XAL Workshop Guide

Thomas Pelaia II

XAL Workshop 2010

May 3 - 5, 2010

Managed by UT-Battelle
for the Department of Energy

OAK RIDGE National Laboratory

XAL Workshop

- Welcome
- Goals
- Agenda
- Quick Start Guide

Welcome

- First XAL Workshop
- Six different labs represented

Workshop Goals

- Guide to XAL
- Review Current Development Efforts
- Develop Active Collaboration

Agenda

- Quick Start Guide to XAL
- XAL Data Representation
- Application Development Guide
- Featured Presentations
- Collaboration Discussion

Quick Start Guide

- XAL Overview
- Getting the XAL source code
- Building XAL at the command line
- Running XAL applications and scripts
- XAL Common Look and Feel
- Tour of XAL Applications
- More Details at:
  http://www.ornl.gov/~t6p/Main/XALQuickStart.html

XAL Overview

- Developed for SNS
- Application Framework
- Accelerator Physics Modeling
- Channel Access Client Support

- http://www.ornl.gov/~t6p/Main/XAL.html

Minimal Requirements

- Java J2SE 6 with JDK
- Ant 1.7
- Subversion 1.6
- Jython 2.1
- JRuby 1.4

Getting the Source Code

```
svn co https://YourID@xaldev.svn.sourceforge.net/svnroot/xaldev/trunk xal
```

Get Missing Third Party Jars

File	Version	Location
Jama-1.0.2.jar	1.0.2	http://math.nist.gov/javanumerics/jama/
jcommon.jar	1.0.10	http://www.jfree.org/
jfreechart.jar	1.0.10	http://www.jfree.org/
JSciCore.jar	0.944	http://jsci.sourceforge.net/
junit-4.1.jar	4.1	http://junit.sourceforge.net/
ojdbc6.jar	11g release1	http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

Configure your Environment

Insert into ~/.bashrc or ~/.profile:

- **export XAL_HOME=/my/path/to/xal**
- **export CLASSPATH=${XAL_HOME}/build/jar/xal.jar:${XAL_HOME}/build/jar/ext.jar**

Command Line Builds

- ant jar-ext
- ant
- ant build-services
- ant build-apps
- ant deploy-scripts

Running Applications and Scripts (Command Line)

- java –jar ${XAL_HOME}/build/jar/apps/launcher.jar
- jruby ${XAL_HOME}/build/scripts/orbit_viewer/orbit-viewer.rb
- jython ${XAL_HOME}/build/scripts/scl_waveform_capture/capture_scl_waveforms.py

Running Applications and Scripts (Launcher)

- Run the Launcher
- Configuration (First Time Only)
  - Edit the Rules to specify the Java, JRuby and Jython Commands if necessary
  - Save the Launcher document
  - Make the Launcher document the default
- Double click (or press Run button) to launch the selected application or script

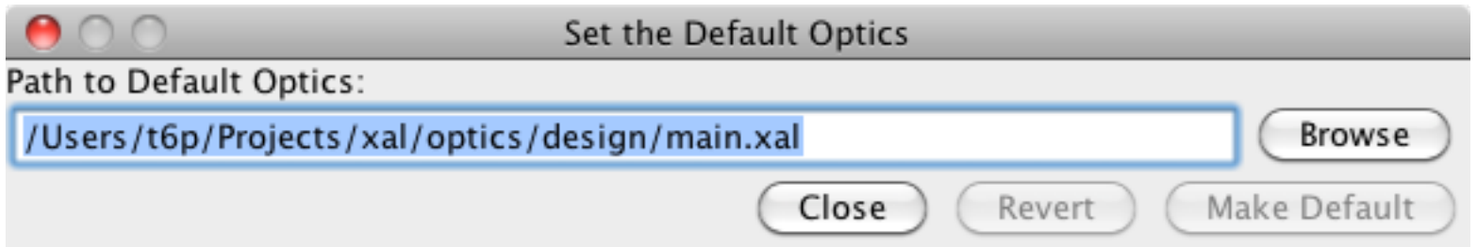
Database Configuration

- Create file with “dbconfig” extension
- Run DB Browser and set URL in Preferences

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<dbconfig date="Thu Jul 23 10:46:03 DST 2009" version="1.0.0">
   <adaptors default="oracle">
        <adaptor name="oracle" class="gov.sns.tools.database.OracleDatabaseAdaptor" />
   </adaptors>
   <servers default="production">
        <server name="production" adaptor="oracle" url="jdbc:oracle:thin:@snsdb1.sns.ornl.gov:1521:prod" />
   </servers>
   <accounts default="reports">
        <account name="reports" user="XAL_2010" password="sns" />
        <account name="score" user="XAL_2010" password="sns" />
        <account name="personal" user="YourIDHere" />
   </accounts>
</dbconfig>
```

Default Accelerator Optics

- Run Optics Switcher
- Navigate to the main optics file and make it the default



Channel Access Configuration

- Download the JCA Library Properties file from:
  http://www.ornl.gov/~t6p/Main/XALQuickStart_files/JCALibrary.properties.zip
- Place JCALibrary.properties at:
  ~/.JCALibrary/JCALibrary.properties
- Modify com.cosylab.epics.caj.CAJContext.addr_list property value:
  - SNS CA: ics-srv-cagate1.sns.ornl.gov
  - Local CA: 127.0.0.1
- Please don't hit the SNS Channel Access Gateway after the workshop

JCALibrary.properties

```
# define the location of the epics shared libraries and caRepeater executable
gov.aps.jca.jni.epics.darwin-x86.library.path = /Library/EPICS/Base/lib/darwin-x86
gov.aps.jca.jni.epics.darwin-x86.caRepeater.path = /Library/EPICS/Base/bin/darwin-x86
gov.aps.jca.jni.epics.linux-x86.library.path = /url/local/epics/base/R3.14.4/lib/linux-x86
gov.aps.jca.jni.epics.linux-x86.caRepeater.path = /usr/local/epics/base/R3.14.4/bin/linux-x86

# define default values for both JNI_THREAD_SAFE and JNI_SINGLE_THREADED contexts.
gov.aps.jca.jni.JNIContext.preemptive_callback = true

# Channel Access address list for JNI context
gov.aps.jca.jni.JNIContext.addr_list = ics-srv-cagate1.sns.ornl.gov
#gov.aps.jca.jni.JNIContext.addr_list = 127.0.0.1

gov.aps.jca.jni.JNIContext.auto_addr_list = false
gov.aps.jca.jni.JNIContext.connection_timeout = 30.0
gov.aps.jca.jni.JNIContext.beacon_period = 15.0
gov.aps.jca.jni.JNIContext.repeater_port = 5065
gov.aps.jca.jni.JNIContext.server_port = 5064
gov.aps.jca.jni.JNIContext.max_array_bytes = 16384

# define default values only for JNI_SINGLE_THREADED context
gov.aps.jca.jni.SingleThreadedContext.event_dispatcher = gov.aps.jca.event.DirectEventDispatcher

# define default values only for JNI_THREAD_SAFE context
gov.aps.jca.jni.ThreadSafeContext.event_dispatcher = gov.aps.jca.event.QueuedEventDispatcher
gov.aps.jca.jni.ThreadSafeContext.priority = 5

# define default values for QueuedEventDispatcher components
gov.aps.jca.event.QueuedEventDispatcher = 5

# Channel Access address list for CAJ context
com.cosylab.epics.caj.CAJContext.addr_list = ics-srv-cagate1.sns.ornl.gov
#com.cosylab.epics.caj.CAJContext.addr_list = 127.0.0.1

com.cosylab.epics.caj.CAJContext.auto_addr_list = false
com.cosylab.epics.caj.impl.reactor.lf.LeaderFollowersThreadPool.thread_pool_size = 30
```

Running the Virtual Accelerator

- Configure JCA Library Properties for local Channel Access
  - com.cosylab.epics.caj.CAJContext.addr_list = 127.0.0.1
- Launch the Virtual Accelerator application and select an accelerator and a sequence from the Accelerator menu
- Hit the Start VA button to begin updating the process variables

XAL Common Look and Feel

- Common Menus
- Copy, Cut and Paste
- Console and Persistent Logging
- Online Help
- Standard Java Icons + Custom XAL Icons for XAL specific buttons and menu items
- Application Demo

Tour of a few XAL Applications

- Launcher
- Virtual Accelerator
- Online Model
- Scan 1D/2D
- Orbit Correction
- Score
- MTV
- Knobs

XAL Services

- PV Logger
  - Logs groups of PVs to database on demand and on schedule
- MPS Trips
  - Orders and logs First Faults
- Trip Monitor
  - Generic trip monitoring and logging

XAL Machine Description

- Accelerator
  - sequences and power supplies
- Accelerator Sequences
  - subsequences and nodes (no drifts)
- Accelerator Nodes (Devices)
  - ID, type, position, length, status, channel suite, type dependent data (e.g. power supply, magnetic length, ...)
  - control devices (dipole, quadrupole, sextupole, RF Cavities)
  - diagnostics (position monitor, loss monitor, current monitor)
- Power Supplies

Database Machine Representation

- Integrated in the SNS Global Database
- Design Machine
- http://snsapp1.sns.ornl.gov/SNS_Data_Model/index.htm
  - SNS Global Database -> XAL_OPTICS -> Main Model
- Jeff Patton is our Database Administrator

XAL Optics File Structure

- design under ${XAL_HOME}/xal_xmls
- main.xal
  - main optics file
  - Reference to device mapping, optics source, hardware status, timing source and model parameters files
- sns.impl
  - associates device type (and soft type) to device classes
- sns.xdxf
  - accelerator device definition
  - generated from database using db2xal

XAL Optics File Structure (continued)

- hardware_status.xdxf
  - Overrides optics file
  - Additions, Exclusions and Parameter changes
- timing_pvs.tim
  - Timing specific channels
- model.params
  - Beam initialization parameters
  - Probe initialization

Application Framework

- Rapid Application Development
  - Bricks GUI Builder
  - Document based Application Model
  - Leverages Design Patterns
- Common Look and Feel
  - Modeled after Netscape Navigator / Mozilla Firefox
  - Consistent platform accelerators
- Familiar Modern Features
- Stable, Careful Development

Common Design Patterns in XAL

- Adaptor
- Factory Methods
- Key-Value-Coding
- Model-View-Controller
- Notification Center
- Proxy
- Singleton

Primary Application Components

- Application Adaptor
- Document(s)
- Document Window (may be dynamic)
- Resources
  - HTML based Online Help
  - Info properties
  - Menu Definition (optional)
  - GUI Bricks file (dynamic views)

Accelerator Based Applications

- Configuration of Default Accelerator
- Loading of Default Accelerator
- Accelerator Menu
- Code Demonstration

Demo: Creating an Application

- Start with Template
- Edit the project name in the Ant build file
- Edit “About” Info
- Modify package reference in Application adaptor and Document
- Specify Application Name
- Update the Help
- Print list of Electromagnets
- Add Custom Menu

Bricks

- Encourages Model-View-Controller
- Application for building a user interface
  - simplifies or eliminates tedious code
- Inspired by NextStep's Interface Builder
  - not as sophisticated and no connections to controller
- Stores views in an XML file
- Window references are loaded from the XML file and views are instantiated
- Views can be retrieved by ID
- Code Demonstration

Demo: Defining the User Interface

- Add a Vertical Container
- Add a Filter Field and fix the vertical height
  - Set the tooltip
- Add a Scroll Pane and Table for Magnets
- Add a Control Box with a Run button
  - Set the Run button's Icon to media:Play24.gif
  - Set the tooltip for the button

Key-Value Table Model
gov.sns.tools.swing.KeyValueTableModel

- Rapidly create record based table models
- Support for viewing and editing
- Handles nested objects
- Intelligent defaults (e.g. column naming)
- Highly Configurable
- Filtering with KeyValueFilteredTableModel
- Code Demonstration

Demo: Create a Table Model

- Instantiate a KeyValueFilteredTableModel
- Configure the key paths
- Configure the column names
- Configure column editing
- Configure the filter component
- Configure the matching key paths

Data Adaptor

- Archive and Unarchive objects to Persistent Storage
- Abstracts procedure from implementation of storage
  - Group of key-value pairs
  - Nested Lists of tagged groups
  - Accessors and Setters for primitive types and tagged groups
- XML Data Adaptor implementation
- Code Demonstration

Demo: Implement Save and Open

- Provide file extension
- Monitor Change Handling
- Add Data Label
- Implement DataListener
  - dataLabel, update, write
- Edit the Primary Constructor
- Edit saveDocumentAs

Channel Access gov.sns.ca

- Wrapper of JCA API
  - defense against JCA API modifications
  - value transformations
  - factory singleton channels for PV and Transform
- Works with either JCA/JNI or JCA/CAJ
- Blocking and Non-Blocking Connect, Get and Put
- Monitors
- Scalar and Array types
- Code Demonstration

Demo: Channel Access

- Blocking Connect and Get
- Non-Blocking Connect and Monitor

Plot Framework

- Presentation by Andrei Shishlo

Scanning Framework

- Presentation by Andrei Shishlo

Online Model

- Presentation by Chris Allen

Featured Presentations

- Future Upgrades for the Online Model
- High Level Applications for Spiral 2
- Use of XAL at Spiral 2
- Survey of Current XAL Challenges

Collaboration

- Discussion
- Coding Standards and Quality Control
- Source Code Management
- Proposal
- Assignments

Thank You!