

Table of Contents

Distributing gLite clients to the Application Area.....	1
Introduction.....	1
Process.....	1
Scripts.....	1
create-aa-repo.....	1
create-aa-list.....	1
Update example.....	2
Whenever a patch affecting cert-glite-AA goes into certification.....	2
Whenever a patch affecting glite-AA goes into production.....	2
Contact.....	2

Distributing gLite clients to the Application Area

Introduction

For the 3.1 releases we are introducing an extra distribution route for a subset of the clients. We will maintain a section of the Application Area repository in order to give the experiments early access to updates and to enable them to use these clients in building their own distributions.

The Application Area repository is a directory hierarchy in AFS, holding binary installations of a number of packages which are organised according to version and architecture. It is not an APT or YUM repository, and does not contain packages in the usual sense. You can see it on AFS at `/afs/cern.ch/sw/lcg/external/`.

We release gLite 3.1 on SL3 and SL4 (32 and 64 bit) - it is important to note that this means the binaries on SL3 are not the same as those released to production through the standard release process (which is gLite 3.0). For more information on this subtlety, see the README file in the repository -

`/afs/cern.ch/sw/lcg/external/Grid/README`

Process

The client list is held in the file `glite-AA` and this should be maintained just like any other rpm list. The Application Area repository should be populated with any client updates which enter certification, and the list defining what has been released should be updated when a release is made.

- `cert-glite-AA` represents the what is in certification
- `glite-AA` represents what is in production

Scripts

There are a couple of scripts to manage the process, they are in CVS [↗](#) alongside the other integration scripts.

create-aa-repo

```
create-aa-repo -r rpm_list.txt -t tar_list.txt -d destination_directory
```

The command updates the repository in the directory given by the `-d` argument. It takes two lists as further arguments, both of which should be generated by `check-etics-repository` (see later). Wherever possible a binary tarball will be used to populate the repository, but if this is not available a 'fake' tarball will be created from the rpm.

create-aa-list

As numerous versions of a particular package will exist simultaneously in the AA repo, a version file must be produced whenever the clients are updated in production. This version file will list the components which make up the latest release.

```
create-aa-list -l <list file> -v <version> -d <target directory>
```

This is a very simple script that just reformats `glite-AA` and labels the new file with a version.

Update example

Whenever a patch affecting cert-glite-AA goes into certification.

The `cert-glite-AA` list should be updated and tagged for each platform (`slc4_ia32_gcc346`, `slc4_x86_64_gcc346` and `slc3_ia32_gcc323`).

```
./check-etics-repository -p slc4_ia32_gcc346 -a tar.gz -l ../cert-lists/cert-glite-AA > tar_list.txt
./check-etics-repository -p slc4_ia32_gcc346 -a rpm -l ../cert-lists/cert-glite-AA > rpm_list.txt
./create-aa-repo -r rpm_list.txt -t tar_list.txt -d /afs/cern.ch/sw/lcg/external/Grid
```

and also

```
./check-etics-repository -p slc3_ia32_gcc323 -a tar.gz -l ../cert-lists/cert-glite-AA > tar_list.txt
./check-etics-repository -p slc3_ia32_gcc323 -a rpm -l ../cert-lists/cert-glite-AA > rpm_list.txt
./create-aa-repo -r rpm_list.txt -t tar_list.txt -d /afs/cern.ch/sw/lcg/external/Grid
```

and also

```
./check-etics-repository -p slc4_x86_64_gcc346 -a tar.gz -l ../cert-lists/cert-glite-AA > tar_list.txt
./check-etics-repository -p slc4_x86_64_gcc346 -a rpm -l ../cert-lists/cert-glite-AA > rpm_list.txt
./create-aa-repo -r rpm_list.txt -t tar_list.txt -d /afs/cern.ch/sw/lcg/external/Grid
```

Whenever a patch affecting glite-AA goes into production

The software is already in the AA repository, but it has not been identified as being certified.

For an update to go to production, the `glite-WN` list will be updated and tagged. The `glite-AA` list should be updated in the same way and tagged **with the same version as the WN**. The point is that we are making a subset of the WN available in a different form and the versioning must be consistent.

Then;

```
./create-aa-list -l ../lists/glite-AA -v <version> -d /afs/cern.ch/sw/lcg/external/Grid
```

where the version is the version of the WN release.

As the SL3 binaries here will **never** go to production (see the note in the introduction), this step is performed only once and has no platform specific actions.

Contact

Contact Oliver Keeble if you have problems.

-- Main.okeeble - 10 May 2007

This topic: EGEE > ApplicationAreaRepository
Topic revision: r6 - 2007-08-06 - OliverKeeble



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback