



... for a brighter future

What's Coming in EPICS Base R3.14.11

*Andrew Johnson and Eric Norum,
AES Controls Group, Argonne*



U.S. Department
of Energy

UChicago ►
Argonne_{LLC}



A U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC

Highlights

- Record aliases
 - Long Strings and Field Modifiers
 - Monitor event type DBE_PROPERTY
 - Link flags MSI and MSS
 - More initHooks
 - PINI field values, PHAS support
 - Analog Record Types
 - General Time API changes
 - Dynamic Loading on Unix platforms
 - Other stuff
-
- Beyond R3.14.11

Record aliases

- Records can now have multiple aliases, defined at DB load-time:

```
– record(ai, "record") {  
    alias("alias1")  
}  
alias("record", "alias2")
```

- Aliases are just additional names for their record instance

- Stored in the database
- Must be created in or after the record instance
- Can be loaded from a different .db file
- All names are visible to CA clients

- NB:

- You cannot make an alias point to a specific field of a record
- Info items are attached to a record/alias name, not to a record instance

Record aliases cont.

- The IOC's `db1` command lists aliases as well as records
- A new IOC command `db1a` lists aliases with their records:
 - `epics> db1a`
alias1 -> record
alias2 -> record
- A record's NAME field can be used to get its canonical name via CA:
 - `tux% caget record.NAME alias1.NAME alias2.NAME`
record.NAME record
alias1.NAME record
alias2.NAME record
- Record aliases can help when
 - making changes to a site record naming convention
 - different groups want different names for control system PVs
- But they can only solve *some* naming issues

Long Strings and Field Modifiers

- Record fields holding strings have never been limited to 40 characters
 - The NAME field has been 61 characters long since R3.14.1
 - The 40 character limit is imposed by the CA protocol
- The IOC can now make long string fields accessible via CA
 - Enabled by appending '\$' to the field part of the PV name
 - *e.g. record.NAME\$*
 - The '\$' is a Field Modifier which changes the reported type of the PV from DBR_STRING to an array of DBR_CHAR
 - The '\$' suffix only works on string and link fields

Long Strings cont.

```
- tux% cainfo record.NAME record.NAME$
record.NAME
  State:          connected
  Host:          tux.aps.anl.gov:5064
  Access:        read, no write
  Native data type: DBF_STRING
  Request type:  DBR_STRING
  Element count: 1
record.NAME$
  State:          connected
  Host:          tux.aps.anl.gov:5064
  Access:        read, no write
  Native data type: DBF_CHAR
  Request type:  DBR_CHAR
  Element count: 61
```

Long Strings cont.

- MEDM and EDM have supported the display of char arrays in text widgets for many years, no changes are needed
 - Must configure the MEDM widget format as ‘string’
- The catools programs in Base (caget, caput, camonitor) now support long strings using the new -S option
- CA client authors: Please support long strings in your applications
 - They are compatible with all versions of the CA client library
 - However, *do not* try to recognize them by parsing the PV name; have the user tell you when to treat a PV as a string
 - *The \$ suffix is not the only way to make a long string PV*
- Post questions about long strings to tech-talk, Andrew will answer them

Monitor event type DBE_PROPERTY

- A new CA monitor event trigger has been added, DBE_PROPERTY
- This event will be triggered by the IOC whenever the properties (extended attributes) of the channel change
 - Enum strings, Display and Alarm limits, Units string, Precision etc.
- Requires support from the record types
 - Currently only the mbbi and mbbo record types trigger this on their VAL field when the xxST fields are written to
 - Later versions of Base will trigger it automatically, based on additional information added to the record.dbd files

DBE_PROPERTY cont.

- CA Client authors: Please start using DBE_PROPERTY
 - At the moment you probably use `ca_get()` or `ca_get_callback()` after a channel connects to fetch its extended attributes
 - From R3.14.11 use `ca_create_subscription()` with an event mask of DBE_PROPERTY instead
 - Even if the record type never explicitly triggers this event, your callback will be called once when the subscription connects
 - Your client will be prepared for future releases when these events will get triggered

Link flags MSI and MSS

- DB links and CA input links now have 4 possible choices for the Maximize Severity flag:
 - NMS Non-Maximize Severity (default)
 - MS Maximize Severity
 - MSI Maximize Severity Invalid
 - MSS Maximize Status and Severity
- MSI is like MS but only for an alarm severity of Invalid Alarm
 - Useful for a CALC of an input record value which should be Invalid when the read operation fails
- MSS works like MS but copies the Alarm Status along with the Severity
 - Makes an I/O record's alarm status appear in another record

More *initHooks*

- We have added some new *initHook* states
 - Backwards compatible as long as you used the state names from *initHooks.h* and don't rely on their numerical values
- The old states `initHookAfterInterruptAccept` and `initHookAtEnd` are now deprecated, but still work as before
 - Use `initHookAfterDatabaseRunning` and `initHookAfterIocRunning` instead if possible
- Some of the new states can occur multiple times
 - Every time the `iocPause` and `iocRun` commands are used
- See *initHooks.h* or the R3.14.11 documentation for more details
 - Discussed in Release Notes and Application Developers' Guide

PINI field values, PHAS support

- Record processing at initialization has also been updated
 - The PINI field can now take one of four choices:
 - *NO, YES, RUN, PAUSE*
 - The new values RUN and PAUSE cause records to be processed at the start of the iocRun or iocPause commands
 - *Note: iocRun is executed as the second half of iocInit*
 - CA links are not active when PINI=RUN records are processed
 - PINI processing now honors the PHAS field, records with lower PHAS will be processed before higher ones

Analog Record Types

- Alarm limits and `get_alarm_double` changes
 - Where HHSV, HSV, LSV or LLSV are set to NO_ALARM, the analog record types now return a NaN for the corresponding alarm level
 - CA clients can now omit displaying alarm levels that are not used
 - Applies to ai, ao, calc, calcout, dfanout, sel and sub record types
- Alarm filtering (code from Bernd Schoeneburg, not yet in CVS)
 - New field AFTC (Alarm Filter Time Constant) specifies time constant
 - Slows the rate at which level alarms are raised
- Soft channel device support for the ai record type
 - Now implements SMOO input filtering

General Time API Changes

- The previous event system API allowed an Interrupt Service Routine to request the current or an event time
- The General Time versions of `epicsTimeGetCurrent()` and `epicsTimeGetEvent()` cannot be called from Interrupt context
- New routines `epicsTimeGetCurrentInt()` and `epicsTimeGetEventInt()` can be called from Interrupt context
 - They only check the most recently successful provider, do not traverse the priority list of time/event providers
 - If the provider has not registered an interrupt-safe routine, they fail
 - Routines are registered with `generalTimeAddIntCurrentProvider()` and `generalTimeAddIntEventProvider()`
- We also added `generalTimeHighestCurrentName()` which returns the name of the highest priority current time provider, so you can check that an IOC is getting time from the source it's supposed to be using

Dynamic Loading on Unix Platforms

- Experimental feature prompted by Elliott Wolin (JLAB)
- New IOC command `dlload` on Unix platforms
 - Must include the `dllload.dbd` file when building an IOC to enable this command
 - Imports a loadable library into the IOC process while it's running and executes any C++ static constructors found in it
 - After `dllload` run a `xxx_registerRecordDeviceDriver` command from the library to register its components with the IOC
 - Can be used to load new sequence programs and sub and aSub record subroutines after `ioclnit`
- See R3.14.11 Release Notes for more details

Other Stuff

- RTEMS version 4.9.2 or later is now required
 - Added support for MVME2700 and MVME2400
- Old support files for hpux, sgi and VMS have been deleted
- Changes in libCom:
 - New string hashing functions
 - epicsRingBytes now rejects messages that are too big for its buffer
 - New routines for escaping and un-escaping strings
 - Portable epicsNAN and epicsINF defined in epicsMath.h
- Catools changes (caget, caput, camonitor, cainfo):
 - New option `-p priority` to set CA priority
 - New option `-F separator` to set the output field separator
 - Non-printable characters now use C `\` escape sequences

Beyond R3.14.11

- R3.14.11 may be the last R3.14 version of Base
 - Released this summer
- Field Modifiers are going to be big in future versions
 - Access to sub-arrays (fixed indexes):
 - `caget 'MyWaveform.VAL[100:200]'`
 - Get/put multiple fields of a record as an array
 - `caput -a MySub.A,B,C,D 4 1 2 3 4`
 - Use JSON encoding for even more complex things:
 - `caget 'Sema4.OWNER{"put":{"TAKE":"ANJ"},"WDOG":30}{'`
 - `camonitor 'MyBpm.X,Y{"beam":"BLUE"},"maxrate":2}'`
- Will probably split Base into 3 separate modules: Core, CA, IOC
 - Permit Jeff Hill to work on new versions of CA without affecting development of the IOC, and vice versa