GLUE Metapackaging Group

A joint project between LCG, EDG and iVDGL

Flavia Donno
Flavia.Donno@cern.ch

LCG-EIS

Outline

GLUE Metapackaging Group

Recommendations for installation and configuration

Status and Plans

Software installation on demand

Proposed solutions

The Group

The GLUE Metapackager group is composed of representatives from EDG, iVDGL and LCG:

Olof Barring[1], Germán Cancio[1], Flavia Donno[1], Saul Youssef[2], Jorge Rodriguez[3], Alain Roy[4]
(1) CERN, (2) Boston University, (3) Florida University, (4) Wisconsin University

The group started meeting end of **November 2002**.

The group came up with a set of **recommendations** for ***software developers***, ***software distributors*** and ***release managers***.

Such recommendations are described in the document:
https://edms.cern.ch/document/385786/1.2/TAB3
made public on February 2003.

While the distribution recommendations are clear and accepted, still the **configuration procedure** needs to be agreed and accepted, and translate into a practical plan.

Software development and deployment

- During the development phase, developers set up a directory structure for the code, put it in a CVS repository, and use ***software building tools*** to build the software on a specific OS or on multiple platforms with multiple flavors.
- Once the software is built, ***packaging tools*** are used to create a binary software distribution that can be installed at other locations.
- different ***software distribution tools*** can be deployed to fetch software from a repository, which could be local, and install the software on the machines managed by a system administrator. The repository can also allow local users to install the software on private desktops.
- Once the software has been installed, ***software configuration tools*** are used to properly configure the software reflecting the local site policies and configurations.

The Mandate

The mandate of the group is to provide solutions for the following problem that LCG has to solve: give support for **distribution, installation** and **configuration** of **GRID Middleware** and **Application Software** for large and middle size farms, whether manually or automatically managed as well as for desktop users.

No building tools are considered.

In particular solutions should be provided for *first time installations, updates, patches, upgrades and un-installation* of software bundles.

The *resulting installation must be the same* in terms of functionality and setup no matter what is the size of the installation and the procedure followed.

The *freedom of choosing the most appropriate packager* and packaging format at a given site is given.

Glossary

Package	A unit of software that physically contains files and provides certain functionalities. It could offer client and/or server components, APIs and CLI interfaces. Also configuration scripts could be part of a Package.
Package Format	The format chosen to prepare a given software package for distribution. Examples are cpio, tar, rpm, and pkg.
Management Packager	Used to install, upgrade, or uninstall a given package distributed in a particular packaging format.
System Packager	The management packager together with the supported packaging format used natively by the operating system (eg. RPM for most Linux distributions, PKG for Solaris).
Meta-packager	A packaging tool built on top of one or more management packagers. A meta-packager is used to manage (install/uninstall/upgrade) software bundles by invoking the management packager with the right operations. Examples are pacman and updaterpms.
Software Configuration	The process of adapting the software to individual virtual organization and site-specific settings. This usually implies changing configuration files.

State of the Art

- At CERN, LCG plans to rely on the new software developed by the EDG/WP4 team for installation and management of large PCs farms.
  - LCFG offers a complete solution for farm installation and management starting from the installation and configuration of the OS, local accounts and services management, software installation, upgrade and un-installation, etc.
  - The new solution will rely on system packagers such as rpm for Linux and pkg for Solaris.
- Many sites in Europe and US use farm management tools that provide installation and configuration tools for software bundles distributed in system native formats.
  - Rocks
  - Systems based on RedHat Kickstart
- Many sites in US use management tools not based on rpms
  - PACMAN
- A user sitting at his/her desktop should be able to install application software in user space – no root privileges.
  - This allows for support for installations on demand in the GRID environment at job execution
  - User Interface software
  - Development environment

Installation recommendations

For Software Developers

- **Software developers should provide their software in at least three formats:**
  - **Source format that can be compiled**
  - **Binary format that uses the appropriate packaging technology**
  - **Binary format that does not require any extra software (like RPM) to install.**

- **Complete instructions for installing and uninstalling each package must be provided.**
- **Software packages must be relocatable, except in cases when they absolutely cannot be.**

Installation recommendations

For Software Bundle or Release Managers

- **Describe the contents of the software bundle in a machine-readable format that can be automatically translated into descriptions that will work for different meta-packages.**
  - **The format is specified by the GLUE meta-packaging group.**
  - **Translations may be imperfect, but will allow software bundlers to automate most of the work in building a bundle that can be distributed by a meta-packager.**
  - (Meta-Packagers: pacman, LCFGng updaterpms, ROCKS specific tools, etc.).

Installation recommendations

For Site Administrators

- **Site administrators should install the software however they like.**

Configuration recommendations

For Software Developers

- **The software should be able to be installed separately from configuring the software**
  - **Software and configuration files can be distributed in the same or different packages**
  - **The installation process should be able to run without interaction from a user**
- **For system-wide configuration (as opposed to per-user configuration), there should be a configuration tool that allows the configuration to be edited easily**
  - **This tool should allow reconfiguration of the software after it has been configured for the first time.**
- **The location of log files and state files (such as process id files), and the port numbers used by programs that access the network should be configuration.**
  - **reasonable defaults (such as /var for log files used by privileged processes) should be used when configuration values are not supplied.**
- **For software that requires processes to continually run, scripts should be provided to start and stop by the scripts.**
  - **These scripts (System V) should check to make sure that the software is properly configured.**
  - **Upon completion of the configuration step, scripts should exist that allow a user to set up their environment to use the software.**

Configuration recommendations

For Software Bundle or Release Managers

- **Software/Release Managers should preserve the separation between installation and configuration steps.**
  - **This is a requirement for installation on farms of computers**
  - **Meta-packagers that are not intended for installation on farms do not need to maintain this separation, if it provides a better user interface.**

Configuration Scripts

- **Software developers should provide a configuration script for their software**
  - **By using some conventions, we hope we can generate an implementation that simplifies what a person needs to change when creating a metapackager setup.**
  - **From the meta-packager independent description generate a partially complete and correct description for any meta-packager to use to make an installation with no editing.**
  - **This would allow, for example, for an easier and more automatic creation of LCFG configuration components or Pacman configuration packages.**

Configuration Scripts

- Configuration scripts that meet this recommendation are assumed to work with configuration data that can be expresses as name-value pairs.
- Configuration scripts that conform to this specification are required to take at least four types of parameters, and each one can be specified as many times as desired.

--value <name> <value>

This sets a single named attribute to have the specified value.

--empty <name>

This specifies that the given attribute has the empty value. Not all software will need this.
Ex. GLOBUS_LOCATION = “”

--undefined <name>

This specifies that the given attribute should be undefined. Not all software will need this.
Ex. #GLOBUS_LOCATION=”” The attribute GLOBUS_LOCATION is not defined.

--input <file>

This specifies that there is a file that defines values. Each line of the file can contain a single --value, --empty, or --undefined , argument. These lines look just like they do on the command line, except that they are not subject to the substitutions performed by the shell. Names and values may be quoted (single or double) to allow spaces.

A proposal under discussion

- **Each software bundle should come with a meta-package independent description or *bundle template***
  - **Each bundle template can include *other templates* and/or list *package specifications*.**
  - **A bundle template can be *OS specific* or *generic*.**
  - ***Package specifications* refer to the specific releases (name, version, release) of the packages that one is trying to describe.**

- **A bundle template contains the following information:**
  - **Name (and version) of the bundle template**
  - **Short description**
  - **List of included templates, if any (filenames)**
  - **Package specifications. For every package, we have:**
  - **Package name, version and release**
  - **Package architecture**
  - **Package file name**
  - **Package pre-install commands, if any (depends on packager)**
  - **Package install commands, if any (depends on packager)**
  - **Package post-install commands, if any (depends on packager)**
  - **Package configure commands, if any**
  - **Package configure arguments, if any**
  - **Package pre-uninstall commands, if any (depends on packager)**
  - **Package post-uninstall commands, if any (depends on packager)**

First implementation example

- VDT released on Feb 10th Globus 2.2.4 in use by EDG (VDT 1.1.7)
- This distribution is in RPM format as produced by NMI and in PACMAN format as produced by VDT.
- The package list has been agreed between VDT and EDG.
- The RPMs do not contain configuration steps.
- The configuration follows the EDG conventions and is distributed in auxiliary RPMs by EDG.
- The configuration strategy as treated in the document will be discussed between VDT and NMI from a practical point of view. However, discussions among EDG, LCG, VDT and NMI will take place in order to guarantee a correct distribution that fits all needs.
- For the EDG middleware, a first attempt of meta-packager independent description should be made.

Status

- Nothing has been done to enforce that a PACMAN and RPM installation will produce the same results in the first VDT 1.1.7 exercise. Need work in this direction.
- The problem of distributing/applying patches to software and configuration has not been fully addressed.
- The meta-packager independent description needs a lot more thinking.
- It is not clear whose responsibility this is: developers or release managers or both. All developers need to be involved (Globus, NMI, Condor Team, EDG,…). Not enough representation in the group.
- It would be very useful to have a set of tools to convert meta-packager independent information into configuration info.
    - Grid Config (from NMI)

      http://www.nsf-middleware.org/documentation/NMI-R2/0/gridconfig/

Plan

- Get feed-back from LCG Application Area and LHC Experiments
- Continue the discussions to finalize the configuration open issues and come up with an operational plan.
- Evaluate existing tools (PACMAN, EDG/WP4, Grid Config) and propose solutions.
- Possibly design and build a “meta-package independent configuration tool”.
    - Working group ???

On demand installation and configuration

On demand installation

- Requirements from the experiments to be able to install/configure software “on demand”

  (pre-installation on farms could be not convenient)

  - Tools in place: PACMAN, PIPPY, etc.

- Application Software should be installable from a running grid job.
  - Alice installs software starting from CVS.

User and Job requirements

- Only **Software Managers** are allowed to manage official application software (special administrative grid role).
  - … But also ordinary users
  - … (Management for simultaneous installations … ?)
- **Special Grid Tools** should be provided to manage the application software. If they use "Grid jobs" than they should be tagged as special.
- The Software Manager or the user can install software **per site** (???).
- Software Management tools should be **"Information System" aware**.

Software requirements

- No root privileges required when installing software.
- Software must be relocatable.
- It should come with procedures to verify the installation.
- It should come with procedures to verify the version installed.
- It should publish requirements:
  - External software dependencies
  - System dependencies
  - Installation path
  - Environment
- It should come with configuration scripts that allow for reconfiguration and verification.
- It should be possible to declare a given "version" of the software "current" and others old/new. It should be possible to restrict the use of old/new versions only to special users.

Information Requirements

- Software bundles will be "uploaded" to the Grid and made available advertising the availability of the bundle in a proper Grid catalogue.
- Once the software has been installed, information about the version installed and the space occupied should be made available.
- Tools should be provided to "flexibly" query the information system (list of "VO" sites, which sites have a given version current,…)
- Software Information should be published (sw manager, installation info, status, dependencies, environment,…)

Software Management

- Tools to install and disinstall software, list, query, …
- Software installation verification
- Handling of requests coming from several managers
- Failure recovery
- More ???

Space Management

- Space dedicated to application software bundles
- Space dedicated for application installation/configuration
- Tools for space management

- Users should be able to install in their home areas
- Management of temporary user’s space
- …

Proposal

- The CERN IT/FIO Group is coming up with a proposal for LCG to accommodate installation on demand that satisfies the requirements stated by the experiments.

- Other proposals/prototypes exists based on **EDG replication** and **Globus information tools**, **PACMAN** and **PIPPY**

    http://physics.bu.edu/~youssef/pacman/index.html

    http://heppc12.uta.edu/~mcguigan/pippy/

Conclusions

- A lot of efforts in software distribution, installation and configuration solutions.
- It is absolutely necessary to guarantee consistency across installation methods and provide uniformity of configuration solutions
- The mandate of GLUE Meta-packaging Group is to get consensus and come up with a proposal and an implementation solution independent of the meta-packager used.
- Request for more flexible “on-demand” installation and configuration solution on the Grid.