

Tour of XAL

Thomas Pelaia II, Ph.D.

XAL Lecture Series

Lesson 2

July 3, 2008



XAL Lecture Series

1. Java Primer
2. **Tour of XAL**
3. Building Java Applications in XAL
4. Ruby Primer
5. Scripting Applications in XAL
6. Channel Access in XAL
7. XAL Accelerator Hierarchy

Tour of XAL

- **XAL Overview**
- **Building XAL**
- **Application Development**
- **Scripting XAL**
- **Services**
- **Tools**
- **Accelerator Physics**
- **Channel Access**
- **Resources**

What is XAL?

- **Development environment for creating pure Java accelerator physics applications, scripts and services**
 - Control room applications
 - Analysis applications
- **Application framework**
- **Toolbox of Java packages**
- **Ant based build system (independent of IDE)**
- **Developed for the Spallation Neutron Source (SNS) commissioning and operations**
- **Evolving collaboration among several labs**

Collaboration

- **Source Forge Project: [xaldev](http://sourceforge.net/projects/xaldev)**
 - <http://sourceforge.net/projects/xaldev>
 - Source code managed using Subversion
- **Open Source development**
- **Dozens of developers among several sites**
 - SNS, SLAC, BNL, JPARC, GANIL and others
- **Contact us to participate**

Development Requirements

- **Java JDK J2SE 5**
 - *XAL is not compatible with J2SE 6 on Linux*
- **Ant 1.7**
- **Your favorite editor or IDE**
- **Subversion client (svn) for source code control**
- **JRuby and Jython for scripting**

Runtime Requirements

- **Java J2SE 5**
 - *XAL is not compatible with J2SE 6 on Linux*
- **JRuby and Jython for scripting**
- **EPICS Channel Access Client libraries (including JCA/JNI library)**
 - needed if you want to use JCA/JNI for channel access
 - required on SNS controls network
- **Portable Channel Access Server**
 - needed for virtual accelerator
- **JDBC compatible database and driver**

Documentation

- **Most XAL classes provide Javadoc documentation**
- **A few XAL packages provide Javadoc documentation**
- **An XAL Setup guide and a README file are provided at the top level of the source tree**
- **The external jar files are documented in the README file of the external jar directory**

Repository

- **Subversion source code management**
- **Contains XAL source code**
- **Contains third party packages**
 - **Packages are stored in jar files under the ext_jars directory**
 - **Please respect the licenses of the third party packages**
 - **Use of jcchart libraries is deprecated and the license restricts development to the physics 1 server only**

Installation Summary

- Confirm the **development requirements**
- Checkout XAL from the SNS subversion repository
- Define *XAL_HOME* environment variable to point to the root of the XAL project
- Optionally install JCA/JNI library
- Place *JCALibrary.properties* file in *~/.JCALibrary* directory and configure the properties file
- Change directory to the root of the XAL project for executing Ant builds

Build Phases (using Ant)

- `ant jar-ext`
 - Merges external jar files into a single external jar file
- `ant`
 - Builds the core XAL packages
- `ant build-services`
 - Builds all XAL services
- `ant build-apps`
 - Builds all XAL applications

XAL Package Hierarchy

Item	Package (Location)
Application Framework	gov.sns.application
General Tools	gov.sns.tools
Channel Access	gov.sns.ca
Accelerator Hierarchy	gov.sns.xal.smf
Online Model	gov.sns.xal.model
Applications	gov.sns.apps
Scripts	gov/sns/scripts
Services	gov.sns.services

Constructing Applications

- **Application Framework**
 - foundation for modern XAL applications
- **Bricks**
 - graphical user interface construction
 - runtime for generating user interfaces
- **Options for applications**
 - **Mix Bricks with Application Framework**
 - recommended for new applications
 - **Use Bricks alone**
 - ideal for scripts with user interfaces
 - **Use application framework alone**
 - common among older applications

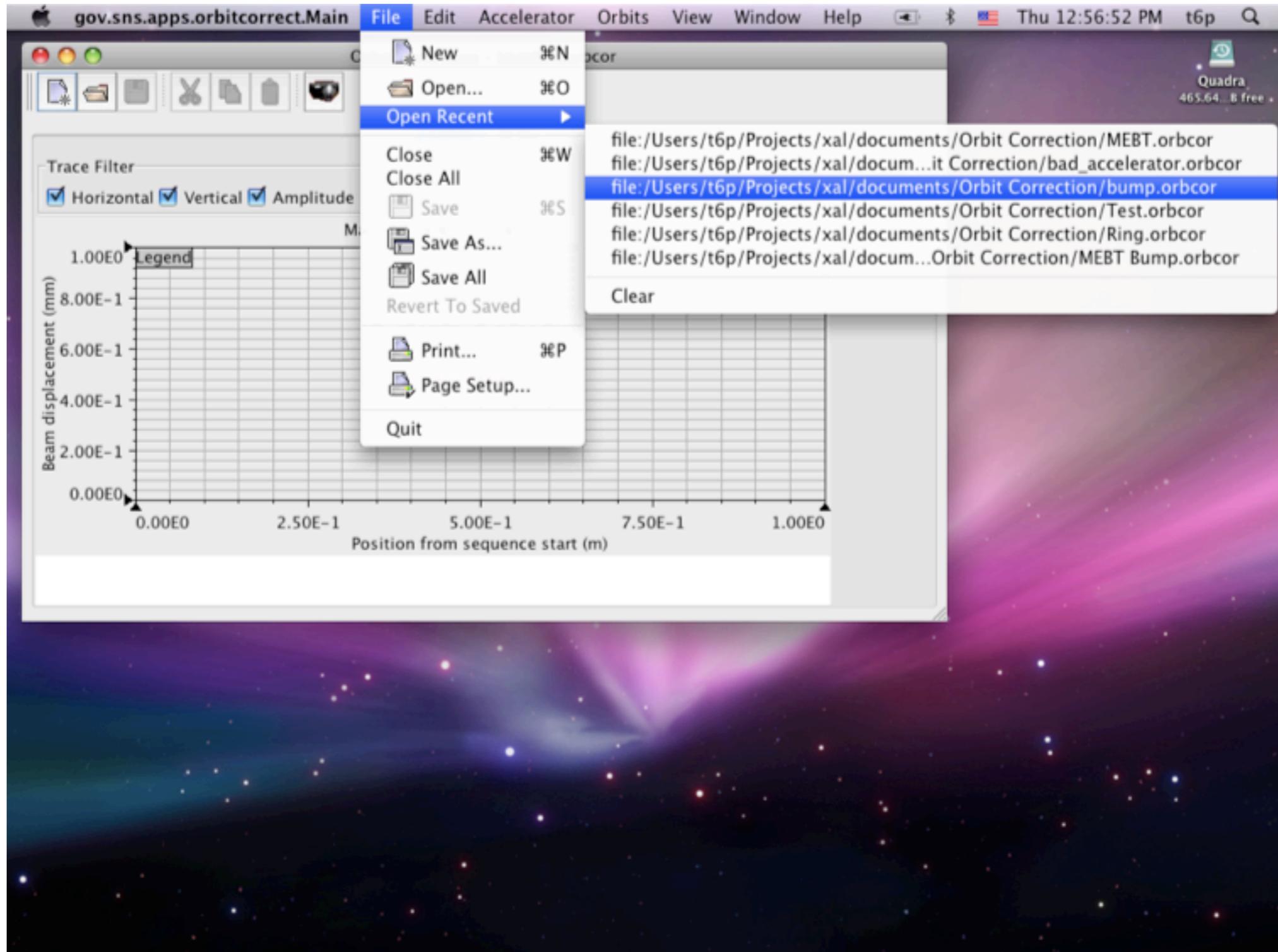
Application Framework

- **Foundation for document based applications**
- **Consistent look and feel across applications**
 - standard, familiar menu items
 - free automatic behaviors
- **Rapid application development**
- **Broadly customizable**
- **Integrated online application help**

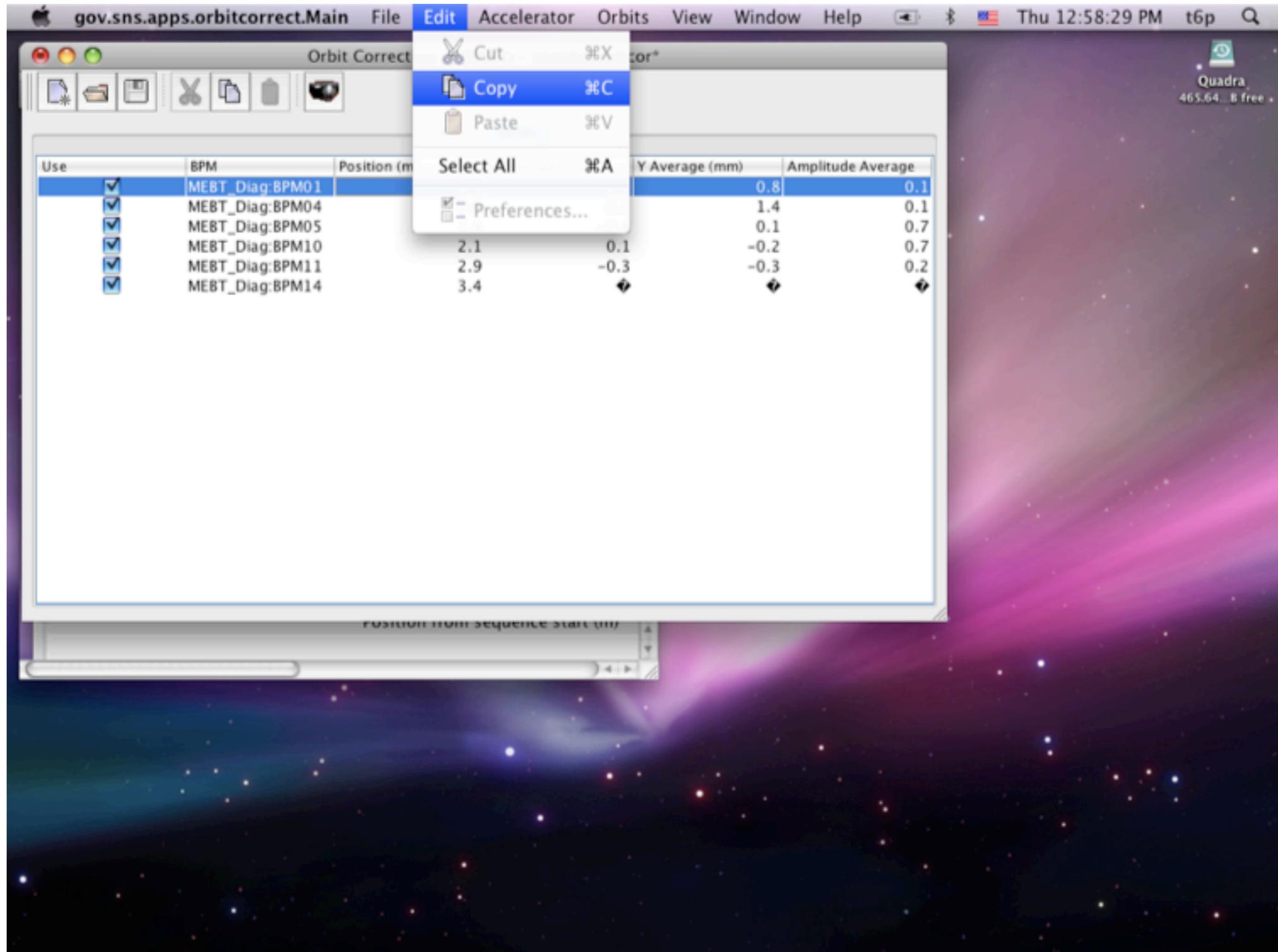
Automatic Application Behaviors

- **Windows marked for document modification**
- **List of recently opened documents maintained**
- **Warning dialog when closing modified documents**
- **Copy, Cut and Paste on transferable components**
- **Standard output and standard error captured and displayed in console view**
- **Automatically enables/disables standard menu and toolbar items based on state**
- **Standard icons and platform appropriate shortcuts for menu and toolbar items**

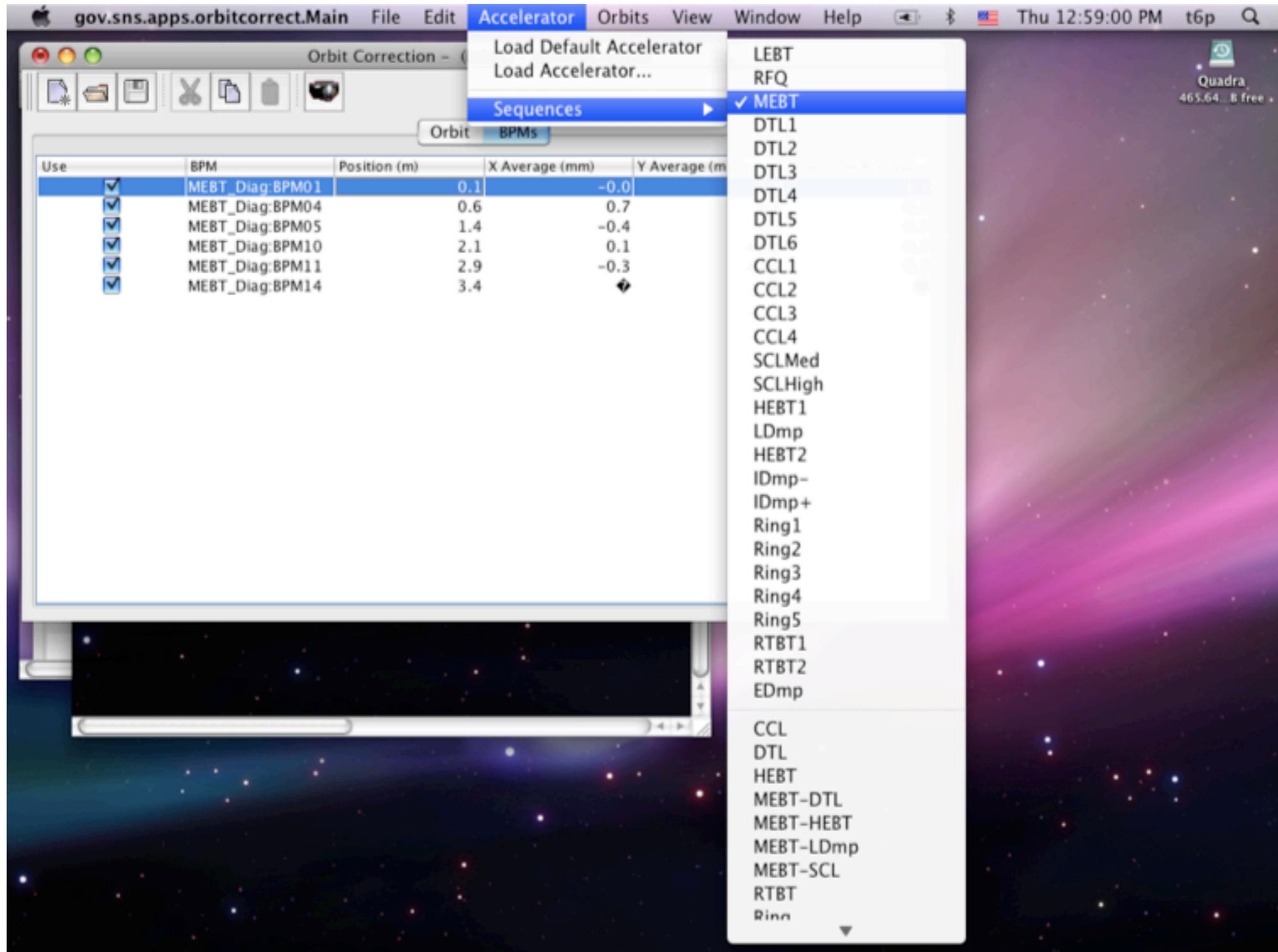
Standard Menu Items (plus custom Orbits menu)



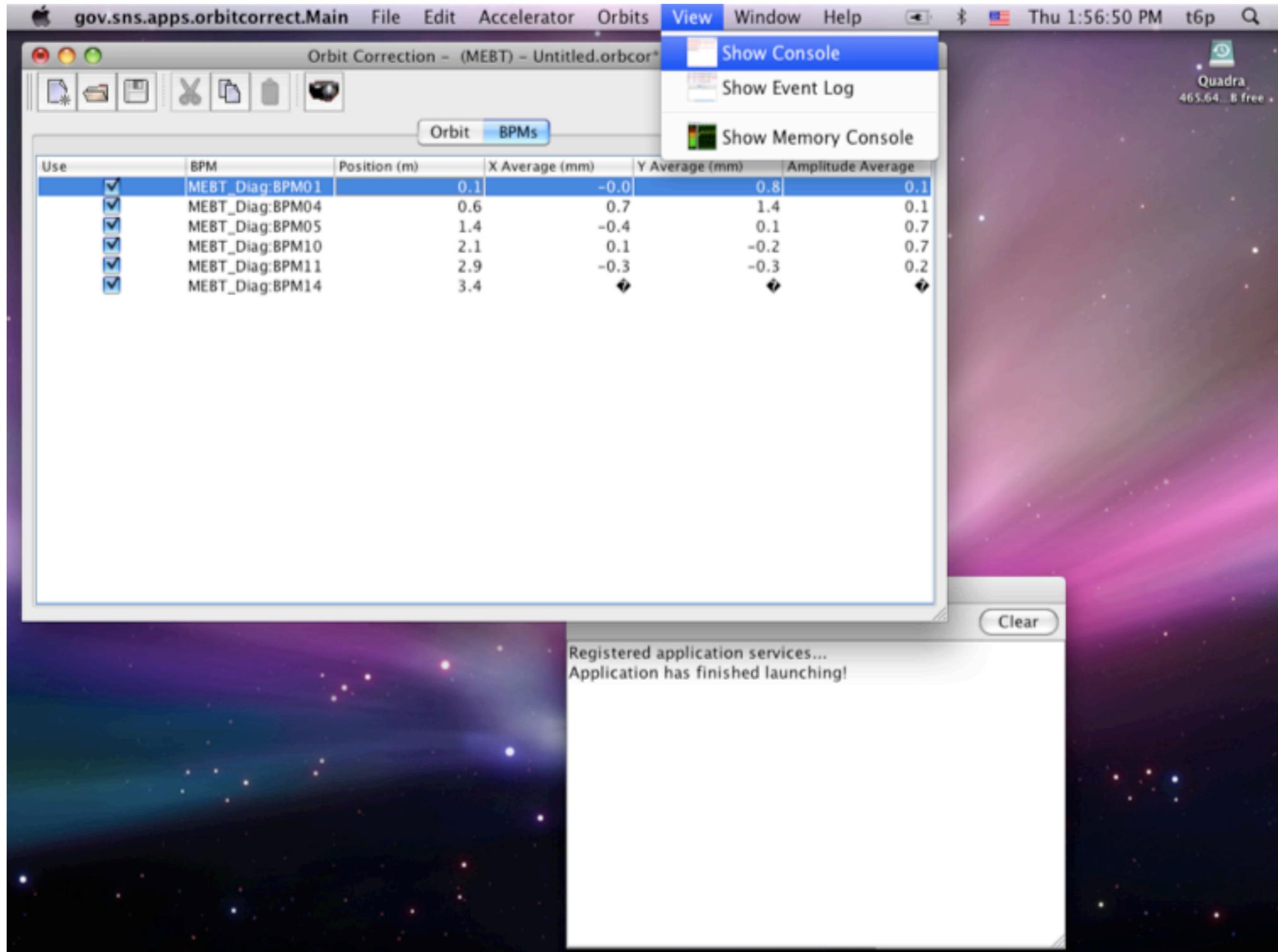
Standard Menu Items (plus custom Orbits menu)



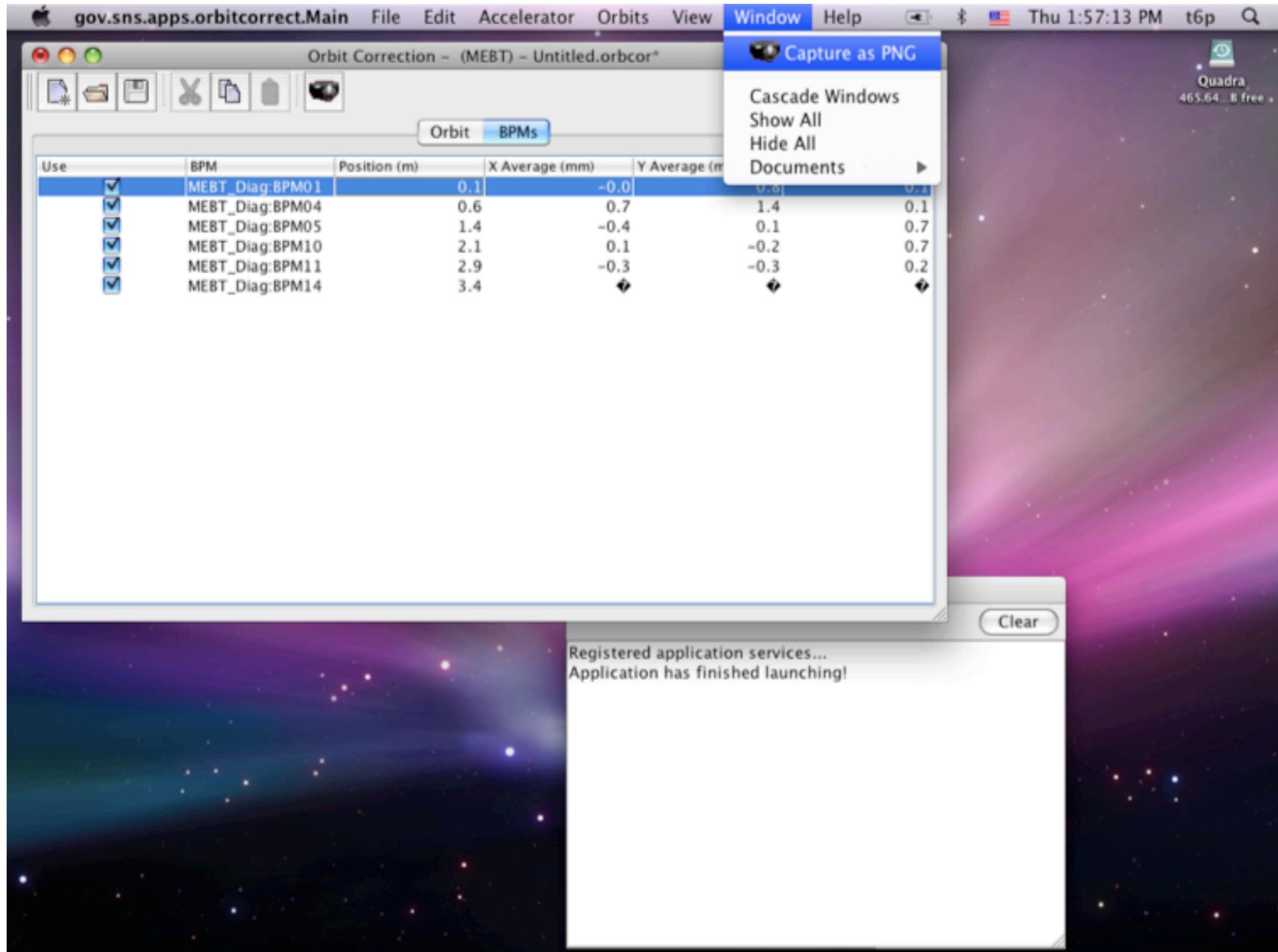
Standard Menu Items (plus custom Orbits menu)



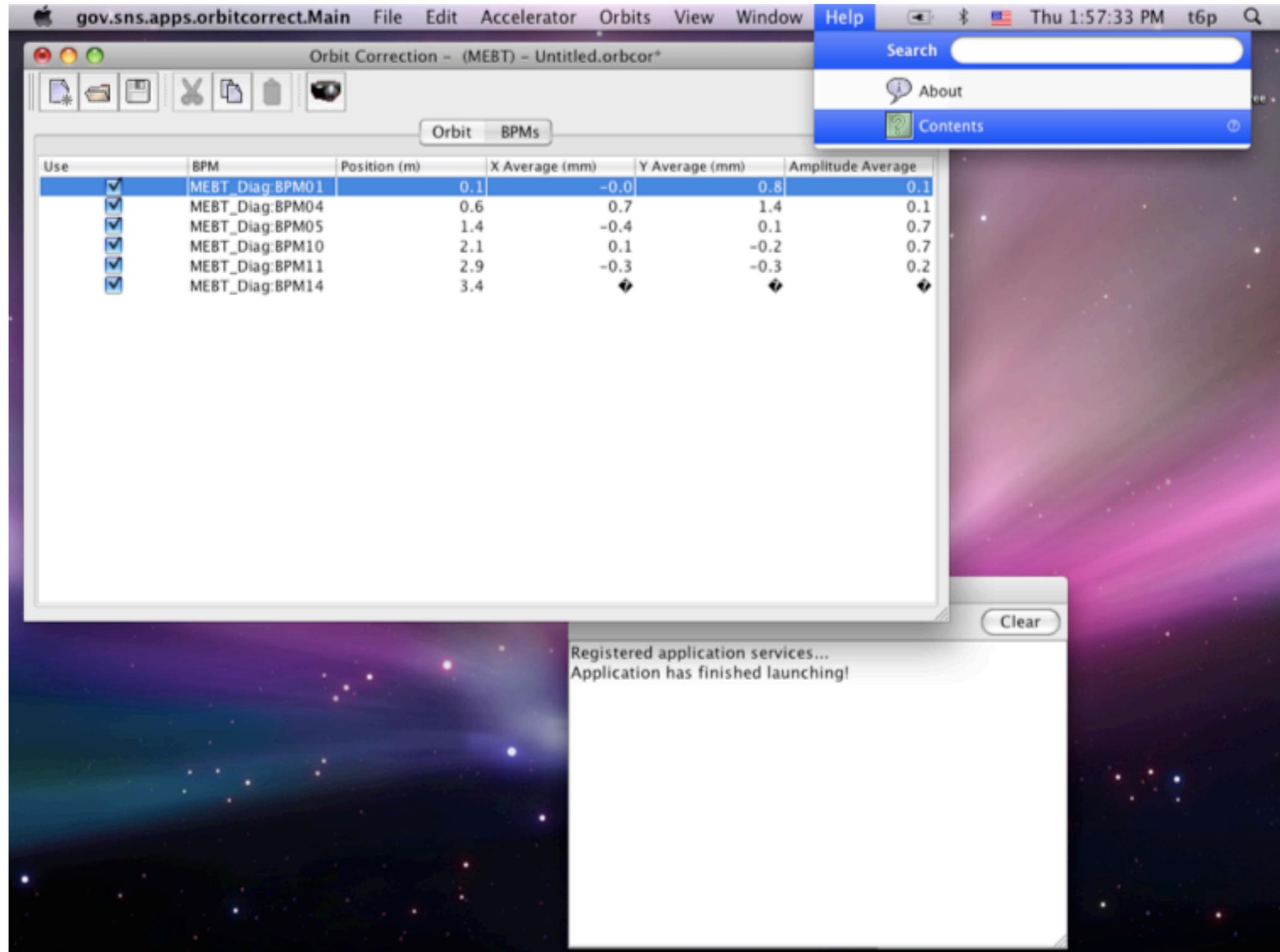
Standard Menu Items (plus custom Orbits menu)



Standard Menu Items (plus custom Orbits menu)



Standard Menu Items (plus custom Orbits menu)



Core Application Components

Base Class	Role of Custom Subclass
ApplicationAdaptor	Provide application specific information and hooks
XalDocument	Document specific attributes, data and hooks
XalWindow	View for the document

Resource	Role
menundef.properties	Defines menus, menu items, toolbar and toolbar buttons
About.properties	Basic application information
Help.html	Online documentation

Bricks

- **Graphical User Interface construction application**
- **Stores views within an XML file**
- **Runtime generates swing views dynamically**
- **You provide the controller code**
 - **Inherently Model-View-Controller compliant**
 - **More flexible than traditional swing hand coding**
- **Compatible with your favorite editor or IDE**
- **Allows for compile free development with scripting languages (e.g. JRuby or Jython)**
- **Integrates with the Application Framework**

Bricks-based Application Framework Components

Base Class	Role of Custom Subclass
ApplicationAdaptor	Provide application specific information and hooks
XalDocument	Document specific attributes, data and hooks

Resource	Role
gui.bricks	XML file with definition of windows and dialogs
menundef.properties	Defines menus, menu items, toolbar and toolbar buttons
About.properties	Basic application information
Help.html	Online documentation

XAL Applications

(sample from over four dozen)

Package	Description
bricks	Build user interfaces
energymanager	Generate new optics variations
knobs	Control groups of PVs together
lossviewer2	Monitor losses from detectors
mpx	Run online model simulations
orbitcorrect	Measure and correct the orbit
Scan2D	Scan PVs in two dimensions
score	Save and restore machine state
slacs	Measure and set RF phases
virtualaccelerator	Simulate a live machine

Application Packaging

- Applications are packaged with relative references to the *xal.jar* and *ext.jar* files
- They are placed in the *build/jar/apps* directory under the XAL root
- To run an application named “someapp” type:

```
java -jar ${XAL_HOME}/build/apps/someapp.jar
```

Scripting

- **JRuby and Jython**
 - JRuby is much more powerful than Jython
 - Access to most of XAL and Java API
- **Rapid development without compiling**
- **Great for prototyping and simple applications**
- **Integrates well with Bricks**

XAL Scripts

(sample from numerous scripts)

Script	Description
ccl_orbit_correction	Correct CCL orbit based on alignment
fit_injection_kickers	Measure and correct injection kicker closure
make_ring_bump	Generate bumps within the ring
ring-tune-monitor	Monitor and display the horizontal and vertical spectra
scan1d_scatter_plot	Plot Scan 1D data as a scatter plot using selected PV pairs
waveform-monitor	Monitor and display arbitrary waveforms and their spectra

Service Infrastructure

- **Services are headless executables (no GUI)**
- **Run continuously in the background**
- **Typically on one instance runs**
 - **Sometimes more than one instance can run across servers**
- **Provides remote communication with user interfaces**
 - **XAL defines an API for communication between client and service**
 - **Uses open standards**
 - **XML-RPC for communication**
 - **mDNS for dynamic service discovery**

XAL Services

Service	Package	Description
MPS First Faults	mpstool	Monitors and logs Machine Protection System (MPS) Faults
PV Logger	pvloggertool	Monitors and logs groups of Process Variables. Both continuous and on-demand logging.
Trip Monitor	tripmonitor	Monitors and logs arbitrary trip conditions

XAL Tools

(sample from among several)

Tool	Description
Solver	Optimization using a collection of algorithms
Plot	Collection of scientific plotting tools: scatter, line, etc.
Formula Interpreter	Extensible interpreter of formulae
Statistics	Common statistical operations: mean, standard deviation, etc.
Math	Complex, Fourier Transform, Trigonometry
Data Adaptor	Abstracts archiving and unarchiving object trees

Accelerator Hierarchy

- **Stored in XML file generated from database**
- **Accelerator**
 - top level container of sequences and power supplies
- **Accelerator Sequence**
 - container of sequential nodes
- **Accelerator Node**
 - represents a physical accelerator device
 - magnets (Dipole, Quadrupole, Sextupole, etc.)
 - diagnostics (Beam Position Monitor, Profile Monitor, etc.)
 - provides channels for monitoring and control
- **Magnet Power Supply**

Online Model

- **gov.sns.xal.model**
- **simulates charged particle dynamics through specified accelerator sequences**
- **supports both linear sequences and rings**
- **Calculates Twiss parameters, energy and orbit distortions**
- **Six dimensional phase space propagation**
- **Optics field input can be from design optics, live machine or custom values**
- ***model.params* file provides initialization parameters**

Online Model Classes

Class	Description
ProbeFactory	Instantiates a probe from model parameters
Probe	Holds transient state as it propagates through a lattice
Scenario	Generates a lattice and manages the run configuration
Trajectory	Container of result states from a scenario run
ProbeState	Result information (Twiss, energy, ...) at an element

Running the Online Model

- **Generate a probe from the probe factory**
- **Instantiate a scenario for an accelerator sequence**
- **Set the scenario's probe**
- **Set the scenario synchronization mode**
 - live, design, live with RF from design
- **Apply custom fields if any**
- **Synchronize the scenario with the sources**
- **Run the model**
- **Get the states from the resulting trajectory**

XAL Channel Access

- **gov.sns.ca**
 - package abstracts channel access
 - provides some insulation from API changes to underlying access layer
 - allows for both blocking and concurrent calls
 - supports connection, get and put operations (with and without callbacks) and monitors
- **gov.sns.jca**
 - implements XAL Channel support using JCA
 - wraps Java Channel Access (JCA) calls
 - can switch between JCA/JNI and JCA/CAJ
 - JCA/JNI is recommended for production due to known, serious issues with JCA/CAJ

XAL Channel Access Classes

Class	Description
ChannelFactory	Creates channels on demand and caches them
Channel	Manages state of PV and provides access operations
Monitor	Manages state of a monitor for a PV
ChannelRecord	Channel callback result with value and conversions
ChannelStatusRecord	Adds status and severity to channel record
ChannelTimeRecord	Adds event time stamp to channel status record

Resources

- **XAL Project**

- <http://sourceforge.net/projects/xaldev>
- <http://www.ornl.gov/~t6p/Main/XAL.html>

- **EPICS**

- <http://www.aps.anl.gov/epics/>

- **JCA/JNI**

- <http://jca.cosylab.com/downloads.html>

- **JCA/CAJ**

- <http://caj.cosylab.com/>