

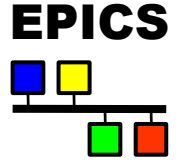
# EPICS Base 3.14 Introduction

## APS Controls 22JAN2003

Marty Kraimer  
APS/ANL



# Base 3.14 Overview



## •Main Goal for 3.14: port iocCore

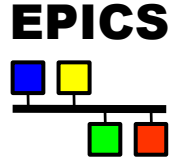
- vxWorks
- RTEMS - Open Source Real Time Operating System
- Solaris
- Linux
- Win32
- Darwin - Open Source for Mac OSX
- HPUX11 - Release 3.14.1
- Base software is organized to minimize porting effort
- Marty Kraimer, Jeff Hill, Janet Anderson, Eric Norum, and Ralph Lange primary developers.

## •Hardware Support

- vxWorks support unbundled
- RTEMS used at CLS and SSRL. Thus hardware support growing.
- Requires major changes for non VME/vxWorks platforms



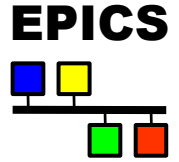
## 3.14 Channel Access Features



- Large Arrays
  - Both client and server must be 3.14
  - Set `CA_MAX_ARRAY_BYTES` on both client and server
  - Transparent to existing clients but must be relinked
  - Purpose is ease of use not performance
  - PCAS also supports large arrays.
- Multipriority CA servers
  - Client must request, i.e. not transparent to existing clients.
  - Provides incremental improvement for performance degradatation. Purpose is to allow inter IOC communication to have higher priority than other clients.
- Channel Access Reference Manual available via Release WWW page



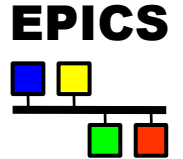
# Getting Started



- ◆ Application Developer's Guide for 3.14.1
  - ◆ Chapter 2: New Features for 3.14
  - ◆ Section 2.2: Example Application
    - ◆ IocCore example for any supported platform
    - ◆ Example SNL that uses unbundled sequencer
    - ◆ Two channel access client examples
  - ◆ Chapter 4: EPICS Build Facility
- ◆ Other Documentation available via EPICS Home Page
  - ◆ <http://www.aps.anl.gov/epics/>
  - ◆ Under IOC Software, select R3.14
  - ◆ Then select Release R3.14.1



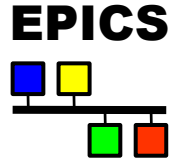
# Example IOC Application



```
mercury% mkdir example
mercury% cd example
mercury% /usr/local/iocapps/R3.14.1/support/base/3-14-
1/bin/solaris-sparc/makeBaseApp.pl -t example example
mercury% /usr/local/iocapps/R3.14.1/support/base/3-14-
1/bin/solaris-sparc/makeBaseApp.pl -i -t example host
The following target architectures are available in base:
    solaris-sparc
    vxWorks-68040
What architecture do you want to use? solaris-sparc
mercury% ls
Makefile      configure     exampleApp   iocBoot
mercury% gnumake
```



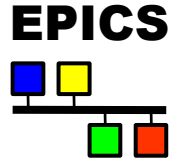
# Running Example



```
Mercury% cd iocBoot/iocHost
mercury% ../../bin/solaris-sparc/example st.cmd
dbLoadDatabase("../db/example.dbd",0,0)
registerRecordDeviceDriver(pdbbase)
dbLoadRecords("../db/dbExample1.db","user=mrkHost")
dbLoadRecords("../db/dbExample2.db","user=mrkHost,no=1,scan=1
second")
dbLoadRecords("../db/dbExample2.db","user=mrkHost,no=2,scan=2
second")
dbLoadRecords("../db/dbExample2.db","user=mrkHost,no=3,scan=5
second")
dbLoadRecords("../db/dbSubExample.db","user=mrkHost")
iocInit()
Starting iocInit
#####
### EPICS IOC CORE built on Jan 9 2003
### EPICS R3.14.1 $R3-14-1$ $2002/12/12 15:56:59$
#####
iocInit: All initialization complete
epics>
```



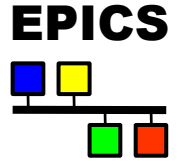
# Major Porting Problems



- ◆ VME/VXI - Hardware Support Unbundled.
- ◆ vxWorks libraries - Define/Implement OSI interfaces.
- ◆ vxWorks dynamic loading - Registry.
- ◆ Build Environment - Major changes.
- ◆ vxWorks shell - iocsh ( ioc shell)
- ◆ Interrupt Level support -  
Use a global mutex if OS doesn't allow.



# Operating System Independent Interfaces

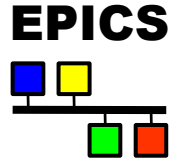


epicsRing	rngLib	Generic
epicsTimer	wdLib,osiTimer	Generic
epicsAssert	epicsAssert	Default, vxWorks
epicsEvent	semLib	RTEMS,WIN32,POSIX,vxWorks
epicsFindSymbol	symFindByName	Default,vxWorks
epicsInterrupt	intLib	RTEMS,default,vxWorks
epicsMutex	semLib	RTEMS,WIN32,POSIX,vxWorks
epicsThread	taskLib	RTEMS,WIN32,POSIX,vxWorks
epicsTime	tickLib,osiTime	RTEMS,WIN32,POSIX,vxWorks
osiPoolStatus	memLib	RTEMS,WIN32,default,vxWorks
osiProcess	osiProcess	RTEMS,WIN32,POSIX,vxWorks
osiSigPipeIgnore	osiSigPipeIgnore	WIN32,default,POSIX,vxWorks
osiSock	osiSock	Linux,RTEMS,WIN,default,solaris,vxW





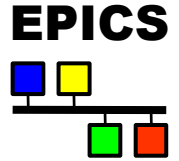
# Some Details



- ◆ Registry
  - ◆ vxWorks symFindByName - bind to global symbol.
  - ◆ iocCore dynamically binds record/device/driver/etc support.
  - ◆ While building application:
    - ◆ A perl program generates a C function
    - ◆ C function is linked with application. During startup the C function is called. It registers the support.
- ◆ Build Environment
  - ◆ Extensive changes, more functionality, easier to use.
- ◆ Iocsh – simple command interpreter, built in commands.
- ◆ Interrupt Level Support
  - ◆ IocCore has minimal use.
  - ◆ vxWorks, RTEMS real support. Default uses global mutex.



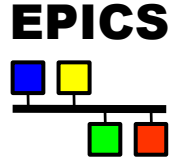
# Compatibility



- ◆ Building R3.13 IOC applications with 3.14.
  - ◆ The old 3.13 build rules are still supported.
  - ◆ Using old build rules requires few changes. However all hardware support is unbundled.
  - ◆ Should convert to new rules ASAP
- ◆ Old CA client interface still supported
  - ◆ Many CA client applications have been built with 3.14.
- ◆ Converting R3.13 IOC Applications to new build rules
  - ◆ New rules in <top>/configure
  - ◆ Single Makefile in application directories
  - ◆ Conversion instructions available on WWW



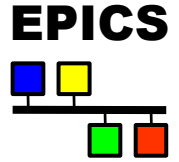
# Two Killer Apps



- ◆ Sequencer - Unbundled version (KECK and now SLAC)
  - ◆ Works on all supported platforms
  - ◆ Works as standalone process or as part of iocCore
- ◆ GpibCore
  - ◆ Benjamin Franksen's version ported to R3.14.
  - ◆ HP LAN supported on all platforms
  - ◆ NI1014 supported on vxWorks
- ◆ Both are built as support apps in  
`/usr/local/iocapps/R3.14.1/support`



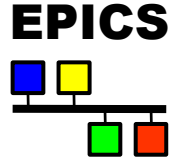
# Hardware Tested with 3.14



- ◆ The following support has been tested (vxWorks only)
  - ◆ Allen Bradley 6008 Scanner, 1771 I/O, etc.
  - ◆ Stepper motor support from base 3.13 (NOT Tim Mooney's)
  - ◆ Mizar 8310
  - ◆ XYCOM: 566, 210, 220, 240
  - ◆ Analogic dvx2502
  - ◆ Burr Brown mpv902 and mpv910
  - ◆ VMIC 4100
  - ◆ Acromag avme9440
- ◆ Bitbus was tested but not for 3.14.1
- ◆ None of this is currently built in 3.14.1/support



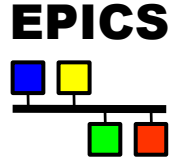
# Converting Existing Apps



- ◆ R3.14 uses more memory
  - ◆ A test example with 4000 records
    - ◆ 3.13 uses 7.67 megaBytes
    - ◆ 3.14 uses 9.01 megaBytes
  - ◆ If already short on memory then a big problem
- ◆ CA client appears to use much more CPU time
  - ◆ Being investigated



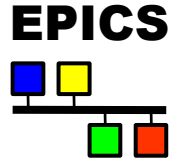
# New Build System



- ◆ Major task for porting iocCore
  - ◆ Before Makefile, Makefile.Host, Makefile.Vx.
  - ◆ What now? In particular Makefile.Vx
- ◆ Solution
  - ◆ Single Makefile builds everything.
  - ◆ Many things can be built in a single directory
- ◆ Makefile.Vx
  - ◆ Did not use regular object libraries
  - ◆ Now everything uses regular libraries as defined by platform
- ◆ A single executable is created even for vxWorks iocs.



# Build Host Product



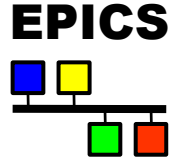
```
PROD_HOST += caExample
caExample_SRCS      += caExample.c
caExample_LIBS     += ca
caExample_LIBS     += Com
```

```
PROD_HOST += caMonitor
caMonitor_SRCS      += caMonitor.c
caMonitor_LIBS     += ca
caMonitor_LIBS     += Com
```

```
ca_DIR           = $(EPICS_BASE_LIB)
Com_DIR          = $(EPICS_BASE_LIB)
```



# Building Database Files



```
# xxxRecord.h will be created from xxxRecord.dbd  
DBDINC += xxxRecord
```

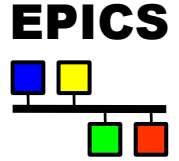
```
# <name>.dbd will be created from <name>Include.dbd  
DBD += example.dbd
```

```
# name<i>.db will be created from name<i>.substitutions  
and <name>.template  
DB += name1.db name2.db
```





# Building IOC product



```
PROD_IOC += example
example_SRCS += xxxRecord.c
example_SRCS += devXxxSoft.c
example_SRCS += dbSubExample.c
```

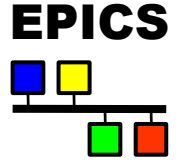
```
# <name>_registerRecordDeviceDriver.cpp will be created from
<name>.dbd
```

```
example_SRCS += example_registerRecordDeviceDriver.cpp
example_SRCS_DEFAULT += exampleMain.cpp
example_SRCS_vxWorks += -nil-
```

```
#The following adds support from base/src/vxWorks
example_OBJS_vxWorks += $(EPICS_BASE_BIN)/vxComLibrary
```



# Building IOC Product cont.



```
# Use win32 object libs for registered
support
example_LIBS_win32 += recIocObj
example_LIBS_win32 += softDevIocObj
example_LIBS_win32 += testDevIocObj
example_LIBS_DEFAULT += recIoc
example_LIBS_DEFAULT += softDevIoc
example_LIBS_DEFAULT += testDevIoc
example_LIBS += iocsh
example_LIBS += miscIoc
example_LIBS += rsrvIoc
example_LIBS += dbtoolsIoc
example_LIBS += asIoc
example_LIBS += dbIoc
example_LIBS += registryIoc
example_LIBS += dbStaticIoc
example_LIBS += ca
example_LIBS += Com
```