Table of Contents

How to certify a patch *within a product team*

(Taken from Gianni's Revision 22 Feb 2010)

Introduction

In general, all gLite middleware patches are managed by CERN Savannah - https://savannah.cern.ch/. For each individual patch, the link on Savannah is https://savannah.cern.ch/patch/?XXXX.

The workflow for product teams to release patches is at Integration Guide - The main changes from the pre-product team era are that ProductTeams now assign the patch internally to a certifier (although Andrew Elwell will still assist to find someone at the moment), and that they are responsible for creating the yum repositories themselves (no longer a 'ready for integration' stage)

When a new patch is submitted to Savannah by the developer of a product team it can either be as a component needed by another product team (Release Path: **Internal**) or a full metapackage (Release Path: **Standard**). The certification process is slightly different in the two release path; in the following sections we will highlight the main differences.

General guidelines

These guidelines are always valid, no matter the release path of the patch. You should have an account on CERN Savannah. If not, please create it and join the groups *jra1mdw* and *egee-sa3*. (In case of Savannah group problems, please contact Andrew Elwell).

Use SL and **not SLC** on the machines used to certify patches. If you want to keep a detailed report of the certification, please use the CertificationReportTemplate, and link it to the right section in the SA3Testing page.

In principle, any high priority patch should be certified as soon as possible. Throughout all the process regularly add comments about the patch on Savannah, for example to state the results, problems you found during the certification; unexpected messages etc etc. Don't hesitate to contact the developer if needed. For any problems found in certification, please submit Savannah bugs for them and change the "Bug detection area" to *Certification*. A certifier should also be aware of the current bugs open for the component he/she is certifying in order not to open bugs that have already been reported. In any case, before opening a new bug, it is always suggested to have a look at the list of open bugs for a given component.

Most of the certification at CERN is done using the vNode servers which can be quickly commissioned and configured using yaimgen. Partner sites should have their own methods of building a testbed (physical or virtual machines). Any reference to 'machine' should be interpreted as components of your testbed, virtual or otherwise. To integrate your nodes into a shared testbed as described at EGEE09, please contact Tomasz Wolak or Louis Poncet.

Finally, if there are any parts of the certification process or documentation that is unclear then raise a bug against it!

When a patch is assigned to you:

1. Change the task and patch status

As soon as you start working on a patch, update the task to **In Progress** in Savannah and change the patch status to **In certification**.

2. Check patch details

Check carefully if the relevant information provided by the developer or patch submitter about the patch is correct and complete on https://savannah.cern.ch/patch/?XXXX.

HowToFillAPatch contains full details and the mandatory fields.

The *Release Path* field can be

- *Standard*: there should only be **one** metapackage listed in the *Affected metapackages* field.
- *Internal*: there can be multiple metapackages listed in the *Affected metapackages* field.

3. Install and test the patch

3a. Standard Release Path

In what follows *the service* is the metapackage listed in the *Affected metapackages* field in Savannah.

You have to install and test the service performing both clean installation and upgrade from production. The steps involved are described in the following sections. For installing the services follow GenericInstallGuide310 or GenericInstallGuide320.

3a-1. Test clean installation

Install the service (*yum install -metapackage name-*) on a clean SL machine with both the production repository and the patch repository (the one created by ETICS, whose link has to be provided within the patch) enabled. Check that the RPMs installed by the patch match the list in the *RPM Name(s)* field in Savannah. Configure the service using Yaim.

Run the certification tests has explained in the section #Running_certification_tests. All the mandatory tests have to pass.

Record all the steps performed and the outcome of the tests in the certification report or as a comment in a Savannah patch.

3a-2. Test upgrade from production

Install the production version of the service (*yum install -metapackage name-* with the production repository enabled) and verify that it works properly (perhaps running the certification tests explained in section #Running_certification_tests).

Upgrade the service (*yum upgrade*) enabling the patch repository created by ETICS, whose link has to be provided within the patch.

If required by the field *Metapackages to be reconfigured* in Savannah, re-run YAIM to configure the node type. In the case that a new YAIM is required, please make sure the new YAIM is available and installed.

Restart services if it is required by the Savannah field *Metapackages to be restarted*. Testing by rebooting machine may be needed as well. (do all services start as planned? is chkconfig correct?).

Run the certification tests has explained in the section #Running_certification_tests. All the mandatory tests have to pass.

Record all the steps performed and the outcome of the tests in the certification report or as a comment in a Savannah patch.

3b. Internal Release Path

In this release path the patch provides rpms that will be used by one or more product teams that are different from the one who provided the patch. The patch will not be release to production, it will instead be used by a node-type patch following the standard release path. The patch can have multiple items in the *Affected metapackages* field in Savannah.

In this case, it is not mandatory for the certifier, to test the patch on ALL the affected metapackages, since this could imply an effort to big in case of components that are used by many node types.

Nonetheless the certifier has to run all the necessary tests that the product team responsible for the patch has available and that are necessary to consider the patch good for production use.

4. Verify bug fixes

Check that each bug listed in the field *Depends on the following items* of the patch in Savannah is fixed. For some bugs, checking the fix can be too complex or not necessary; in this case you can skip this step. The page BugNonVerification contains some hints to make a proper judgment. You should also see if you can create any Regression Tests

4.a Update each Savannah bug record

For every bug add a comment into the Savannah bug page explaining how you tested the fix. If you decided to skip the bug fix checking, mention the reasons (e.g "This bug fix cannot be tested due to...").

Update the status of each bug attached to the patch:

- if the bug is fixed, change the status to *Fix Certified*
- if the bug is not fixed (i.e, it can be reproduced after installing the patch) there are two possibilities:
  - ♦ change the status back to *None* and remove it from the list of attached bugs for the patch. Go on with the certification.
  - ♦ change the status back to *None* and reject the patch changing the patch status to *Rejected*

The choice between these two possibilities has to be agreed within the Product Team. In any case also add a comment to the patch ('rejected due to bug ### not fixed' or 'detached bug ###, not fixed') and in the bug explaining the steps you did to check the bug fix.

- If you are unable to verify the bug fix then you can change the bug status to *Fix not Certified* and go on with the certification. In the bug, add a comment explaining the reason why it was not possible to check this bug.

5. Change patch status

Change the patch status to *Certified* when you have confidence that the patch works fine. You should have, at least:

- successfully run the mandatory tests
- verified all the bug fixes and moved the bugs status to either *Certified* or *Fix not certified*.

If the patch makes the situation worse than before, or does not fix an important bug, then you should reject the patch or change the status to *With provider* so that the developers can work on it. (The other transition that may occur here is the move to *Obsolete*, ie a new patch is now in *Ready for Certification* that replaces this one: The Patch Coordinator can advise on this).

If you produced a certification report, please link it as a comment in the Savannah patch, and in the right section of the SA3testing page.

The transition of a patch to *Certified* will notify the verification team who will then use the process in PatchVerification before moving the patch to *Verified* ready for signing. (see the Developers Guide A list of verified patches is maintained at InternalPatch). A product team is also responsible for notifying the EMT so that the glite_3_X_cert branch can be updated in ETICS to reflect the new patch availability.

6. Change the task status

Once all the patches associated with the task you have been assigned, then update the task with the effort spent on certification (in hours) and change the status to *Done* (any overdue tasks are tracked in the weekly phone meeting).

7. Terminate all your machines

Terminate any running machines that you were using to certify this patch to free resources for others (especially valid for vNode users at CERN).

Running certification tests

Test the basic functionalities to check whether the patch breaks any service. Verify the new functionalities if the patch provides them, and check that that they are properly documented. Due to their complexity, performance and stress tests should be done only in special cases. If the patch affects a service which advertises in the information system, check that the information is valid (more info in the following section).

For each service, the page SA3Testing provides the list of mandatory tests that must be executed. All these tests must pass before declaring a patch certified.

You **must** record all the tests run and their outcome as a comment in the Savannah patch, or as a wiki page linked to patch. A template for the CertificationReportTemplate is available.

Check published information

If applicable, please also check if published information are correct after upgrading to the new patch (perhaps including a diff between the versions of published info), and also log and error messages are meaningful. The Gstat2 validation scripts can be used to check the information published by a service. Use gstat-validate-ce to check information published by a Computing Element, gstat-validate-se to check information published by a Storage Element, and gstat-validate-service for any other service.

-- GianniPucciani - 09-Feb-2010

This topic: EGEE > HowToCertifyAPatch
Topic revision: r58 - 2010-06-10 - GianniPucciani