EPICS Version 3: Future Directions

Andrew Johnson
AES/SSG, Argonne National Laboratory
APS Controls Status  Richard Farnsworth

Accelerator IOC EPICS Versions

%
Outline

- Recent EPICS Base Releases
- Main Features included in 3.15.0.1
  - Demonstrations
- Some Features in Development
- Future Plans
Recent EPICS Base Releases

- **3.14.12.2 was released 2011-12-12**
  - The Known Problems page currently documents 12 important issues reported since this release (10 patch files)
  - Many other minor issues have been fixed on the 3.14 branch
  - A new patch release (3.14.12.3) is due sometime soon
    - Let me know if you have specific deadlines coming up for which coordination would be advantageous (ITER CODAC?)

- **3.15.0.1 was released 2012-07-31**
  - The Known Problems page currently links to 6 patch files
  - Many more changes and additions have been made on the 3.15 branch
  - New development releases may come out at any time
    - I welcome discussion of release dates or frequency, including timing for the first operational 3.15.1 release
Main Features Included in Base 3.15.0.1

- Compatibility with 3.13 build system removed
- Source code reorganized, fewer shared libraries
- Parallel builds improved — `make -j`
- Soft events can be named instead of just numbered
- Attributes of non-VAL fields improved
- Alarm filtering added to input record types
- Server-side per-channel filtering
  - Dead-band, synchronization, sub-array, time-stamp
  - New filters can be written and plugged in as needed
- Asynchronous process-get support
  - New “Async Soft Channel” device support for input records
  - Processes the target record (DB link only) and waits for completion before fetching the value from the addressed field
Some Features in Development

- **Record Type Enhancements**
  - 16 links for both fanout (was 6) and sequence (was 10) record types
    - Backwards compatible, despite adding a link 0
    - New OFFSet and SHiFT fields can modify selection

- **New printf record type (not yet merged)**
  - Long string VAL field, length configurable
  - Uses format string to determine data type for INP$n$ links
    - Supports variable field-width and precision, long strings
  - Output link, device support follows stringout record type
    - Soft Channel, Async Soft Channel, stdio

**Demo**
More Features in Development

- Record Reference Documentation from DBD files
  - Replace the Record Reference Manual Wiki
  - Added converter to generate HTML files from text added to DBD
    - Using Perl’s Plain Old Documentation (POD) format with a few extensions
    - No additional software dependencies
  - Extracts field descriptions and menu choices from DBD data, keeping tables in generated document up to date
  - dbdExpand.pl strips the text from expanded DBD output, POD is not seen (or understood) by the IOC’s DBD parser
  - Not yet merged, there might be objections...
Example of DBD to HTML conversion

=head1 Analog Input Record (ai)

This record type is normally used to obtain an analog value from a hardware input and convert it to engineering units. The record supports linear and break-point conversion to engineering units, smoothing, alarm limits, alarm filtering, and graphics and control limits.

=head2 Parameter Fields

The record-specific fields are described below, grouped by functionality.

-recordtype ai

=cut

recordtype(ai) {

=head3 Input Specification

These fields control where the record will read data from when it is processed:

-fields DTYP, INP

The DTYP field selects which device support layer should be responsible for providing input data to the record. The ai device support layers provided by EPICS Base are documented in the Device Support section. External support modules may provide additional device support for this record type. If not set explicitly, the DTYP value defaults to the first device support that is loaded for the record type, which will usually be the C<Soft Channel> support that comes with Base.

The INP link field contains a database or channel access link or provides hardware address information that the device support layer to determine where the input data should come from. The format for the INP field value depends on the device support layer that is selected by the DTYP field. See Address Specification for a description of the various hardware address formats supported.

=head3 Units Conversion

These fields control if and how the raw input value gets converted into engineering units:

-fields RVAL, ROCF, ASLO, AOFF, LINR, ESLO, POFF, EGUL, EGUF

The RVAL field contains the raw value as read from the hardware device and passed directly into the VAL field. The ROCF field contains the raw offset as read from the hardware device and passed directly into the C<Soft Channel> device support, and is also fairly common for GPB and similar high-level device interfaces.

If the device support sets the RVAL field, the LINR field controls how this gets converted into engineering units and placed in the VAL field as follows:

1. RVAL is converted to a double and ROCF is added to it.
2. If ASLO is non-zero, the value is multiplied by ASLO.
3. AOFF is added.

=recordtype ai

This record type is normally used to obtain an analog value from a hardware input and convert it to engineering units.

The record supports linear and break-point conversion to engineering units, smoothing, alarm limits, alarm filtering, and graphics and control limits.
Future Plans

- Continue to develop 3.15 as I have time
  - Documentation needs updating, not just DBD to HTML text
  - Add new features as they come up

- Release timetable for a 3.15.1 production release?
  - Talk to me if you have a specific deadline
  - Will expect help with testing

- Merge the V4 pvIOC plugin so V3 IOCs support pvAccess?
- Jeff Hill’s replacement CA server based on Data Access?