



*... for a brighter future*

# *Eclipse as an Application Workbench*

*Kenneth Evans, Jr.*

*Presented at the Software Collaboration Meeting*

*January 24, 2007*

*Oak Ridge National Laboratory, Oak Ridge, TN*



U.S. Department  
of Energy

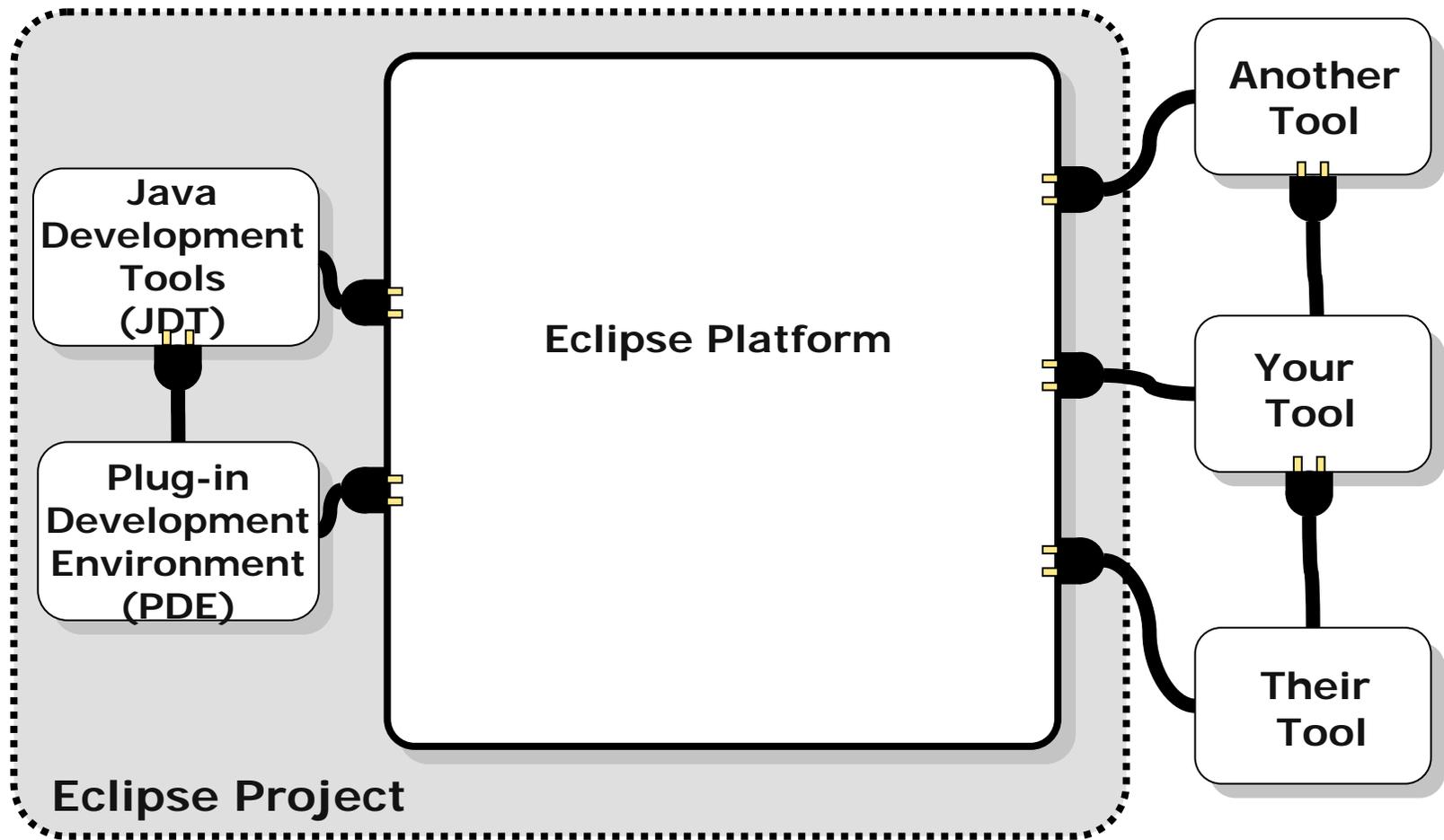


A U.S. Department of Energy laboratory  
managed by The University of Chicago

# Eclipse

- Eclipse is an Open Source community
- It was started in 2001 by IBM
  - IBM donated a lot of research
  - Controlled the early development, but later relinquished control
- Out of the box it looks like a Java IDE
- It is really a Plug-in manager
  - That happens to come with Java Development plug-ins.
  - You can take these out and put your own (and/or others) in

# Eclipse is Very Extensible and Very Flexible



Modified From: Tony Lam, ICALEPCS Presentation, October 2005

## *Eclipse Foundation Membership*

- Strategic Developers (13 as of Jan 2006)
  - At least 8 developers assigned full time to developing Eclipse
  - Contribution up to \$250K
- Strategic Consumers (4)
  - Contribution up to \$500K
  - Can reduce the dues by contributing 1-2 developers
- Three other tiers
  
- Bottom line
  - \$\$\$ and Developers (currently > 150 full time)

# Eclipse Consortium Strategic Members



\* Strategic Consumer

# Eclipse as a Java IDE

The screenshot displays the Eclipse IDE interface for a Java project named "JProbe". The main editor shows the source code for `JProbe.java`, which includes a constructor and an `addPropertyChangeListener` method. The Package Explorer on the left shows the project structure, including the `src` directory and various Java files. The Outline view shows the class hierarchy, and the Java Beans view shows the `JProbe` object. The Problems view at the bottom shows several deprecation warnings for the `getenv(String)` method. The Search view on the right shows search results for "parseInt", indicating 288 occurrences in the workspace.

```
27 public static final boolean useCAJ=true;
28 private MainFrame frame = null;
29 private ListenerList listenerList = new ListenerList(1);
30
31 /**
32  * Constructor for JProbe. Creates the MainFrame.
33  */
34 public JProbe(Composite parent)
35 {
36     if(printThread) {
37         System.out.println("JProbe: " +
38             Thread.currentThread().toString());
39     }
40
41     frame = new MainFrame(parent, SWT.NONE, this);
42
43     // Set window decorations
44     // Make it exit when the window manager close button is clicked
45     // Center it on the screen
46
47     // Make the menu (This will override the plug-in menu)
48     // frame.makeMenu(parent.getShell());
49
50     // Display the MainFrame
51     frame.pack();
52     frame.setVisible(true);
53 }
54
55 /**
56  * Add a PropertyChangeListener to the list
57  * @param listener
58  */
59 public void addPropertyChangeListener(IPropertyChangeListener listener)
60 {
61     if(listenerList != null) {
62         listenerList.add(listener);
63     }
64 }
```

Problems: 0 errors, 48 warnings, 0 infos

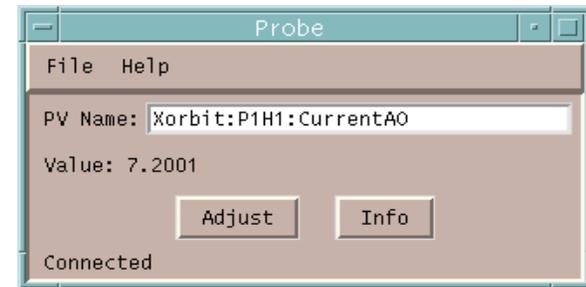
Description
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
The method <code>getenv(String)</code> from the type <code>System</code> is deprecated
Discouraged access: The type <code>EclipseEnvironmentInfo</code> is not accessible due to restriction on required library <code>C:\eclipse\plugins\or</code>
The local variable <code>view</code> is never read

## *Rich Client Platform (RCP)*

- “Rich Client” is a term from the early 1990’s that distinguished applications built with Visual Basic and the like from “Console” or “Simple” applications
- Eclipse is particularly suited to Rich Client applications
- The possibility of using the Eclipse platform for applications was there from the beginning, but foreshadowed by its use as an IDE
  - In the early days it required hacking to make Rich Clients
- RCP is now (as of Eclipse 3.1) supported by the interface and encouraged
- You essentially use Eclipse as a framework for your application
  - You inherit all of its built-in features
  - As well as those from other community plug-ins
- You include only the plug-ins you need
- Is a very extensible development platform
  - You can use plug-ins developed by others as needed
  - Others can use yours and extend them

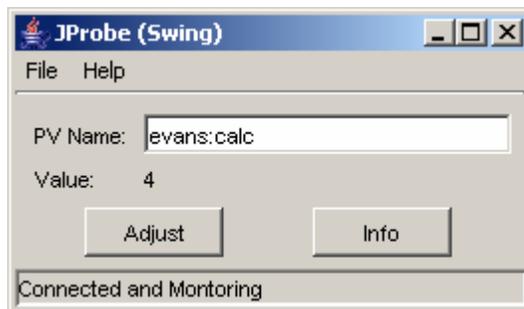
# Eclipse as a Rich Client Platform

- Looks like an application, not an IDE
- Inherits a lot of functionality
  - Persistence (Properties and Preferences)
  - Help
  - Featured About dialog (like Eclipse's)
  - Splash screen
  - Dockable windows, and much more ...

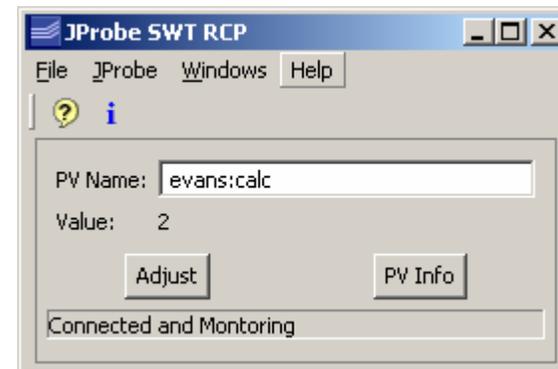


Pyre Application

`probe.py --frame.pvName=Xorbit:P1H1:CurrentA0`



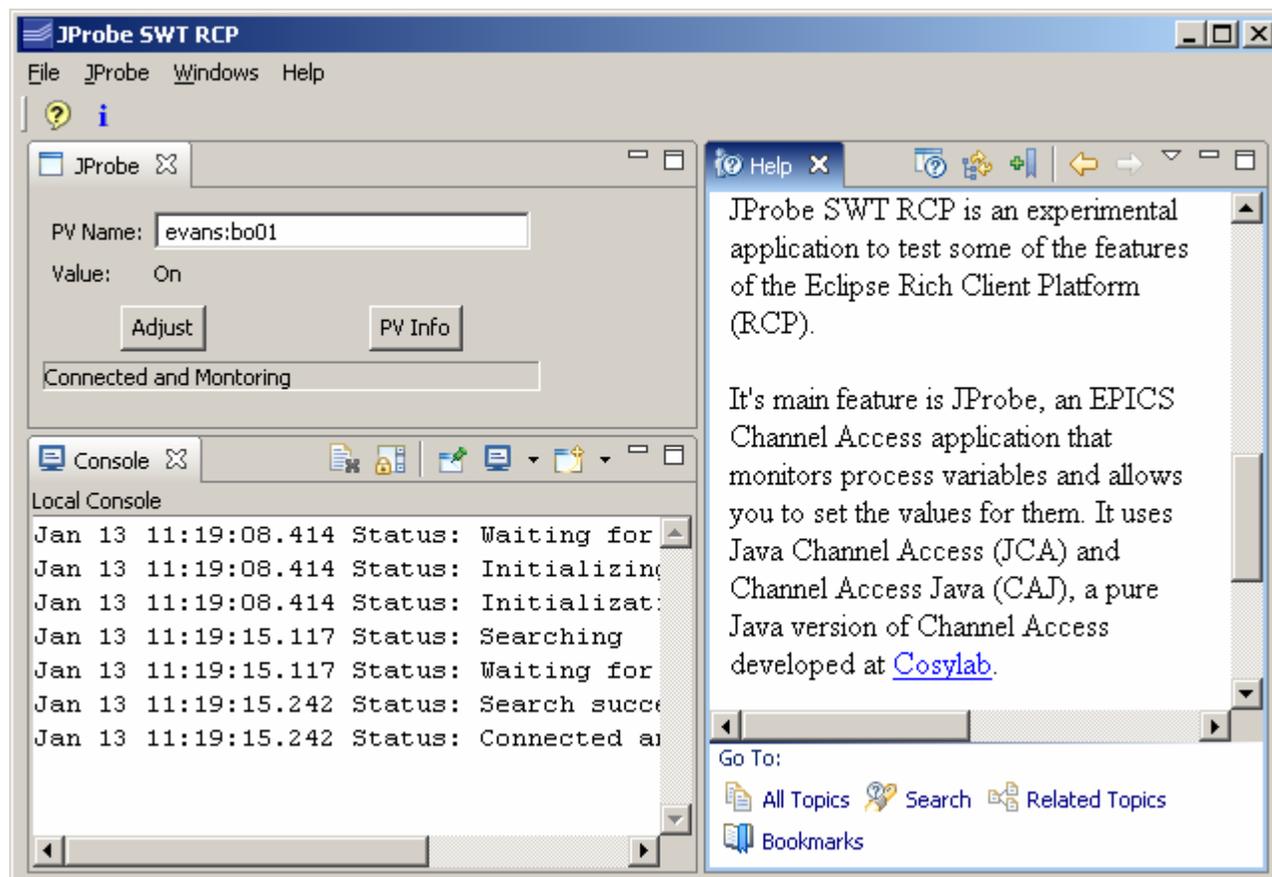
Java Application



RCP Application

# Probe on Steroids

## Leveraging the Eclipse Framework



# An RCP Application is Also a Plug-In

The screenshot displays the Eclipse IDE interface for a C/C++ project named "Test IOC". The left-hand side shows a project navigator with a tree view containing folders like "bin", "include", and "iocBoot", along with files such as "Makefile", "README", and "envPaths". The main editor window shows the "st.cmd" file with the following content:

```
#!/.../bin/cygwin-x86/test

## You may have to change test to something else
## everywhere it appears in this file

#< envPaths

## Register all support components
dbLoadDatabase("../db/test.dbd",0,0)
test_registerRecordDeviceDriver(pdbbase)
```

The bottom-right pane shows the "Problems" and "Console" tabs. The console output includes the following text:

```
Test IOC [C/C++ Local Application] C:\Documents and Settings\evans\My Documents\Eclipse\workspace\Test IOC\bin\cygw
Starting iocInit
#####
### EPICS IOC CORE built on Apr 20 2006
### EPICS R3.14.8.2 $$Name: R3-14-8-2 $$ $$Date: 2006/01/06 15:55:13 $$
#####
iocInit: All initialization complete
## Start any sequence programs
{seq sncExample,"user=evans"
epics> db1
evansHost:aiExample
evansHost:aiExample1
evansHost:aiExample2
evansHost:aiExample3
```

The bottom-left pane shows the "JProbe" window, which is circled in red. It displays the following information:

PV Name: evansHost:aiExample  
Value: 6  
Adjust [button] PV Info [button]  
Connected and Monitoring

## Bottom Line

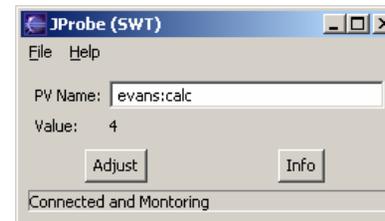
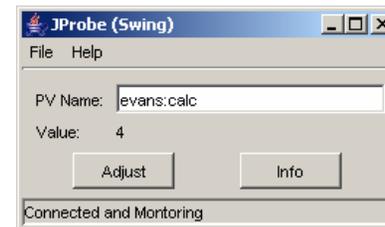
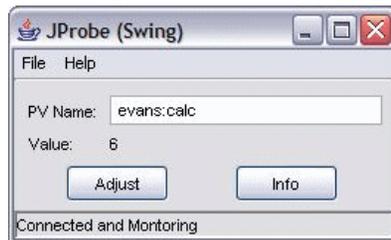
- Is a very powerful and extensible IDE and Framework
- Is also an IE - A way to organize your work
- Is Open Source
- Has a community
- Is supported by most of the industry
- Has a large number of developers (>150)
- Has significant financial backing
- Are many 3<sup>rd</sup>-party Plug-ins, both free and commercial
- Are more than 60 open-source projects
  - From Web Tools to Code Profilers
- Is continuing to expand and improve rapidly
- Is free
- Downsides
  - Is a continually changing, moving target

# AWT vs. SWT - You Have to Decide

- AWT / Swing (Abstract Windowing Toolkit)
  - Write once, run anywhere
  - Formerly ugly, with bad performance
  - Now look and work well
  - Use garbage collection
  - Come with the JDK and JRE
- SWT / JFace (Standard Window Toolkit)
  - The important fact is that Eclipse uses SWT, not AWT
  - Supposed to look better, run faster
  - A thin wrapper around native widgets
  - SWT components must be disposed (vs. garbage collected)
    - *Owing to need to free native resources*
  - Need JNI libraries for each platform
  - Distribution is through the Eclipse Foundation, not Sun

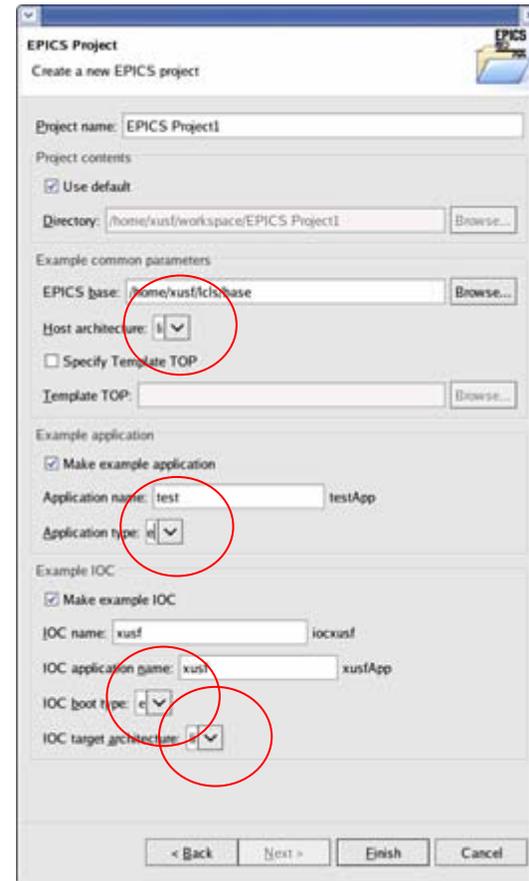
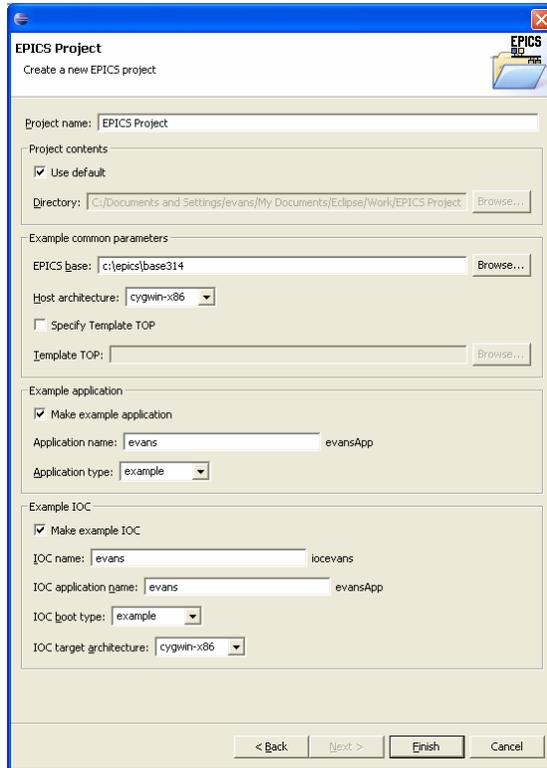
# AWT vs. SWT - More Considerations

- It is not easy to convert between them
- The SWT look is not obviously better
- The performance difference may not be there either, today
- Eclipse uses SWT
  - They are supposed to mix and match, but ???
- Sun is unlikely to include SWT support in the JDK and JRE soon



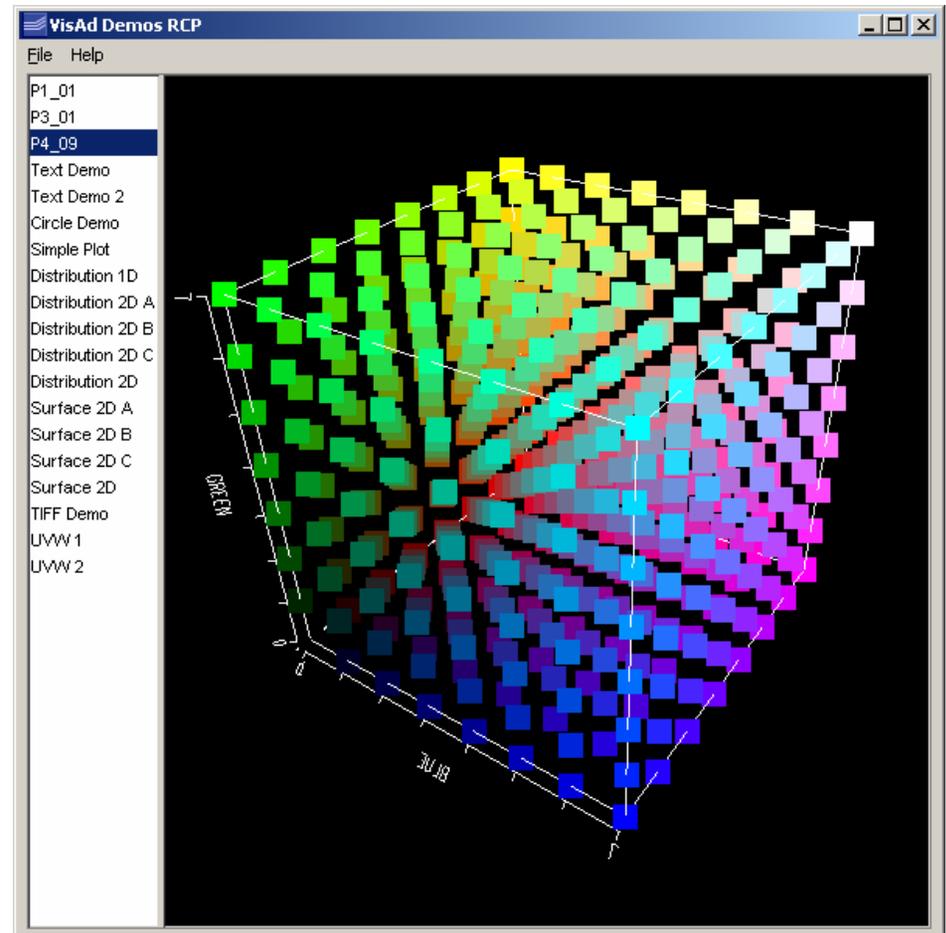
# SWT Platform Dependence

- Example: Working Windows dialog doesn't work right on Linux



# Combining Swing and SWT - SWT\_AWT Bridge

- ContentPane of JFrame is embedded in an SWT Composite
- Menu Initialization is separate from other UI initialization
  - Standalone Swing version uses Swing menus
  - RCP versions uses RCP workbench menus
  - Both can call same instance methods (or not)
- This application also uses JAI and J3D
  - Both are Java extensions
  - Don't play well with Eclipse



# *X-Ray Software Development at the APS*

- Best described as “Uncoordinated”
- Wide variety of languages
  - FORTRAN, C, C++, Perl, Tcl/Tk, Python, Java, ...
- Visualization relies on (different) commercial products
  - IDL, IGOR, Matlab, ...
- Each beamline tends to do its own thing
- Modeling and Analysis is not well integrated with Data Acquisition
- Lack of real-time data reduction
- Little high-performance computing
- Little remote access
- No common data format
  
- The APS has been running for ten years
  - People are getting their work done
  - They are fixed in their ways

# Scientific Software Web Page



## Advanced Photon Source Scientific Software Section

A U.S. Department of Energy, Office of Science,  
Office of Basic Energy Sciences national synchrotron x-ray research facility

[About](#) ▾ [Science](#) ▾ [Media Center](#) ▾ [User Information](#) ▾ [Education](#) ▾ [Facility](#) ▾  

[Argonne Home](#) > [Advanced Photon Source](#) > [APS Engineering Support Division](#) > [Scientific Software](#) >

[APS Home Page](#)  
[APS Engineering Support Division](#)  
[2006 XSD Scientific Software Workshop](#)

### Scientific Software Section

The Scientific Software Section is a newly-created section that is responsible for scientific software for the APS. This group has a two-fold mission: (1) It is to assist in integrating existing analysis and modeling codes with existing data acquisition software, and (2) It is to develop new codes for both existing and new techniques. The group will implement the recommendations of the [2006 XSD Scientific Software Workshop](#).

The acronym **XRAYS** stands for X-Ray Analysis Software or X-Ray Analysis Software.

A list server has been set up for XRAYS. Information about the list, including how to subscribe, is available at <http://www.aps.anl.gov/mailman/listinfo/xrays>, and the archives are available at <http://www.aps.anl.gov/Mailman/archives/public/xrays/2006-June/date.html>.



U.S. Department of Energy



Office of Science  
Department of Energy



Office of Basic Energy Sciences  
Serving the Present, Shaping the Future

[Privacy & Security Notice](#) | [Contact Us](#) | [Site Map](#)

# Scientific Software Section

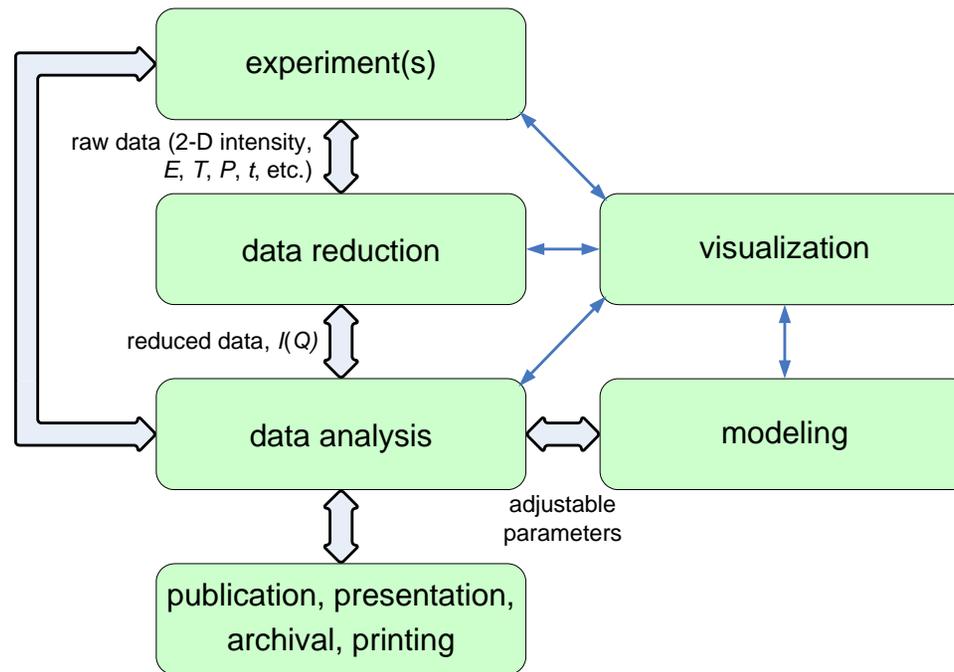
- Specific goals:
  - Combine existing analysis and visualization codes with beamline data acquisition software and transform these codes into easy-to-use software
  - Provide a scientific workbench program that is easy to use and learn and from which users can access all the software that is necessary to manage the entire scientific work flow
  - Create new analysis and visualization applications that can be used on all beamlines and that are easily integrated into the standard workbench
  - Develop a software framework, perhaps more than one, that provides tested and debugged scientific routines, such as fitting and visualization, which can be used by developers to create applications
  - Create an interface to the facilities necessary to provide high-performance computing
  - Provide documentation, distribution, maintenance, and support

# *Rationalization for Eclipse*

- Providing coordination is a primary goal
- Resources are limited
- Have to choose something
  - Eclipse seems like the best choice
  - Powerful, flexible, extensible
  - Open-source
  - Huge community with many projects
  
- Java development environment leads to high productivity
- Deployment via plug-ins appears to solve many problems
  
- Downsides
  - Most X-Ray beamline staff and users are not using Eclipse now

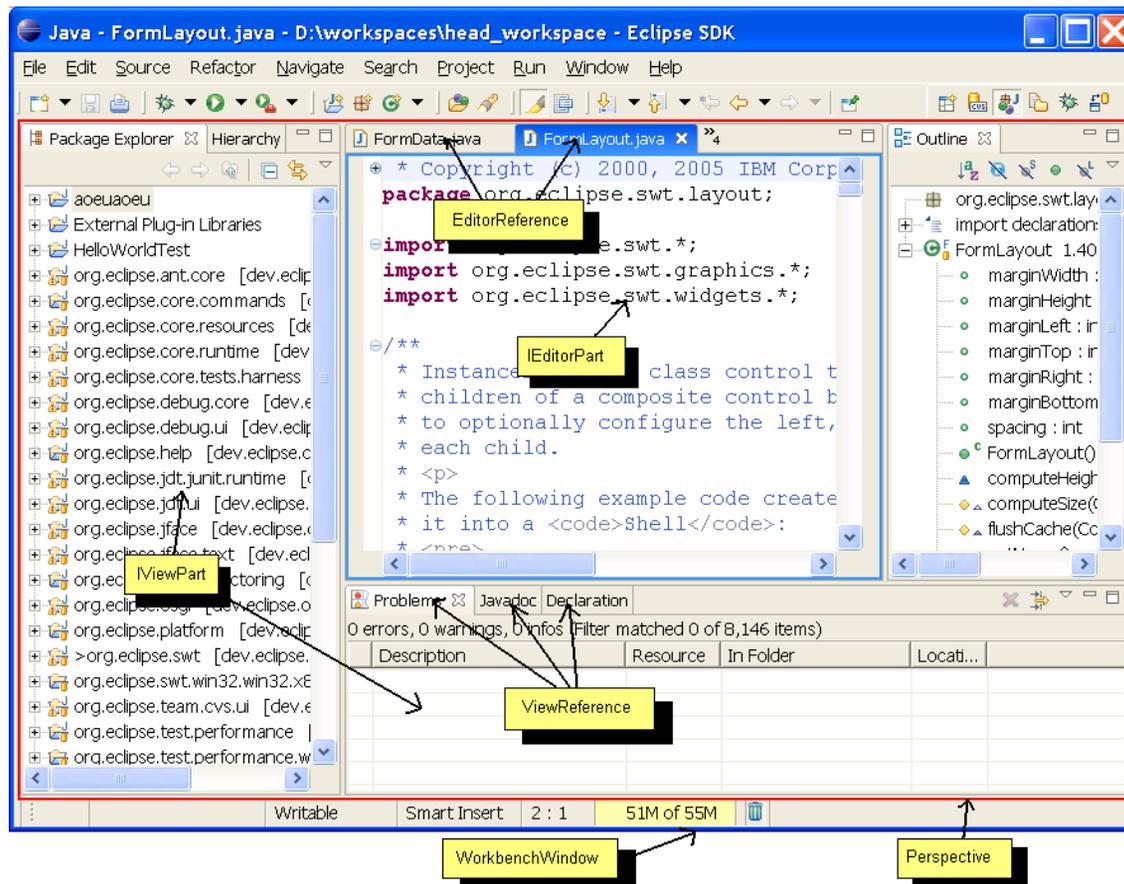
# Want to Manage the Entire Experimental Data Flow

- Eclipse should provide the workbench to do this
  - An IDE without the “D”
  - A way to organize your work
  - A tool for anything in general and nothing in particular



# Eclipse Layout Fundamentals

- Perspective: A particular layout of a Workbench window
  - Has zero or one editor area and zero or more surrounding Views.



From: Stefan Xenos, Inside the Workbench - A guide to the Workbench Internals  
<http://www.eclipse.org/articles/Article-UI-Workbench/workbench.html>

# GumTree (ANSTO)

The screenshot displays the GumTree Platform interface with several active windows:

- Device Navigator:** A tree view showing hardware components like DataNumber, Environment Monitor, HistMem, Macro, Motor (with sub-items s1 and tth), PerfMon, RS232 Controller, ScanObject, SicsRuenStack, SicsVariable, and SingleCounter. Below it is a table of properties and values.
- Command Line Terminal:** Shows terminal output including a welcome message, a warning about hardware, and a directory listing for 'dir'.
- TwoD Plot View:** Displays a 2D plot of a sample (NaCl) at 120x120 units. It includes a color palette set to 'Heat 2' and a smaller inset plot.
- MS Word:** Opened to a document titled 'GumTree - A Java Based GUI Framework for Beamline Experiments' by Tony Lam, showing an abstract.
- SANS Browser:** Shows a web page for 'Neutron & X-ray Data Format' from ANSTO.

At the bottom, the status bar indicates 'SICS scan : point = 3 , x = 0.3 , y = 198.83623' and the system tray shows the start button, taskbar icons, and the time 10:52 AM.

From: Tony Lam, ICALEPCS Presentation, October 2005

# Wind River Workbench

The screenshot displays the Wind River Workbench interface for application development. The main window is titled "Application Development - BrowserView.h - Wind River Workbench".

- Project Navigator:** Shows a list of files and folders, with "BrowserView.h" selected. The "Name Filter" is set to "browser".
- Code Editor:** Displays the source code for "BrowserView.h". The code includes preprocessor directives, class declarations, and a class definition for "BrowserView".

```
#ifndef _GNUCC
#pragma once
#pragma interface
#endif

#define BrowserView_First

#include "Box.h"
#include "CodeTextView.h"

class PrettyPrinter;
class ChangeDirDiag;
class FileList;

//----- BrowserView -----

class BrowserView: public VBox {
protected:
    SeqCollection *path, *directories;
    CompositeVObject *fileLists; // shown file lists
    int nShown; // number of shown file lists
    int left; // index of left most file list in
    VObject *shiftLeft, *shiftRight; // buttons

    ChangeDirDiag *changeDir;

public:
    void LoadFile(int at, FileList *fl);
    void Shell(int at, char *path, char *cmd= 0);

public:
    MetaDef(BrowserView);
    BrowserView(EvtHandler *dp, int numFileLists);
    ~BrowserView();
};
```
- Type Hierarchy:** Shows a supertype hierarchy for "BrowserView", including "VBox", "Box", "CompositeVObject", "VObject", "EvtHandler", and "Object".
- Members of BrowserView:** Lists various members such as "s\_accessMembers: AccessMembers\*", "s\_an: int", "align: VObjAlign", "changeDir: ChangeDirDiag\*", "contentRect: Rectangle", "directories: SeqCollection\*", "fileLists: CompositeVObject\*", "gap: Point", "gisa: Class\*", and "left: int".
- Call Tree:** Shows a tree of callers for the "LoadFile" method, including "Control(int, FileList\*) - BrowserView", "Control(int, int, void\*) - BrowserView", "ClearSelection(bool) - CollectionView", "ClearSelection(bool, bool) - MultiSelCollView", "Control(int, int, void\*) - FileList", "Control(int, int, void\*) - ActionButton", "Control(int, int, void\*) - MarkToggleButton", "Control(int, int, void\*) - Clipper", and "Control(int, int, void\*) - ColorSlider".
- Retriever:** Shows search results for "BrowserView", finding 10 instances in the workspace. The results include "class BrowserView" and "BrowserView \*bv".
- Outline:** Shows a tree view of the project structure, including "Includes", "Macros", "BrowserView", "FileBrowserPPrinter", "FileBrowserTextView", and various class and operator definitions.
- Include Browser:** Shows a list of files included in the project, such as "BrowserView.h - /filebrowser", "Box.h - /filebrowser/et3", "CompVObject.h - /filebro", "VObject.h - /filebro", "CodeTextView.h - /filebrowsi", "VObjectTextView.h - /filebr", "TextCmd.h - /filebr", "CompVObject.h - /fi", "TextFormatter.h - /fileb", "Font.h - /filebrowsi", and "Mark.h - /filebrowsi".

# EPICS Control System Studio

The screenshot displays the EPICS Control System Studio interface. The main window shows a data browser with three plots: 'evans:bo02' (green), 'evans:bo01' (red), and 'evans:calc' (blue). The x-axis represents time from 2006/12/08 11:14:53 to 11:15:15. The y-axis ranges from -10 to 3.0. The 'evans:calc' plot shows a sawtooth pattern, while the 'evans:bo01' and 'evans:bo02' plots show square waves.

Below the data browser is the 'Data Browser Config' window, which includes a 'Process Variables' table and an 'Archives used for selected Process Variable' table.

Process Variable	Min	Max	Axis	Color	Wi...	Type
evans:calc	-10.27	6.04999...	0	Blue	0	linear
evans:bo01	-1.1320...	2.132	1	Red	0	linear
evans:bo02	-0.0100...	3.25400...	2	Green	0	linear

The 'Archives used for selected Process Variable' table is currently empty.

On the right side of the interface, the 'EPICS PV Tree' window shows the hierarchy of process variables. The selected PV is 'evans:calc', which is calculated as '4.0'. The tree includes various input process variables (INPA through INPL) and their associated calculations.

At the bottom left, there is a 'Clock' window displaying a circular analog clock showing approximately 12:15.

# EPICS IDE : IOC Development

The screenshot displays the Eclipse IDE interface for EPICS IOC development. The main editor shows the `st.cmd` file with the following code:

```
1#!/usr/bin/cygwin-x86/test
2
3### You may have to change test to something else
4### everywhere it appears in this file
5
6#< envPaths
7
8## Register all support components
9dbLoadDatabase("../db/test.dbd",0,0)
10test_registerRecordDeviceDriver(pdbbase)
11
12## Load record instances
13dbLoadRecords("../db/dbExample1.db", "user=evansHost")
14dbLoadRecords("../db/dbExample2.db", "user=evansHost, no=1, scan=1")
15dbLoadRecords("../db/dbExample2.db", "user=evansHost, no=2, scan=2")
16dbLoadRecords("../db/dbExample2.db", "user=evansHost, no=3, scan=5")

```

The console output shows the execution of the `st.cmd` file:

```
Test IOC [C/C++ Local Application] C:\Documents and Settings\evans\My Documents\Eclipse\Work\Test IOC\bin\cygwin-x86\st.cmd
dbLoadRecords("../db/dbSubExample.db", "user=evansHost")
## Set this to see messages from mySub
#var mySubDebug 1
iocInit()
Starting iocInit
#####
### EPICS IOC CORE built on Apr 20 2006
### EPICS R3.14.8.2 $$$Name: R3-14-8-2 $$$Date: 2006/01/06 15:55:13
#####
iocInit: All initialization complete
## Start any sequence programs
#seq sncExample, "user=evans"
epics> db1
evansHost:aiExample
evansHost:aiExample1
evansHost:aiExample2
evansHost:aiExample3

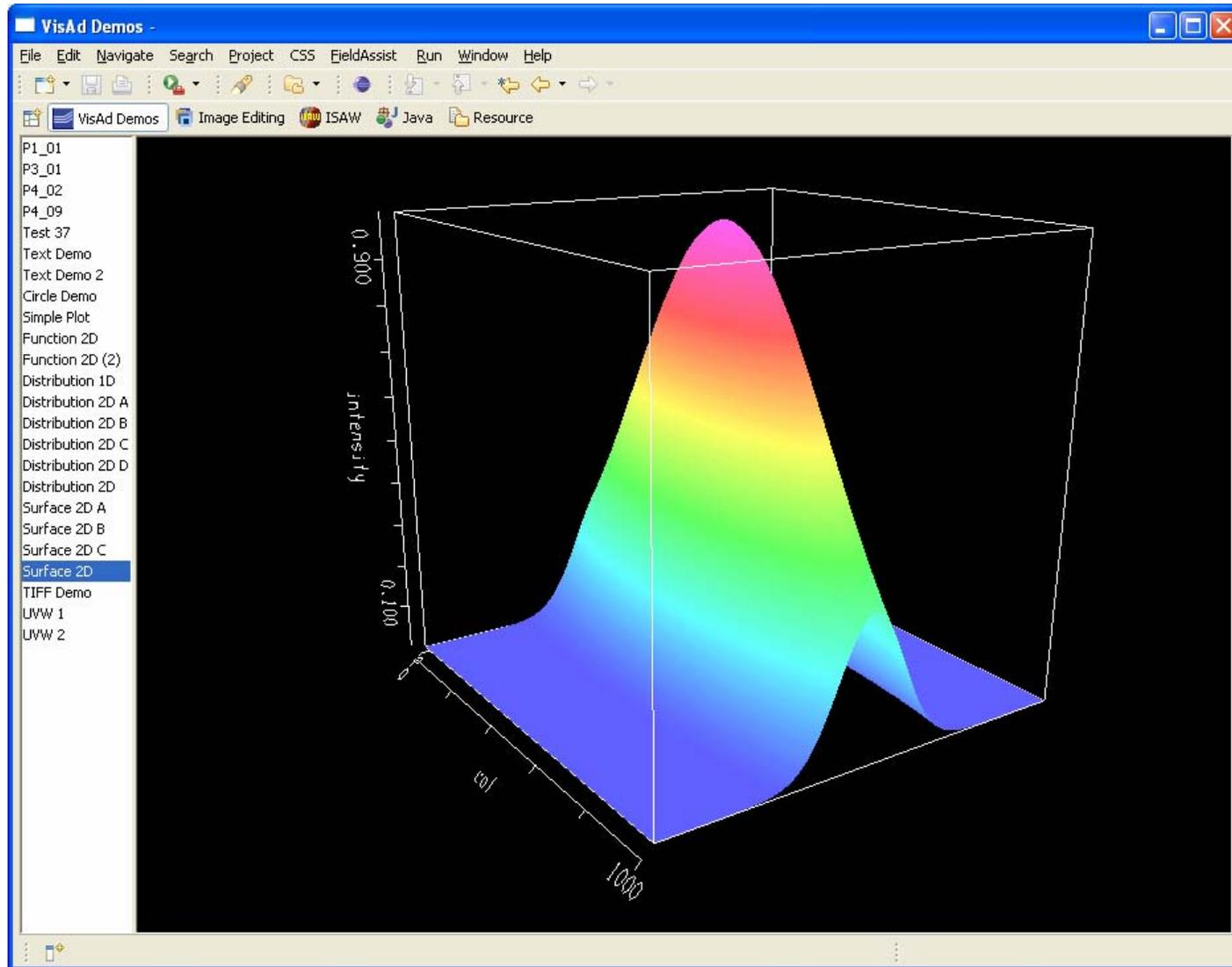
```

The 'New Project' dialog is open, showing the 'EPICS Project' wizard selected. The 'Wizards' list includes:

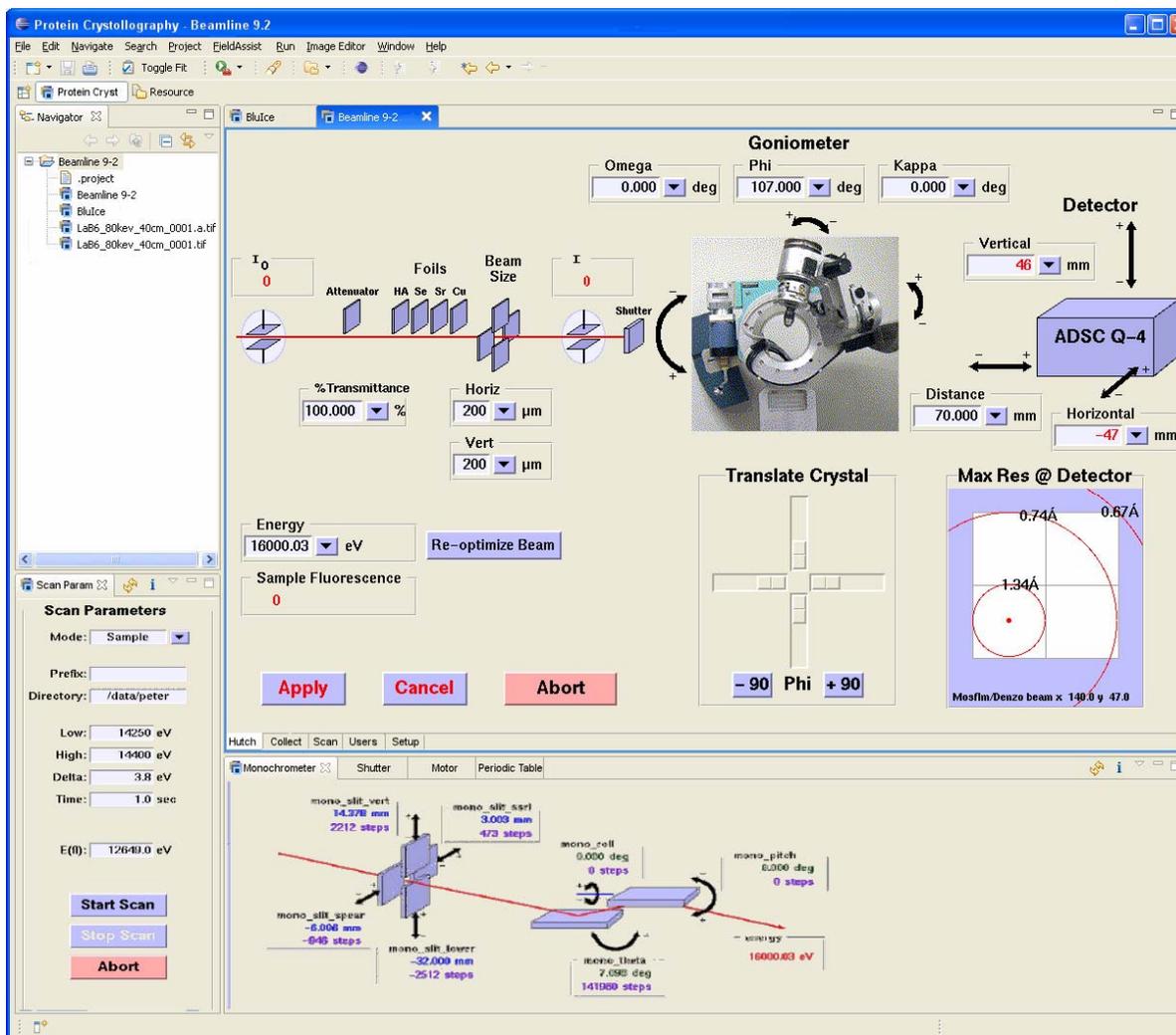
- C++
- Managed Make C++ Project
- Standard Make C++ Project
- CVS
- Eclipse Modeling Framework
- EMF Project
- Empty EMF Project
- EJB
- EPICS
- EPICS Project
- Graphical Modeling Framework
- J2EE
- Java
- Java Project
- Java Project from Existing Ant Buildfile

The 'EPICS Project' wizard is highlighted, and the 'Next >' button is visible at the bottom of the dialog.

# VisAD: A Perspective Can be a Single Application

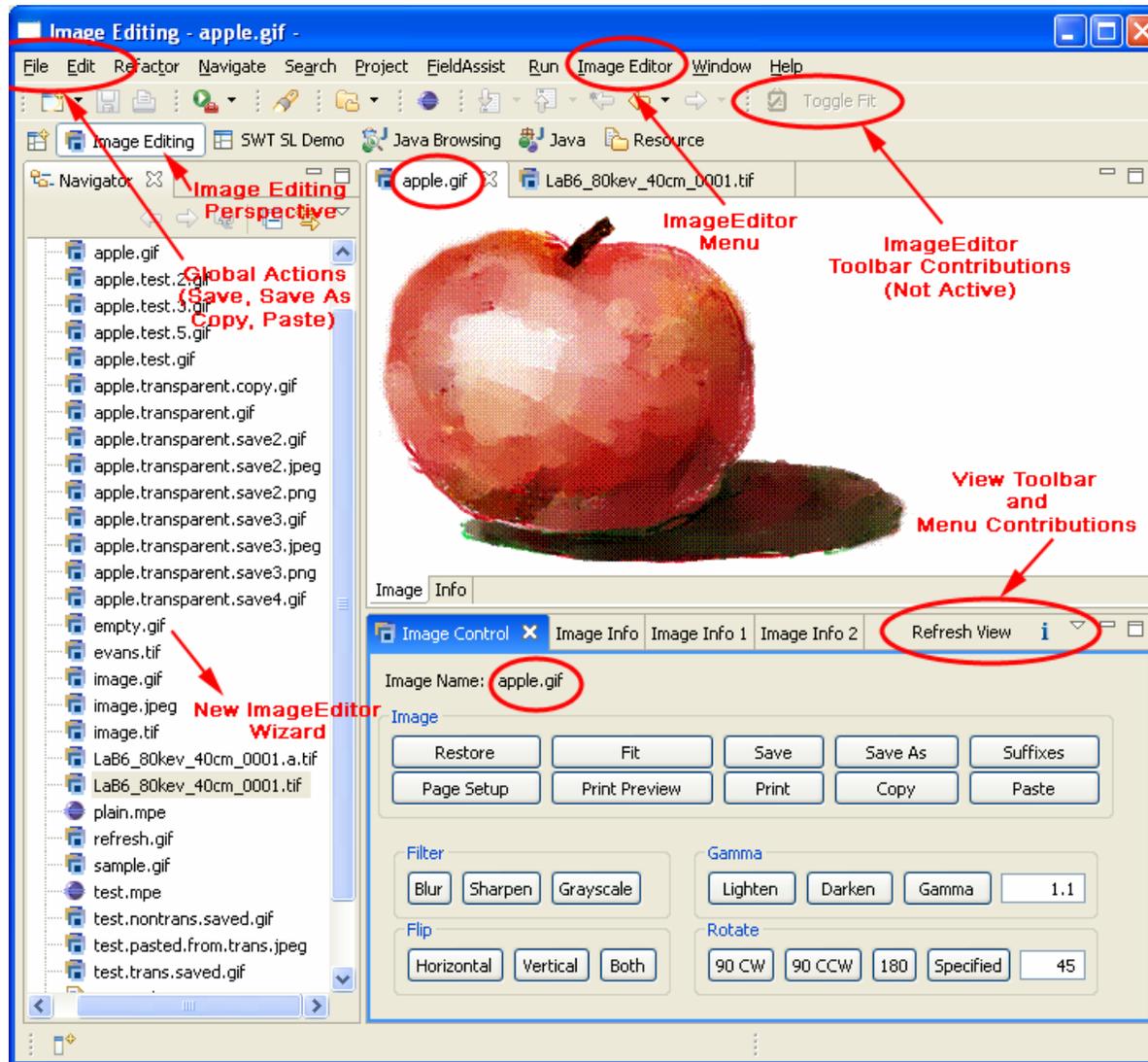


# X-Ray Experiment



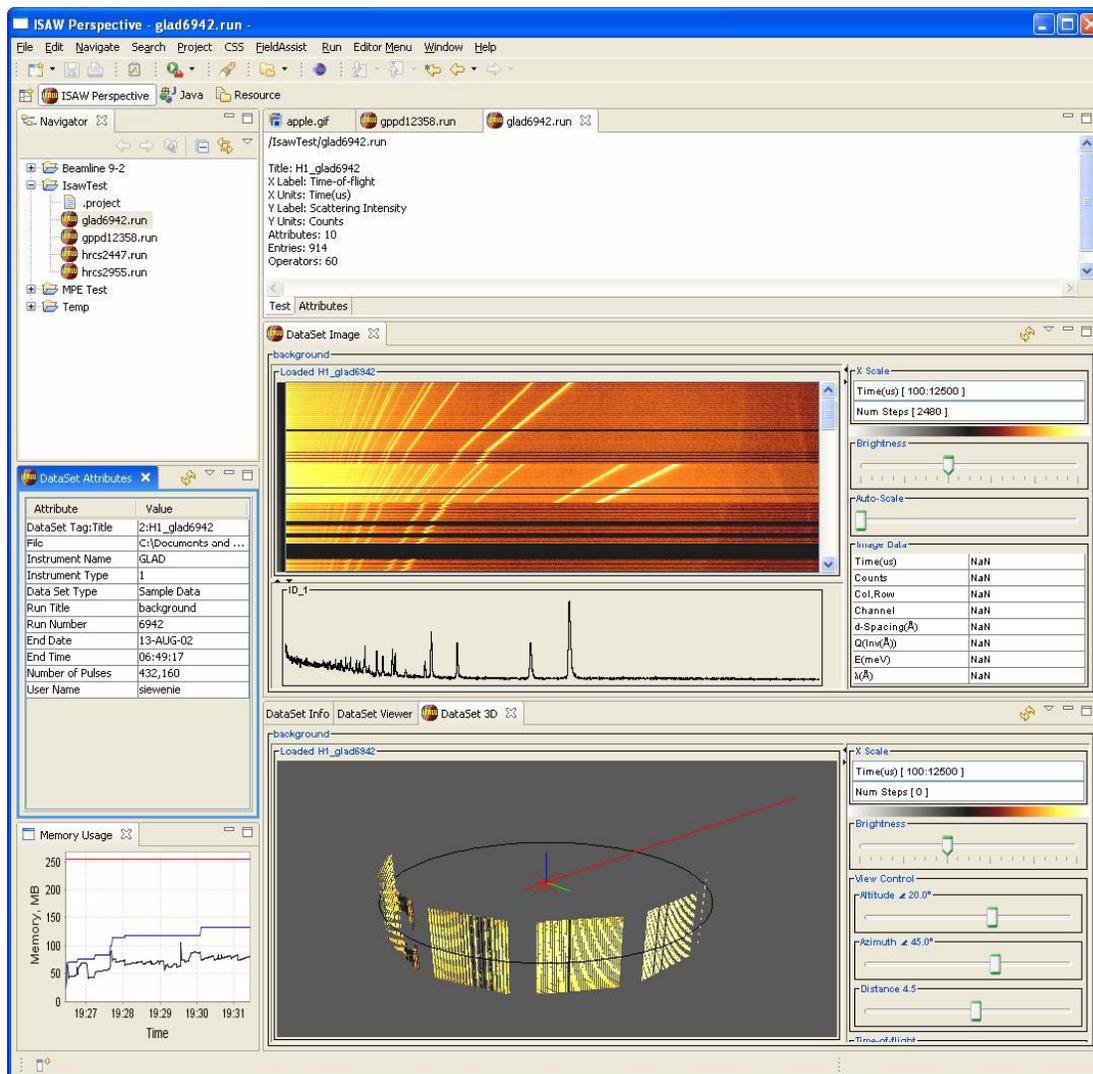
Images from: BLU-ICE and the Distributed Control System, NOBUGS III, January 2000

# Image Editor as an Experiment Prototype

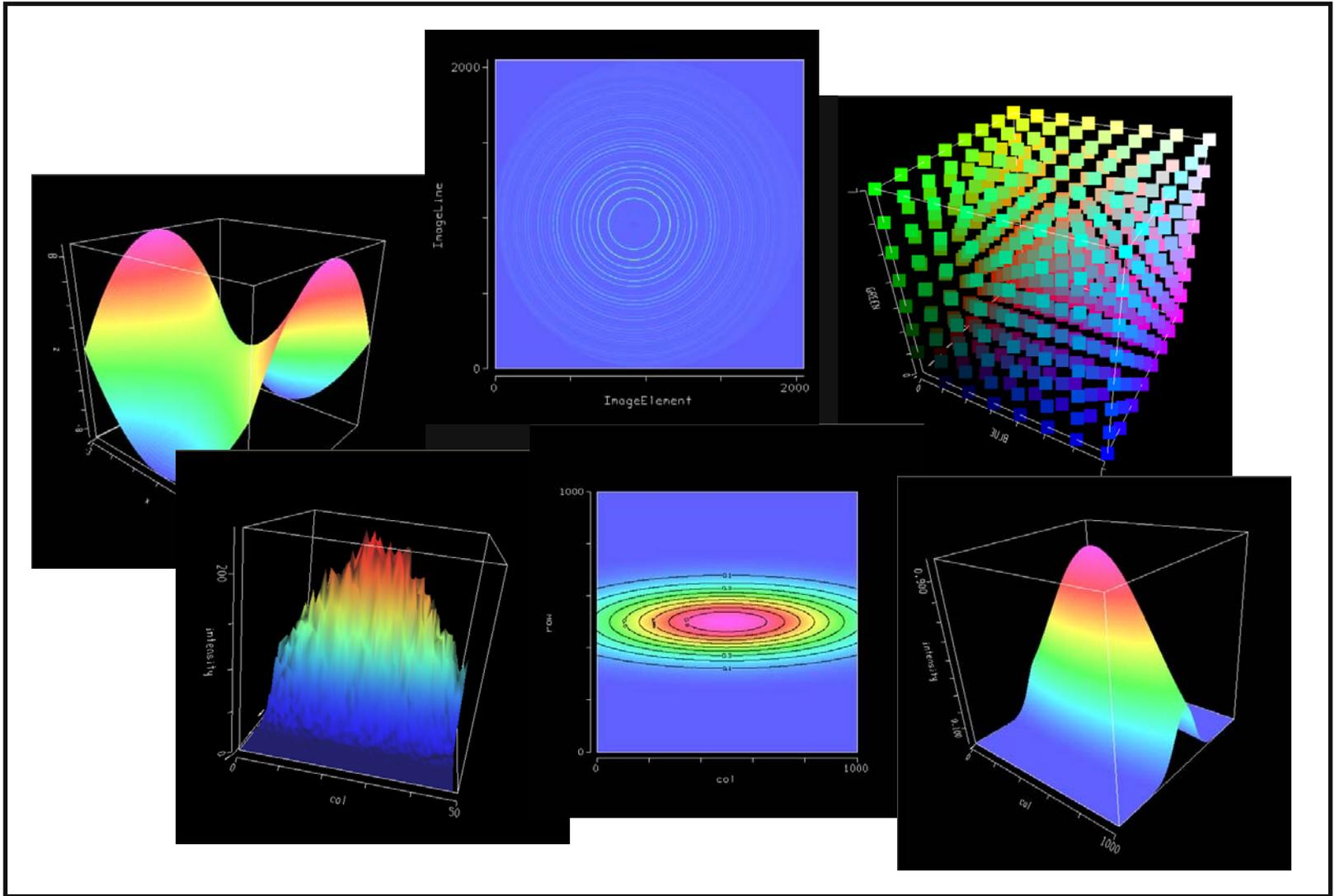


# Prototype Implementation of ISAW in Eclipse

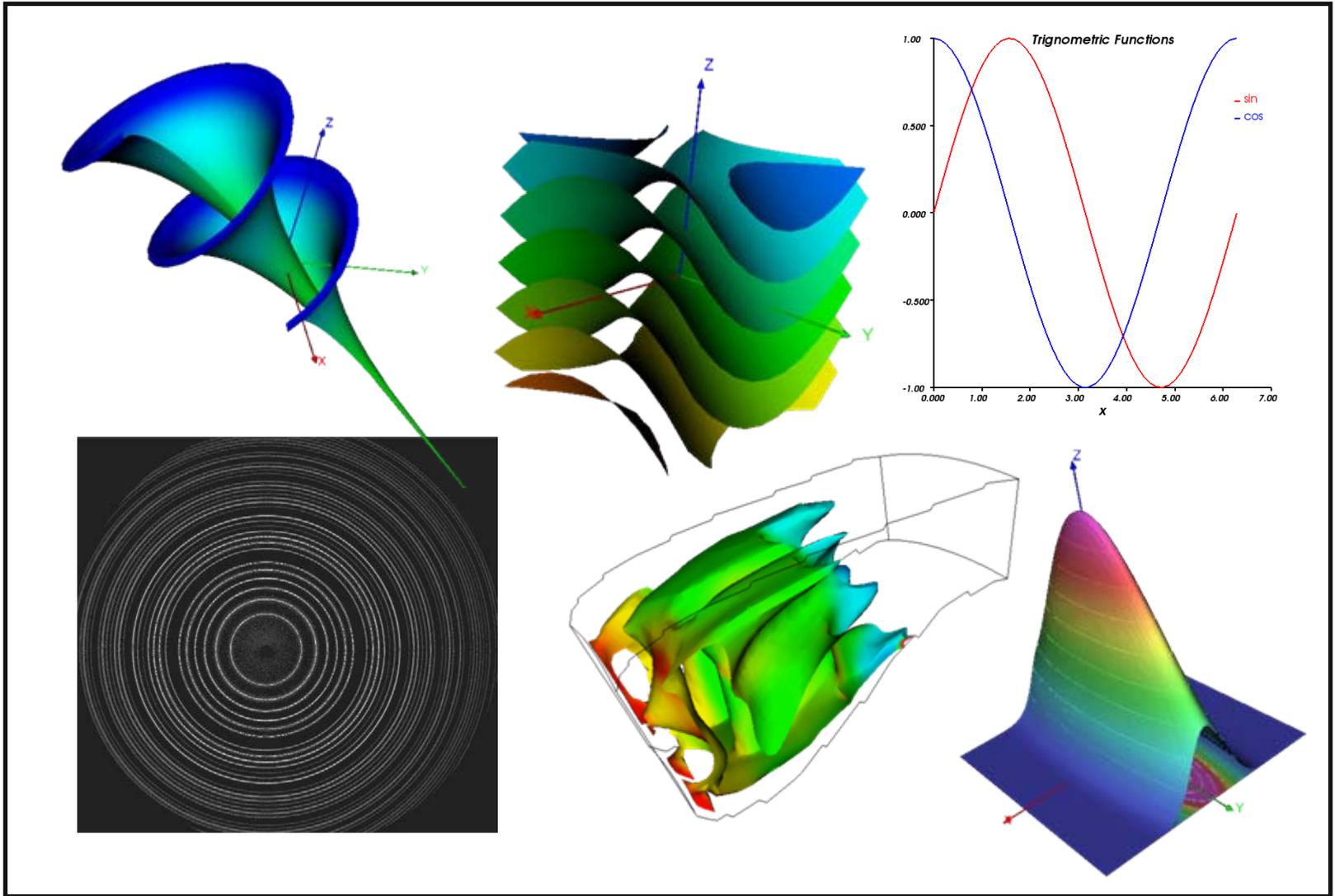
- Perspective
- Editor for DataSets
  - .run, .isd
- Some Views
  
- All work together



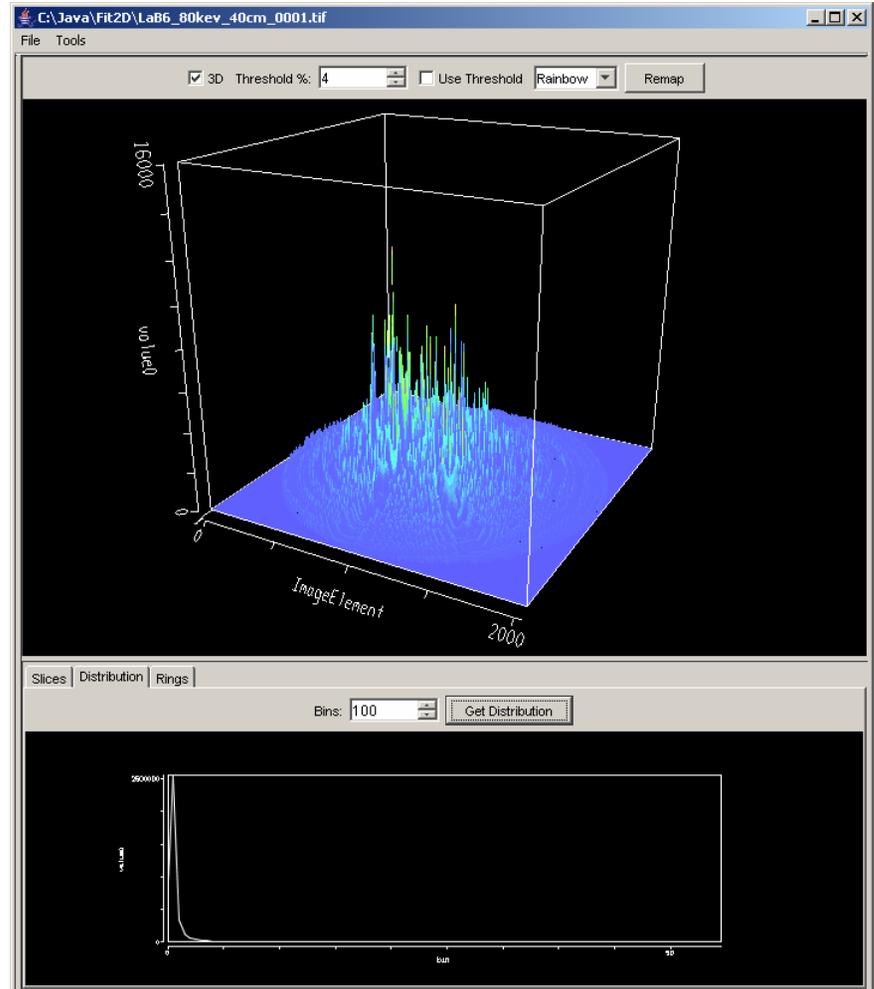
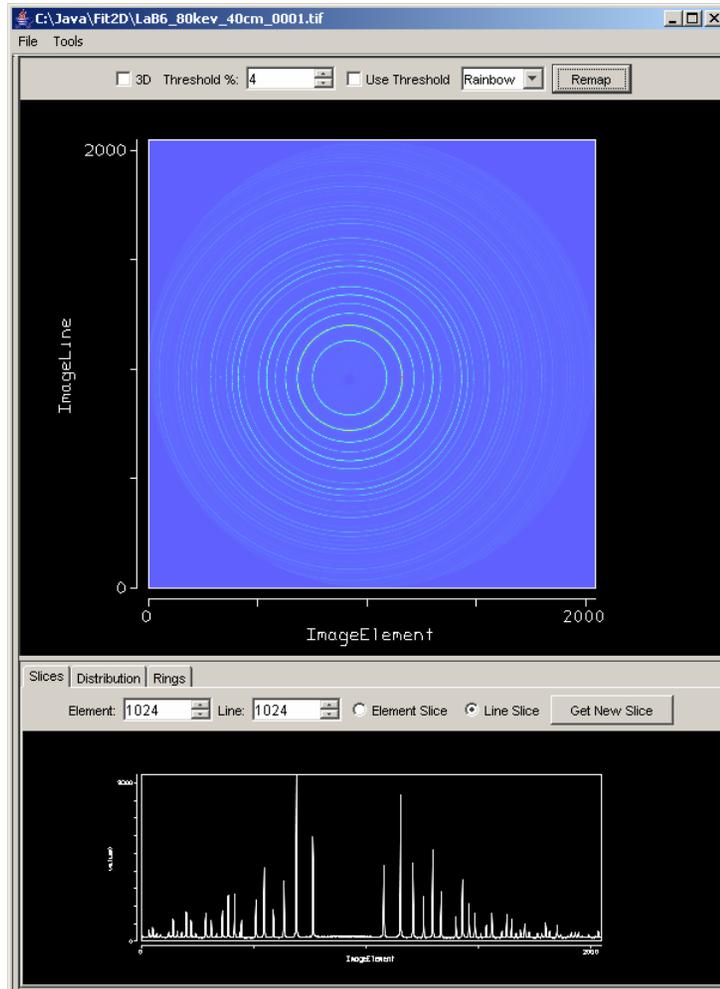
# Visualization: VisAD



# Visualization: VTK



# Prototype Image Analysis Tool using VisAD Graphics



# *Thank You*

*This has been a  
Scientific Software Presentation*

# *Thank You*

*This has been a  
Scientific Software Presentation*