EUROPEAN MIDDLEWARE INITIATIVE

# EUROPEAN

# MIDDLEWARE INITIATIVE

## PACKAGING POLICY

| | |
|---|---|
| Document Version: | **2.1** |
| Date: | **23.01.2012** |
| Document Status: | **APPROVED** |

This work is co-funded by the EC EMI project under the FP7 Collaborative Projects Grant Agreement Nr. INFSO-RI-261611.

# Table of Contents

# EMI Packaging Policy

## 1. Introduction

This document describes the EMI policy to be followed when creating packages for an EMI software component.

## 2. Definitions

- **Component**: It refers to any of the EMI software components. The list of EMI software components can be found in Section 4 `EMI Components and Product Teams`, page 11 of DNA1.3.1 - Technical Development Plan [R1].
- **Binary packages**: executable packages.
- **Source packages**: package files that contains everything needed to recreate not only the programs and associated files that are contained in the binary package file, but the binary and source package files themselves.

## 3. Package Formats

Product Teams must support all package formats of the EMI supported platforms.

The mandatory package formats for SL5 and SL6 platforms are:

- `PACKAGE-VERSION-RELEASE.PLATFORM.rpm` - The binary RPM for every platform.
- `PACKAGE-VERSION-RELEASE.PLATFORM.src.rpm` - The source RPM
- `PACKAGE-VERSION-RELEASE.PLATFORM.tar.gz` - The binary tarball for every platform.

The mandatory package formats for the Debian 6 platform are:

- `PACKAGE_VERSION-REVISION_PLATFORM.deb` - The binary DEB
- `PACKAGE_VERSION.orig.tar.gz` - A tarball containing a checkout from the version control repository.
- `PACKAGE_VERSION-REVISION.debian.tar.gz` - A tarball containing the debian directory (in Debian Source 3.0 format)
- `PACKAGE_VERSION-REVISION.dsc` - The control file
- `PACKAGE_VERSION-RELEASE_PLATFORM.tar.gz` - The binary tarball

In addition, a source tarball containing a checkout from the version control repository will be provided to the users under the name: PACKAGE-VERSION.tar.gz

Buildable source packages must be produced.

All source packages (src.rpm) will be validated by a Mock build in a pristine environment for SL5 and SL6 platforms. The `orig.tar.gz`, `debian.tar.gz` and `dsc` files will be validated by a Pbuilder build for the Debian 6 platform.

# 4. Metapackages

Product Teams must provide meta-packages to define sets of packages to be installed together.

The metapackage names must start with the string `emi-` and then have the rest of the name in LOWECASE (i.e. `emi-voms`). The LOWERCASE is mandatory in the Debian packaging policy.

The following rules must be followed for meta-packages:

- The meta-package must be empty. No files included.
- The meta-package must include only first level dependencies (that is the minimal set of packages allowing the complete installation of the product).
- The meta-package must not set any version constraints for the dependencies
- The meta-package version must not change when any new versions of its dependencies are released. A new meta-package version is released only when dependencies are added or removed.

NOTE: Metapackages do not allow the installation of 32bit packages on 64bit boxes and therefore are not suitable for packages (i.e. clients) that have this requirement.

No policy exists yet for the metapackages on Debian.

# 5. ETICS Package configuration

- Each project configuration can use the property `package.repo` to specify a YUM repo file from where pick the EMI packages produced during the nightly builds.
- The `package.autoreqprov` must be set only in project configuration and its value must be `yes`.

- The `package.prefix` property must **NOT** be set by components. This property is set at project level as `/`.
- When using the ETICS Pakager:
    - In the install target of the build commands, install the files in the ${prefix} directory as if ${prefix} would be /. For debian, as de debian/rules file is written in Makefile syntax, all the build commands have to be separated using "&&".
    - The ETICS Packager removes the Prefix set to `/` making the package non relocatable.
- When not using the ETICS Packager:
    - Make sure not to have the Prefix specified in your specfile
    - Make sure to produce a tar.gz matching the content of the RPM (therefore including the full path /usr, /usr/bin, etc.) in the case of the SL5 and SL6 platforms. A correct tar.gz and a matching dsc file are expected for Debian platform.
    - For SL5 and SL6, binary and source tarballs shoud be placed under the `tgz` directory, and binary and source rpm under the `rpms` directory.
    - For Debian, binary (`tar.gz`), debian (`debian.tar.gz`) and source (`orig.tar.gz`) tarballs, as well as the `dsc` file, should be placed under the `tgz` directory. The binary deb file should be placed under the `debs` directory.

Examples of SPECFILE (show) Examples of SPECFILE (hide)

- SPECFILE: dcap
- SPECFILE: VOMS
- SPECFILE: root

# 6. Package Contents and Metadata

The EMI project has adopted the well known packaging guidelines and policies defined by Fedora, EPEL and Debian.

Please refer to:

- Fedora Guidelines and Policies [R2]
- EPEL Guidelines and Policies [R3]
- Debian Policy Manual [R4]

The EMI project adopts the acceptance criteria of such documents to validate packages on these specific platforms.

ETICS provides the RPMLint plugin to verify that the RPM generated complies with the Fedora Guidelines.

## 6.1. Build in pristine environments

- From EMI 1 RC1 it is possible to build in pristine environments leveraging the Mock integration of the ETICS client 1.5.0 and the SUDO capability of the ETICS worker nodes.
- This is possible by adding the option `--repackage=[MOCK_OR_DEBIAN_CONFIGURATION]` in the etics-build command.
- The option is enabled only if the user running the build has root or sudo privileges.
- This option collects all the src.rpm generated during the normal ETICS build and executes Mock in a pristine environment rebuilding all packages one by one.
- This repackaging happens at the end of the ETICS normal build.
- In addition to the OS and external repositories such as EPEL YUM, the Mock and Pbuilder build makes use of a locally generated and EMI RCX YUM or APT repositories.
- In the Package List section of the report, it is possible to see what packages have been built successfully via ETICS and/or Mock/Pbuilder.
- From where the dependencies are installed:
    - Dependencies which are part of the same build, get built before and are then installed from the locally generated YUM/APT repository.
    - Dependencies part of EMI but which are not part of the same build, get installed from the current nightly YUM/APT repository (defined in `package.repo`).
    - External dependencies are taken from EPEL or OS YUM repositories in the case of SL5 and SL6 or from the official Debian repository for Debian 6.

# 7. Managing EMI Packages in EPEL

Please, follow the instructions below to distribute EMI packages into the EPEL repository:

- The person responsible for maintaining EMI products in the EPEL repository, the EPEL maintainer, **must request approval** if new versions of EMI packages need to be uploaded in EPEL. Changes need to be described and justified.
    - **Approval Request** mustl be done using a GGUS ticket under the category "Change Request". The ticket must be assigned to the corresponding PT support unit that is developing the relevant EMI product.
    - PTs must acknowledge and answer the GGUS ticket stating whether they agree or not with the proposed changes.
    - Only if a **positive answer** is received from the PT, the EPEL maintainer can implement the change and submit new packages to EPEL.

        - ◊ In parallel, the PT can start implementing the proposed changes to include them also in the EMI release, tracking this activity by opening a corresponding RfC in the PT internal tracker. The priority is decided by the PT, but the recommended one is `Medium`.
        - ◊ the GGUS ticket will be treated as any other GGUS ticket - State should be `On hold` and the corresponding RfC should be attached. The state `Solved` will only be used when the new packages are available in the EMI repository.
    - A **Negative answer** must be justified by the PT.
        - ◊ Once an agreement is reached between the EPEL maintainer and the PT, the GGUS ticket will be closed as `Unsolved`. No changes will be implemented and submitted to EPEL.
    - Packages prepared by the EPEL maintainer must be checked by the PT to verify that they don't impact the EMI release.
- During the EMI project lifetime **no packages** developed and maintained within the project will be made available directly to EPEL.
    - all EMI products will be made available under consistent releases, according to EMI policies;
    - only after the EMI products are released within the EMI release, PTs can claim their ownership and further maintain those packages in EPEL.

No policy exists yet on how to manage EMI packages in the Debian repositories.

# 8. Contacts

EMI SA2

# 9. Table of References

| Reference | URL |
|---|---|
| R1 | DNA1.3.1 - Technical Development Plan<br>https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDNA131 |
| R2 | Fedora Guidelines and Policies<br>http://fedoraproject.org/wiki/PackagingGuidelines |
| R3 | EPEL Guidelines and Policies<br>http://fedoraproject.org/wiki/EPEL/GuidelinesAndPolicies |
| R4 | Debian Policy Manual<br>http://www.debian.org/doc/debian-policy/ |

# 10. Logbook

Details Hide Details

## v2.1 (Approved on 23.01.2012)

- *20.01.2012*: Added the directories to leave the packages (SL and Debian) whan not using the ETICS packager.
- *16.01.2012*: Final updates for EMI2 and Debian
- *08.11.2011*: Updated some sections to be compliant with Debian

## v2.0

- *05.09.2011*: Added section on managing EMI packages in EPEL as requested by the release manager.

## v1.0 (approved on 11.03.2011)

- *25.08.2011*: No change in contents: added section numbers, cover and table of references.
- *11 February 2011*: First draft prepared by Lorenzo.
- *11 March 2011*: Maria applies feedback given by PEB.