Channel Access and Client Tools

Author: Kenneth Evans, Jr., August 2004
Modified: Kay Kasemir, October 2006
Andrew Johnson, January 2007
EPICS Overview

Client Tools

- EDM
- MEDM
- Client
- Client
- Probe
- ALH

Channel Access

- Server
- IOC
- IOC
- IOC

- Meter
- Power Supply
- Camera
Channel Access

- The EPICS “software bus”
- Used to read and write values to/from Process Variables
- To many people, Channel Access is EPICS
  - Especially those that have no IOC experience
  - “Integrate X into EPICS” often means “Be able to control X via CA”
- CA is not defined by a protocol specification
  - Jeff Hill (LANL) maintains the CA client and server libraries
  - A single expert maintainer for both ensures very robust control systems
What is a Process Variable (PV)

“A named item of data, with associated optional attributes”
- Data is an Integer, Floating point number, enumeration value or string, or an array of any of those types
- Attributes include timestamp, alarm status/severity, precision, engineering units string, list of enumeration strings, operator/control/alarm limits
Channel Access in One Slide

CA Client

CA Server

Process Variables:

S1A:H1:CurrentAO
S1A:H1:CurrentAO
S1:P1:x
S1:P1:y
S1:G1:vacuum

Who has a PV named “S1A:H1:CurrentAO”?

I do.

Change its value to 30.5

OK, it is now 30.5

Notify me when the value changes

It is now 20.5 AMPS

It is now 10.5 AMPS

It is now -0.0023 AMPS

“connection request” or “search request”

“get” or “caGet”

“put” or “caPut”

“set a monitor”

What is its value?

25.5 AMPS

30.5 is too high. It is now set to the maximum value of 27.5.

30.5 is too high. It is now set to the maximum value of 27.5.

You are not authorized to change this value

“put complete”

or

or

“post an event” or “post a monitor”

You are not authorized to change this value
Tools Covered in This Presentation

- Command-Line Tools
  - caget, caput, camonitor, cainfo
- Probe
- StripTool
- MEDM
- ALH
More Information

- There is a wealth of information in the EPICS web pages
- Each of the Extensions covered here has its own page with much additional information
  - Including tar files of the latest releases
- There are many other tools described there as well

- The Extensions *executables* are typically located at
  - …epics/extensions/bin/<platform>/<executable>
  - e.g. /opt/epics/extensions/bin/solaris-sparc/edm
  - Platforms are solaris-sparc, linux-x86, win32-x86, etc.
- The Base command line tools are typically at
  - …epics/base/bin/<platform>/<executable>
EPICS Extensions Web Page

Experimental Physics and Industrial Control System

Extensions

The following list gives access to individual pages for most of the standard EPICS host tools and CA clients. Note that some of the minor pages linked below do not appear in the sidebar on the left.

Some of this software can be downloaded from the individual web pages linked below, and the collection of tools from AES are also available bundled together. See the Extensions Download page for details.

If your extension does not appear in this list, or there’s something wrong with an entry on this page, please send me an email, giving a URL for your website if applicable.

Config Files

- Extensions build config files (R3.13)
- Extensions build config files (R3.14)

Standalone CA Clients

- ADT, Array Display Tool
- ALH, Alarm Handler
- AR, Data Archiver (the original, deprecated)
- BURT, Backup and Restore Tool
- CAEX, Channel Access Examples
- CASE, Host-based Save/Restore
- CAU, Channel Access Utility

Argonne National Laboratory
Command-Line Tools

- There used to be several versions of these tools
- We will discuss the ones that now come with EPICS Base
- The tools we will cover are:
  - caget
    - Gets the value of one or more process variables
  - caput
    - Sets the value of one process variable
  - camonitor
    - Monitors the value changes of one or more process variables
  - cainfo
    - Gets information about one or more process variables
- All accept –h to display usage and options
Caget Example

- Get the values of two process variables
  ```
  caget S35DCCT:currentCC S:SRlifeTimeHrsCC
  ```
- Returns
  ```
  S35DCCT:currentCC    102.037
  S:SRlifeTimeHrsCC    7.46514
  ```
Caput Example

- Set the value of a process variable
  \[ \text{caput Xorbit:S1A:H1:CurrentAO 1.2} \]

- Returns
  \[
  \begin{align*}
  \text{Old} &: \text{Xorbit:S1A:H1:CurrentAO} & 0 \\
  \text{New} &: \text{Xorbit:S1A:H1:CurrentAO} & 1.2
  \end{align*}
  \]
Camonitor Example

- Monitor two process variables
  
camonitor evans:calc evans:bo01

- Returns
  
<table>
<thead>
<tr>
<th>Process</th>
<th>Timestamp</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:04.623245</td>
<td>1</td>
</tr>
<tr>
<td>evans:bo01</td>
<td>2004-08-05 17:23:04.623245</td>
<td>On</td>
</tr>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:05.123245</td>
<td>2</td>
</tr>
<tr>
<td>evans:bo01</td>
<td>2004-08-05 17:23:05.123245</td>
<td>Off</td>
</tr>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:05.623245</td>
<td>3</td>
</tr>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:06.123245</td>
<td>4</td>
</tr>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:06.623233</td>
<td>5</td>
</tr>
<tr>
<td>evans:calc</td>
<td>2004-08-05 17:23:07.123183</td>
<td>6</td>
</tr>
</tbody>
</table>

- Use Ctrl-C to stop monitoring
Cainfo Example

- Get information about a process variable
  
cainfo S35DCCT:currentCC

- Returns
  
  State: connected
  Host: ctlapps4l188:5064
  Access: read, no write
  Data type: DBR_DOUBLE (native: DBF_DOUBLE)
  Element count: 1

- Some additional information can be found using Probe
**Probe**

- Simple way to get information about a single process variable
- Combines the features of caget, caput, camonitor, and cainfo in a graphical interface
- Very useful in diagnosing problems
Probe Demo

Macromedia Flash Demonstration of Probe
StripTool

- Plots process variables in real time on a strip chart
- Widely used
StripTool Demo

Macromedia Flash Demonstration of StripTool
MEDM

- Stands for Motif Editor and Display Manager
- The principal human interface to the APS control system
- Used worldwide at many facilities
- Creates and runs control screens
MEDM Screens

And thousands of others
MEDM

- MEDM is very reliable at both design and run-time
- However it is very hard to extend
  - Not written in Object-Oriented style
  - Maintainer has recently moved to other responsibilities
- APS will only fix major bugs found in MEDM now
- Not a good choice for a new control system
  - Use EDM instead (upcoming lecture)
ALH

- Stands for Alarm Handler
- Important GUI application in the APS Control Room
- Brings alarms to the operators’ attention
  - It dings and flashes
- Can be configured to require the operator to acknowledge alarms
- Provides a hierarchical display
  - Allows managing alarms in overview or in detail
- Provides guidance for handling specific alarms
- Logs alarms and displays alarm history
ALH

- ALH will be covered more fully in another lecture