

“workload” software towards SCM

CVS

- Ask an AFS account to EGEE segretary
 - Fill up a module and fax it
- Before download software
 - check that **java** (suggested version 1.4.2_04) and **ant** (suggested version 1.6.1) be installed
 - export the following variables:
 - CVSROOT=:ext:<afs login>@jra1mv.cvs.cern.ch:/cvs/jra1mv
 - CVS_RSH=ssh
 - JAVA_HOME=/usr/java/j2sdk1.4.2_04 (java location)
 - ANT_HOME=/opt/ant-1.6.1 (ant location)

“workload” software CVS tree used in the porting exercise

- A sub-subsystem represents the union of several modules. All of them are at the same level in the CVS tree.
- In the porting exercise, we have:
 - as sub-systems
 - “org.egee.test3.wms”
 - as modules
 - “org.egee.test3.wms.common”
 - “org.egee.test3.wms.jcommon”
 - “org.egee.test3.wms.thirdparty-bypass”
 - “org.egee.test3.wms.thirdparty-loki”
 - “org.egee.test3.wms.thirdparty-globus_gridftp_server”
 - “org.egee.test3.wms.thirdparty-globus_ssl_utils”

How to set up a working area 1/3

- e.g login as egeeuser
- Fill up the file “**egee.build.properties**” with the following values:
 - platform=rhel30_gcc32
 - repository=<TOP location>/repository (see next slide)
- “**egee.build.properties**” stays in the user home
 - it contains all the properties specified with -D (see next slide)

How to set up a working area 2/3

- `mkdir jra1; cd jra1`
 - `cvs co org.egee; cd org.egee/project`
 - case 1: download all the packages and all the external packages
 - `ant -f egee.csf.xml`
 - case 2: download just “wms” sub-component and its external dependencies (at the moment, it does not work well, so we need to use case 1)
 - `ant -f egee.csf.xml test3.wms`
 - “test3.wms” is a temporary name used for the porting exercise.
 - “wms” should be the permanent name. **Is it fine?**
 - case 3: download just a “wms” module like `test3.wms.thirdparty-bypass` (possible, but it does not work well yet)
 - `cd ..; cd org.egee.test3.wms`
 - case 4: using the default target, that's “stage”
 - `ant`
 - case 5: using a different target from the default
 - `ant compile`
 - `ant stage`
 - `ant stage dist` (**it is possible to put more than one target after “ant”**)
 - case 6: using a particular property
 - `ant -Dbootstrap=true` (**the properties are the prefix -D**)

How to set up a working area 3/3

- case 1: download all the packages and the external packages under the directory called “repository”
 - ant -f [egee.csf.xml](#)
- Each developer during the setting up of his/her working area downloads the external packages under the directory “repository”.
- If we still want to use the same account (called “egeeuser” instead of “edguser”) for all developers on the same machine, and if we do not want to have N directories “repository”:
 - A master developer has to download the whole package under <TOP location>, running the following command:
 - ant -f egee.csf.xml
 - The other developers have to run the following command:
 - ant -f [egee.csf.xml](#) -Drepository=<TOP location>/repository
 - Another possibility is to put the following line
repository=<TOP location>/repository
in the file [egee.build.properties](#)
- The directory that contains **external packages** can be called differently by the standard (that's “repository”).

Version 1/2

- File version.properties contains:
 1. module.version 0.0.0
 2. module.age 0
 3. module.build 0
- The first two value are changed according to the release manager
 - module.age is constraint by the module.version (see next slide)
- The last value is automatically increased after every build process on the central server

Version 2/2

- Looking at how the “wms” release version goes ahead
- Suppose that “wms” is composed by the modules “common”, “tp”, and “ui”:

1. wms 1.0.0-0

- common 1.0.0-0
- tp 1.0.0-0
- Ui 1.0.0-0

When the first digit of the version changes, the fourth digit will come back to zero

2. wms 2.0.0-0

- common 2.0.0-0
- Tp 1.0.0-0
- Ui 1.0.0-0

One tag will be called e.g. As
wms_R_2_2_0_2,
wms_B_2_2_0_2,
wms_T_2_2_0_2.

3. wms 2.1.0-1

- common 2.1.0-1
- tp 1.0.0-0
- ui 1.0.0-0

B means baseline.

4. wms 2.2.0-2

- common 2.1.0-1
- tp 1.1.0-1
- ui 1.0.0-0

A changed version web interface will be ready within the end of June

CVS Access

- Developers have access to their own modules
- Team member and cluster manager have access to the sub-component and all its modules

Stage and Dist

- Stage is the target “make stage”. It is equivalent to a “make install prefix=<stage location>” (at the moment, we have “include” instead of “interface”)
- Dist contains two targets:
 1. “make distsrc” is equivalent to a “make dist” with the exception that the **egee-<module>_<version>_src.tar.gz** file is moved under the dist location;
 2. “make distbin” is equivalent to a “make stage” with the addition that the distbin location is tar gzcd in a file called **egee-<module>_<version>_bin.tar.gz** and moved under the dist location.
- Stage and dist location are at the same level of the working area
- Each module must be deployed via a src and bin tar files.

Close issues

- Stage and install trees have the same structure, because in stage we have include instead of interface.
- Lib and bin have a flat structure.
- Html documentation files, README file, and so on are installed under /opt/egee/share/doc/egee-<modulo>. **(The documentation should be only provided via Html files)**
- Start up script files, and other system files are installed under /opt/egee/sysroot/{etc/rc.d/init.d,proc/var,.....}.

Open issues

- Test locations?
- Include structure?
- In the next days, the porting made will be redone using the latest code version, (those in INFN FORGE CVS), and the test3.wms will be changed in wms. Do you agree?
- Then, when we will decide how to separate the current modules in simple ones, I will start porting the other modules under the new structure.
- Others ?