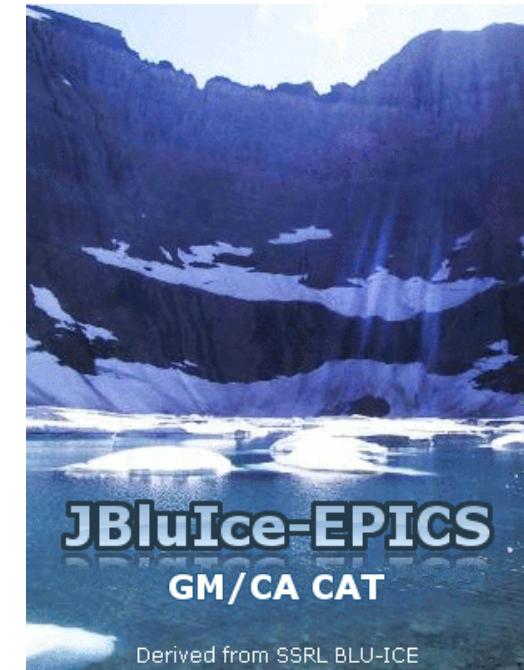


JBlulce-EPICS: A New GUI for Macromolecular Crystallography at the Advanced Photon Source

Mark Hilgart, Sudhir Babu Pothineni, Sergey
Stepanov, Oleg Makarov, Nagarajan Venugopalan,
Michael Becker, Ruslan Sanishvili, Craig Ogata, Derek
Yoder and Robert Fischetti

GM/CA CAT at the Advanced Photon Source
Biosciences Division of Argonne National Laboratory
USA



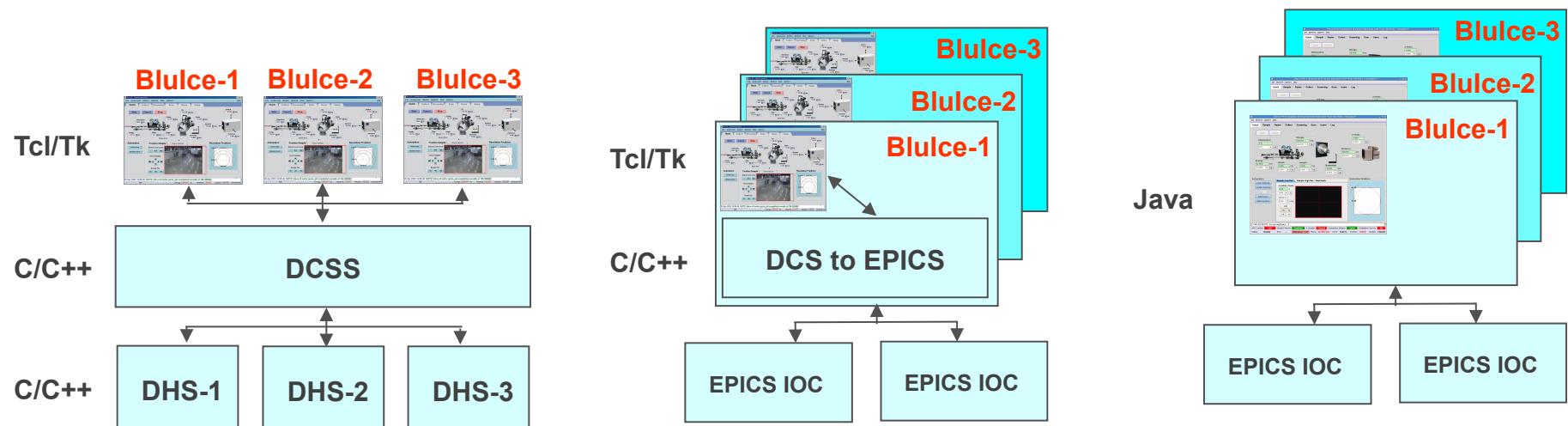
Derived from SSRL BLU-ICE

Control Software Requirements

- GM/CA is a macromolecular crystallography beamline
- Our users require:
 - Familiar interface
 - Crystallographers are highly mobile
 - Need to use software at different beamlines with little training
 - Rapid, reliable feature development
 - Crystallographers have many choices of beamlines
 - Advances from other beamlines must be incorporated quickly
 - New features must be released quickly before others do – yet must be high quality
- Goals for the transition are therefore:
 - Make the client architecture as simple and flexible as possible
 - Provide a reliable client during the transition without interruption



History of JBlulce



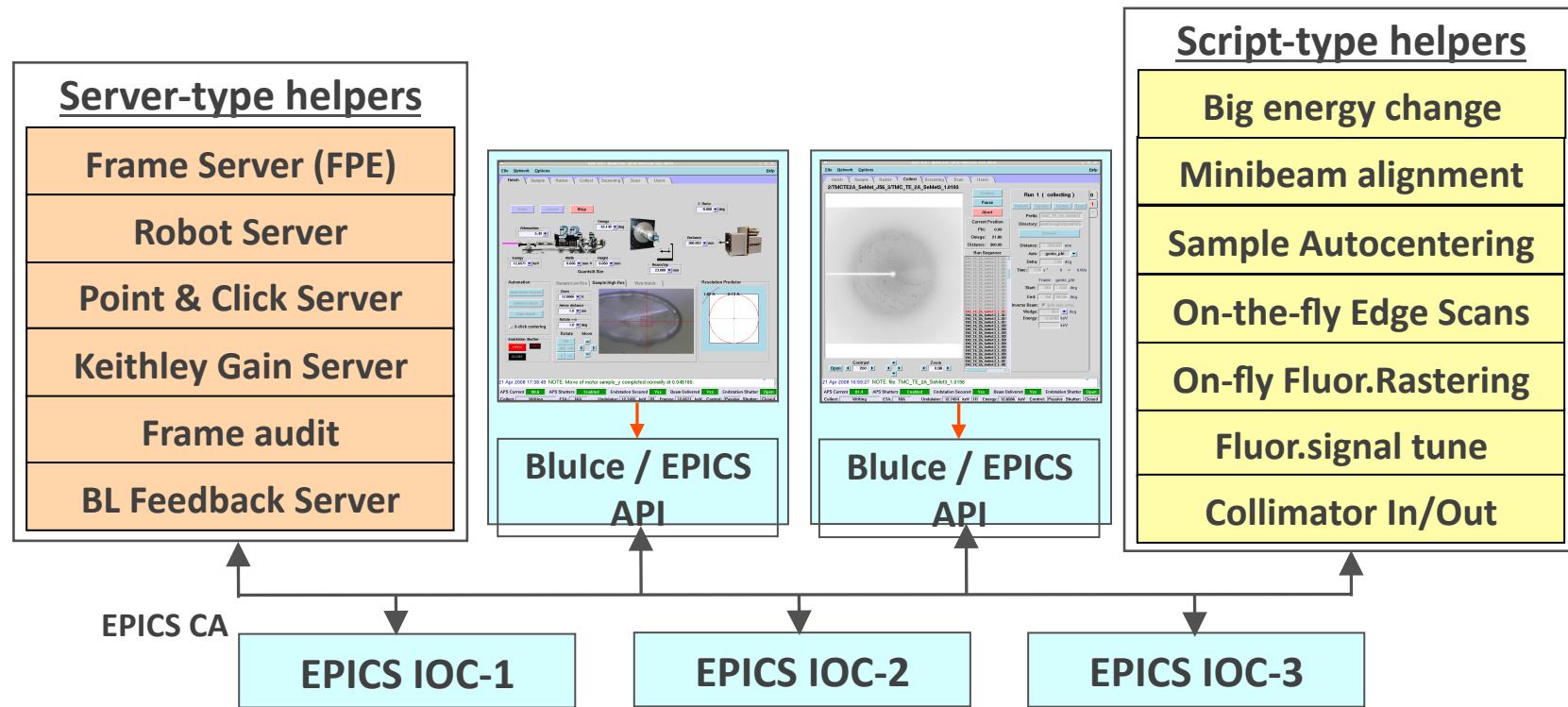
Blulce Architecture (SSRL)

Blulce-EPICS Architecture

JBlulce architecture

- SSRL Blulce: 1997
- GM/CA replaced SSRL Blulce backend: 2003-2005
- GM/CA developed JBlulce: 2008-2010

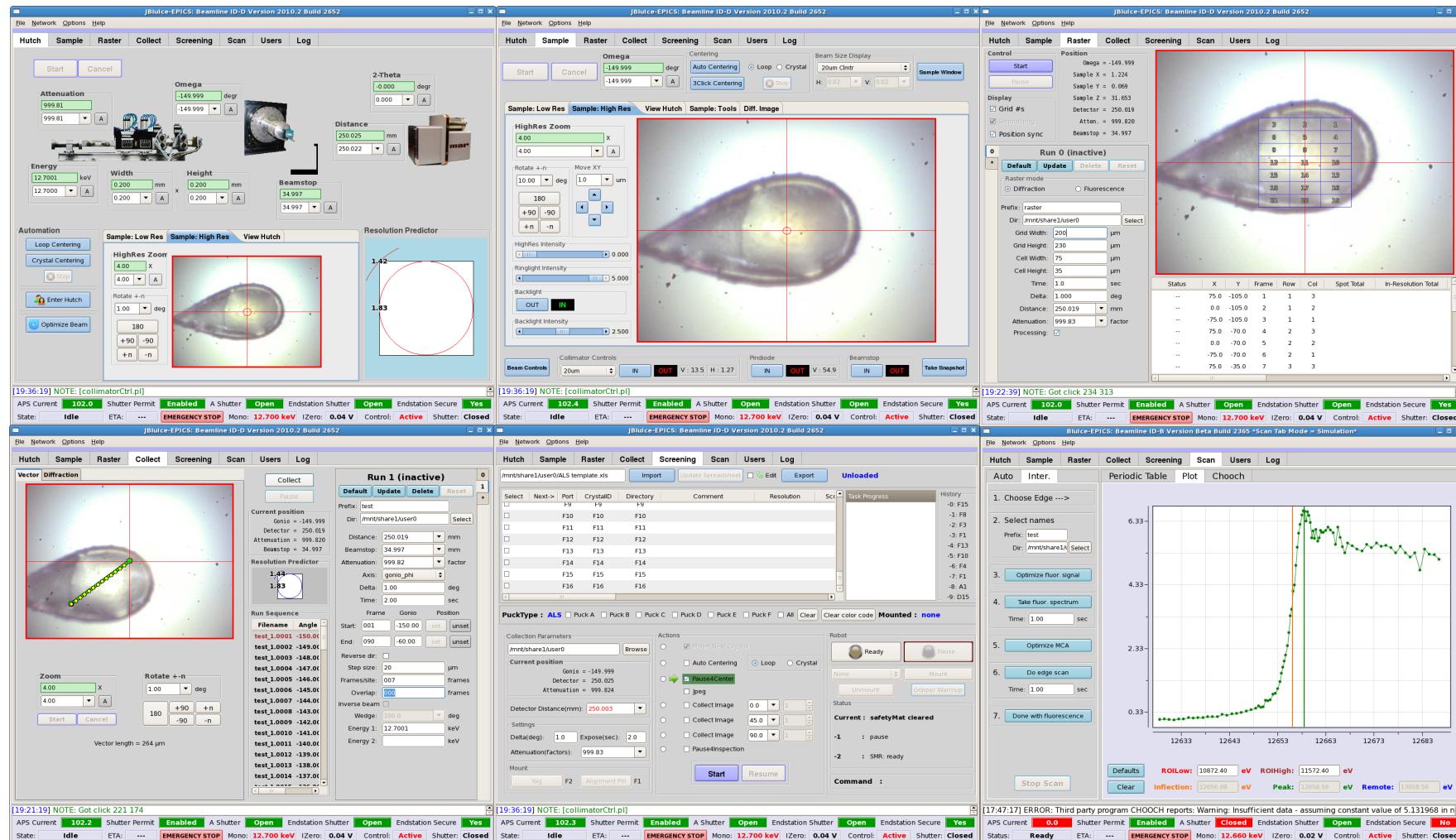
Interaction between JBlulce-EPICS and helpers



- Client performs some operations
- Large or complex operations are performed by scripts or servers
- Allows staff to work independently with EPICS as the interface



JBlulce-EPICS: 100% Java in June 2010



Mark Hilgart - JBlulce-EPICS - October 11, 2010



What changed from Blulce to JBlulce?

- **Multiple languages -> single language**
- Tcl/C++ -> Java
- Architecture changes
- Features



Single language: Advantages

- Easier debugging
 - Full stack trace is always available
 - Step-through debugging always works
- Higher reliability
 - Thanks to easier debugging and JVM protection from memory errors
- Faster development
 - No protocol layer means changes are easier



What changed from Blulce to JBlulce?

- Multiple languages -> single language
- **Tcl/C++ -> Java**
- Architecture changes
- Features



Tcl/C++ to Java

- Java removes uncertainty about memory corruption
 - Tools like valgrind and electric fence, and classes like checked pointers, are not required
 - Time is saved in testing and debugging
- Java (SWT) allows custom widgets in the same language
- Eclipse IDE has more features and is more reliable than ActiveState's IDE for Tcl



What changed from Blulce to JBlulce?

- Multiple languages -> single language
- Tcl/C++ -> Java
- **Architecture changes**
- Features

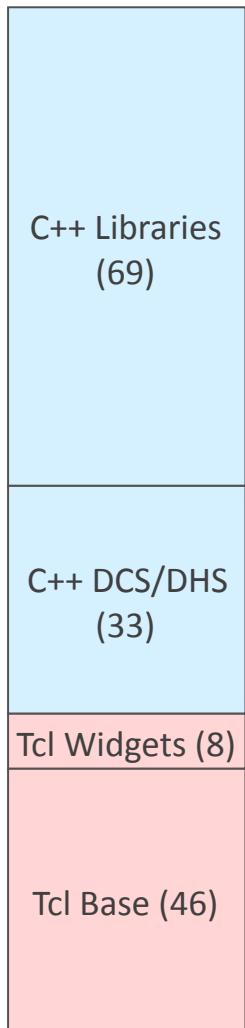


Architecture: Thread-based actions

- Blulce-EPICS had some threaded commands, but now all are converted
- Single 1-2 page methods tell the whole story of an action
 - In a thread, the line number tells you a lot about what's happening
 - In a callback-based class, which timers are running, and which callback will be called next?
- Helper scripts are written this way, so porting methods (and keeping in sync) is straightforward



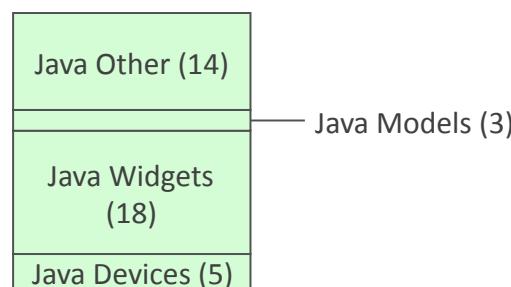
Architecture: 3x LOC reduction



Blulce-EPICS (156)

JBlulce-EPICS (40)

(# = kLOC)



- Proprietary protocols are replaced by standard ones
 - TCP sockets
 - HTTP
 - Math
- Replaced DCS layer with direct method calls

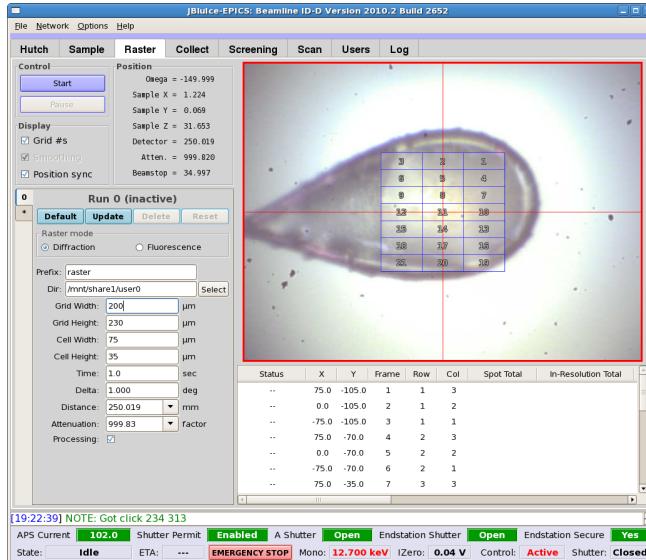


What changed from Blulce to JBlulce?

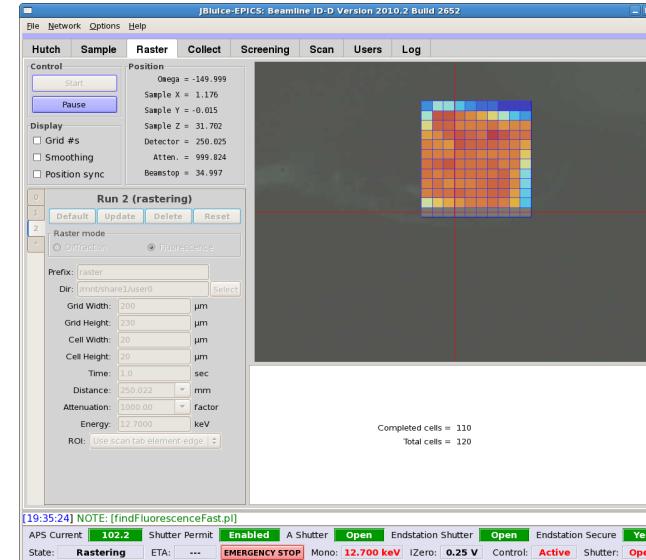
- Multiple languages -> single language
- Tcl/C++ -> Java
- Architecture changes
- **Features**



Features: Raster



Diffraction raster



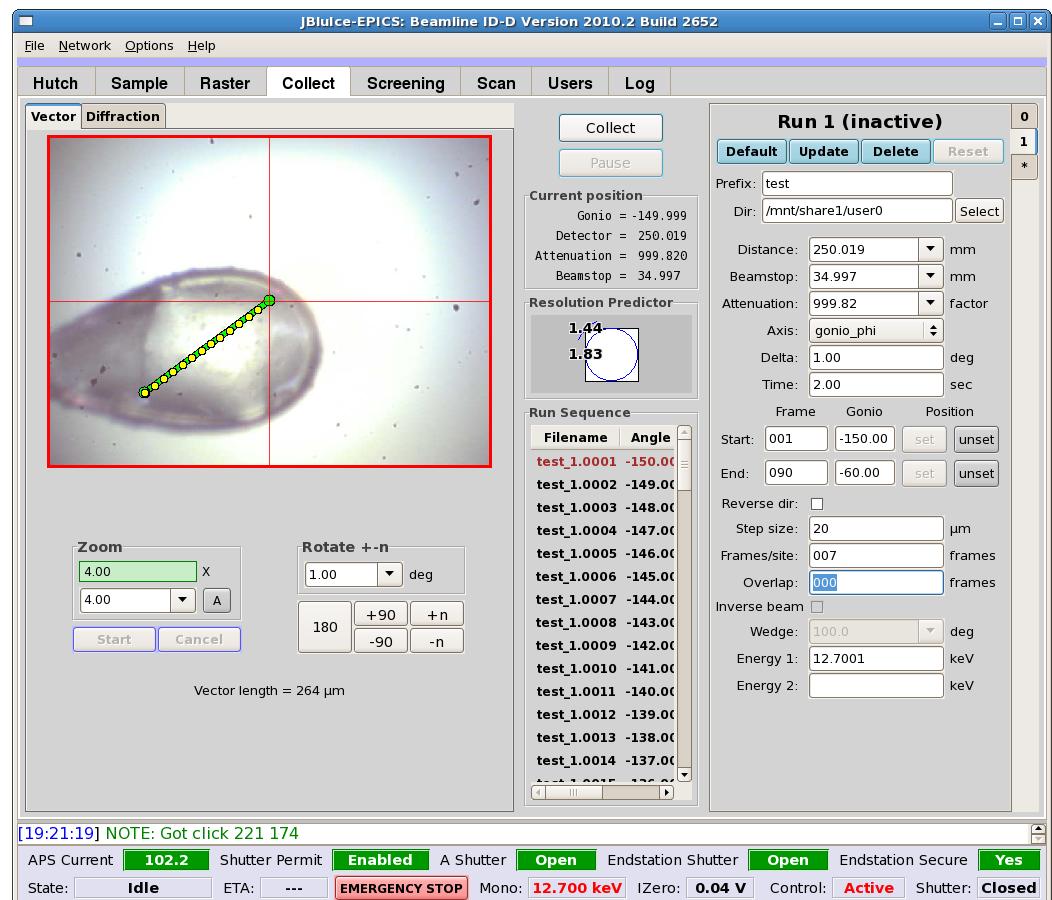
Fluorescence raster

- Release schedule
 - 2008-2: Initial release
 - 2009-2: Fl. raster
 - 2010-1: Multiple run tabs
- Two uses
 - Find invisible crystals
 - Find the best diffracting areas of large crystals
- Diffraction vs. fluorescence
 - Diffraction is always relevant
 - Fluorescence is 4x faster and causes less radiation damage



Features: Vector collect

- Data collection along an arbitrary 3D vector
 - To define, center endpoints and click “set”
- Uses
 - Collect along rod-shaped crystals
 - Raster crystals that don't fit the rigid raster tab grid
- Visualization over live video
- Parameters for:
 - Frames per site
 - Site spacing
 - Overlap



Features: Editable spreadsheet and Weblce integration

The figure consists of three panels. The top panel is a screenshot of the BlJBlulce-EPICS software's control interface, showing a grid of ports (A1-A8) and their status. The middle panel is a screenshot of a web browser displaying a spreadsheet titled 'User Cassettes | BL1-5 Cassettes | Cassette Summary | Cassette Details'. The spreadsheet contains data for three samples (A8, A7, A2) with columns for Port, CrystalID, Protein, Images, IceRings, Comment, Score, UnitCell, Mosaicity, Rmsd, BravaisLattice, and Resolution. The bottom panel is a screenshot of the Weblce interface, showing crystal images and processing tools.

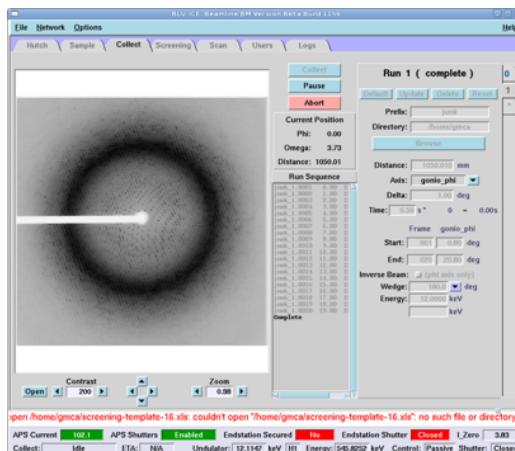
Port	CrystalID	Protein	Images	IceRings	Comment	Score	UnitCell	Mosaicity	Rmsd	BravaisLattice	Resolution
A8	A8	myo	A8_003.img A8_004.img	0 0	myoglobin, 9.5% xylitol, 9.5% glucose	0.669	90.25 90.25 45.35 90.00	0.75°	0.095 mm	P3,P312,P321,P6,P622	1.38 Å
A7	A7	myo	A7_003.img A7_004.img	3 3	myoglobin, 9.5% xylitol, 9.5% glucose	0.739	90.34 90.34 45.33 90.00	0.70°	0.057 mm	P3,P312,P321,P6,P622	1.34 Å
A2	A2	myo	A2_003.img A2_004.img	0 0	Collected several data sets on 7-1 - probably trashed, myoglobin, sucrose cryo	0.825	90.15 90.15 45.14 90.00	0.08°	0.086 mm	P3,P312,P321,P6,P622	1.28 Å

- Weblce is a web-based viewer for experiment results which also performs sample scoring
- Integration automatically sends experiment results to Weblce for processing

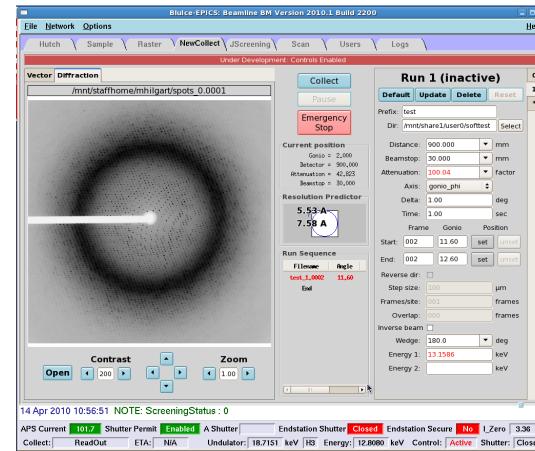


Tcl/Java integration

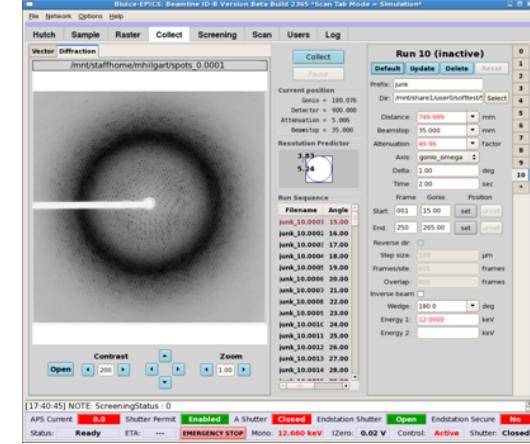
100% Tcl: 2008-1



During conversion



100% Java: 2010-2



Tcl tabs and shell

Some Java tabs,
Tcl shell

Java tabs and shell

- Raster was an initial test
- 1-3 tabs were converted per run
- TkXext embedded Java windows in Tcl
- Named pipes were used for sending commands between processes



Release

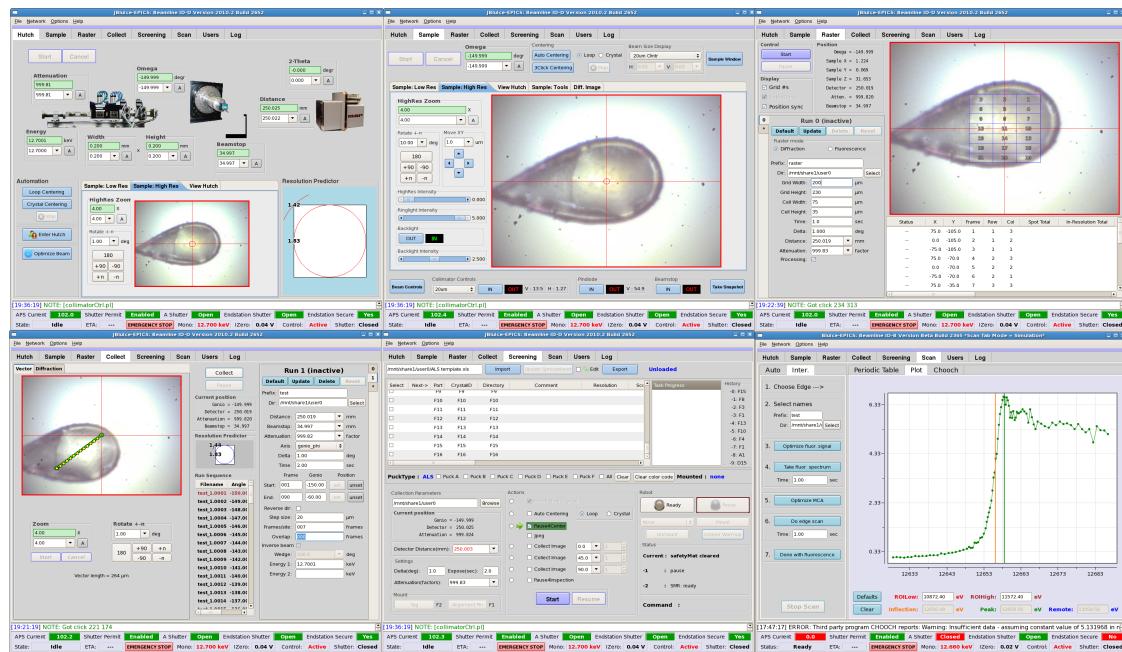
- 2.5 years of development
 - Raster: 2008-2
 - Fl. raster: 2009-2
 - Screening: 2009-3
 - Collect: 2010-1
 - Hutch, Sample, Scan: 2010-2
- Testing
 - Fully functional 4 weeks before users
 - Change cutoff 2 weeks before users
- Smooth release
 - No critical bugs reported in run 2010-2



Conclusion

- Users' demands guide development
 - Familiar interface
 - Throughout development, interface is kept the same
 - Rapid, reliable feature development
 - Architecture has been streamlined to enable fast development and high reliability
- Future plans: concentrate on features





JBlulce-EPICS

JBlulce Developers

Mark Hilgart
Sudhir Pothineni

EPICS Developers

Sergey Stepanov
Oleg Makarov

Design Suggestions and Testing

Craig Ogata
Ruslan Sanishvili
Michael Becker
Nagarajan Venugopalan
Derek Yoder

Management

Janet Smith
Robert Fischetti

Mark Hilgart - JBlulce-EPICS - October 11, 2010

