ETICS
The Grid Quality Process

All you ever wanted to know about ETICS, but never dared to ask

Alberto Di Meglio (CERN)
ETICS Project Manager

www.eu-etics.org

Information Society and Media

Overview

- Current Status
- Overview of common mistakes
- New functionality
- Your use cases

Current Status of ETICS

- Current version of the client: 0.7.4-1
- Main server: etics.cern.ch

- Deployment of ETICS 1.0 RC being done at CNAF
- Client version: 0.8.4-1
- Main server: etics-06.cnaf.infn.it

- It can be used for tests, but data on etics-06.cnaf.infn.it will not be preserved

- It should become a production server during the week as soon as some last firewall problems are solved

Current status of gLite

- **Regular builds done on SLC3/32, SLC4/32 and SLC4/64**
- **Until now managed by ETICS people, now being transferred to SA3**
- **Current build success rate:**
  - SLC3 (ia32): 86%
  - SLC4 (ia32): 70%
  - SLC4 (x86_64): 62%
- **Regular builds done for metapackages and subsystems, managed by SA3**

Most common issues

- **Missing dependencies: the old build systems was attaching all dependencies to the subsystem, components didn't need to know about them. This has changed, all components must manage their own list of dependencies. If a component fails, first check if it has the correct information**

Most common issues

- **Failed dependencies: A dependency doesn't build and caused a chain reaction**

Most common issues

- **Install command: in the old system the install command was rarely used. Now a working install command is mandatory to produce well-behaved packages and to stage the files. If a component fails, check if its dependencies are installing all the necessary files**
- **This is often a problem with include files for example. In the past having them in the stage was enough. Now they need to be handled by the installed command or building from pre-compiled packages doesn't work**

Most common issues

- **Compilation errors: this is a problem especially on SLC4/32 and SLC4/64. The compiler is different and stricter. If a component fails and all dependencies are in place and all necessary files are staged, check if old warnings have now become errors**

Most common issues

- **Missing Externals: this is mainly a problem on platforms different from SLC3/32 and SLC4/32. Externals for SLC4/64 are being added, for other platforms most of them are probably still missing**
- **Building the externals from source is of course possible, but sometimes the result is not the same as that obtained by using a package from a distribution (e.g.: boost or mysql)**

Most common issues

- **Missing Documentation: this is certainly a problem. We were planning to release the documentation already two weeks ago, but it always gets delayed. This is now my personal priority**

Most common issues

- **What has changed?: this question recurs frequently. There are two different issues:**
  - Sometimes things do change, but proper auditing and logging is not accessible to users: we are fixing this, it will be available in April
  - Other times things don't change, but the system behaves differently because of the different commands used.

Most common issues

- **The main example of the second case is the difference between**
  - etics-checkout –c glite_branch_3_1_0; etics-build org.glite.wms
  - etics-checkout –-project-config glite_branch_3_1_0 org.glite.wms; etics-build org.glite.wms
- **In both cases the current configuration of org.glite.wms in the 3.1.0 tree is built**
- **However, in the first case ALL glite components are built from source, in the second case only WMS components are built from source**
- **Therefore if a non-WMS package was built at least once, but now is failing, the second command works, the first one doesn't**

Most common issues

- Of course at this point your question is: why was that component built once and now doesn't build anymore
- The answer is: because the way gLite builds are managed is not fully reproducible
- All changes are always injected in the branch glite_branch_3_1_0, therefore dependencies may change overtime and introduce differences in the build
- A possible solution is to create new project and components configurations every time a change affects the way a component is built.

Most common issues

- **Another solution is to use the --frombinary option**
- **Using this option all components for which a package already exist are taken from binary and the sources are only used if the package has never been built**
- **Also this makes the two commands shown earlier to be equivalent**
- **As a general rule:**
  - Use --frombinary to verify a build and maintain consistency
  - Use the default behaviour for development when you need to checkout the sources
- **So, why this is not done? Simply because during the migration it was necessary to run full builds to make sure that compilation problems were not overlooked**

Real examples

- **org.glitergma.api-cpp**
  - cp /home/condor/execute/dir_23685/userdir/org.glite.rgma.api-cpp/build/scripts/rgma-client-check/producer /home/condor/execute/dir_23685/userdir/org.glite.rgma.api-cpp/etics-tmp/libexec/rgma-client-check/C++
    cp /home/condor/execute/dir_23685/userdir/org.glite.rgma.api-cpp/build/scripts/rgma-client-check/consumer /home/condor/execute/dir_23685/userdir/org.glite.rgma.api-cpp/etics-tmp/libexec/rgma-client-check/C++
    03/06/07 08:38:54.485 INFO main [write] - cp: cannot stat `/home/condor/execute/dir_23685/userdir/org.glite.rgma.api-cpp/build/scripts/rgma-client-check/producer': No such file or directory
- **Cause: Install command**

Real examples

- **org.glite.wms.classad-plugin**
  - 03/06/07 08:51:03.844 INFO main [write] -
    03/06/07 08:51:03.937 INFO main [write] - glite/wms/ism/ism.h:
    No such file or directory
    retrieveCloseSEsInfo.cpp:30: unknown namespace
    `glite::wms::ism‘
- **Cause: Failed dependency**

Real examples

- **org.glite.wms.ns**
  - 03/06/07 08:56:13.386 INFO main [write] -
    03/06/07 08:56:14.274 INFO main [write] -
    CommandFactoryServerImpl.cpp:70:49:
    glite/security/proxyrenewal/renewal.h: No such file or directory
    03/06/07 08:56:14.276 INFO main [write] -
    03/06/07 08:56:14.372 INFO main [write] -
    CommandFactoryServerImpl.cpp:76:37:
    glite/wms/purger/purger.h: No such file or directory
    03/06/07 08:56:14.374 INFO main [write] -
    03/06/07 08:56:15.163 INFO main [write] -
    CommandFactoryServerImpl.cpp:91: unknown namespace
    `glite::wms::purger‘
- **Cause: Missing dependency**

Real examples

- **org.glite.wms.wmproxy**
  - 03/06/07 09:38:21.058 INFO main [write] - ./../src/utilities/.libs/libglite_wms_wmproxy_utilities.so: undefined reference to `glite::wms::common::utilities::quota::getFreeQuota(std::basic_string, std::allocator > const&)' ../../src/utilities/.libs/libglite_wms_wmproxy_utilities.so: undefined reference to `glite::wms::common::utilities::quota::getQuota(std::basic_string, std::allocator > const&)' collect2: ld returned 1 exit status
- **Cause: ???**

Real examples

- **org.glite.wms.wmproxy**
  - 03/06/07 09:38:21.058 INFO main [write] - ./../src/utilities/.libs/libglite_wms_wmproxy_utilities.so: undefined reference to `glite::wms::common::utilities::quota::getFreeQuota(std::basic_string, std::allocator > const&)' ../../src/utilities/.libs/libglite_wms_wmproxy_utilities.so: undefined reference to `glite::wms::common::utilities::quota::getQuota(std::basic_string, std::allocator > const&)' collect2: ld returned 1 exit status
- **Cause: ???**

Real examples

- **org.glite.wms-utils.tls (SLC4/64)**
  - 03/06/07 08:42:25.571 INFO main [write] - checking for /home/condor/execute/dir_17683/userdir/repository/externals/glob
    us/4.0.3-VDT-1.6.0/slc4_x86_64_gcc346/include/gcc64pthr/lber.h... no
    checking for ldap thr... "GLOBUS found no"
- **Cause: Missing external (VDT on x86_64 doesn't provide ldap components, need to add openldap)**

Real examples

- **mysql-client (SLC4/64)**
  - 03/06/07 07:11:33.156 INFO main [write] - The following error has been detected on configuration 'mysql-client v. 4.1.1 1': Failed to extract either binaries or sources for configuration 'mysql-client v. 4.1.11'
    03/06/07 07:11:33.157 INFO main [write] -
- **Cause: Wrong configuration (mysql on x86_64 must be v. 4.1.20, v. 4.1.11 doesn't exist)**

Main Changes in 0.8/1.0

- Different method to cache configuration information, only the configurations that are really used are stored, not the entire project. This considerably speeds up working with individual components or subsystems
- Plugins: independent programs that can be deployed in the client to extend the functionality (a la ECLIPSE). We provide some basic ones, but everybody is encouraged to produce their own (metrics, tests, mock objects, etc) following the ETICS plugin specifications
- Redesigned build reports: now are statically generated in the client, no need to use tomcat to display them and available also in the local builds
- Added dependency navigations to the report and anything produced by the plugins

Release Plans for 1.1

- Add usage logging and auditing
- Add configuration locking
- Add full support for Debian and Windows
- Refactor some bottlenecks to increase performance of the client
- Add the scheduler to manage automation of remote builds
- More plugins (but plugins are normally released asynchronously from the client)

Release Plans after 1.1 (April-September)

- Add the functionality to make local changes to modules and configurations and synch to database later (like the CVS commit)
- Introduce full Software Repository web service
- Redesign the web applications to be more intuitive and add the user portal
- Add co-scheduling to the test system, that is the possibility of executing remote tests that require the execution of commands on different machines

Testing

- ETICS is not only for building, but also (and especially) for testing
- The gLite metapackage deployment tests are being implemented in ETICS on SLC3/32, SLC4/32, SLC4/64
- After the deployment we will work with SA3 to implement more system and functional tests on all required platforms

Ales' Use Case

- Start with local build of a specific (last known working) configuration of a subsystem
- Fix a reported bug, check the fix, and commit into CVS
- Let another developer checkout the current code from the branch, and augment the bug fix eventually
- Create new tags and configurations, assuming the bug fix spans multiple components

Ales' Use Case

- **Subsystem: Catania_All_Hands (in testProject)**
- **Two components: HelloCPP, DepCPP**
- **HelloCPP depends on DepCPP**
- **Start with all HEAD configurations**
- **Run standard commands:**
  - etics-get-project testProject
  - etics-checkout Catania_All_Hands

Ales' Use Case

- **Case 1:**
  - Modify DepCPP as needed
  - Tag DepCPP using the command:

    etics-tag –c <config-name> --config-version <x.y.z-r> DepCPP

  - This command tags the code in CVS and creates the new configuration in ETICS cloning from the current one (HEAD in this case)

Ales' Use Case

- **Case 2:**
  - Modify DepCPP and HelloCPP as needed
  - Tag the Catania_All_Hands subsystem using the command:

    etics-tag –c <config-name> --config-version <x.y.x-z> --childlist <filename> Catania_All_Hands

  - This command tags the code in CVS and creates new configurations and parent-child relationships in ETICS cloning from the current configurations (HEAD in this case)
  - The file passed with the --childlist option contains one entry for each child to be attached to the subsystem in the form:
    - Component-name config-name x.y.z-r
  - Can be an extension of the gLite dependencies.properties.xml

Ales' Use Case

- **Note:**
  - The etics-tag command is not meant to be a solution for all possible cases
  - It is more an API that project maintainers can use to implement their own tagging scripts
  - For example you may need to tag several components at the same time, but not as part of a subsystem
  - It supports additional options like a configuration input file (the same format as used by etics-configuration) to create completely new configurations instead of cloning from the current one
  - We are adding some way of generating automatically the version info, for example options like --increasemajor, --increaseminor, etc