

Table of Contents

| | |
|---|----------|
| Release Management tasks..... | 1 |
| Client Integration..... | 1 |
| Using rpm lists..... | 1 |
| Using ETICS..... | 2 |
| ETICS project configuration management..... | 3 |
| VDT configurations in ETICS..... | 4 |
| EMT agendas..... | 5 |

Release Management tasks

Client Integration

Responsible: Maria Alandes **Backup person:** David Smith

Using rpm lists

The list of dependencies included in a client release can be found in SVN under https://svn.cern.ch/repos/grelscripts/glite_UI_and_WN_rpm_lists/. Check in the gLite web pages which version is the last released version in production.

In order to prepare gLite UI/WN/VOBOX releases, make sure a task and a patch are created under the following trackers:

- gLite Middleware patch tracker.
- EGEE JRA1 workplan. The task should contain a reference to the patch.

Once it is clear what will trigger the next client release, estimate a deadline for having the patch certified in the corresponding task. In the meantime and until the new release is actually prepared:

- Attach any patches that may affect the gLite client release in the corresponding patch.
- Modify the the rpm lists.
 - ◆ Download the rpm lists: `svn co https://svn.cern.ch/repos/grelscripts/ --username your_login`
 - ◆ Apply your changes and commit them using: `svn ci -m "Comment"`, where comment is:
 - ◇ For new versions of an existing dependency: `PATCH:XXXX. NEW package_name1, ..., package_nameN`
 - ◇ For new dependencies: `PATCH:XXXX. ADDED package_name1, ..., package_nameN`
 - ◇ For removed dependencies: `PATCH:XXXX. REMOVED package_name1, ..., package_nameN`

When you are ready to prepare the release, then follow the steps below for the WN and the UI:

- Copy the list of rpms from the last version of the files in SVN in the corresponding patch.
- Access machine `lxbrf2713`
- Delete any previous patches in `/afs/cern.ch/project/gd/www/glite/prepare/internal/3.2/glite-UI` or `/afs/cern.ch/project/gd/www/glite/prepare/internal/3.2/glite-WN`.
- Run: `./grelscripts/patch_to_internal.py -r 3.2 -a x86_64 -o sl5 --no-etics patch_number`
- Since you will have to sign the packages, you need your gpg key installed in the machine, as explained in the SR release instructions.

For the VOBOX release, you need to create the metapackage in ETICS first.

- Clone the `org.glite.glite-clients.glite-VOBOX` component configuration matching the latest version in production.
- Increase the version and add or remove new or obsolete dependencies if needed.
- Lock the new component configuration against `glite_3_2_cert`.
- Build in SL5/64bits component configuration so that the new metapackage is registered in the permanent repository.
- Add the metapackage in the `glite-VOBOX` file in SVN.

- Then follow the same instructions as for the UI/WN.

The repo files to be used to do the certification are:

- http://grid-deployment.web.cern.ch/grid-deployment/glite/repos_certification/3.2/glite-UI.
- http://grid-deployment.web.cern.ch/grid-deployment/glite/repos_certification/3.2/glite-WN.
- http://grid-deployment.web.cern.ch/grid-deployment/glite/repos_certification/3.2/glite-VOB.

Copy these URLs in the corresponding patch and move the patch into Ready for Certification.

Using ETICS

[▣ Details](#) [▣ Hide Details](#)

ETICS configurations corresponding to the latest production versions:

- **Subsystem:** `glite-clients_Jan2011`
 - **glite-WN:** `glite-WN_R_3_2_10-2`
 - **glite-WN-version:** `glite-WN-version_3_2_10`
 - **glite-UI:** `glite-UI_R_3_2_9-4`
 - **glite-UI-version:** `glite-UI-version_3_2_9`
- **SL5 integration:** dependencies for SL5/64bits and SL5/32bits are maintained in ETICS. ETICS can't be used to generate the YUM repositories since it doesn't work with groups. A manual step is therefore needed to create the certification repository.
 - ◆ Create patches for `glite-UI/glite-WN` in SL5/64bits. Make sure you attach all the relevant patches that release new versions of the clients and summarise the changes in the release notes.
 - ◆ Go to `org.glite.glite-clients`.
 - ◆ Clone the `glite-UI-version/glite-WN-version` component configurations matching the latest version in production.
 - ◆ Increase the version.
 - ◆ Lock and build in SL5/64bits the new configurations so that the new packages are registered in the permanent repository.
 - ◆ Clone the `glite-UI/glite-WN` component configurations matching the latest version in production.
 - ◆ Increase the version and add or remove new or obsolete dependencies if needed.
 - ◆ Create a new `glite-clients` subsystem configuration.
 - ◆ Edit the subconfigurations and add the new versions for `glite-UI-version/glite-WN-version` and `glite-UI/glite-WN`.
 - ◆ Lock the new subsystem configuration against `glite_3_2_cert`.
 - ◆ Build in SL5/64bits and SL5/32bits the `glite-UI/glite-WN` component configurations so that the new metapackages are registered in the permanent repository.
 - ◆ Copy and paste the package lists of the two builds in the Savannah patches for `glite-UI` and `glite-WN`. Use this list of packages to create a repository containing a group install.
 - ◆ Use the comps files from the production repository: `glite-UI`, `glite-WN`. Remember to modify these files if you have added or deleted dependencies.
 - ◆ In order to create a repository with a group, run: `createrepo -g comps_file $directory_where_the_repo_is_located`
 - ◆ Create a repo file that the certifier can download to use this repository and add the link in the patch.
 - ◆ **Known issues:**
 - ◇ The AMGA clients in ETICS are not correct. Therefore, manually remove the amga client package from the list of rpms and add the one currently distributed in production:
 - http://glitesoft.cern.ch/EGEE/gLite/R3.2/glite-GENERIC/sl5/x86_64/RPMS

◇ The following packages are not defined as dependencies in the UI/WN component configurations. This is OK since the installation is via groups and they are part of the comps file. However, they need to be added manually in the savannah patch so they are part of the repository:

```
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/download/registered/org.gli
. http://etics-repository.cern.ch/repository/pm/registered/repomd/id/755
. http://etics-repository.cern.ch/repository/pm/registered/repomd/id/755
. http://etics-repository.cern.ch/repository/pm/registered/repomd/id/755
```

- **SL4 integration:**

- ◆ Create patches for `glite-UI/glite-WN` in SL4/32bits and one patch for `glite-WN` in SL4/64bits. Make sure you attach all the relevant patches that release new versions of the clients and summarise the changes in the release notes.
- ◆ Go to `org.glite.glite-clients`.
- ◆ Clone the `glite-UI/glite-WN` component configurations matching the latest version in production.
- ◆ Increase the version and add or remove new or obsolete dependencies if needed.
- ◆ Lock the new component configurations against `glite_3_2_cert`.
- ◆ Build in SL4/32bits the `glite-UI/glite-WN` component configurations, and in SL4/64bits the `glite-WN` configuration, so that the new metapackages are registered in the permanent repository. Make sure you create a yum repository as well.
- ◆ Copy and paste the package lists in the corresponding Savannah patches for `glite-UI` and `glite-WN`.
- ◆ Copy and paste the YUM repo URL in the Savannah patch.

ETICS project configuration management

Responsible: Maria Alandes **Backup person:** David Smith

ETICS project configurations should be updated every time there is a `Verified` patch. The release manager should also track build problems in patches so the change of version in the affected packages can be understood in the release notes of a service. It's enough to mention `Build fixes` and consider the patch `Verified` after testing the build is succesful.

The steps to update a project configuration are:

- Query for any `Open` patch in status `Verified` (`Internal` and `Standard`).
- Make sure the patch includes a subsystem configuration.
- Clone `glite_3_*_cert` and rename the clone as `glite_3_*_new`.
- Increase the version in the general configuration properties.
- Increase the version in the `gLiteProjectVersion` property defined in the `Default Platform`.
- Put a comment in the description field summarising the changes.
- Edit the subconfigurations of `glite_3_*_new` and include the new subsystem configurations of all `Open` and `Verified` patches.
- Launch a build and wait for the result.
- Update the `glite_3_2_cert` Log or `glite_3_1_cert` Log twiki pages by adding a comment: `In the queue - The following configurations have been requested and the build is being tested right now:.` Then include a list of subsystem configurations with their corresponding patches (when the patch exist), together with a short summary of the change.

- Update the Verified patches twiki by adding an entry per patch in red status.
- If the build is succesful, rename `glite_3*_cert` to the corresponding version, something like `glite_3*_X_Y`.
- Delete `glite_3*_cert`
- Clone `glite_3*_X_Y` and rename the clone as `glite_3*_cert`.
- Lock `glite_3*_X_Y`.
- Update the `glite_3_2_cert` Log or `glite_3_1_cert` Log twiki pages by adding the date and the `glite_3*_X_Y` as a title.
- Update the Verified patches twiki by adding the same title and moving the status to green.
- Only for Internal patches: close the patches with a comment like `ETICS project configuration has been updated. I close the patch.`
- Announce the change to the EMT. You can use the template below:

Subject: `glite_3*_cert` update

Hi all,

`glite_3*_cert` has been updated to `=glite_3*_X_Y=`:

```
* Summary of the changes
* ...
* ...
```

Details can be found in:
Link to the corresponding Log twiki.

VDT configurations in ETICS

Follow the recipe below to add new VDT versions in ETICS:

- **SL5** (the instructions are the same for 32bit and 64bits):
 - ◆ Copy the following VDT packages in an SL5 machine:
 - ◇ `vdt_globus_data_server`
 - ◇ `vdt_globus_essentials`
 - ◇ `vdt_globus_jobmanager_common`
 - ◇ `vdt_globus_rm_client`
 - ◇ `vdt_globus_rm_essentials`
 - ◇ `vdt_globus_rm_server`
 - ◇ `vdt_globus_sdk`
 - ◆ Create a tarball for each package
 - ◆ Install the production version of `gpt`
 - ◆ Install all the previous packages + `gpt`.
 - ◆ Run the following commands:
 - ◇ `export GPT_LOCATION=/opt/gpt`
 - ◇ `export GLOBUS_LOCATION=/opt/globus`
 - ◇ `export GLOBUS_INSTALL_LOCATION=$GLOBUS_LOCATION`
 - ◇ For 32bits: `/opt/gpt/sbin/gpt-build -force -nosrc -builddir=/tmp/globus_core.$$ gcc32 gcc32pthr gcc32dbg gcc32dbgpthr`
 - ◇ For 64bits: `/opt/gpt/sbin/gpt-build -force -nosrc -builddir=/tmp/globus_core.$$ gcc64 gcc64pthr gcc64dbg gcc64dbgpthr`
 - ◆ Create a tarball of the `/opt/globus` directory: `tar -C /opt/globus -czvf globus-4.0.8-VDT-1.10.1-5.tar.gz .`
 - ◆ Copy the packages + tarballs under:
 - `/afs/cern.ch/project/etics/repository/vdt/globus/VDT-x.y.z/platform`
 - ◆ Create the subsystem configurations in ETICS by cloning the last production version configurations.
 - ◆ Build the packages in the permanent repository.
 - ◆ Update the `glite_3*_cert` configurations to point to the new versions.

EMT agendas

EMT agendas should be prepared for the meetings taking place on:

- Monday at 14h30 in 28-R-015.
- Wednesday at 15h30 in 600-R-03.

You can reuse and clone any of the existing agendas under Indico[?].

The main sections you would like to discuss with developers are:

- gLite Releases:
 - ◆ status of Staged rollout and Production. Current and Upcoming releases.
 - ◆ any other issue related to the repositories and release notes.
 - ◆ relevant people: Mario David, Staged rollout responsible.
- Project Configuration:
 - ◆ status of glite_3_*_cert and glite_3_*_dev
 - ◆ relevant people: Eamonn Kenny, glite_3_*_dev responsible
- Documentation
 - ◆ changes on Integration guide, Developers guide and Savannah.
 - ◆ track open bugs in status `None`.
 - ◆ track major or critical bugs valid for all platforms.
 - ◆ copyright and license report.
 - ◆ relevant people: Francesco Giacomini who produces the copyright and license reports.
- Product Teams:
 - ◆ track any blocking issues that needs coordination among the different product teams.
 - ◆ track progress on missing SL5 services.
 - ◆ relevant people: PTs representatives
- ETICS:
 - ◆ track any bug blocking the developers.
 - ◆ announce new releases.
 - ◆ relevant people: Lorenzo Dini/Andres Abad.
- Bug priority classification:
 - ◆ Track any major or critical bugs opened in the last week and set the correct priority.

-- MariaALANDESPRADILLO - 26-Apr-2010

This topic: EGEE > ReleaseManager

Topic revision: r11 - 2011-02-07 - MariaALANDESPRADILLO



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