

CMS SOFTWARE INSTALLATION

K. Rabbertz, *University of Karlsruhe, Germany;*

M. Corvo, *INFN, Padova, Italy;*

Sh. Muzaffar, *Northeastern University, Boston, USA;*

A. Nowack, *RWTH Aachen, Germany;*

A. Sciabà, *CERN, Geneva, Switzerland;*

J. Weng, *CERN, Geneva, Switzerland & University of Karlsruhe, Germany;*

St. Wynhoff, *Princeton University, USA*

Abstract

For data analysis in an international collaboration it is important to have an efficient procedure to distribute, install and update the centrally maintained software. This is even more true when not only locally but also grid accessible resources are to be exploited. A practical solution will be presented that has been successfully employed for CMS software installations on systems ranging from physicists' notebooks up to LCG2 enabled clusters. It is based on perl for an automated production of RPM packages¹ and `xcmsi`, a tool written in perl and perl/Tk, to facilitate installing, updating and verifying the RPM packaged software.

GENERATION OF RPM PACKAGES

The basic development and final testing of CMS software is performed on the central repositories at CERN. Therefore, all RPMs are based on this central installation of all experiment and external software. In total, this comprises the experiment specific programs of the CMS collaboration, the installed versions of packages of the LCG project [2] and the CMS specific installations of other external programs. In addition, an RPM of the SCRAM tool (Software Configuration, Release And Management) [3] and the necessary compilers, currently the GNU compiler collection GCC 3.2.3 [4], are provided.

Concerning the preparation of the RPMs, two classes of packages are differentiated:

1. Software projects, managed by SCRAM that have to be registered in a local SCRAM database and
2. Tools, which means all the rest that does not need a special treatment during installation.

The central script to gather the additional intelligence from SCRAM and to prepare the dependency information is `ProjectDist.pl`, a schematic view of the procedure is presented in fig. 1. Further scripts are employed to perform consistency checks, produce tar archives, take proper care of symbolic links and to finally generate the RPMs.

The code can be accessed via a link to the CVS repository on the common project page of `xcmsi/RpmGen`². There is also a download page for the CMS software³.

INSTALLATION OF CMS SOFTWARE

`xcmsi` is a tool to ease the installation of the entire CMS software on computing resources ranging from physicists' notebooks up to grid-enabled clusters. Its main parts are `xcmsi.pl`, a perl script using perl-Tk to provide a graphical interface for a user friendly configuration of the environment, and `cmsi.pl`, a perl script to do the actual installation. In addition, `cmsv.pl` serves to validate the installed software. The latter two can also be called in batch mode provided they are given access to a proper configuration file.

By design, the following requirements are met by this tool:

- Arbitrary installation directory `$CMS_PATH` to relocate packages
- No root privileges required
- Network access for automated downloads is helpful but not mandatory
- Command-line (batch mode) installation possible
- Separate validation suite included
- A graphical interface (GUI) for a concise configuration also by less experienced users
- Save-able and (re)loadable configuration files
- Configuration files directly usable to set the CMS environment for users
- Multi-platform support
- Support for multiple non-interfering installations

¹Named after the RedHat package manager[1]

²<http://savannah.cern.ch/projects/xcmsi>

³<http://cern.ch/cms-xcmsi>

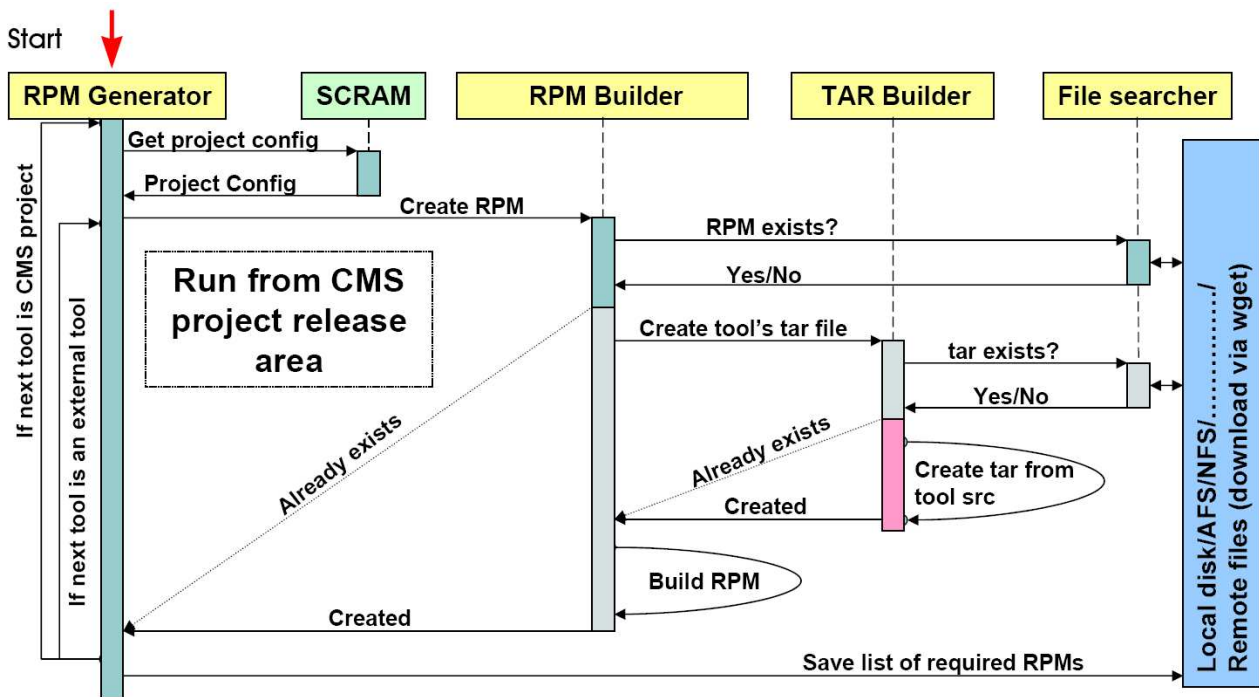


Figure 1: Schematic view of the iterative RPM generation procedure initiated by `ProjectDist.pl`. It differentiates between SCRAM managed projects, where additional dependency information is extracted in order to run the proper SCRAM set-up in the installation phase, and other tools where no additional treatment is necessary.

A typical installation comprises 1 GCC compiler package, 1 SCRAM package, about 40 CMS RPM packages and 30 LCG RPM packages. In total, the 72 packages themselves take around 1 GB, the complete unpacked installation 4 GB of disk space.

The source code is available within the CVS repository of the `CMSToolBox`⁴ as `XCMSInstall` or as gzipped tar archive, again from the CMS download page.

Configuration with `xcmsi.pl`

The main window of `xcmsi` fig. 2 provides six rows and columns of push buttons representing approximately the corresponding sequence of steps to follow for a successful install. Focusing the discussion here on the most important steps we start with the second row which first gives the possibility to look up the current configuration. In the `Select Tags` section, the user can choose between download tags representing different CMS software projects — all necessary software for the chosen project, even additional projects if required, is downloaded and installed. The `Select Archives` section shows the already installed RPMs and offers the possibility to choose new RPMs for the update mode. The last three buttons serve to load a previously saved configuration file and to edit and save the current settings. There are about 30 configurable settings but usually only three of them have to be adapted to the users' needs

and only these three are shown in the first configuration window:

1. `CMS_SRC`, the location where `xcmsi` is to be found
2. `CMS_RPMS`, the directory for depositing all the RPMs
3. `CMS_PATH`, the software installation area

The additional settings are hidden behind an "Expert Options" push button and should only be changed if necessary.

Pushing the `Options` button additional features can be activated which are required for certain install versions. The `Select Tags` window actually enters the software projects to install in the "Download additional RPMs" entry field. If other Linux distributions than RedHat or Mandrake are tried, say SuSE or even Debian, it might be necessary to pass additional arguments to the RPM commands for the installation.

Finally, you can start, verify or update the installation and quit the program. Since the graphical user interface is employed only for the configuration itself, the settings have still to be passed to the perl scripts `cmsi.pl` and `cmsv.pl` which do the real work. This is achieved by saving the actual configuration into a file, e.g. `cmsset.default.csh`, containing all necessary information and whose name has to be specified as `-f` option to these scripts. It can be written in either flavour, bash or c-shell. Thereby it is ensured that the whole procedure can be performed from

⁴<http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/CMSToolBox/XCMSInstall/?cvsroot=CMSToolBox>

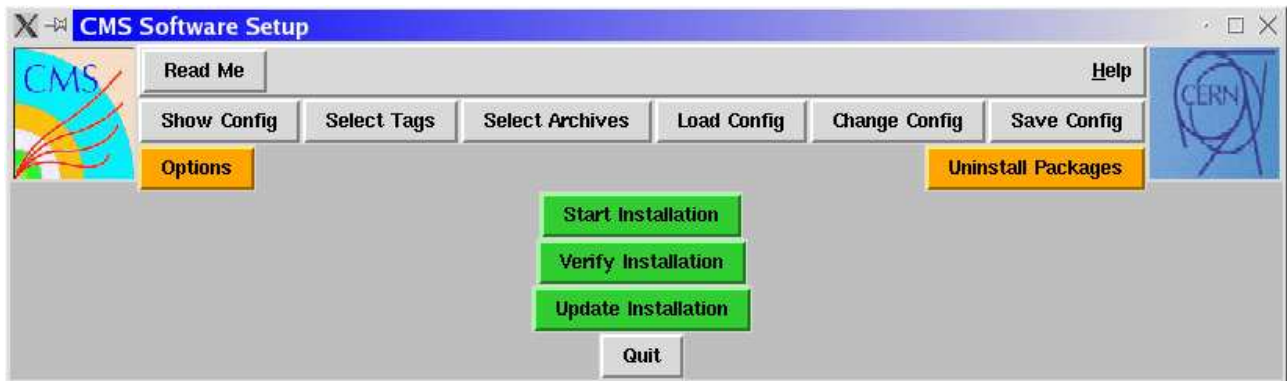


Figure 2: The main window of `xcmsi.pl`. The arrangement of push buttons from top to bottom and left to right represents approximately the sequence of steps to follow for normal usage.

the command line (or in batch mode) without duplicating functionalities. In addition, the configuration files are designed in such a way that they can directly be used to set up a users environment for developing its code with the installed software. More details can be found in the online documentation.[5]

Installation and verification with `cmsi.pl/cmsv.pl`

As explained `xcmsi.pl` serves only for a concise configuration of the CMS software installation and once, this is accomplished, the corresponding command line is composed and executed. Currently, a "default" could look like:

```
./cmsi.pl -f cmsset_default.csh -g config
-d "ORCA_8_4_0 OSCAR_3_4_0"
```

where the `-g config` option is mandatory to fix a problem in SCRAM's treatment of environment variables necessary for GCC compilers that are not found in the usual places. In total, `cmsi.pl` accepts the following options:

```
cmsi.pl Version V0.8
Usage: cmsi.pl [switches/options]
        [list of rpm's]
-d software set Download/install or erase
                pre-defined software set(s),
                multiple sets have to be
                quoted ""!
-e Erase a set of rpms
-f configfile Use configfile instead of
                cmsset_default.csh
-g gcc-path Adapt SCRAM to use gcc-path
                or with special keyword
-g config adapt SCRAM to use gcc-path
                from configfile, the new
                default!
-h Print this text
-i Overrule installation abort
                in case of existing rpms
-m mirror Use mirror instead of
                CERN download server
```

```
-r rpm options Additional options to rpm
                install command, multiple
                options have to be
                quoted ""!
-u Run in update mode
```

The verification process just needs to know which configuration file to use. Taking all necessary information from this file, the verification can be run independently and repeatedly on any installation or update performed. The only option other than `-f configfile` is `-h` which prints a short help text. In the current state, a set of checks on consistency in the general set-up and on the reproducibility of results achieved with the CMS reconstruction program ORCA is performed. There are many more aspects which could and should be tested, this is a point for future improvement.

INSTALLATION VIA LCG

Every Virtual Organization (VO) in the LCG scope has an Experiment Software Manager (ESM) who is mapped onto a privileged grid account with write permissions on a specific software area. Currently, LCG sites have two distinct policies for software installation. The first one allows the ESM to install software on a shared area common to all Worker Nodes where it is mounted via a network file system (NFS, GPFS). This shared area may reside on the site's Computing Element (CE). The second one without a shared area, foresees an installation procedure where the experiment software is first installed on one WN and then propagated to all the others. This second mechanism is still under test. From the point of view of disk space consumption and fault tolerance it is less favourable.

The grid installation job runs on a WN creating, on the shared area, the directory structure containing CMS software. The core tool which performs installations on LCG sites is `cmsi.pl`. Around it, a perl script (`CmsSwGridInstall.pl`) has been developed which provides some features to automatize the creation and submission of installation jobs.

Basically, the script queries the ldap server (GRIS) of every LCG Computing Element to retrieve published tags and decides the action to perform, either a brand new installation or an update. Once all information has been collected, it creates shell scripts, which are wrappers of `cmsi.pl`, and the necessary job description files (jdl files). All jdl's are then submitted and their status monitored to retrieve output files as soon as they finish.

Once the job finishes, the Information Service is made aware of the newly installed distribution publishing a new tag describing it. Every release is actually marked with a specific tag which describes the software contents. This system allows users to drive production or analysis jobs only on sites publishing the needed tag.

The experiment can also provide the RPMs in order to have them installed by the site manager with LCFGng, as it has been done at the Imperial College site. However, this approach would require agreement and coordination between the experiment and the various site managers. The CMS software manager installation procedure is thus the preferred option.

CURRENT DEPLOYMENT IN CMS

The described procedure has routinely been used to distribute and install new releases of CMS software on more than 20 LCG2 sites in France, Germany, Great Britain, Italy, Pakistan, Russia, Spain, Switzerland (CERN) and Taiwan, on NorduGrid [6] it has successfully been tested. Since by design the installations can be performed on any PC or notebook complete statistics on the usage are not available. For the last two weeks the corresponding http download server has counted about 1000 hits per day, the counter for SCRAM downloads is at 185 meaning about 12 base installations per day. This, however, does not take into account double or incomplete downloads.

OUTLOOK

xcmsi/RpmGen has proven to be a reliable and flexible solution to the problem of distributing, installing and updating the complete CMS software necessary for development and analysis. It has been successfully employed for this purpose on stand-alone systems like notebooks up to LCG2 enabled clusters during the CMS Data Challenge this year.

Foreseen improvements concern the grid installation and the validation procedures, automated updates and the extension to platforms that are not binary compatible to the CMS operating system of reference via source RPMs.

ACKNOWLEDGEMENTS

For their help in designing the posters of this contribution we would like to thank Corinne Beroujon, Grand Théâtre de Genève, Martin Erdmann, RWTH Aachen, and Steffen Kappler, University of Karlsruhe.

REFERENCES

- [1] <http://www.redhat.com/docs/books/max-rpm>
- [2] <http://lcg.web.cern.ch/lcg>
- [3] <http://spi.cern.ch/scram>
- [4] <http://gcc.gnu.org>
- [5] <http://cern.ch/cms-xcmsi/doc/xcmsi>
- [6] <http://www.nordugrid.org>