Virtual Accelerator Setup Instruction (Linux)

C. Chu
08/03/2006
Updated 04/20/2008

1. Get, build and install EPICS BASE from APS (http://www.aps.anl.gov/epics/).
2. Download (from XXX), put Portable Channel Access Server (PCAS) on local disk and set environment variable $PCAS_HOME to this directory.
3. In the $PCAS_HOME directory, edit the *Makefile* file to replace the "EPICS_BASE" path with yours (set in step 1).
4. Type "make" to build PCAS.
5. Download XAL from http://sns.gov/APGroup/appProg/xal/xaldev.tgz. Un-zip and un-tar the xaldev.tgz.
6. Set $XAL_HOME environment variable to the path you generated in step 5.
7. Follow the README file in the XAL package to build the framework and apps.
8. (recommended but not required) Edit the VirtualAccelerator script to reflect "EPICS_CA_ADDR_LIST"
9. Save the "JCALibrary.properties" file in your $HOME/.JCALibrary directory and make necessary changes (e.g. file paths).
10. When first time running the VA, you will be asked to set a path to the PCAS launching script. Please point it to $XAL_HOME/gov/sns/apps/virtualaccelerator
11. You can then prepare an accelerator file for your own institute (see $XAL_HOME/xal_xmls/ for examples) or use existing SNS files (can set as default accelerator or user defined).
12. Virtual accelerator may run into "out-of-memory" problem if you use a longer beam-line and not allocate enough memory for Java VM.

Note: Because in step 10 the path to the local $PCAS is set in Java Preferences, it is not working with Sun Java 1.6, i.e. you have to set the $PCAS path every time you run VA.

Sample JCALibrary.properties file:

```
# define the location of the epics shared libraries and caRepeater executable
gov.aps.jca.jni.epics.linux-x86.library.path=/home/chu/epics/base-3.14.8.2/lib/linux-x86
gov.aps.jca.jni.epics.linux-x86.caRepeater.path=/home/chu/epics/base-3.14.8.2/bin/linux-
x86

# Overwrite the default addr_list to point to a specific ca server
gov.aps.jca.jni.JNIContext.addr_list=localhost
gov.aps.jca.jni.JNIContext.repeater_port=5065
gov.aps.jca.jni.JNIContext.server_port=5064

# overwrite the default auto_addr_list so that network interfaces will not be
automatically added to the addr_list.
gov.aps.jca.jni.JNIContext.auto_addr_list=false

# toto is going to use very large array, so we want to increase the maximum array size.
gov.aps.jca.jni.JNIContext.max_array_size=32768

# define default values only for JNI_THREAD_SAFE context
gov.aps.jca.jni.ThreadSafeContext.event_dispatcher=gov.aps.jca.event.QueuedEventDispatche
r
gov.aps.jca.jni.ThreadSafeContext.priority=1

# for CAJ
com.cosylab.epics.caj.CAJContext.addr_list=localhost
com.cosylab.epics.caj.CAJContext.auto_addr_list=false
com.cosylab.epics.caj.CAJContext.max_array_size=32768
com.cosylab.epics.caj.impl.reactor.lf.LeaderFollowersThreadPool.thread_pool_size=20
com.cosylab.epics.caj.CAJContext.beacon_period=15
```