

Table of Contents

gLite Release Procedures	1
Workflows	2
Certification Staged Roll-out Production.....	2
Roll-back during staged roll-out.....	4
Release of a subset of the patched in a bundle (partial release).....	5
Templates	7
Template for announcing the release of the update to the production service (using the CIC Portal broadcast tool).....	7

gLite Release Procedures

Workflows

Certification Staged Roll-out Production

Actors:

- MU: The gLite Release Team
- OU: The Operation Managers
- EA: The Early Adopter site(s)
- The production site(s)

Summary:

- MU certify and distribute a new update.
- OU request the EA sites to upgrade.
- EA sites update the relevant services and report.
- After a quarantine period the release is rolled-out to the wide production service.

Objects:

- *patch*: the Savannah document used to track a change in the middleware (Ref. the [jra1mdw patch tracker](#))
- *delta repository*: AKA "Beta" repository yum repository containing the newly introduced rpms
- *production repository*: yum repository used by the production sites * *preview release pages*: web pages with release notes for the updates object of the staged roll-out including lists of rpms (Ref. the [preview release pages](#))
- *production release pages*: web pages with release notes for the updates to production including lists of rpms (Ref. the [production release pages](#))
- *roll-out task* : the Savannah document used to track a roll-out work order to the EA sites. it includes the task report as an attribute (Ref. the [sa1dep task tracker](#))
- *task reports page (management view)*: web page summarising the status and outcome of the Roll-out tasks (Ref. the [task report \(management view\)](#))
- *task reports page (production view)*: web page summarising in a node-type-oriented fashion the status and outcome of the Roll-out tasks for patches actually moved to production (Ref. [\[\[\[the task report \(production view\)\]\]\]](#))

Steps:

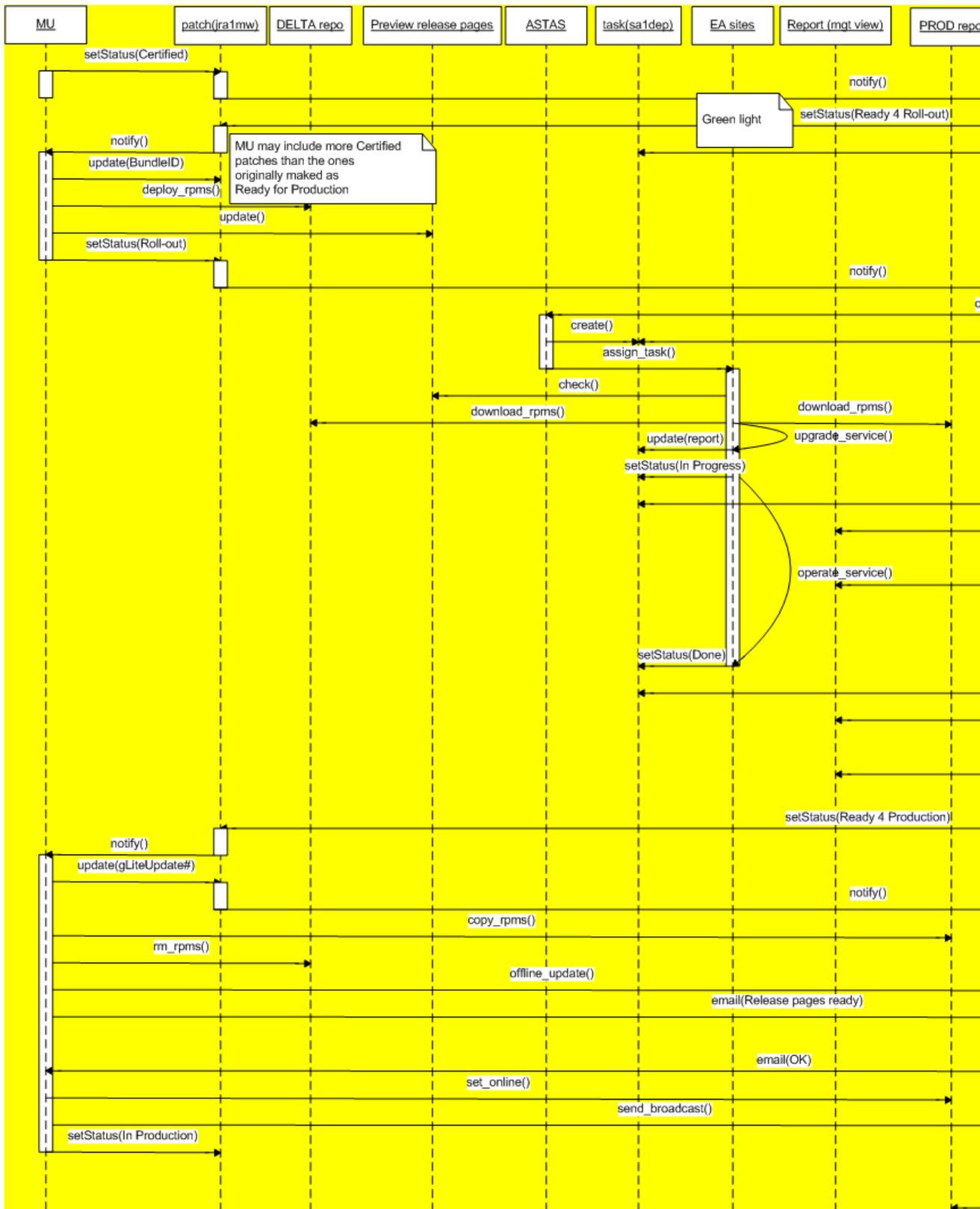
1. MU: Set status of one or more patches to 'Verified'
 - ◆ Savannah notification(s) sent to `grid-deployment-managers@cern.ch`
2. OU: If the staged roll-out can be accepted (e.g. non conflict in use of the repositories) set status of one or more patches to 'Ready for Rollout'
 - ◆ NOTE: this is not a validation of the certification by the OU. This transition of at least one patch to 'Ready for Rollout' has the value of a green flag to the MU for the deployment in the delta repository
3. MU: Assign a BundleID to the patch(es)
 - ◆ NOTE: at this stage MU can decide to insert other verified patches in the bundle (even if not marked by the OU as 'Ready for Rollout')
4. MU: Deploy the rpms corresponding to the patches identified by the BundleID in the delta repository
5. MU: Update the preview release pages
6. MU: Set the status of the patch(es) to 'Rolling out'
 - ◆ Savannah notification(s) sent to `grid-deployment-managers@cern.ch`

7. OU: If the Bundle is complete (patches in 'Rolling out' = patches in preview release notes) Create the Roll-out tasks and assign them to the EA Sites
 - ◆ IMPORTANT NOTE: before submitting the tasks with ASTAS OU needs to wait a couple of hours in order to let the information about changed status and bundleID be propagated to the local copy of the Savannah database
8. EA sites: Update service following the preview release notes and using both the production and the delta repository
 - ◆ Issues to be reported a GGUS tickets and Savannah bugs (if applicable) using Bug Detection Area= "Beta Service" AND mentioned in the task report (field in the Roll-out task).
 - ◆ After the update set the status of the Roll-out task to "In Progress" and an appropriate value for the outcome [Success|Fail|Warning]
 - ◆ After the quarantine period indicated in the task (typically one week) set the status of the Roll-out task to "Done" and an appropriate value for the outcome [Success|Fail|Warning]
9. *[In parallel]* OU: Follow-up the staged roll-out taking appropriate actions if needed
 - ◆ Check the task reports page (management view)
 - ◆ Eventually reject patches and Start Roll-back procedures
10. OU: Set the status of the patch(es) to 'Ready for Production'
 - ◆ NOTE: at this stage OU can decide to move forward patches in 'Rolling-out' which were originally included in other bundles
11. MU: Verify that the Bundle is complete (patches in 'Ready for Production' = patches in preview release notes)
 - ◆ If not corrective actions are needed affecting the delta repository and the preview release pages
12. MU: Assign a gLite Update Number to the patch(es)
 - ◆ Savannah notification(s) sent to `grid-deployment-managers@cern.ch`
13. MU: Move the rpms identified by the patches in 'Ready for Production' from the delta repository into the production repository
14. *[In parallel]* OU: Publish relevant task reports in the task reports page (production view)
15. MU: Update the production release pages (off-line copy)
16. MU: Notify the OU that the release notes are ready
 - ◆ e-mail sent to `grid-deployment-managers@cern.ch`
17. OU: Approve the release notes
 - ◆ reply to e-mail sent to `gd-release-team@cern.ch`
18. MU: Set the production release pages and the production repository on-line
19. MU: Announces the update to the production sites via the CIC portal broadcast tool using the broadcast template.
20. Production sites: Update service following the production release notes, eventually consulting the task report page (production view) and using the production repository

▣ Show Sequence Diagram ▣ Hide Sequence Diagram

- Interaction MU/OU (sequence diagram):

ReleaseProcedure < EGEE < TWiki



Roll-back during staged roll-out

Actors:

- MU: The gLite Release Team
- OU: The Operation Managers

Roll-back during staged roll-out

- EA: The Early Adopter site(s)

Summary:

- A patch is rejected during the staged roll-out
- The MU change delta repository and documentation accordingly
- Some EA sites have to roll-back
- Some EA sites need to synchronise

Objects:

- *patch*: the Savannah document used to track a change in the middleware (Ref. the [jra1mdw patch tracker](#))
- *delta repository*: yum repository containing the newly introduced rpms (Ref. [\[\[the delta repository\]\]](#))
- *preview release pages*: web pages with release notes for the updates object of the staged roll-out including lists of rpms (Ref. [\[\[the preview release pages\]\]](#))
- *roll-back task* : the Savannah document used to track a roll-back work order to the EA sites. it includes the task report as an attribute (Ref. the [sa1dep task tracker](#))

Steps:

1. OU: Set status of one or more patches to 'Rejected'
 - ◆ Savannah notification(s) sent to `gd-release-team@cern.ch`
2. MU: Delete the corresponding rpm(s) from the delta repository
3. MU: Re-create the affected meta-package(s) to include the previous version of the offending rpm(s) increasing the meta-package version number (e.g. the build number)
4. MU: Update the preview pages (same bundle ID) with the time-stamp of the change and possible specific instructions to the site for the rollback
5. MU: Notify the OU that the preview release pages are ready
 - ◆ e-mail sent to `grid-deployment-managers@cern.ch`
6. OU: Based on the affected services of the rejected patch(es) Create the Roll-back tasks and assign them to the EA Sites
7. EA sites: Roll-back the service following the instructions in the preview release pages
 - ◆ Issues to be reported a GGUS tickets AND mentioned in the task report (field in the Roll-out task)
 - ◆ After the roll-back set the status of the Roll-back task to "In Progress" and an appropriate value for the outcome [Success|Fail|Warning]
 - ◆ After the quarantine period indicated in the task (typically three days) set the status of the Roll-out task to "Done" and an appropriate value for the outcome
 - ◆ *NOTE: with the current structure of the repositories EA sites running node types only indirectly affected by the change are requested to roll-back for consistence (e.g. problem in YAIM core affecting the UI indirectly affects the WMS) . This should be dramatically mitigated if the repositories are split*

Release of a subset of the patched in a bundle (partial release)

Actors:

- MU: The gLite Release Team
- OU: The Operation Managers
- EA: The Early Adopter site(s)

Summary:

- A subset of the patch(es) released with a bundle are moved to production while the others are left some more time with the EA sites (not rejected)
- MU re-build the affected meta-package(s) (e.g the UI includes a VOMS client that is released with another patch)

Objects:

- *patch*: the Savannah document used to track a change in the middleware (Ref. the jra1mdw patch tracker [☞](#))
- *delta repository*: yum repository containing the newly introduced rpms (Ref. [\[\[\[the delta repository\]\]\]](#))

Steps:

- MU: Recreate the meta-package fixing the dependencies in the way that versions of dependent rpms which are not released to production are not included. The version number of the meta-package is left unchanged
 - ◆ *NOTE: the decision to change the content of the mpkg without changing the number has as main motivation not to invalidate the site reports produced up to that stage (to be checked by production sites). The possibility to use different mpkg names (e.g. pps-glite-*) is excluded because after the process EAs must have a fully commissioned production site with no need to update again*
 - ◆ *NOTE: This is a sub-optimal situation that introduces some risks. So it should be possibly avoided. e.g. by identifying very focused bundles that can be moved back and forth as a whole. This issue should be dramatically mitigated if the repositories are split per node-type*

Templates

Template for announcing the release of the update to the production service (using the CIC Portal broadcast tool)

TO:

Publish in the CIC Portal news, CIC-on-Duty, OSG, ROC managers, VO managers, Production site administrators
support-eis@cern.ch, glite-announce@cern.ch

SUBJECT:

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

MESSAGE:

Dear members of the EGEE Grid Production Service,

UPDATE **nn** for gLite 3.**x** has been released to the EGEE Grid production service.
The priority of the updates is: **[Normal/HIGH/**URGENT**]**

All details of the update can be found in:

<http://glite.web.cern.ch/glite/packages/R3.x/updates.asp>
(services supported on 32b architectures)

and

http://glite.web.cern.ch/glite/packages/R3.x/x86_64/updates.asp
(services supported on 64b architectures)

The updated middleware services with new version numbers are given on the Updates web page.

For more detailed information including fixed bugs, updated RPMs, configuration changes and how to deploy, please go to the 'Details' link next to each service on the 'Updates' web page.

**** PLEASE ALSO READ THE NOTES AND 'KNOWN ISSUES' SECTION FOR EACH SERVICE. ****

Note that the UI/WN tarball is available from the individual service pages
(click on the service name on the main page).

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

Best regards,
The gLite Middleware Teams

This topic: EGEE > ReleaseProcedure

Topic revision: r7 - 2010-03-09 - unknown



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