MSA3.6 – Developers' guide

Andreas Unterkircher, CERN

JRA1/SA3 All Hands Meeting

www.eu-egee.org

e-infrastructure

Content

- Updates to the current process based on current version of ETICS. Deliverable MSA 3.6 „Developers‘ Guide“
- Possible future updates based on desired ETICS features (currently not available).

Developer Guide

- **EU deliverable MSA 3.6 „Developers‘ Guide“ https://twiki.cern.ch/twiki/bin/view/EGEE/DevelGuide**
- **Contains Section Release Workflow**
  - Understanding the release
  - Project configurations
  - Integration scenarios
  - How to handle bugs
  - How to handle patches
- **Proposed updates of the process based on current status of ETICS**
  - Might improve if ETICS implements certain features

Current status & problems

- **gLite arranged as subsystems in ETICS. Subsystems should**
  - Group together components with synchronized evolution
  - Contain no other unreleated components
- **Not always the case**
- **Production release is approximated in ETICS by a project configuration (glite_branch_3_1_0)**
- **Subsystem configurations used to select the versions of components that should be in the release**
  - Additional layer of abstraction
- **Environment per platform**
  - Specify the DEFAULT property for externals

Current status & problems

- **Default scenario (majority of cases):**
  - Developer works on one (some) component(s) and builds against glite_branch_3_1_0.
  - No conflicts with other subsystems or externals.
  - Described in detail in the Developers‘ Guide.
- **Difficult scenarios**
  - Developer works on a component that depends on other components not (yet) in glite_branch_3_1_0.
  - Component conflicts with other subsystems.
  - Component needs update of externals that affect other components.
  - Recents cases: WMS/LB, CREAM.

Updated process

- **Three configurations maintained by release team.**
- **glite_branch_3_1_0**
  - Production release.
  - Updated every time a release to production is being made.
- **glite-branch_3_1_0_cert**
  - Contains recently certified patches („certified“ status & PPS).
  - Not yet released to production.
- **glite_branch_3_1_0_dev**
  - Development branch, latest tags of interest to developers.

Updated process

- **Default scenario**
  - Build against glite_branch_3_1_0.
  - Even for the case of updating a single component a new subsystem configuration is needed.
  - Subsystem configurations should be consistent with production. Preferable is one configuration per patch.
  - If several versions of the same subsystem configuration are in production the newest is being added to glite_branch_3_1_0.
  - ETICS deployment test required before setting the corresponding patch to „Ready for Integration“.

Updated process

- **Difficult scenarios**
  - Build against glite_3_1_0_cert if not yet released components are needed.
  - Developers can request a clone of glite_branch_3_1_0 and do their own modifications. A patch 1234 „With provider“ has to be created and the cloned configuration is named glite_branch_3_1_0_patch1234.
  - Be aware that glite_branch_3_1_0 may change while you work on glite_branch_3_1_0_patch1234.
  - Conflicts have to be discussed in EMT.
  - ETICS deployment test required before setting the corresponding patch to „Ready for Integration“.
  - When the patch finally is in production the corresponding configuration will be deleted.

Updated process

- Guidelines on how to handle patches and bug also described in the Developers' Guide (cf. Oliver's talk).
- Guidelines on how to fill a patch: https://twiki.cern.ch/twiki/bin/view/EGEE/HowToFillAPatch

Future updates: deployment tests

- **ETICS deployment test**
  - Prototype available but no fully working version (promised since May 08).
  - Ensures that newly generated rpms are installable in production.
  - Installs a production node type (repo file as argument) and updates this node type with the newly generated rpms (volatile area).
  - A test has to be submitted for every node type (ETICS cannot dynamically create several subtests).
  - Developer must kow which node types are affected by the produced rpms (Is this the case?). We could produce a meta script that launches several „etics-test“ commands.

Deployment tests

- **Deployment test command example:**
  - `etics-checkout -c upgrade-test_R_1_1_0_1 upgrade-test`
  - `etics-submit test --runasroot -e HTTPD_USER=nobody -e HTTPD_USER=nobody -p nodeName=glite-WMS -p repoURL="<URLs for repos to install prod node> -p pkgListUrl="<URL to rpm list provided by ETICS>" -c upgrade-test_R_1_1_0_1 upgrade-test`
  - `etics-submit test --runasroot -e HTTPD_USER=nobody -e HTTPD_USER=nobody -p nodeName=glite-WMS -p repoUrl="http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/glite-WMS.repo,http://grid-deployment.web.cern.ch/grid-deployment/download/relocatable/test/cern.repo" -p pkgListUrl="http://grid-deployment.web.cern.ch/grid-deployment/download/relocatable/test/org.glite.wms-glite-wms_R_3_1_100-rpm.txt" -c upgrade-test_R_1_1_0_1 upgrade-test`

Future updates: individual components

- Currently a component must be part of one and only one subsystem (well, Akos knows about a „bug“ in ETICS...).
- In the EGEE 08 conference ETICS announced a planned feature to let subsystems become systems of independent groups of selectable components. That allows a component to be in multiple subsystems.
- Developer creates a subsystem „patch1234“ that contains only the updated components.
- With this we would not need a diff tool for ETICS configurations?

Future updates: multiple rpms per subsystem

- **Currently only one package (rpm) can be created by a configuration within ETICS**
  - But subsystems can produce several rpms (VOMS, DM).
  - These rpms need be addressed individually for runtime dependencies.
- **On the EGEE 08 conference ETICS announced a planned feature „virtual-packages“ to deal with this issue. Allows a component configuration to have package subconfigurations.**

Future updates:installation of externals

- **ETICS externals dependencies don‘t correspond to system dependencies.**
  - E.g. Installing the mysql-devel rpm triggers installation of several other rpms (mysql, openssl-devel,...)
  - ETICS installes different externals in different directories (different from, say, /usr/lib)
  - Current workaround: fat tarballs (mysql-devel tarball contains parts of mysql). Time consuming, not generic.
  - Solution: ETICS provides chroot environment or (better) a virtual machine & installation of dependencies via OS mechanisms (yum install).