Table of Contents

The WLCG Grid Applications area

Location of WLCG Applications area

It is stored on AFS under:

/afs/cern.ch/sw/lcg/external/Grid (read only)

/afs/.cern.ch/sw/lcg/external/Grid (read and write)

When changing the AFS area make changes under the read/write directory and the changes should become visible under the read only directory at most 1 hour later. If the changes do not appear open a ticket to AFS support.

Relevant Files

Apart from the actual software the following are updated each time software is added:

- ./README_emi_or_later - summary of latest versions of currently maintained packages
- ./0.emi/packagelist - mapping of rpms-versions to directory/version
- /afs/cern.ch/project/gd/www/glite/aa-updates.rss - webpage with update notices (visible at http://grid-deployment.web.cern.ch/grid-deployment/glite/aa-updates.rss)

Structure of component directories

<Name/Version/arch/content>

Generally the following architecture directories are made:

mkdir i686-slc5-gcc41-opt

mkdir x86_64-slc5-gcc41-opt

mkdir x86_64-slc6-gcc44-opt

mkdir x86_64-centos7-gcc48-opt

ln -s i686-slc5-gcc41-opt i686-slc5-gcc34-opt

ln -s i686-slc5-gcc41-opt i686-slc5-gcc43-opt

ln -s x86_64-slc5-gcc41-opt x86_64-slc5-gcc34-opt

ln -s x86_64-slc5-gcc41-opt x86_64-slc5-gcc43-opt

ln -s x86_64-slc5-gcc41-opt x86_64-slc5-gcc46-opt

ln -s x86_64-slc6-gcc44-opt x86_64-slc6-gcc46-opt

ln -s x86_64-slc6-gcc44-opt x86_64-slc6-gcc48-opt

ln -s x86_64-slc6-gcc44-opt x86_64-slc6-gcc49-opt

ln -s x86_64-slc6-gcc44-opt x86_64-slc6-gcc62-opt

ln -s x86_64-centos7-gcc48-opt x86_64-centos7-gcc49-opt

ln -s x86_64-centos7-gcc48-opt x86_64-centos7-gcc62-opt

i.e. three architectures corresponding to 32bit/slc5, 64bit/slc5 and 64bit/slc6. The symbolic links are for alternate compiler versions; we don't recompile the packages hence using symbolic links to indicate that the versions installed are in fact identical.

The cream, FTS, WMS packages are not available for 32bit/slc5. The FTS3 package is only available for 64bit/slc6.

The meaning of the version number in <Name/Version> directory layout

The <Version> is supposed to identify the software release in some way. The number is given in notices when new software is available and experiments will need to use the number in their configuration (e.g. to setup PATH). However there is no fixed format for the number; it is usually a 1.2.3, 1.2.3-4 or 1.2.3.4string format.

For data management packages it is usually what the team calls it in the release notes. For packages taken from EMI it is usually the number referred to in the EMI release announcement page.

Unpacking of packages

Is done per rpm:

cd <Name/Version/arch>

rpm2cpio /tmp/rpm1_name.rpm | cpio -dimv

...

rpm2cpio /tmp/rpm5_name.rpm | cpio -dimv

remove the /usr/ level mv usr/* . rmdir usr

Addition of symbolic links in some packages

Links: WMS/<Version>/x86_64-slc5-gcc41-opt/lib64/python2.4 -> python

WMS/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.6 -> python

WMS/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.7 -> python

mjf/<Version>/i686-slc5-gcc41-opt/lib/python2.7 -> python2.4

mjf/<Version>/x86_64-slc5-gcc41-opt/lib/python2.7 -> python2.4

mjf/<Version>/x86_64-slc6-gcc44-opt/lib/python2.7 -> python2.4

xrootd/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.7 -> python2.6

fts3/<Version>/x86_64-slc6-gcc44-opt/lib/python2.7 -> python2.6

fts3/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.7/site-packages/fts -> ../../python2.6/site-packages/fts

arc/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.7 -> python2.6

arc/<Version>/x86_64-slc5-gcc41-opt/lib64/python2.7 -> python2.6

arc/<Version>/i686-slc5-gcc41-opt/lib64/python2.7 -> python2.6

htcondor/<Version>/x86_64-slc6-gcc44-opt/lib64/python2.7 -> python2.6

VOMS version 2, >= 2.0.12

```
bin:
ln -s voms-proxy-init2 voms-proxy-init
ln -s voms-proxy-destroy2 voms-proxy-destroy
ln -s voms-proxy-info2 voms-proxy-info

share/man/man1:
ln -s voms-proxy-destroy2.1.gz voms-proxy-destroy.1.gz
ln -s voms-proxy-info2.1.gz voms-proxy-info.1.gz
ln -s voms-proxy-init2.1.gz voms-proxy-init.1.gz
```

RPM Packages which make one software package in the AA

These were determined in an ad-hoc way, in some cases by adding packages until simple tests of the software didn't show up any missing runtime dependencies. In other cases the provider lists a set of rpms or the rpm package name suggests that they form a group.

The list of RPMs is recorded in the ./0.emi/packagelist file. The file is maintained manually although it follows a column structure:

e.g. to see which RPMs were installed in the canl/2.2.2 directory:

cat 0.emi/packagelist | grep canl/2.2.2

canl-c++ 1.0.0-1 canl/2.1.2 (i386 1.0.0-3) canl/2.2.1 (i386 1.0.1-1 from epel) canl/2.2.2 (i386 1.0.1-1 epel)

canl-c++-devel 1.0.0-1 canl/2.1.2 (i386 1.0.0-3) canl/2.2.1 (i386 1.0.1-1 from epel) canl/2.2.2 (i386 1.0.1-1 epel)

canl-java-tomcat 0.1.18-1 canl/2.2.1 canl/2.2.2

canl-c 2.1.4-1 canl/2.2.2

canl-c-devel 2.1.4-1 canl/2.2.2

canl-c-doc 2.1.4-1 canl/2.2.2

canl-c-examples 2.1.4-1

canl/2.2.2 canl-java-javadoc 1.3.2-1

canl/2.2.2 canl-java 1.3.2-1 canl/2.2.2

shows that there were 9 RPMs installed. The first column is the RPM name, the second the RPM version number. The third or more columns are directories under which the RPM has been unpacked; the RPM may

be unpacked under several versions if that particular RPM did not change between versions.

It was assumed that the same package-version number and source repository is available for all architectures, so the entry in packagelist isn't qualified by architecture. The above shows an instance where this was not true, hence the ad-hoc indication of the different version and source of the RPM.

RPMs are found from:

There are a few sources: The data management software from cern are usually taken from EPEL stable, or exceptionally a teams might directly indicate a version and special source repository from which to install a version.

(cern data management components:)

gfal, gfal2, FTS, FTS3, lcg-dm-common, DPM, LFC, gridftp-ifce, is-ifce, srm-ifce

"dcap" is also taken from EPEL. The Grid AA "epel" package is another selection of packages from epel; principally globus/myproxy, davix, gsoap, xrootd and a few miscellaneous dependencies.

mjf is provided by Stefan Rosier.

Most of the rest are from EMI: canl, cream, gridsite, lb, voms, WMS.

Components and the repository source:

```
arc EPEL
gfal2 EPEL + LCGUTIL_EXTRAS_REPO for gfal2-python py2.7.4
gridftp-ifce EPEL
is-ifce EPEL
srm-ifce EPEL
voms (v2 + v3) ITALIANGRID_REPO for voms* + bouncycastle146-mail, EPEL for jakarta-commons-io, EM
gridsite EPEL
canl EPEL for canl-c*, EMI for canl-java*
dcap EPEL
cream EMI
lb EMI
WMS EMI
epel EPEL, (+EGI for xerces-c, classads)
davix EPEL
xrootd XROOTD_REPO
dm-util EPEL, LCGUTIL_EXTRAS_REPO for lcg-util-python py2.7.4 build
DPM [no longer available, same release version (but different packaging version) now in EPEL], LC
FTS [taken from an internal build repo no longer available; rpms for this version available in EM
gfal EPEL, LCGUTIL_EXTRAS_REPO  for gfal-python py2.7.4
lcg-dm-common [taken from an internal build repo; same version (but different packaging) now in E
lcg-infosites EPEL
LFC [no longer available, same release version (but different packaging version) now in EPEL], LC
mjf [from Stefan Roiser]
FTS3 FTS3_REPO
```

The package name and versions themselves are recorded in a text file in the Grid AA. e.g. to see which RPM packages (and versions) goes into the gfal/2.6.8/ directory:

grep gfal2/2.6.8 /afs/cern.ch/sw/lcg/external/Grid/0.emi/packagelist

Links for the repositories, or one mirror of them, for the repositories mentioned above are given below for sl6 x86_64. The sl5 x86_64/i386 where available can be found by changing the repository url in the expected way.

EPEL: http://mirror.switch.ch/ftp/mirror/epel/6/x86_64/

EMI: http://emisoft.web.cern.ch/emisoft/dist/EMI/3/sl6/x86_64/[baselupdates]/

LCGUTIL_EXTRAS_REPO:
http://grid-deployment.web.cern.ch/grid-deployment/dms/lcgutil/repos/el6-extras/x86_64/

LCGDM_PY27_REPO :
https://grid-deployment.web.cern.ch/grid-deployment/dms/lcgdm/repos/py27/el6/x86_64/

ITALIANGRID_REPO: see http://italiangrid.github.io/voms/download.html, currently:
https://ci-01.cnaf.infn.it/download/voms/emi3/sl6/x86_64

EGI: http://repository.egi.eu/sw/production/umd/3/sl6/x86_64/updates/

XROOTD_REPO: http://xrootd.cern.ch/sw/repos/stable/slc/6/x86_64/

FTS3_REPO: http://grid-deployment.web.cern.ch/grid-deployment/dms/fts3/repos/el6/x86_64/

Notification about updates in the WLCG grid applications area

Edit the text file ./README_emi_or_later to indicate the latest version of all the current Grid AA packages.

However the main notification is a web page. The source file is at

/afs/cern.ch/project/gd/www/glite/aa-updates.rss

It is formatted as an RSS feed. This file is updated to include an article (content) each time there is a set of updates available. It should only be updated once the new files are visible in the read-only AFS WLCG Grid AA space.

Editing aa-updates.rss: It's suggested to copy the existing file to a new file "aa-updates-new.rss", edit that, verify it by visiting the web page before moving it into place, replacing the original file.

e.g.

http://grid-deployment.web.cern.ch/grid-deployment/glite/aa-updates-new.rss

http://grid-deployment.web.cern.ch/grid-deployment/glite/aa-updates.rss

Editing consists of:

1. replicating the topmost <item>..</item> and editing the new, topmost item.
2. In the new item; change the comment at the top about what has changed. The table below it summarises all the current "most recent" versions of all packages, with changed packages listed in <b>(bold). Then another table below that gives links to any available release notes for the changed components.
3. Update the <pubData> and <guid> at the end of the item. For the guid generate a random one with "uuidgen -r".
4. Also update the <pubDate> at the top of the document above the first <item>

-- Main.David.Smith - 3 June 2014

-- AndreaManzi - 3 June 2014

This topic: LCG > WLCGGridApplicationsArea
Topic revision: r12 - 2017-01-19 - unknown