

Building Level3

Farrukh Azfar
Sarah Demers
Todd Huffman
James Loken
Mat Martin
Kevin McFarland
Armin Reichold
Kirsten Tollefson
David Waters

Contents

- Building Level3
 - Premises and Assumptions
 - The Level3 Package
 - Building Calibrations
 - Build Structure
 - Exporting and Shelving
 - Example
- Validation
- Debugging
- Documentation
- Organisation

Premises and Assumptions

- Procedures presented here can be done by **anyone** – any collaborator can develop a Level3 trigger
- **Reproducibility** is “the prime directive”. If **L3exe_47** crashes, we can rebuild it any time.
- There are two types of builds:
 - **development**: no guaranteed reproducibility, does not appear in run-configurations database (RCF-DB)
 - **production**: must start from frozen release + frozen patches; guaranteed reproducibility, will be visible in RCF-DB, can be flagged for use in run-control (after validation)

The Level3 Package

- Contains **all you need** to prepare a test release (`l3Setup.ksh`) in which you can:
 - build and run a Level3 executable (generate exe's, tcl's, calib's)
 - export the build results to the Level3 machines
- Level3 is **different** from other packages
 - the **code** in the package **does not uniquely define** the outcome of **a build**
 - Trigger-DB, Calib-DB and RCF-DB hold information defining outcome
 - **different outcomes are selectable** by the user (physics table name & tag)
 - RCF-DB has a dynamic role in this (see 2 slides ahead)
 - DB's use **templates** in Level3 package to create actual build targets (e.g. `L3exe.cc`)

13Setup.ksh

- Contains **list of all patches** to each frozen release from which Level3 can be built
- It itself is always added from the **head**
- It is **used to check** that a test release from which a build is attempted conforms to the prescription in this file. If this or any other check fails no tags are issued for this build.
- Other checks include:
 - Base release consistent with cdfsoft2 version
 - Base release is not development
 - No locally modified files
 - All files up to date
 - No modifications from head \Leftrightarrow all files have sticky tag (except **13Setup.ksh**)

Building Calibrations

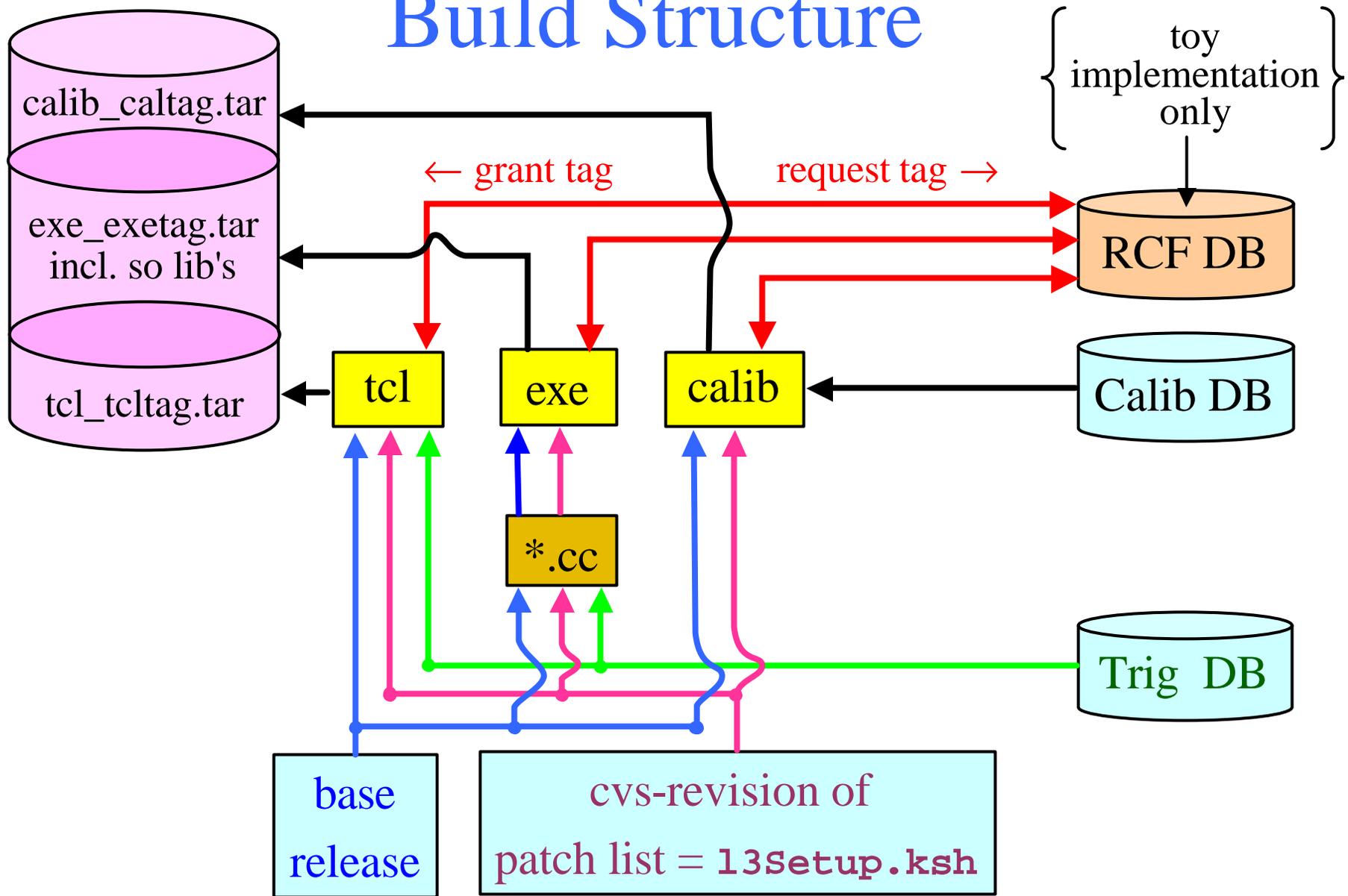
Why export calibrations as text files to processor nodes ?

- Level3 requires **D to E** conversion constants, e.g.:
 - ADC counts to GeV in calorimeters
 - drift times to distances in CMX.
- Currently calibration constants are written to an **Oracle DB** on a dedicated Sun 4500.
- We **CAN NOT** have 150 processor nodes logging in to this database at start of a run ! This would **take too long**.
- Therefore software has been written for the build phase of Level3 to:
 - log into the Oracle database
 - read the calibrations and **export them in text format**.
- All this is done during **gmake Level3.tbin**
- CalibrationManager module allows online Level3 processes to read calibrations from text files.

Building Calibrations

- Greater involvement of detector groups is needed
- Calibration offers **uniform and simple API** to read/write either
 - text
 - ORACLE
 - MSQL
- Detector groups should provide calibration access/write code.
- Currently only calorimetry and the muon groups have written the 10 lines of code necessary to use the interface.
- Unfortunately most calibration **constants are still hardwired**.
- Most of the ORACLE calibration **tables are already defined** but the code using them is missing.
- Work is in progress for the COT constants.

Build Structure



Exporting and Shelving

- All build items are **individually** exported into one tar file each
 - Executable(s) and shared lib's → **exe_\$EXE_TAG.tar**
 - Tcl's → **tcl_\$TCL_TAG.tar**
 - Calibrations → **calib_\$CALIB_TAG.tar**
- All tar files are kept **on the shelf** on the Level3 build platform (currently on **oxpc01.fnal.gov**; could move to DB)
- **Access** from Level3 computers via anonymous ftp
- Identification of tar files through **tags** e.g. **EXE_TAG**
- Run Control selects **specific set of tags** (held in RCF-DB) and picks up corresponding tar files
- Export mechanism works. Tested yesterday in development

Example

- > `./getL3Setup.ksh` # got this from head of Level3
- > `source makeLevel3TestRelease <dir> <release name>`
`<release version>`
- > `cd <dir>/<release name>`
- > `source Level3/getTags <physics_table_name>`
`<physics_table_tag>`
- > `gmake` # make Calib DB access utilities, tcl's, .cc's
- > `gmake Level3.tbin` # make executable(s) and calib text files
- > `gmake Level3.EXPORT_L3exe` # make tar balls

Validation

- Build procedure is validated by **running** previous **example** every night in **development**.
- Process a standard data file with resulting **default Level3 executable, tcl and calibration**. (only one set of (exe,tcl,calib) gets validated)
- Single trigger paths as contributed by **physics groups** should get validated by physics groups.
- Single path results should be **compared to full Level3 executable results**.

Bug Reporting/Fixing Procedure

1. Enter bug in **e-log**.
2. **Put tar file** with CORE, executable, shared libs, tcl and calibrations on oxpc01.
3. **Page** the Level3 on-shift person. 24hr x 7d coverage started last week.
4. **On-shift** person debugs, delegates or takes immediate action (switch off module via tcl)
5. Acknowledgement and follow up in the e-log

Documentation

- Starting point for documentation is:
<http://www-cdf.fnal.gov/internal/people/links/TraceyPratt/my.html>
- contains shift list, contact info, e-log, code & build documentation, debugging/error reporting procedures, this talk
- Soon to be linked to Level3 trigger home page
- Code and build procedures documentation maintained via cvs in Level3/doc.
- Chris Green to use this link:
<http://www-cdf.fnal.gov/internal/people/links/TraceyPratt/my.html>

Level 3 Organisation

