Release Notes for APS Release 3.11.6
May 26, 1994

New and/or Updated Documentation:

EPICS Application Source/Release Control
EPICS Add On Products Source/Release Control
Channel Access Security Design
Archiving Reference Manual
Channel Access Reference Manual
Application Developer's Guide
Synchronized Time Stamp Support

A. Directions for switching to production APS release R3.11.6

NOTE: IOC Memory

This release will use more memory on your iocs than previous releases (mainly because of access security). John Winans has modified the board support package to support > 8 MegByte memory boards. See accessSecurity Document for details.

PART I. All developers and users

Add the following early in your .cshrc file:

```
source /usr/local/epics/cshrc
```

This file creates definitions for Unix environment variable EPICS_ADD_ON and changes your path so that it is searched in the order: .epicsUnix, $PVT_ADD_ON, $EPICS_ADD_ON, $path. If you are in an epics application area the epics release bin directory will be searched first, then PVT_ADD_ON (if you have defined this variable), then the epics add on area, then the previously defined path.

PART II. Application System Area Managers
Follow the directions given in the Application Source/Release Control document, Chapter 3 - Procedures for Application System Area, Section 4 - Integration.

PART III. Application Shadow Area Developers
The following directions are identical to the direction given in the Application Source/Release Control document Chapter 4 - Shadow Node Procedures, Section 2 - Synchronizing an application shadow node.

PART IV. Changes to Record Support

All ascii record definitions have to be changed to add the access security level to each field description. The following script can be used to perform the conversion (It assumes that the files are named <type>Record.ascii)

```
#! /bin/sh
for file in *Record.ascii
do
ex - $file <<\EOF
1,$s/DBF_/ASL1 DBF_/
wq
EOF
done
```


The source file for each record support module must be edited to change all calls of the form

```
tsLocalTime(&precord->time);
```

to

```
recGblGetTimeStamp(precord);
```

All calls to gts_trigger_counter should be removed. For example the ai record previously had the code

```
if (pai->scan == SCAN_IO_EVENT){
        if ((pai->evnt != 0)  && (gts_trigger_counter != 0)){
                if ((gts_trigger_counter % pai->evnt) != 0){
                        return(0);
                }
        }
}
```

This is no longer executed.

B. Unbundling of Channel Access Clients

The channel access clients as well as GDCT have been unbundled from epics. See the EPICS Add On document for details

C. Access Security

Access Security is now implemented.
See the Channel Access Security Document for details

D. Channel Access Additions

Caching and Blocking Channel/Database Access

The requirements resulted from discussions between Jeff Hill, Claude Saunders, and Marty Kraimer.

We agreed that the following changes should be make to channel access and/or database access:

1) Cached db_puts.
2) Locked db_gets
3) ca_puts with notification of record processing completion.
4) Allow tracking of multiple batches of synchronous Channel Access Requests
.

Cached database puts

Previously the following was true. If
1)a dbPut is directed to a passive record
2)the put is supposed to cause the record to be processed.
3)The record is already active (PACT is true)
then the request is just ignored.

Changes were made so that the following is true

1) If a dbPutField, which is a request from outside the database such as from Channel Access, is directed to a field with Process Passive True and the record is already active, then, after the record finishes processing, it will again be processed.

2) If a dbPutLink with Process Passive true is directed to a record that is already active because of a dbPutField request to that record then after the record finishes processing it will again be processed.

Locked db_gets

Previously a record was not always locked while data is fetched.
If an array record is being read this could result in part of the data
coming from one scan and the remainder from the next scan.
Records are now locked during channel access gets.

ca_puts with confirmation of record processing.
Batched synchronous Channel Access Requests

Claude requested that channel access clients be able to receive
notification of completion from ca put requests. By completion it
means that if the put results in a request for record processing
(the record is passive and the field is PP) then a completion
message is generated only when that record and all records processed
as a result of that record being processed complete processing.

In addition Claude has requested that blocking
for completion of a group of put requests be possible.
This request is satisfied by a new CA IO synchronization
facility. The new facility allows the programmer to create
any number of synchronization groups. The programmer can
interleave IO requests within any of the groups. Once The
IO operations are initiated then the programmer is free to
block for IO completion within any one of the groups as needed.

E. Synchronous Time Support

New software has been added to IOC core to maintain time stamps.
The new software has the ability to maintain time stamps over all
IOCs on a network. This software replaces the previous ioc
time stamp support. The main differences are that the new version
support multiple hardware boards and allows multiple time stamp
events. See the manual listed above for details.

F. Description of major differences between R3.11.3 and R3.11.6

1. IOC Software Components

alarm.h

Added definitions for READ_ACCESS and WRITE_ACCESS

Archiver

The ARR_kbd and AR_kbd programs are removed.
Use ARR_cmd and AR_cmd.

libCom/cvtFast.c
---
replaced gcvt by sprintf with e format.
cvtFast.h - removed support for non-ansi prototypes
The ExpToString routines now just call sprintf. Bug for .1 fixed

libvxWorks
---
removed iocFloatLib.c

dbStaticLib

It is now ok if DTYP does nor have associated INP or OUT
Check for record names too long.

Added dbReportDeviceConfig

dbTest

Added dbhcr (Hardware Configuration Report)
To use type (on ioc)
dbhcr > report
Then use the Unix sort command
sort < report > reportSort

dbAccess
---
dbPutLink no longer calls recGblRecordError (error message storms)

taskwd

Now allows callback to call taskwdRemove

errPrintf

Fixed bug that caused message to be suppressed
if status=0 and errno=0.
Now calls logMsg instead of printf.

Scan Tasks
---
added scanOnce.
If scanAdd fails set precord->scan=SCAN_PASSIVE
Fixed bug that causes failure if first record in database
has 28 character name.
Will only call taskwdRemove if restarting

Analog Output Record

Added AOFF and ASLO

Binary Output Record

When the binary output record support module processes the record
because of the HIGH field, it calls dbProcess rather than the
process entry itself. This if the record becomes disabled between
the time VAL is set to 1 and the time specified by HIGH occurs then
the record will not be processed. Before it would get processed.

Event Record and soft device support.

Made the VAL field a prompt field. The constant link value is
now moved to the VAL field only if VAL is zero.

Histogram Record

Fixed bug in callback mechanism

Calc Record

Modified postfix.c so it returns a non zero value when it can
detect improper usage of a comma in a calc record expression.

Added support for constants CONST_PI CONST_R2D and CONST_D2R
(PI, Deg to Radians, Radians to Degrees)

Calc record now allows usage of lower case anywhere in expression.

Sequence Record

Added monitor release code for the values that are fetched and
then put when the links are processed.

compress record

Fixed th updating of the inx field for the AVERAGE algorithm

dbAccess.c

Float/Double conversions to string were incorrect for value>1e7.

Allen Bradley IXE Millivolt Input

This was not correctly performing linear conversions. It is fixed.

Hardware Error Messages and Error Message Storms:

A problem arose because Allen Bradley Analog Outputs caused error messages to be generated every time a record was processed if the Analog Output did not exist or if the Adapter was down.

A check was made of many of the existing device support modules. They were changed to implement the following algorithm:
1) If a hardware error was detected during record processing the associated record is put into alarm. This was true before.
2) If errVerbose is TRUE and the alarm status and/or sevrity is being changed then an error message is issued.

NOTE: errVerbose has the value False. It can be set true by issuing the vxWorks shell command:
errVerbose=1
This can be done in an ioc startup file just before iocInit or at any other time.

The device support modules that use this new algorithm are:
devAiDvx2502.c
devAiXy566Di.c
devAiXy566DiL.c
devAiXy566Se.c
devAllenBradley.c
devAoVmiVme4100.c
devBoMpv902.c
devBiXVme210.c
devMbboMpv902.c
devBiMpv910.c
devBoXVme220.c
devMbbiMpv910.c
devMbboXVme220.c
devMbbiXVme210.c

drvOms

Replaced itob by cvtLongToString

Subtool, dbLoadRecords, and dbLoadTemplate

Fixed a bug in the variable substitution software. Subtool, dbLoadRecords, and dbLoadTemplate had a buffering problem.

VXI resource manager

A bug was fixed in the VXI resource manager. The bug only effects systems with multiple VXI crates interconnected by the MXI bus. The resource manager was allowed to discover the presence of another MXI bus repeater while searching for devices in a MXI bus repeater. This resulted in a bus addressing conflict. The solution was to search for and discover the presence of all devices on the MXI bus prior to opening the LA window of the first MXI found and searching for devices underneath of it.

Bitbus driver

Both the PEP and Xycom bitbus drivers were updated such that a possible race condition is eliminated in WdTask. A bug fix was added to the bitbus driver.

Allen Bradley RTD module

Support is now available for the Allen Bradley RTD Analog Input module. Two new device types now appear; one for Platinum and the other for Copper. The default is to convert to degF. The parm field of INP can be set to C if you want degC and to O if you want ohms.

2. Channel access bug fixes

A 32k memory leak was discovered in the CA client library. The leak occurred each time that a communication linkl with an IOC was lost.

A bug was fixed in the vxWorks version of the CA client library relating to connection cleanup. Symptoms of the problem are bus errors when a vxWorks CA client exits.

It was observed that the CA server would stop accepting new CA client connections if an IOC reached its maximum file descriptor limit even if steps were taken to close exiting CA client connections to free up exiting file descriptor allocations. The bug was fixed.

A <50 byte memory leak was fixed in the CA server. The leak occurred each time a client disconnected.

3. Application Source/Release Control

makefile (top)

Changed logic - The three major invocations are:
command: make
just invokes make in each application directory
command: make world
invokes doGets to get any out-of-date sccs controlled files
then invokes makesdr to bring sdr files up-to-date
then invokes make world in each application directory
command: make dotar
builds a compressed tar file of this whole structure
command: make tarinfo
explains how to unpack the compressed tar file.

makefile (app)

Changed logic - The two major invocations are:
command: make
just does makes in the target arch, Db's
and if present invokes makefile.pvt
command: make world
invokes doGets to get any out-of-date sccs controlled files
then rebuilds Makefile(s) from Imakefile(s)
then invokes make on the rebuilt Makefile(s)
then invokes make in each <*>Db directory
then invokes make -f makefile.pvt (if present)

makefile (dct)

Makefile has been changed to use default.dctsdr, default.sdrsum and <file>.rpt dependencies to determine when to create <file>Db.database. It has been changes to use atdb/dbta instead of DCT to create the database and the short form reports. The makefile now also allows usage "make <file>Db.database" to make only one database.

makefile (gdct)

This makefile does nothing unless there is a makefile.pvt present in the same directory. It is invoked by makefile.gdct

makesdr

Makesdr has been changed to use dependencies on <epics>/share/ascii/*.ascii, cat_ascii/*.ascii, rep_ascii/*.ascii and recursively on any files included by any of the ascii files to determine when to create default.dctsdr and default.sdrsum.

apSccsScan, doGets
---
apSccaScan has been replaced by new tool doGets which performs the same function as apSccsScan and now works recursively from any directory in which it is invoked.

apFixDctRpt

New file to convert DCT reports to atdb/dbta format.

apCreateIocName

Now puts .epicsUnix links in all ioc directories.

apCreateDbDir

Reworked to allow creation of a plain Db directory.

archList

New file for list of target architectures for the build objects in an application.

apBlds

Changed to use archList for list of target architectures to be built.

apFixLinks

Corrected handling of makefile.pvt.

<appName>/op/doc

Added op/doc directory to application template directory structure.

<appName>/src/ImakefileVx

Added epicsH/rec to INCLUDE path.
Changed default definintion of SNCFLAGS to null n the ImakefileVx template.
Added rules MakeObjRuleForStateCpp and MakeObjRuleForStateCppFlags for using the c pre-processor, cpp, prior to the state notation compiler, snc.

dbsccs

Added dbsccs create and replaced DCT usage with atdb/dbta calls.

4. Unix Tools

alh

Added startup file selection boxes for log files and changed alh to use dbDefs.h to define pvname and field name lengths.

DCT

For CT_HEX fields a leading 0x is now printed.

gdct

Added the ability to put substitution variables into ".db" files using gdct. Gdct now allows substitution variables to be put in to any dct editable field.
gdct bugs fixed, including duplication of many records at a time.

dbreport

A ".db" file reporting program has been added, this program produces reports on process variables and fields from a ".db" file. A man page for dbreport has also been added.

snaps

The snaps program has been modified to use the file description manager to control the pending events, all the input data channels are monitored, the timestamp is also returned for the monitored channels, the time scale for the plot with accuracy to milli-seconds for pulse mode.

toolComm/ca

Removed the restriction of putting non null string array to IOC.
Fixed the problem of entering multiple put enum waveform records.
Allowed an option of automatic flush pend events before checking for wait events. Monitored channel also returns timestamp.

CaMath

A new function CaWaitFlush is added to CaMath which allows the user to set automatic event flush to on or off before performing the check event.

xmsnaps

A new tool, xmsnaps, will be available to EPICS users in production release R.3.11.4. Xmsnaps is an integrated Motif application which can provide a view of the most current history for a number of monitored process variables(PVs). Any changes in the monitored process variable values will be automatically captured by the snaps program. It uses an adjustable circular buffer to store the most current snapshots of the moitored process variables.
Xmsnaps can operate in three different modes: pulse mode, periodic scannning mode, and periodic event-driven scannning mode. It can snap the values of the monitored PVs, export captrued data to disk files, display text data files, and plot the results.

The user guide for "xmsnaps" is available in the user drawer of ApsDoc Interleaf cabinet.

stripTool

A new tool, stripTool, is now available to EPICS users in production release R.3.11.6. StripTool is a menu driven X-Window Motif application which allows users to create strip charts to monitor up to ten specified channels on a single strip chart. StripTool is available as one of the EPICS unbundled products in the epics/add_on/bin directory.

5. Motif-based Editor/Display Manager

NEW FEATURES:
=============

PRIVATE COLORMAP
----
a private colormap can be specified for use in MEDM (complete with flashing) so that MEDM can run even when color-greedy applications like PV-WAVE run. this option is enabled by a command-line option "-cmap". the default is not to use a private colormap...

TRIGGER CHANNEL FOR CARTESIAN PLOTS
----
trigger channels for Cartesian plots has been implemented and is available for testing. existing (non-triggered) plots should work as they currently do, but, by specification of a trigger channel PV name, plot visual updates occur only when the trigger channel changes. this can be used to throttle the visual updates to some modest rate, although the internal data updating will occur at whatever the plotted channel data rate is...

EPICS_DISPLAY_PATH

The EPICS_DISPLAY_PATH environment variable can now use colon-separated format for multiple search directories. In other words, setting the EPICS_DISPLAY_PATH environment variable to a string of the format a:b:c:d... will result in file access attempts in the directory order specified, and, upon successful access, will use the first accessible directory/file pathname.

COMMAND-LINE MACRO-SUBSTITUTION

The macro-substitution mechanism in place for related displays has been extended for "top level" displays. A new command line option "-macro" has been introduced which allows passing of a string of macro substitutions for display startup time, to be applied to the specified displays on the MEDM command line. This option is only honored for EXECUTE mode MEDM startup (-x) and applies only to command line specified displays.

The form of the option is: -macro "a=b,c=d"
with the double quotes being a required part of the syntax, and, again is only honored if -x is also specified.

This option also applies for remote startup of displays. Hence, if an instance of MEDM was started in EXECUTE mode to a display, and another instance is invoked via

```
% medm -x -macro "a=b,c=d" macro1.adl macro2.adl macro3.adl
```

then the display requests for macro[123].adl will be forwarded

to the remote MEDM along with macro substitution information for creation in that context.

MAIN MENU
----
de-sensitize Edit pulldown menu and File->New,Save,SaveAs buttons when in execute mode (prevents saving in execute mode, for instance)

-CMAP (private cmap) changes
============================

the use of BlackPixel(dpy,scr)/WhitePixel(dpy,scr) by xrt/graph, and the xc/ derived widgets causes problems with the private cmap. the private cmap now allocates a black and white pixel which maps to the same value as in the default colormap, hence circumventing the problem for Pseudo-Color visuals... a fix was also introduced into the xc/ widgets to properly get colors out of the widgets colormap not the default colormap

CONTROLLERS
===========

Controller Object updates now allowed while cursor is in controller.

The handling of controller objects as simultaneous monitors has been changed to guarantee accuracy with underlying channels (within the limits of CA's monitor mechanism).
All controllers now will update regardless of presence of cursor in that object. The previous "enter-don't-update" feature was introduced to make valuators predictable to use and was applied to all controllers. In retrospect this was not necessary or sensible for "discrete" controllers like message buttons, etc.

Controllers now update even when the cursor is within the confines of the controller object. However, for valuators, the updates are disabled during DRAG operations, and re-enabled upon conclusion of DRAG. This behavior is appropriate for the valuator and gets around earlier problems. Note that DRAG is initiated by selecting the slider AND MOVING IT; merely selecting the slider via MB1 will not disable updates until it is also moved.

remove <Key>space translations for push buttons/radio box/toggle buttons (per Ned - to prevent unintentional pushing of buttons, e.g., after screen saver kicks in)

alarm modified colors for choice button, message button and menu now consistent with all other monitors/controllers

FILE SAVE OPERATIONS
====================

MEDM now carefully honors file permissions for display files under SAVE operations. Various combinations of directory and file permissions are possible and are now correctly handled.

BUG FIXES:
==========

bug fix in valuator expose/redraw mechanism

```
properly handle window parent chaining mechanism at root window

the "LinacSequence.adl" maximize problem has been fixed

the "cannot allocate colors" colormap/X device-ran-out-of-colors
    problem has been fixed

the "realloc-ing input fd array" problem has been fixed

don't do monitor list traversal if in EDIT mode

add a single caHeartBeat() timer proc in main to handle misc. CA
processing (remove code which added an XtTimerProc caHeartBeat()
for each transition to EXECUTE mode in modeCallback -- this was
clearly a bug and probably is responsible for problems when
people do lots of Edit <-> Execute  transitions...)

An alarm color mode problem in private colormap (-cmap)
instances of MEDM has been identified and fixed.

Controllers as Monitors:
There was a bug in the monitor list traversal (updateMonitors.c)
code that treated controllers as monitors, including not doing
updates when (data->value == data->displayedValue).  Note that
controllers are different than strict monitors in this regard,
because the user can change the visual state of the object
such that data->displayedValue is not accurate.

E.g., for choice buttons, the user can depress a button which
sets a toggle which initiates a ca_put() which eventually
gets back to the monitor and updates its state.  But if a
another mechanism updates that state rapidly such that the
traverseMonitorList() routine doesn't get around to the update
in time, the value can be reset to its old value which
is inconsistent with the visual state, yet the visual state
is not updated since (data->value == data->displayedValue).

This problem has been fixed, and controller's monitors are
keyed off of data->modified alone which should guarantee
the correct visual state regardless of user interactions.
```