Production Release Check List

Responsible: Pablo Guerrero **Backup person**: Maria Alandes

Introduction

The following list must be used in combination with the Integration Procedure twiki that describes all the technical details involved in the release process. This twiki aims to list the necessary steps to create a production release helping to avoid mistakes and decrease human errors.

Check List template

Please, print the following check list templates to go through the production release process:

- Prod-check-list.doc

Pre requirements

Access to machine `lxbrf2713` where you have the necessary scripts to create repositories (https://svn.cern.ch/reps/grelscripts⬈)

The Glite GPG key is needed for signing in 3.2. More information can be found here CertificationSigning

Check List

1. Copy the production repo into the prepare area:

```
cp -pR /afs/cern.ch/project/egee/gLite/R3.x /afs/cern.ch/project/gd/www/glite/prepare/prod/3.x
```

2. For each patch in status `Ready for Production`:

- Check in the list of associated bugs, if any, that the bug status is `Fix certified` or `Fix not certified`. Remove Invalid/Obsolete bugs. Contact certifiers if there are bugs in `Ready for test` since they should update this status after verifying the bug.

- Check the release notes again and ask for an improved text to the patch submitter if the quality is poor or the explanations are not very useful.

3. Update the `Release` field of the patches with `3.X.0 Update *NN*`.

4. Insert the patches in the web DB and insert the updated RPMs. Before, be sure the release, architecture, service and service platform have been created, if any of them are missing the insertion will fail.

Send the savannah patch information to the web page:

```
savannah_to_web.py  -r <release> -o <os> -a <arch> <patch1 [... patchN]>
```

Run the preview_to_production script in test mode and send the updated rpms and urls to the web page:

```
test_preview_to_production_to_web.py  -r <release> -o <os> -a <arch> <patch1 [... patchN]>
```

5. In `lxbrf2713`, run the script to create the production repository:

```
preview_to_production.py  -r <release> -o <os> -a <arch> <patch1 [... patchN]>
```

6. Copy the production repo in the prepare area to the production area:

```
cd /afs/cern.ch/project/egee/gLite
rm -Rf R3.X_old
cp -pR  /afs/cern.ch/project/gd/www/glite/prepare/prod/3.1 R3.X_new
```

7. If the release affects the UI and/or the WN, copy the tarballs from the staged rollout area to the production area .

```
Copy from

/afs/cern.ch/project/gd/www/download/relocatable/beta/glite-UI/(SL4_i686 or SL5_x86_64)
glite-UI-3.X.Y-0.tar.gz
glite-UI-3.X.Y-0-external.tar.gz

/afs/cern.ch/project/gd/www/download/relocatable/beta/glite-WN/(SL4_i686,SL4_x86_64 or SL5_x86_64
glite-WN-3.X.Y-0.tar.gz
glite-WN-3.X.Y-0-external.tar.gz

to
/afs/cern.ch/project/gd/www/download/relocatable/glite-UI/(SL4_i686,SL4_x86_64 or SL5_x86_64)
/afs/cern.ch/project/gd/www/download/relocatable/glite-WN/(SL4_i686,SL4_x86_64 or SL5_x86_64)
```

8. Create the update, attach the patches, format the release notes and copy the short description to the field for every patch.

- Log in into the admin interface: https://glite.cern.ch/admin/⌤
- Add a new update and fill Number, Release, Platform and Priority. Don't mark the published check box !!!
- Patches:
    - Every patch should be already in the web page containing the information from savannah and the repository info
    - In every patch, select the update you just created and save it.
    - Use the release notes editor to edit them. It's a good idea to review the HTML that has been generated.
    - Copy the short part to the short release notes. It's a good idea to copy and paste the HTML instead of doing it directly, keeps better the format.

9. Add any know issues if needed

- Log in into the admin interface: https://glite.cern.ch/admin/⌤
- If there are any known issues, add a new know issue and fill the information.

10. Do a final check of the contents of the release pages:

- Main page (ej. http://glite.cern.ch/R3.2/sl5_x86_64/updates/21/⌤):
    - Check that a summary for each patch is present and well formated.
    - Check the list of patches is complete according to the original mail sent by Operations.
    - Check the priority of patches is correct. If a patch is priority High the corresponding affected services should be in red, stating `High`.

- Service specific update pages (ej. http://glite.cern.ch/R3.2/sl5_x86_64/glite-BDII_top/3.2.10-3/⌤):
    - Check the release notes make sense for the service and that only the patches affecting the service are actually described.
    - Check the list of rpms is correct by comparing it with the rpm list in the relevant patches.
    - Check the list of bugs is correct by comparing it with the bug list in the relevant patches.
    - Check the URLs of the rpms are correct, in the update page and in the rpm list page (both

HTML and txt).

11. Send a preview mail to `project-eu-egee-middleware-emt` and `gd-release-team`. In the subject write `Preview 3.X Production Update XX`. They'll check that everything is OK and they'll give the green light to publish the release. The mail should be something like:

```
Dear all,

Please find a preview of the production release notes here:
<< Preview URLs here >>

Any input/feedback provided as a reply to this mail would be valuable and is therefore welcome.

Cheers,
The Integration Team
```

12. In case any of the patches provide a fix for a security issue identified by GSVS, contact Linda Cornwall to get the URL for the Security Advice. Modify the web pages to include this URL as soon as it's available.

13. Publish the new repository (every 30min past full hour glitesoft is synchronized):

```
cd /afs/cern.ch/project/egee/gLite
mv R3.X R3.X_old
mv R3.X_new R3.X
```

14. Publish the release notes

- Log in into the admin interface: https://glite.cern.ch/admin/ 🡥
- Go the update you are creating and mark the published? checkbox, set the date, and save it.

15. For each release patch change the Savannah status to `In production`. Change the attached bugs status to `Ready for Review`, that's needed also for the bugs attached to the internal patches released. This can be done using the SCLI. Please, download the script in the location where you normally have `org.glite.savannah.viewer/cli/`:

```
svn co http://svnweb.cern.ch/guest/aelwell/savannah_cli/trunk
cd trunk/
 ./ReadyForReview.sh -p PATCH_NUMBER
```

16. If there are new node types released, remember to:

- Create the repo files under
  `http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.X/`.
- Add in the Generic Install Guide any relevant information about the new node type where needed.

17. Announce the Release using the Operations portal.

- Go to the Operations portal 🡥.
- Select glite Annoucement Lists, LCG Rollout, OSG, WLCG Tier-1 contacts, VO managers, Site Administrators.
- Add in the CC: `gd-release-team`.

```
SUBJECT:

gLite Release of UPDATE nn to gLite 3. x. Priority: [Normal/HIGH/**URGENT**]

MESSAGE:

Dear gLite users,
```

```
UPDATE nn for gLite 3.x is now ready for production use.
The priority of the updates is: [Normal/HIGH/**URGENT**]

The highlights of the update are:

-
-

All details of the update can be found in:
[COPY AND PASTE updates URL here]

Note that the UI and WN tarballs are available from the UI and WN pages.

Remember to report any issues with this set of updates using GGUS: www.ggus.org

Best regards,
The gLite Release Team and Mario David EGI TSA1.3
```

19. Check the retirement calendar and update the dates. http://glite.cern.ch/support_calendar/⬚

20. Put a symlink to `empty` in the beta repo for the released services.

21. Create a backup of the release you just created, in /srv/glite_web and logged in afs, execute: ./backup.sh

In the case you need to modify the old pages, mount them with: mount -t davfs https://dfs.cern.ch/dfs/Websites/g/glite⬚ /mnt/glite_repository

The user is: cern\glbuild

This topic: EGEE > ProdRelCheck
Topic revision: r84 - 2011-08-12 - unknown