

So You Want to Run a Job with a Fuse Mount?

Mark Morgan

Abstract

This document covers all of the tasks that must be undertaken in order to set the situation up from end-to-end so that a Genesis II grid job can run with a mounted grid (fuse) file system. Some of the information contained in this document is specific to the user's side of the equation (setting up the job, etc.), and some of it is specific to the system (grid) administrator's role.

The User

On the user facing side of the fuse mount scenario, the user must specify a job that requests a grid file system. This is done using the standard JSDL "Filesystem Mount" resource element. Similar to specifying a **SCRATCH** file system, a **GRID** file system is indicated simply by giving a file system name of **GRID** and indicating as type **normal**. For example:

```
<FileSystem name="GRID">  
  <FileSystemType>normal</FileSystemType>  
</FileSystem>
```

The easiest way to actually create a job that uses a **GRID** file system is to use the **job tool** that comes with Genesis II which has the ability to indicate file system mounts such as **GRID** and **SCRATCH**.

Once the user has create a JSDL description of the job, he or she simply submits that job to the grid like it was any other kind of job. No further changes need to be made on his or her side.

The System Administrator

Running jobs that can mount the grid is a somewhat delicate prospect and while the mechanisms involved go to some large amount of effort to attempt to verify that the process is capable of working, a system administrator should still make the final call about whether or not to turn this feature one for a given container.

Succinctly, there are a few things that must be true before a fuse or grid mount will work for a job running through a Genesis II BES:

- The BES container must have access to a shared file system that the job will be able to see when running (this is already required for all BESs).
- The job must run on a machine that
 - Has read/write access to /dev/fuse
 - Has execute access to the binary fusermount
 - Has execute access to a properly configured Genesis II client install.

Assuming that all these things are true, a system administrator configures a Genesis II container to accept fuse mount jobs by specifying that the Genesis II container exports the GENII_INSTALL_DIR environment variable¹. The system administrator must also explicitly turn on FUSE support in the container by turning on the *Fuse Filesystem* container service as detailed below.

Turning on the Fuse Filesystem Container Service

In order to turn on the *Fuse Filesystem* container service, the Genesis II container in question needs to be restarted with a properly formatted \$GENII/deployments/\$DEPLOYMENT_NAME/configuration/cservices/FuseFilesystemManagerService.xml configuration file. This file follows the standard file format for all container services (cservices) and has a single required property of **fuse-directory** which indicates a shared directory where fuse mounts can be made by the container. An example version of this file is included in the Genesis II code base in the same location with a name that ends in .example instead of .xml.

Once the container is restarted with the new container service, the grid queue will need to update the information for that container in order to get the new information, but after that, jobs that request the **GRID** file system mount in JSDL should match with the new container (assuming that the rest of the resource requirements for the job also match).

¹ For more information on exporting environment variables in Genesis II BES containers, please see the document “Exporting Environment Variables to Genesis II Jobs”.