

CMS PARTIAL RELEASES: MODEL, TOOLS and APPLICATIONS

ONLINE and FRAMEWORK-LIGHT RELEASES

C.Jones, D.Lange, E.Meschi, S.Muzaffar, A.Pfeiffer, N.Ratnikova, L.Sexton-Kennedy



▼ Scope & Motivation

▼ CMS software project CMSSW includes over 1000 *interdependent* packages for data analysis, event display, reconstruction and simulation algorithms, detector description, data formats, framework, and utilities. It depends on about 100 of external software products.

▼ Every external package, including configuration, development, and installation tools, is built and packaged for distribution.

▼ The whole CMSSW release is automatically configured, built and packaged into *one distribution file*.

Thus prepared releases and externals are published for distribution and deployment on the GRID and on the user development machines.

▼ Alongside with the main approach CMS has two important *use cases*, where only small subsets of packages and externals are needed:

- High Level Trigger (HLT) algorithms running online on the Event Filter farm.

- Light-weight Root based physics analysis tool Framework-light.

The model of Partial releases allows to build customized releases for such applications, while satisfying a number of special requirements.

▼ Main technical challenges of the Partial Release model are, first of all, to ensure *by construction* the consistency with the full base release, and to automate all steps of the procedure for optimal support at reduced maintenance cost.

▼ ONLINE Release

▼ The goal of the Online Release is to provide consistent environment for HLT online operation, while minimizing the amount of code to be deployed on the Event Filter Farm machines. It should also offer the possibility of producing patch releases forking from the mainstream CMSSW release sequence to retrofit necessary fixes while guaranteeing stable operation of the online HLT.

▼ **Special requirements:**

- Minimize online Build Set to improve robustness and stability.

- Use system compiler, hence different architecture name and completely separate set of distribution files.

- Use external packages available on the local system: online version of *Xdaq* software and its externals.

- Software distribution must be suitable for use with Quattor tool, which is used to manage software installations on the Event Filter farm, and has a number of constraints compare to regular CMSSW installation method using apt-get.

- Full rebuild may be required in case of major software upgrade on the Event Filter farm.

▼ ONLINE Release Application Set includes HLT reconstruction packages, filter modules, data acquisition, and Data Quality Monitoring tools.

▼ The Model

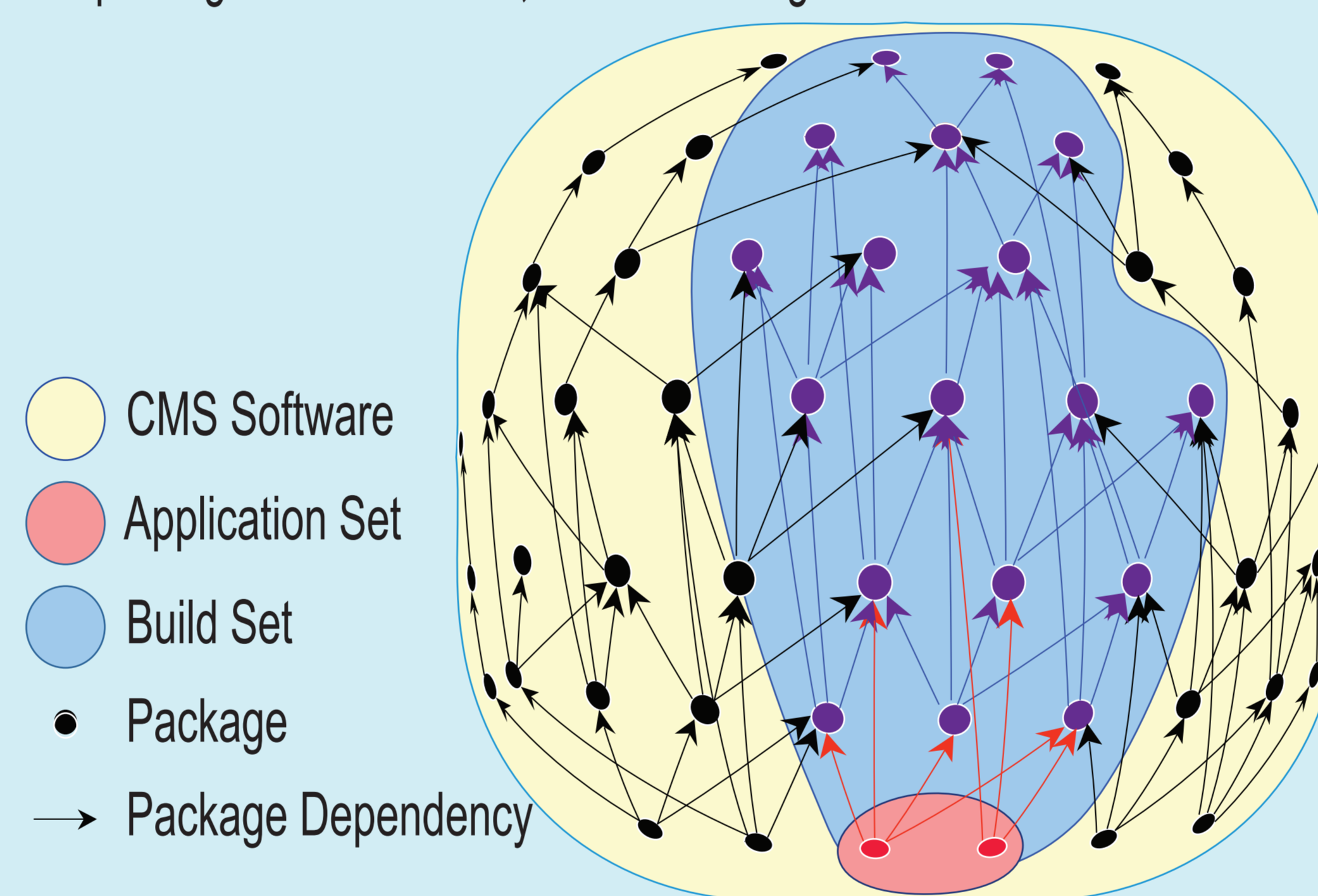
Application Set == a subset of packages of the base software release, that directly provides desired functionality. Application Set is defined by corresponding Application Manager.

Build Set == a complete minimal set of packages of the base release and externals, necessary to build the Application Set.

Partial release == independent software release of packages defined in the application Build Set, built and packaged using standard CMS release management tools.

▼ Most crucial point of finding the correct Build Set is discovery of dependencies between packages.

▼ Dependencies occur at compilation time via included header files, at link time via symbols in libraries, and at run time via dynamically loaded plug-in modules. Some dependencies, such as appearing from the package local unit tests, must be disregarded.



▼ We use Ignominy [1] tool to collect the dependency information for the base CMSSW release, and the BuildSet [2] tool to construct the required Build set based on Ignominy information.

▼ Build and configuration specifications are defined in the CMSDIST [3] repository and are constructed in such a way that Partial release uses source code distribution of the base release, as defined in the Tag Collector [5]. Then only the code included into the Build set is automatically selected.

▼ Build and configuration rules are defined in the common area so that Partial release uses same set of rules as the full CMSSW.

▼ Thus the throughout consistency is insured by construction.

▼ References

1. CMS Experiment, <http://cmsdoc.cern.ch/cms/outreach/html/>
2. CMS Offline Software, http://cms.cern.ch/ICMS/jsp/page.jsp?mode=cms&action=url&urlkey=CMS_OFFLINE
- 3-4. CMS Software Development Tools Workshop, <https://twiki.cern.ch/twiki/bin/view/CMS/SWDevToolsWorkshopApr07>, report: CMS IN 2007/000
5. Ignominy, <http://www.ihep.ac.cn/~chep01/paper/8-024.pdf>
6. CMS Event Filter, <https://twiki.cern.ch/twiki/bin/view/CMS/EventFilter>
7. FWLite Analysis Tutorial, <https://twiki.cern.ch/twiki/bin/view/CMS/SWGGuideFWLiteAnalysis>
8. Online Releases, <https://twiki.cern.ch/twiki/bin/view/CMS/HowToInstallONLINERelease>
9. Quattor <http://quattor.web.cern.ch/quattor/>
10. Xdaq, <https://twiki.cern.ch/twiki/bin/view/XdaqWiki/>
11. CMS packaging system..., <http://indico.cern.ch/contributionDisplay.py?contribId=302&confId=3580>
12. CMS Tag Collector, <https://cmstags.cern.ch/CmsTC/html/>

▼ Tools

[1] **IGNOMINY** is a powerful tool used to analyze software release source code and resulting build products, and to detect different types of dependencies between software packages.

[2] **BuildSet** tool allows to query Ignominy results and to calculate dependencies recursively for a given package or a list of packages.

[3] **CMSDIST** is a repository of specification files for configuration, build, packaging, and installation instructions for CMSSW, Partial Releases, and all external software packages, which are re-built from the source code for distribution and use with CMS software.

[4] **PKGTOOLS** provides a framework for building, packaging and uploading software distributions according to instructions in the CMSDIST specifications.

[5] **TagCollector** is the central interface to publish packages version tags and to manage the contents of CMSSW releases on a package level.

[6] **SCRAM** is software configuration and release management tool used to manage CMSSW release internal configuration and build, and to provide CMS software development environment.

▼ FWLite Release

▼ The goal of the Framework Lite application is to provide the possibility of convenient physics analysis of data in a regular CMS Event Data Model format in plain ROOT environment.

▼ **Special requirements:**

- Minimal distribution set must only include Root and other external packages required by Root, and necessary Data Format packages.

- All dictionaries, necessary for analyzing Analysis Object Data (AOD) directly in ROOT, must be included.

▼ FWLite release Application Set includes all packages containing objects which may appear in the Analysis Object Data (AOD), and corresponding analysis algorithms.

▼ Small distribution size allows easy download and install FWLite application onto the user's laptop.

▼ Minimal number of external dependencies and modest amount of code simplify porting to another platforms. It is already possible to run FWLite application on the MacOS.

▼ Light weight and easy access to the CMS data structures, combined with powerful Root visualization tools, make FWLite attractive and popular among physicists.

