



CMS Software Packaging and Distribution Tools

Stefano Argiro
Shaun Ashby
Volker Büge
Marco Corvo
Ramzy Darwish
Nikolay Darmenov
David Evans
Bockjoo Kim
Andreas Nowack
Natalia Ratnikova
Klaus Rabbertz
Michael Thomas
Joanna Weng
Tony Wildish

CERN
CERN
FZK/University of Karlsruhe
CERN/INFN
FNAL
INRNE/CERN
FNAL
University of Florida
RWTH Aachen University
FNAL
University of Karlsruhe
CALTECH
CERN/University of Karlsruhe
Princeton University



Overview

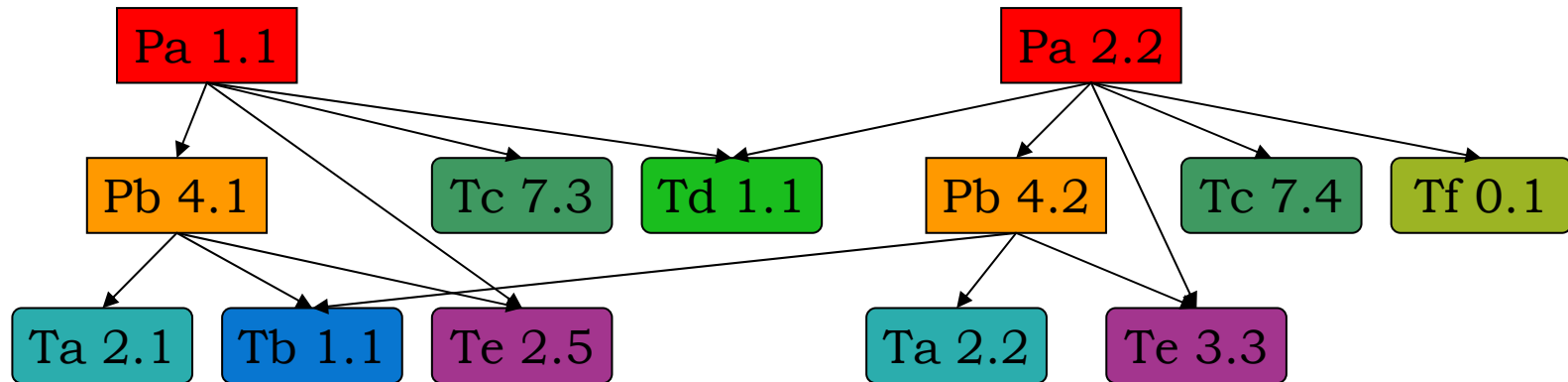
- CMS Software Projects and Its Components
- Distribution Strategies
- Use Cases
- Packaging and Distribution Tools
 - ◆ DAR
 - ◆ RpmGen & XCMSI
- Summary and Outlook



CMS Experiment

- The CMS experiment is a **large international collaboration**
 - ◆ 37 countries
 - ◆ 160 institutes
 - ◆ 2000 scientists and engineers
- **Main goal:**
Analysis of the data from the experiment to find **new, interesting results**
 - ◆ collision rate: 40 MHz
 - ◆ event rate: 150 Hz
 - ◆ events per year: 10^9
 - ◆ raw data volume per year: approx. 1,5 PB
 - ◆ plus massive Monte Carlo simulation of the experiment
- In order to achieve the goal, this needs
 - ◆ **physicists worldwide** developing code for various analyses
 - ◆ **enormous computing resources**
 - ◆ distributed **Monte Carlo simulation** and **data analysis on Grids**
(D. Evans, Id 278, P. Garcia-Abia, Id 271, M. Corvo, Id 273, O. Gutsche, Id 279)
 - ◆ **CMS software installation** on Grids (K. Rabbertz, Id 247),
local clusters, desktop PCs and even laptops

CMS Software Projects



- **Complex hierarchical structure** of the software components:
 - ◆ **Projects (P)** are the main building blocks (e.g. CMSSW, POOL, SEAL, ...)
 - ◆ **External tools (T)** provide already existing functionalities (e.g. dcap, geant4, root, ...)
- Each **project**
 - ◆ exists in **multiple versions**,
 - ◆ uses certain versions of
 - **other projects** and
 - **external tools**
- Multiple versions of projects and external tools **co-exist**
- **Different platforms:**
 - ◆ in the past: Red Hat 7.3
 - ◆ Scientific Linux CERN 3, 32 bit
 - ◆ future: 64 bit? MacOS ? ...



Software Management

- SCRAM (Software Configuration, Release And Management)
 - ◆ management of **different software projects** in CMS and LCG
 - ◆ management of **multiple versions of a project**
 - ◆ configuration of the **dependencies**
 - ◆ **compiling** and **building** of the project (based on make)
 - ◆ more or less **platform-independent** (written in perl)
 - ◆ **installation** of software
 - only for the project itself
 - **no external components**
 - **no binaries**, everything has to be compiled
- **not suitable for easy and fast installation**
- **no installation of all needed components**

- Therefore: Software distribution needs **another approach**



Distribution Strategies

- **DAR** (Distribution After Release):
Minimal set of files
 - ◆ copy only the **needed binaries, libraries, and data files**
 - ◆ all files are stored in a **tar ball**
 - ◆ set the **runtime environment**
- **RpmGen & XCMSI**:
Installation of the full environment
 - ◆ **duplicate completely** a given version of a project and its dependencies
 - ◆ each component is packaged **separately** in **RPM packages** with **proper dependencies**
 - ◆ set all **configuration files**



Pros and Cons

DAR

- **Pro:**
 - ◆ slim and compact distribution
 - ◆ very robust
- **Con:**
 - ◆ only the runtime environment is available
 - ◆ no code development possible
 - ◆ needed components are distributed several times if various versions are installed

XCMSI

- **Pro:**
 - ◆ the full environment is available
 - ◆ maximum flexibility, code development possible
 - ◆ modular design, reuse of already installed components
- **Con:**
 - ◆ huge amount of data is shipped which is in most cases not needed
 - ◆ damage of one component can disturb several projects

Use Cases

- DAR and RpmGen & XCMSI are **complementary** and focus on **different use cases**
- **Monte Carlo production** needs
 - ◆ a **very reliable** environment
 - ◆ a **limited number** of pre-defined executables
 - ◆ **small number of versions** of projects (duplication of files is affordable)
- Monte Carlo production uses DAR
- **Code development** needs
 - ◆ **complete development environment**
 - ◆ several **versions in parallel** (reduction of duplication is desired)
- XCMSI is used for code development on user's resources
- **User's analysis** is not really predictable.
If **rebuilding** of the software on the target machine is needed, **XCMSI** is used for installation.
If **rebuilding is not necessary**, **DAR** can also be used.



DAR (N. Ratnikova, Id 340)

- DAR is a set of python modules for
 - ◆ **creation of archives** (DAR balls) using the **information about runtime environment** for a given application:
 - managing file system objects:
all needed binaries, libraries, data files ...
 - handling runtime environment settings
 - ◆ **creation of incremental DAR balls**
 - only changes with respect to a given DAR ball are collected
 - ◆ **installation of DAR balls**
 - arbitrary directory for installation
 - shell scripts to set the runtime environment
- DAR provides **interfaces to other CMS developed tools**:
 - ◆ **SCRAM-DAR** interface to get runtime environment for SCRAM-managed projects
 - ◆ **RefDB-DAR** interface to CMS Production Reference Database to get request from RefDB and automatically create a new DAR ball
 - ◆ **API** for re-use within **MCPS**, Monte Carlo Processing Service



RpmGen & XCMSI

- **Aim:**

- ◆ Data analysis and software development on
 - laptops
 - desktop PCs
 - local clusters
 - Grid clusters
- ◆ **Easy installation** of the complex software and the complete development environment
- ◆ Installation
 - in an **arbitrary directory**
 - **without root privileges**
 - in a **modular** way

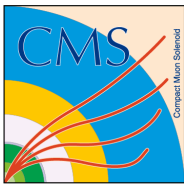
RpmGen

- Creation of an **image of a reference installation**
- Packaging of each **component** into a **single RPM package**
- The package name contains information about
 - ◆ **name** of the component
 - ◆ **type** of the component (SCRAM-based project or external tool)
 - ◆ **affiliation** with CMS or LCG
 - ◆ **platform**
 - ◆ **version number** of the component
 - ◆ **revision number** of the package
- This information gives an **unique directory name** which is used for the installation of this component on the target machine
- Dependencies of the components are mapped to **dependencies between the RPM packages**



Details of RpmGen

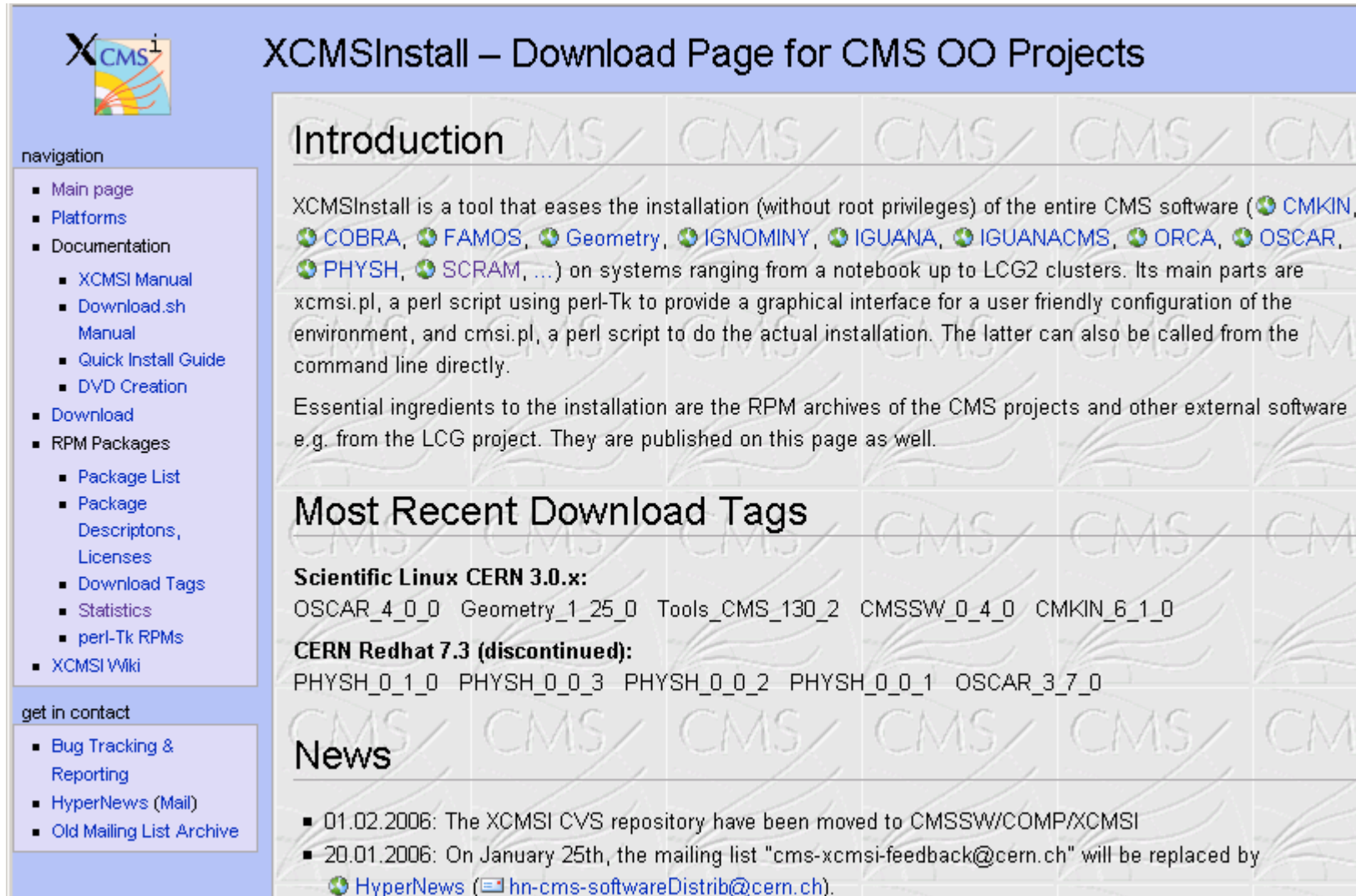
- RpmGen is a set of perl and shell scripts
- **Master script** is called to **create all packages needed** for a given project
 - ◆ **SCRAM** is queried for the **list of projects and external tools**
 - ◆ for **projects**, the master script is called **recursively**
 - ◆ for **each component**, tar balls are created with **an image of the reference installation**
 - **symbolical links** are handled correctly
 - if possible translation into relative links
 - otherwise copy of the file or directory
 - ◆ creation of **RPM packages for each component** using the tar balls
 - **install and uninstall scripts** for configuration of projects
 - **dependencies** to needed packages
 - **no dependencies to system's RPM packages**
- Already existing RPM packages are **not recreated**
- **Consistency checks**
 - ◆ **identical directories** for existing packages
 - ◆ **dependencies**
 - ◆ **version numbers** and directory names



Distribution of RPMs

- New sets of RPM packages are **published on a web server** and archived **by a script**
- **Dynamical web pages** (<http://cern.ch/cms-xcmsi>) show
 - ◆ **all available sets** of packages and the packages themselves,
 - ◆ **statistics** about downloads,
 - ◆ **download script** and the installation tool **XCMSI**,
 - ◆ recent **documentation**

XCMSI Web Page



The screenshot shows the XCMSI web page with a navigation menu on the left and a main content area on the right. The navigation menu includes sections for 'navigation' and 'get in contact'. The main content area features a title 'XCMSInstall – Download Page for CMS OO Projects', an 'Introduction' section, a 'Most Recent Download Tags' section, and a 'News' section.

XCMSInstall – Download Page for CMS OO Projects

Introduction

XCMSInstall is a tool that eases the installation (without root privileges) of the entire CMS software (🌐 [CMKIN](#), 🌐 [COBRA](#), 🌐 [FAMOS](#), 🌐 [Geometry](#), 🌐 [IGNOMINY](#), 🌐 [IGUANA](#), 🌐 [IGUANACMS](#), 🌐 [ORCA](#), 🌐 [OSCAR](#), 🌐 [PHYSH](#), 🌐 [SCRAM](#), ...) on systems ranging from a notebook up to LCG2 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. The latter can also be called from the command line directly.

Essential ingredients to the installation are the RPM archives of the CMS projects and other external software e.g. from the LCG project. They are published on this page as well.

Most Recent Download Tags

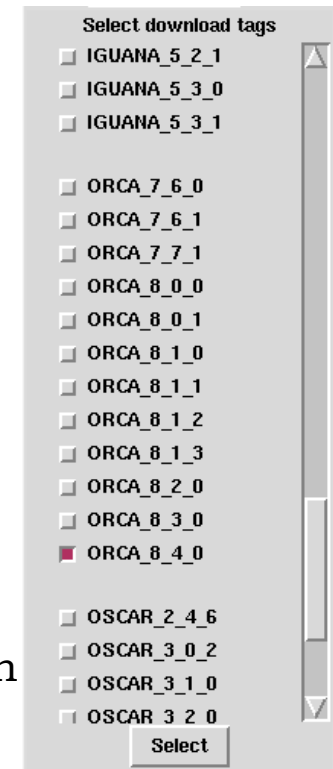
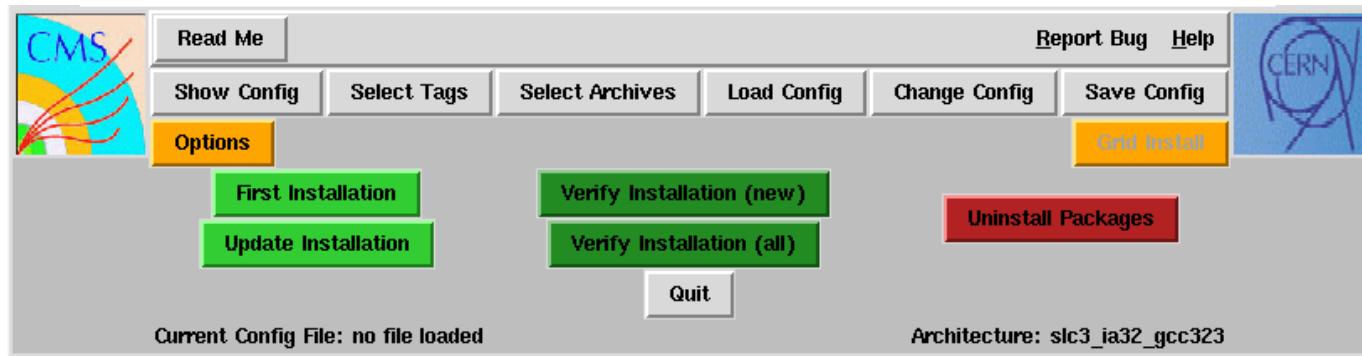
Scientific Linux CERN 3.0.x:
OSCAR_4_0_0 Geometry_1_25_0 Tools_CMS_130_2 CMSSW_0_4_0 CMKIN_6_1_0

CERN Redhat 7.3 (discontinued):
PHYSH_0_1_0 PHYSH_0_0_3 PHYSH_0_0_2 PHYSH_0_0_1 OSCAR_3_7_0

News

- 01.02.2006: The XCMSI CVS repository have been moved to `CMSSW/COMP/XCMSI`
- 20.01.2006: On January 25th, the mailing list "`cms-xcmsi-feedback@cern.ch`" will be replaced by [HyperNews](#) (hn-cms-softwareDistrib@cern.ch).

- XCMSI is a set of perl scripts
 - ◆ **Command line tools** for
 - **download** and **installation** of RPM packages on
 - local machines
 - Grid clusters (DAR balls also supported)
 - **basic verification** of the installation



- ◆ **Graphical user interface** for
 - easy generation of **configuration** files
 - **arbitrary base directory**
 - many expert options which usually do not have to be changed
 - **selection of software projects** for installation
 - **execution of command line tools** for installation and verification
 - **save removal** of not needed packages



Some Statistics about XCMSI

- Around **300 different versions** of various projects
 - ◆ 124 for RedHat 7.3
 - ◆ 177 for Scientific Linux CERN 3 (and compatible)
 - ◆ new versions are usually **available within one day** after publication
- More than **1,300 RPM packages** with **22 GB**
 - ◆ 262 for RedHat 7.3
 - ◆ 392 for Scientific Linux CERN 3
 - ◆ 660 platform independent
- **Downloads** since end of October 2004
 - ◆ 17190 (**around 36 per day**) project downloads
 - ◆ from more than **230 different sites** in about **35 countries**



Summary & Outlook

- **DAR** is **successfully** used for the **CMS Monte Carlo production**
- **RpmGen & XCMSI** are **widely used** for distributing the **CMS software on all kinds of clusters and PCs**
- **Plans:**
 - ◆ **DAR:** building of incremental DAR balls from user-specific applications
 - ◆ **RpmGen:** integration of RPM package creation into the automatic build procedure
 - ◆ **XCMSI:** better validation methods and improved error handling