

EPICS Interface to Area Detectors

Mark Rivers
CARS

APS-TWG
October 16, 2003

Overview

- Goal: Uniform interface for controlling area detectors (CCD, online image plates) from EPICS
 - Any EPICS client (e.g. spec, IDL, scan record) can control (at a bare minimum).
 - Exposure time
 - File name
 - Start collection, wait for completion
 - Much more control for most detectors
- Current status:
 - MAR 165 CCD (complete, in use on Sector 1)
 - Roper CCD detectors (complete, Sector 13 this run)
 - Bruker CCD detectors (in-progress, Sector 13 this run)
 - Will be based on Tim Graber's work
 - MAR 345 online image plate (soon)
 - Will be based on Keith Brister's work

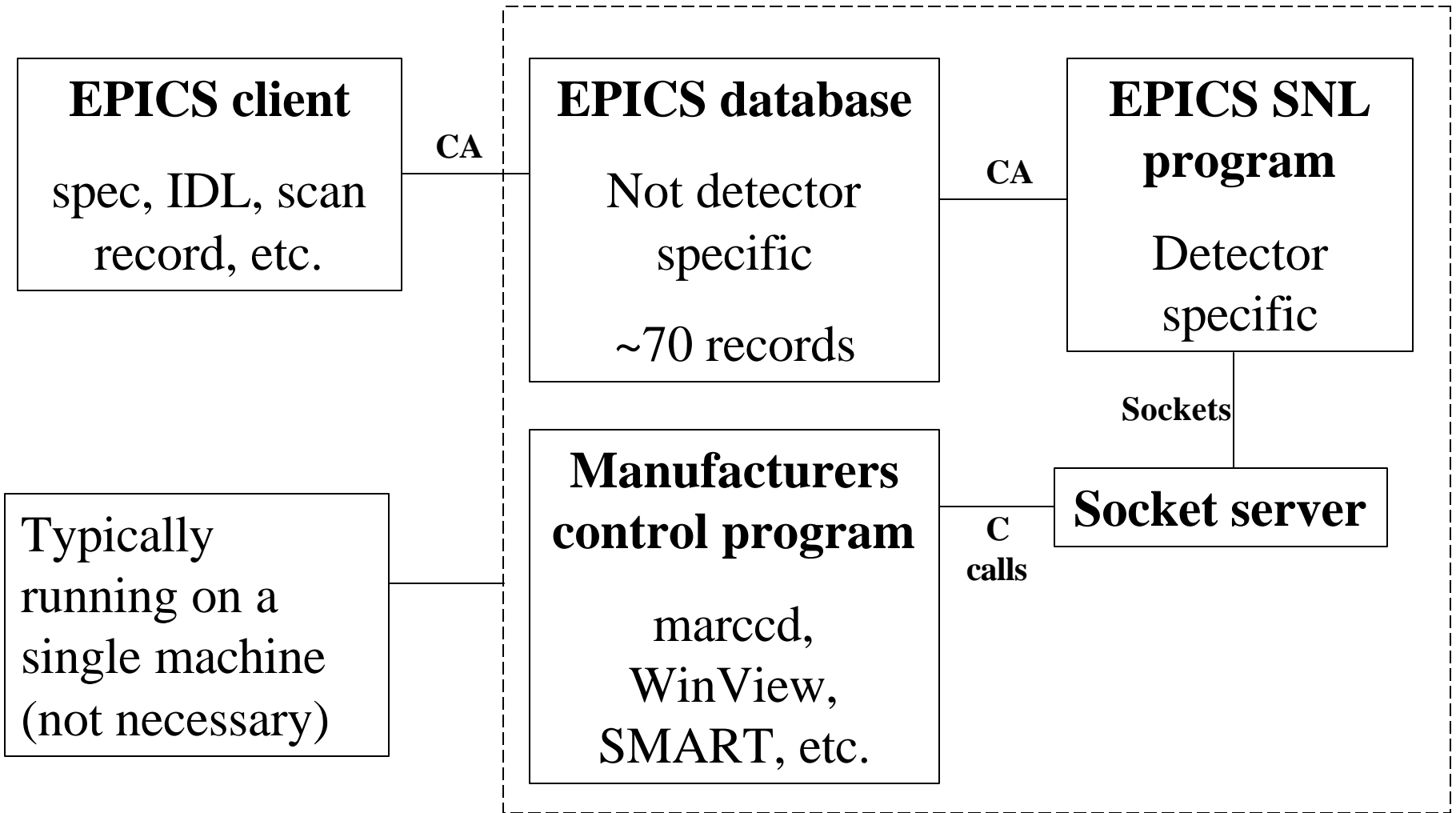
Implementation

- Use manufacturer's software for primary user interface.
 - Minimizes amount of new code
 - Uses existing file formats, unwarping algorithms, etc.
 - These programs include:
 - marccd for MAR165
 - Winview/Winspec for Roper cameras
 - SMART for Bruker cameras (can also use WinView)
 - scan345 for MAR 345 image plate

Implementation

- Control these programs from EPICS
 - Each of these programs has a “remote control” interface, typically using TCP/IP sockets
 - Using EPICS means each client (e.g. spec) does not have to know how to talk to each type of detector. Only has to know how to talk to EPICS.
- EPICS software consists of
 - Database of records (PVs), identical for all detectors
 - State-notation-language (SNL) programs, unique for each detector. Reads/writes PVs and communicates with remote control interface over sockets.
 - The database and SNL programs are typically run on the same machine that the user interface software runs on (e.g. Linux box for MAR detectors, Windows for Roper and Bruker). **No VME crate required.**

Schematic Architecture



Relationship with Brian Tieman's Package

- Complementary to, and compatible with, Brian's package.
 - My software uses the same PV names as Brian's wherever possible, so EPICS clients that work with his software should work with mine
- Brian's uses the EPICS portable channel access server, rather than running a real EPICS database and SNL program on the server machine
 - Only option when Brian wrote his code, before EPICS 3.14.
- Brian's program calls library (DLL) layer directly, bypasses manufacturers user interface program
 - More flexible
 - More code
 - File format is HDF, not manufacturers format that many data processing applications expect
 - Unwarping is not implemented
- Brian's program is a Windows application, and so only works on Windows detectors (Roper, several others). Not MAR or other Unix detectors.

“Expert” medm screen

Area Detector Control

Setup

EPICS name roperCCD:det1:
Manufacturer Roper
Model ST138
SNL Program Running
Server name gselab1
Server port 5001
Connect
Temperature -20.00 1.00
Print debugging
User input
To detector save, c:\temp\test021
From detector OK

Shutter

Shutter Control
Shutter PV roperCCD:det1:Shutter
Shutter status Closed
Open command Open
Close command Closed
Open/Close
Open delay 1.000
Close delay 0.050

File

File path c:\temp\
Base filename test
Next file # 22
Filename format %s%3d.SPE
Comment 1 test comment 1
Comment 2 test comment 2
Comment 3 another comment
Comment 4
Comment 5
Correct bkgnd
Correct flat.
Correct spatial
Auto save
Save file
Last filename test021.SPE

Readout

Binning (X) 2 (Y) 2
Top/bottom (T) 1 (B) 1152
Left/Right (L) 1 (R) 1242
Total counts 2.63258e+008
Net counts 1.29189e+006
Compute counts

Collect

Exposure time 0.200 0.300
frames 1 1
Frame type
Acquire
Detector state Idle
Time remaining 0.0
Polling rate
Abort

Many fields do not apply to all detectors. Simpler screens (e.g. for MAR 165 only) can easily be made.

Process variables (PVs)

\$(P)\$(C)Abort	\$(P)\$(C)AcquireCLBK	\$(P)\$(C)AcquirePOLL	\$(P)\$(C)ActualBinX
\$(P)\$(C)ActualBinY	\$(P)\$(C)ActualNumFrames	\$(P)\$(C)ActualROIBottom	\$(P)\$(C)ActualROILeft
\$(P)\$(C)ActualROIRight	\$(P)\$(C)ActualROITop	\$(P)\$(C)ActualSeconds	\$(P)\$(C)ADC
\$(P)\$(C)AutoSave	\$(P)\$(C)BinX	\$(P)\$(C)BinY	\$(P)\$(C)BitDepth
\$(P)\$(C)CCDManufacturer	\$(P)\$(C)CCDModel	\$(P)\$(C)CloseShutter	\$(P)\$(C)CloseShutterDly
\$(P)\$(C)CloseShutterStr	\$(P)\$(C)Comment1	\$(P)\$(C)Comment2	\$(P)\$(C)Comment3
\$(P)\$(C)Comment4	\$(P)\$(C)Comment5	\$(P)\$(C)Compression	\$(P)\$(C)ComputeROIcts
\$(P)\$(C)ConnectState	\$(P)\$(C)CorrectBackground	\$(P)\$(C)CorrectFlatfield	\$(P)\$(C)CorrectSpatial
\$(P)\$(C)DebugFlag	\$(P)\$(C)DetectorState	\$(P)\$(C)DetInStr	\$(P)\$(C)DetOutStr
\$(P)\$(C)FilenameFormat	\$(P)\$(C)FilePath	\$(P)\$(C)FileTemplate	\$(P)\$(C)FrameType
\$(P)\$(C)FullFilename	\$(P)\$(C)HDFTemplate	\$(P)\$(C)Hours	\$(P)\$(C)Initialize
\$(P)\$(C)MeasuredTemp	\$(P)\$(C)Milliseconds	\$(P)\$(C)Minutes	\$(P)\$(C)NumExposures
\$(P)\$(C)NumFrames	\$(P)\$(C)OpenShutter	\$(P)\$(C)OpenShutterDly	\$(P)\$(C)OpenShutterStr
\$(P)\$(C)PollDetState	\$(P)\$(C)ROIBottom	\$(P)\$(C)ROILeft	\$(P)\$(C)ROINet
\$(P)\$(C)ROIRight	\$(P)\$(C)ROITop	\$(P)\$(C)ROITotal	\$(P)\$(C)SaveFile
\$(P)\$(C)Seconds	\$(P)\$(C)SeqNumber	\$(P)\$(C)ServerName	\$(P)\$(C)ServerPort
\$(P)\$(C)SetTemp	\$(P)\$(C)Shutter	\$(P)\$(C)ShutterMode	\$(P)\$(C)ShutterStatus
\$(P)\$(C)SNLWatchdog	\$(P)\$(C)TimeRemaining	\$(P)\$(C)UserInStr	

marccd remote control

The screenshot displays the marccd software interface. At the top, the menu bar includes File, Edit, View, Configure, Acquire, and Options. The status bar shows: Protocol: None, Detector: Clearing, Intensity: [Stop], Shutter: Closed, Temperature: -79.8 C, Pressure: 0.02 Torr, and Status: Cooler ON. The main window shows a large circular field of view. On the right, there is a control panel with a magnification of 2x, a vertical scale from 0 to 64, and a table of parameters:

	X	Y	
DENZO	512	556	11
Fofter:	63.66	148.47	1.73
Beam:	1036	1036	pixels
	51.47	51.47	mm
Zone:	512	512	pixels
	40.26	40.26	mm
Misset:	-524	-524	pixels
	-41.21	-41.21	mm
	-22.40	-22.40	degrees
Spacing:		11.32	Ang.
Radius:	Layer 0:	741	pixels
	Layer N:	20	pixels
Resolution:		256	pixels
		11.38	degrees
Resolution:			5.21 Ang.

Below the table are buttons for Gonostat and Beam. Further down, a table shows Gonostat settings:

	Current	Target	Units
Distance	0.000	0.000	mm

At the bottom of the control panel, there are buttons for Drive (yellow), Stop (red), Align, and To Header.

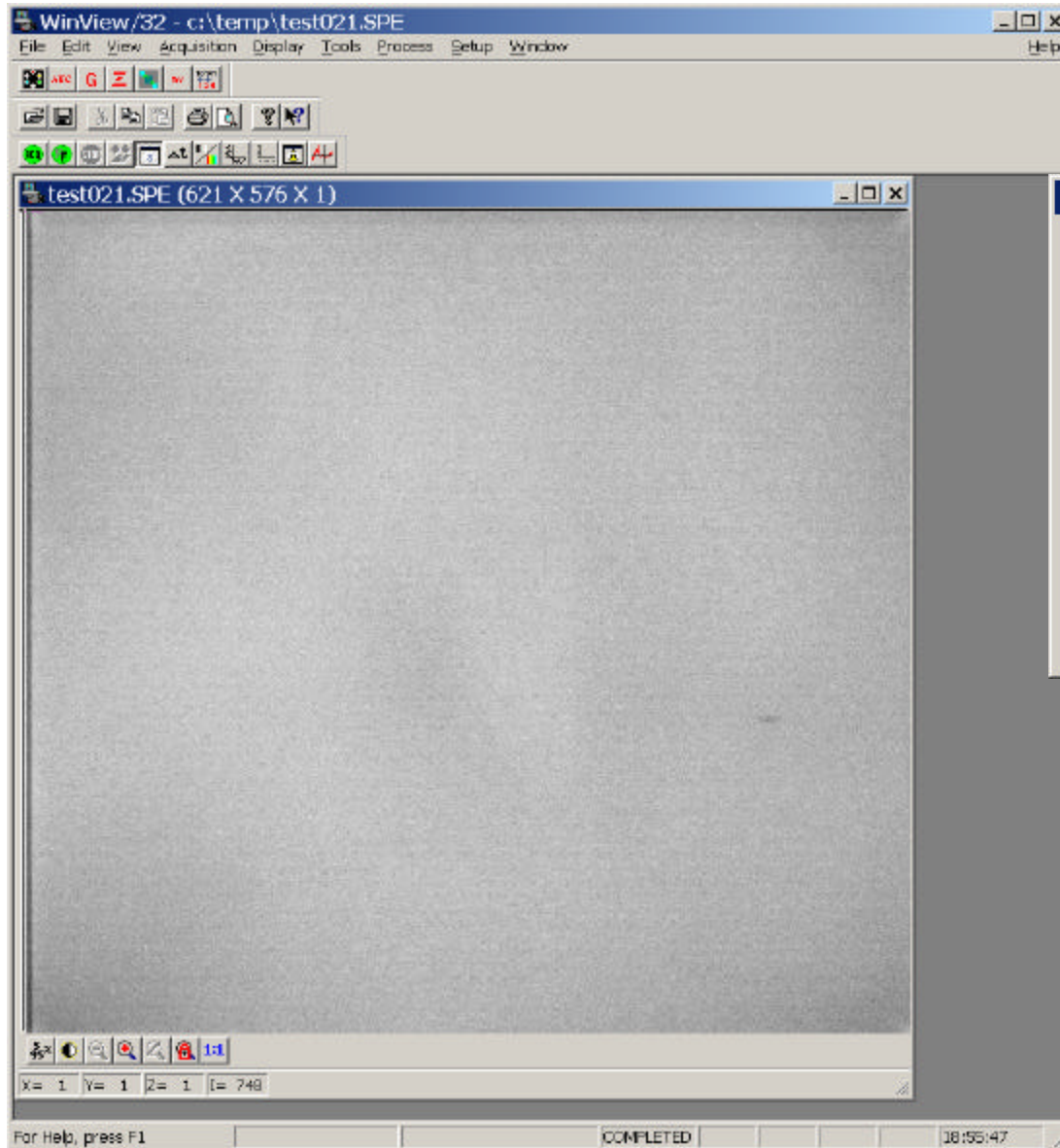
The 'Acquire/Remote Control dialog' is a separate window in the foreground. It contains the following fields and buttons:

- Device Database Host: localhost
- Device Database Server: /home/marccd/source/servers/marccd_server (with a Browse button)
- Device Personal Name: 3000
- Command Log: (with a Browse button)
- Buttons: Start (yellow), Stop (red), Save, Cancel, Dismiss, Help

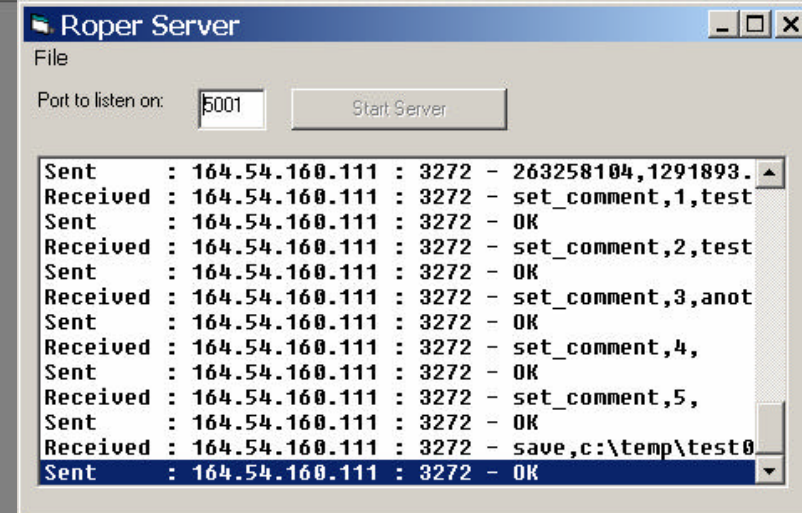
A terminal window at the bottom right shows the following output:

```
File Edit View Terminal Go Help
gunzip: stdout: Broken pipe
gunzip: stdout: Broken pipe
```

Roper Interface



WinView (from Roper)



Socket server written in Visual Basic. Simple ASCII commands. Calls COM interface to automate WinView & WinSpec). Not EPICS specific, other applications can talk to it.