

Packaging and deployment in CODAC Core System

Anze Zagar
ITER Organization

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

ITER CODAC APPROACH

CODAC Core System

- Based on Red Hat Enterprise Linux
- Provides common software for development and operation of ITER control system
- Supports concurrent installation of multiple CCS versions inter-switchable using Red Hat Alternatives mechanism
- Systemd based service management (since CCS 6)

CODAC Core System

- Build & development components: Maven, SDD, Python, Java
- EPICS components: base, modules, CS-Studio...
- Data archiving components: DAN, HDF5, UDA
- Kernel modules for NI devices
- Timing components: TCN, PTP
- Fast controller components: SDN, RTF

EPICS Components

CODAC Core System	5.4	6.0	6.1	6.2	6.3	7.0
RHEL	6.5	7.4	7.4	7.4	7.4	8.5
EPICS Base	3.15.5	7.0.1.1	7.0.2.0	7.0.3.1.ccs3	7.0.4.1	7.0.6.1
EPICS Base Java		7.0.1	7.0.2	7.0.4	7.0.6	7.0.7
Asyn	4.30	4.32.ccs1	4.34	4.37	4.41	4.42
Autosave	5.7.1	5.8	5.9	5.10	5.10	5.10.2
Calc	3.6.1	3.6.1	3.7.1	3.7.3	3.7.3	3.7.4
Sequencer	2.2.4	2.2.5	2.2.6	2.2.8	2.2.8	2.2.9
Std	3.4.1	3.4.1	3.5	3.6.1	3.6.2	3.6.3
Stream Device	2.7.7	2.7.7	2.8.4	2.8.10	2.8.16	2.8.22
Area Detector	2.4	3.1	3.4	3.8	3.10	3.11
OPC UA			0.3.1	0.5.2	0.8.0	0.9.2
CS-Studio	4.4.7	4.5.2	4.6.202	4.6.310	4.7.802	4.8.0

Packaging

- Based on Red Hat's native solution for packaging and distribution of software:
 - RPM packages
 - YUM (DNF since RHEL 8) dependency management utility for software installation from subscribed RPM repositories
 - Satellite Server for centralized deployment and management of RHEL hosts

Pa

jin

- Softw
- Softw
- Simp
- Implic
- const

```
[codac-dev@ccs7 m-example]$ rpm -qlp target/codac-core-7.0-example-myioc-7.0.0.v0.0a1-0.el8.x86_64.rpm
/etc/opt/codac-7.0
/etc/opt/codac-7.0/alt.d
/etc/opt/codac-7.0/alt.d/example-myioc
/opt/codac-7.0
/opt/codac-7.0/apps
/opt/codac-7.0/apps/example
/opt/codac-7.0/apps/example/bin
/opt/codac-7.0/apps/example/bin/linux-x86_64
/opt/codac-7.0/apps/example/bin/linux-x86_64/myapp
/opt/codac-7.0/apps/example/dbd
/opt/codac-7.0/apps/example/dbd/myapp.dbd
/opt/codac-7.0/apps/example/iocBoot
/opt/codac-7.0/apps/example/iocBoot/myioc
/opt/codac-7.0/apps/example/iocBoot/myioc/.iocdesc
/opt/codac-7.0/apps/example/iocBoot/myioc/dbToLoad.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/envPaths
/opt/codac-7.0/apps/example/iocBoot/myioc/envSystem
/opt/codac-7.0/apps/example/iocBoot/myioc/envUser
/opt/codac-7.0/apps/example/iocBoot/myioc/myioc-postSaveRestore.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/myioc-preSaveRestore.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/myioc.req
/opt/codac-7.0/apps/example/iocBoot/myioc/sddPostDriverConf.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/sddPreDriverConf.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/sddSeqToLoad.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/seqToLoad.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/st.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/threadSchedulingConf.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/userPostDriverConf.cmd
/opt/codac-7.0/apps/example/iocBoot/myioc/userPreDriverConf.cmd
/opt/codac-7.0/apps/myioc
/opt/codac-7.0/bin
/opt/codac-7.0/bin/services
/opt/codac-7.0/bin/services/ioc@myioc
[codac-dev@ccs7 m-example]$ rpm -q --requires target/codac-core-7.0-example-myioc-7.0.0.v0.0a1-0.el8.x86_64.rpm
codac-core-7.0-common
codac-core-7.0-epics
codac-core-7.0-epics-autosave
codac-core-7.0-epics-iocmon
codac-core-7.0-epics-std
codac-core-7.0-epics-sysmon
```

```
figure
CONFIG
CONFIG_SITE
Makefile
RULES
RULES_DIRS
RULES_ioc
RULES_TOP
Boot
Makefile
myioc
├── dbToLoad.cmd
├── envSystem
├── envUser
├── Makefile
├── myioc-postSaveRestore.cmd
├── myioc-preSaveRestore.cmd
├── myioc.req
├── sddPostDriverConf.cmd
├── sddPreDriverConf.cmd
├── sddSeqToLoad.cmd
├── seqToLoad.cmd
├── st.cmd
├── threadSchedulingConf.cmd
├── userPostDriverConf.cmd
└── userPreDriverConf.cmd

efile
ppApp
Db
├── Makefile
├── Makefile
src
├── Makefile
└── myappMain.cpp
```

CODAC Yum/DNF Plugin

- CODAC Patches

- Patch RPMs with patch ID suffix

`codac-core-6.0-epics...rpm <-> codac-core-6.0-epics-p11150...rpm`

- Transactional replacement of original RPMs with the patch RPMs and vice versa

```
dnf install-codac-patch 11150
```

```
dnf remove-codac-patch 11150
```


Constraints

- Monolithic installations (all embedded software has fixed versions)
 - => limited number of possible software configurations identifiable by the CCS version
 - + simplifies deployment, maintenance and (remote) support
 - + easier to provide good QA coverage of all possible configurations
 - less development freedom
- Packaging and deployment constraints imposed by the packaging system
 - + simplifies packaging
 - + prevents mistakes and inconsistencies in packaging
 - less development freedom

GENERAL RPM APPROACH

Advantages

- Standard and widely adapted approach with good community support
- Built-in support for: dependency handling, integrity and authenticity checking, ...
- Simple to provision and manage systems using the wide variety of tools provided by community

Limitations

- Rigid in a sense that it makes it difficult to arbitrarily mix different versions of installed software
- Patching requires swapping of the entire RPMs
- Not easily portable to other distributions (even if RPM-based)
- Not easy to build RPMs without some support tools

Possibilities to “Diversify” Installations

- Forcing a specific package version by means of Yum/DNF (package filters, versionlock, DNF Modules)
- Freezing or diversifying of repositories on depl. server
- RPM packages versioned by name
- Red Hat Software Collections
- Containers

DNF Modules (Since RHEL 8)

Applies repository filter to prevent installation of RPMs other than those belonging to selected software version

```
$ dnf module list postgresql
```

Name	Stream	Profiles	Summary
postgresql	9.6	client, server [d]	PostgreSQL server and client module
postgresql	10 [d]	client, server [d]	PostgreSQL server and client module
postgresql	12 [e]	client, server [d]	PostgreSQL server and client module
postgresql	13	client, server [d]	PostgreSQL server and client module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled

```
$ dnf module -y enable postgresql:13
```

RPM Packages Versioned by Name

- Version specific package names and deployment paths to allow **parallel installations** of multiple versions

```
java-11-openjdk-11.0.3.7-0.e17_6.x86_64
java-1.7.0-openjdk-1.7.0.151-2.6.11.1.e17_4.x86_64
java-1.8.0-openjdk-1.8.0.144-0.b01.e17_4.x86_64
java-1.8.0-oracle-1.8.0.141-1jpp.1.e17_3.x86_64
```

- Alternatives mechanism to select the active version

```
$ alternatives --display java | grep -v slave
java - status is manual.
  link currently points to /usr/lib/jvm/java-11-openjdk-11.0.3.7-0.e17_6.x86_64/bin/java
/usr/lib/jvm/java-1.8.0-oracle-1.8.0.141-1jpp.1.e17_3.x86_64/jre/bin/java - family java-1.8.0-oracle.x86_64 priority 180141
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.144-0.b01.e17_4.x86_64/jre/bin/java - family java-1.8.0-openjdk.x86_64 priority 180144
/usr/lib/jvm/java-1.8.0-openjdk-1.7.0.151-2.6.11.1.e17_4.x86_64/jre/bin/java - family java-1.8.0-openjdk.x86_64 priority
170151
/usr/lib/jvm/java-11-openjdk-11.0.3.7-0.e17_6.x86_64/bin/java - family java-11-openjdk.x86_64 priority 1110307
Current 'best' version is /usr/lib/jvm/java-11-openjdk-11.0.3.7-0.e17_6.x86_64/bin/java.
```

Red Hat Software Collections

- Allows **parallel installation** of alternative versions
- Requires special instruction to select the desired version

```
$ python --version
Python 2.7.5
$ scl enable rh-python36 "python --version"
Python 3.6.3
$ scl enable rh-python38 "python --version"
Python 3.8.6
```


Parallel Installation – Containers

- Best suitable for services
- Container images for different versions of a given service (e.g. installed by means of DNF modules)
- Multiple containers can be **installed and running in parallel**