Clear** button. This will remove all input files except the `run.sh` from the job definition directory.

6.1.5 Editing input files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show "garbage" in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

6.1.6 Assigning the main script file

Meaningful usage of this plugin requires that at least one of the input files is a script file. To indicate script files that should be executed by the job, use the **Main script file** drop down listbox to select the main script file.

6.2 Parameter sweep settings

The **Parameters** tab settings for the parameter sweep can be defined. arcjobtool can do parameter substitution of certain parameters in the script and input files. The following substitution parameters can be used:

Parameter	Description
%(name)s	Parameter containing the job name
%(id)d	id of the current task 1-sweepsize
%(sweepSize)d	The size of the parameter sweep

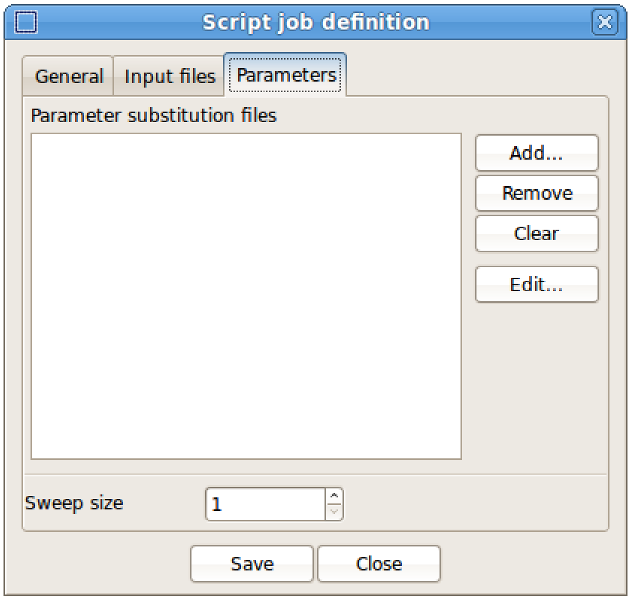
An example of a script with substitution parameters are shown in the following example:

```
#!/bin/sh

echo "Hello, world!"

echo "Jobname %(name)s"
echo "Task id %(id)d"
echo "Sweep size %(sweepSize)d"

sleep 3600
```



Sweep size 1

Figure 6.2: Paramters tab in the script plugin.

arcjobtool will only do parameter substitution of files listed int the listbox **Parameter substitution files**.

6.2.1 Adding files for parameter substitution

Files which should be considered for substitution can be added by clicking on the **Add...** button. This will bring up a dialog showing the existing input files in the definition. Select the file to use for substitution in the list box and click **OK**

6.2.2 Removing files from parameter substituion

Files selected for parameter substitution can be removed by selecting the file in the **Parameter substitution files** listbox and clicking on the **Remove** button.

6.2.3 Clearing substitution files

All files selected for parameter substitution can be cleared by clicking on the **Clear** button.

6.2.4 Editing substitution files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

6.2.5 Setting the sweep size

The sweep size parameter controls how many jobs to submit in the parameter sweep. Changing this parameter will create a subdirectory in the job definition for each parameter. Each subdirectory is a single job that will be submitted to the grid. Changing the sweep size will automatically add and remove these directories. The sweep size is changed by changing the value in the **Sweep size** text box or using the spinner controls on the right side of the textbox.

Note: Currently arcjobtool has a limitation of 9999 jobs for a single parameter sweep.

CHAPTER

SEVEN

NUMPY JOB PLUGIN

The Numpy plugin supports sending Numpy based python jobs to grid resources. The plugin supports submission of parameter sweeps with options of replacing parameters in the python files with information on, task id, job name and sweep size. Each specific job will be created in a folder inside the job definition directory.

7.1 Input files

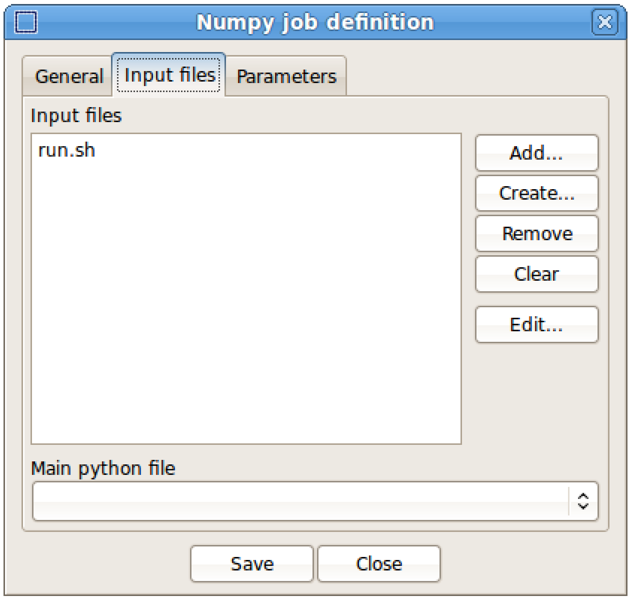


Figure 7.1: Input file tab in the Numpy plugin.

Input files will be transferred to the selected resource by arcjobtool and can consist of python scripts, binary files and textfiles. The **Input files** tab always contains an inputfile called `run.sh` which is the man script that is executed by the job. `run.sh` is maintained by the plugin and should not be modified by the user.

7.1.1 Adding input files

Files are added by clicking on the **Add** button. This brings up a file selection dialog. Select the input file to be included with the job definition. When the file has been selected it is copied into the job definition by arcjobtool. This approach enables the user to move the entire job definition without any dependencies on local files.

7.1.2 Creating new input files

arcjobtool can also create new input files using the **Create...** button. Clicking on this button shows a dialog box asking for a filename. When the filename has been selected an system editor is shown and the file can be edited. arcjobtool will suspend its execution until the editor is closed.

7.1.3 Removing input files

An input file can be removed from the input file list by selecting it in the **Input files** listbox and clicking on **Remove**. This removes the input file from the job definition directory.

7.1.4 Clearing input files

All input files in the job definition can be cleared by clicking on the **Clear** button. This will remove all input files except the `run.sh` from the job definition directory.

7.1.5 Editing input files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show "garbage" in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

7.1.6 Assigning the main python file

Meaningful usage of this plugin requires that at least one of the input files is a main python file. To indicate main python script that should be executed by the job, use the **Main python file** drop down listbox to select the main script file.

7.2 Parameter sweep settings

The **Parameters** tab settings for the parameter sweep can be defined. arcjobtool can do parameter substitution of certain parameters in the script and input files. The following substitution parameters can be used:

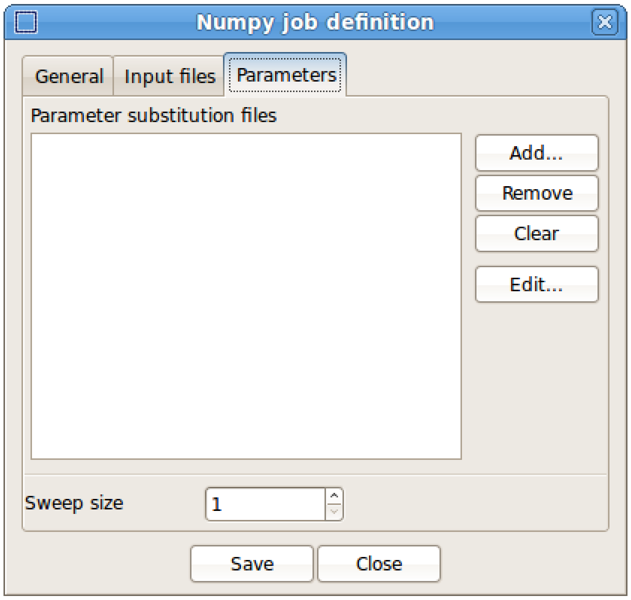
Parameter	Description
%(name)s	Parameter containing the job name
%(id)d	id of the current task 1-sweepsize
%(sweepSize)d	The size of the parameter sweep

An example of a python-script with substitution parameters are shown in the following example:

```
from numpy import *
from pycalfem import *

# ----- ArcGui parameters ----------------------------------------

print "Jobname %(name)s"
print "Task id %(id)d"
print "Sweep size %(sweepSize)d"
```



Sweep size 1

Figure 7.2: Paramters tab in the script plugin.

arcjobtool will only do parameter substitution of files listed int the listbox **Parameter substitution files**.

7.2.1 Adding files for parameter substitution

Files which should be considered for substitution can be added by clicking on the **Add...** button. This will bring up a dialog showing the existing input files in the definition. Select the file to use for substitution in the list box and click **OK**

7.2.2 Removing files from parameter substituion

Files selected for parameter substitution can be removed by selecting the file in the **Parameter substitution files** listbox and clicking on the **Remove** button.

7.2.3 Clearing substitution files

All files selected for parameter substitution can be cleared by clicking on the **Clear** button.

7.2.4 Editing substitution files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

7.2.5 Setting the sweep size

The sweep size parameter controls how many jobs to submit in the parameter sweep. Changing this parameter will create a subdirectory in the job definition for each parameter. Each subdirectory is a single job that will be submitted to the grid. Changing the sweep size will automatically add and remove these directories. The sweep size is changed by changing the value in the **Sweep size** text box or using the spinner controls on the right side of the textbox.

Note: Currently arcjobtool has a limitation of 9999 jobs for a single parameter sweep.

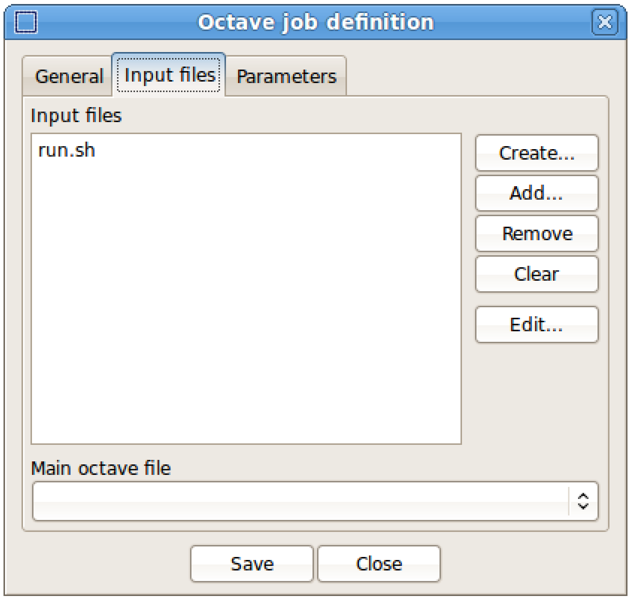
CHAPTER

EIGHT

OCTAVE JOB PLUGIN

The Octave plugin supports sending Octave based jobs to grid resources. The plugin supports submission of parameter sweeps with options of replacing parameters in the .m-files with information on, task id, job name and sweep size. Each specific job will be created in a folder inside the job definition directory.

8.1 Input files



Input files
run.sh

Figure 8.1: Input file tab in the Octave plugin.

Input files will be transferred to the selected resource by arcjobtool and can consist of m-files, binary files and textfiles. The **Input files** tab always contains an inputfile called `run.sh` which is the man script that is executed by the job. `run.sh` is maintained by the plugin and should not be modified by the user.

8.1.1 Adding input files

Files are added by clicking on the **Add** button. This brings up a file selection dialog. Select the input file to be included with the job definition. When the file has been selected it is copied into the job definition by arcjobtool. This approach enables the user to move the entire job definition without any dependencies on local files.

8.1.2 Creating new input files

arcjobtool can also create new input files using the **Create...** button. Clicking on this button shows a dialog box asking for a filename. When the filename has been selected an system editor is shown and the file can be edited. arcjobtool will suspend its execution until the editor is closed.

8.1.3 Removing input files

An input file can be removed from the input file list by selecting it in the **Input files** listbox and clicking on **Remove**. This removes the input file from the job definition directory.

8.1.4 Clearing input files

All input files in the job definition can be cleared by clicking on the **Clear** button. This will remove all input files except the `run.sh` from the job definition directory.

8.1.5 Editing input files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

8.1.6 Assigning the main octave file

Meaningful usage of this plugin requires that at least one of the input files is a main octave file. To indicate main octave m-file that should be executed by the job, use the **Main Octave file** drop down listbox to select the main script file.

8.2 Parameter sweep settings

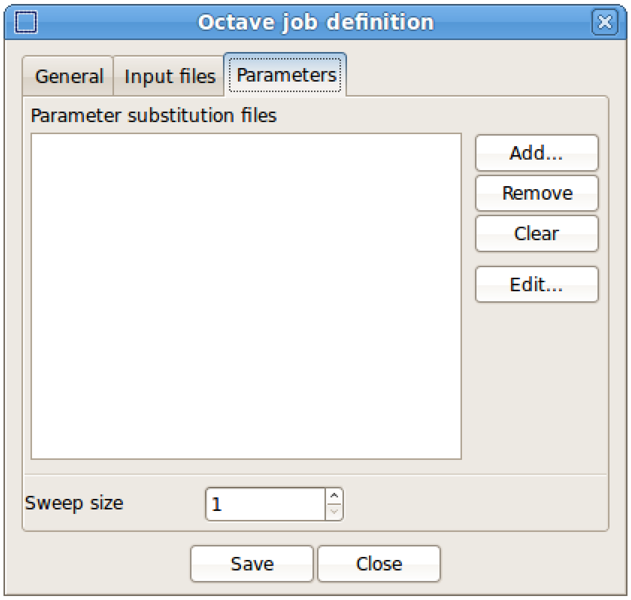
The **Parameters** tab settings for the parameter sweep can be defined. arcjobtool can do parameter substitution of certain parameters in the script and input files. The following substitution parameters can be used:

Parameter	Description
%(name)s	Parameter containing the job name
%(id)d	id of the current task 1-sweepsize
%(sweepSize)d	The size of the parameter sweep

An example of an Octave m-file with substitution parameters are shown in the following example:

```
disp('Jobname %(name)s');
disp('Task id %(id)d');
disp('Sweep size %(sweepSize)d')
```

arcjobtool will only do parameter substitution of files listed int the listbox **Parameter substitution files**.



Sweep size 1

Figure 8.2: Paramters tab in the script plugin.

8.2.1 Adding files for parameter substitution

Files which should be considered for substitution can be added by clicking on the **Add...** button. This will bring up a dialog showing the existing input files in the definition. Select the file to use for substitution in the list box and click **OK**

8.2.2 Removing files from parameter substituion

Files selected for parameter substitution can be removed by selecting the file in the **Parameter substitution files** listbox and clicking on the **Remove** button.

8.2.3 Clearing substitution files

All files selected for parameter substitution can be cleared by clicking on the **Clear** button.

8.2.4 Editing substitution files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

8.2.5 Setting the sweep size

The sweep size parameter controls how many jobs to submit in the parameter sweep. Changing this parameter will create a subdirectory in the job definition for each parameter. Each subdirectory is a single job that will be submitted to the grid. Changing the sweep size will automatically add and remove these directories. The sweep size is changed by changing the value in the **Sweep size** text box or using the spinner controls on the right side of the textbox.

Note: Currently arcjobtool has a limitation of 9999 jobs for a single parameter sweep.

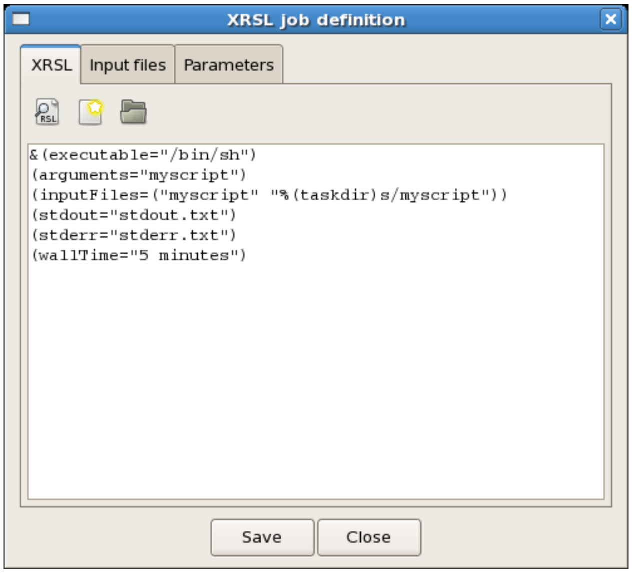
CHAPTER

NINE

XRSL JOB PLUGIN

The XRSL plugin support the creation of jobs where the user can describe the job direct in XRSL. This plugin also supports variable substituion inside the XRSL code as well as in any additional input files. The plugin is divided in 3 tabs, **XRSL**, **Input files** and **Parameters**.

9.1 Describing the job in XRSL



```
&(executable="/bin/sh")
(arguments="myscript")
(inputFiles=("myscript" "%(taskdir)s/myscript"))
(stdout="stdout.txt")
(stderr="stderr.txt")
(wallTime="5 minutes")
```

Figure 9.1: Defining the job in XRSL

In this tab the job description can be entered in the XRSL language. The tab contains a toolbar with 3 buttons described in the following table.

Toolbar icon	Purpose
🔍RSL	validates the entered XRSL.
🗋	edits the XRSL in an external editor.
📁	imports the XRSL from an external file.

9.2 The %(taskdir)s directive

arcjobtool creates directories for each job (task directory) defined in a parameter sweep. To specify local input files the task dir of a specific job has to specified which is not known when the XRSL is edited. To solve this a special directive, taskdir, can be used in the XRSL source code when the location of the jobs taskdir is needed. The following shows an example where the file, `myscript`, which will be retrieved from `%(taskdir)s` directory. The taskdir directive will be replaced with the correct directory before job submission.:

```
&(executable="/bin/sh")
(arguments="myscript")
(inputFiles=("myscript" "%(taskdir)s/myscript"))
(stdout="stdout.txt")
(stderr="stderr.txt")
(wallTime="5 minutes")
```

9.3 Input files

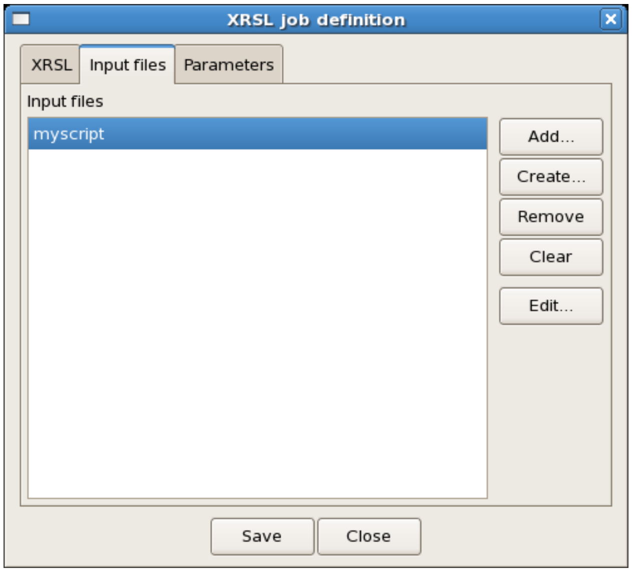


Figure 9.2: Input file tab in the XRSL plugin.

Input files will be transferred to the selected resource by arcjobtool and can consist of scripts, binary files and textfiles.

9.3.1 Adding input files

Files are added by clicking on the **Add** button. This brings up a file selection dialog. Select the input file to be included with the job definition. When the file has been selected it is copied into the job definition by arcjobtool. This approach enables the user to move the entire job definition without any dependencies on local files.

9.3.2 Creating new input files

arcjobtool can also create new input files using the **Create...** button. Clicking on this button shows a dialog box asking for a filename. When the filename has been selected an system editor is shown and the file can be edited. arcjobtool will suspend its execution until the editor is closed.

9.3.3 Removing input files

An input file can be removed from the input file list by selecting it in the **Input files** listbox and clicking on **Remove**. This removes the input file from the job definition directory.

9.3.4 Clearing input files

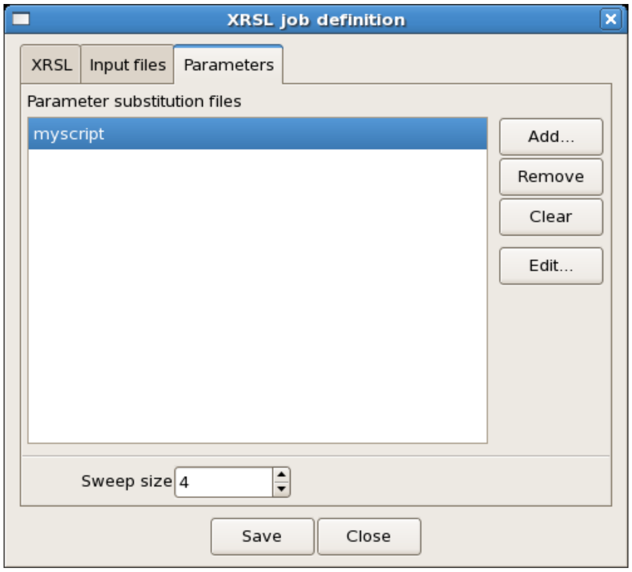
All input files in the job definition can be cleared by clicking on the **Clear** button. This will remove all input files except the `run.sh` from the job definition directory.

9.3.5 Editing input files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

9.4 Parameter sweep settings



myscript

Sweep size 4

Figure 9.3: Paramters tab in the XRSL plugin.

The **Parameters** tab settings for the parameter sweep can be defined. arcjobtool can do parameter substitution of certain parameters in the script and input files. The following substitution parameters can be used:

Parameter	Description
%(name)s	Parameter containing the job name
%(id)d	id of the current task 1-sweepsize
%(sweepSize)d	The size of the parameter sweep

An example of a shell-script with substitution parameters are shown in the following example:

```
#!/bin/sh

echo "Jobname %(name)s"
echo "Task id %(id)d"
echo "Sweep size %(sweepSize)d"
```

arcjobtool will only do parameter substitution of files listed int the listbox **Parameter substitution files**.

9.4.1 Adding files for parameter substitution

Files which should be considered for substitution can be added by clicking on the **Add...** button. This will bring up a dialog showing the existing input files in the definition. Select the file to use for substitution in the list box and click **OK**

9.4.2 Removing files from parameter substituion

Files selected for parameter substitution can be removed by selecting the file in the **Parameter substitution files** listbox and clicking on the **Remove** button.

9.4.3 Clearing substitution files

All files selected for parameter substitution can be cleared by clicking on the **Clear** button.

9.4.4 Editing substitution files

Existing input files in the job definition can be edited by selecting the file in the **Input files** listbox and clicking on the **Edit...** button. This will bring up a system editor enabling the user to edit the input file.

Note: Editing of binary files is not supported. Editing a binary file in the system editor will only show “garbage” in the editor window. Please be careful not saving the binary file when it has been opened in the editor.

9.4.5 Setting the sweep size

The sweep size parameter controls how many jobs to submit in the parameter sweep. Changing this parameter will create a subdirectory in the job definition for each parameter. Each subdirectory is a single job that will be submitted to the grid. Changing the sweep size will automatically add and remove these directories. The sweep size is changed by changing the value in the **Sweep size** text box or using the spinner controls on the right side of the textbox.

Note: Currently arcjobtool has a limitation of 9999 jobs for a single parameter sweep.

CHAPTER

TEN

INDICES AND TABLES

- *Index*
- *Module Index*
- *Search Page*