



(<https://www.eclipse.org/>)

- Configure your **account time zone setting** (</user/3563/edit?destination=proposals/eclipse-advanced-visualization-project#edit-timezone>).
- This proposal is in the Draft state. To open for public review, please send a note to **The Eclipse Management Organization** (<mailto:emo@eclipse.org?subject=Please%20open%20Eclipse%20Advanced%20Visualization%20Project%20for%20pbody=Link%3A%20https%3A%2F%2Fprojects.eclipse.org%2Fproposals%2Feclipse-advanced-visualization-project>)

- Committer Matthew Gerring must **create** (https://dev.eclipse.org/site_login/createaccount.php) an Eclipse Foundation Account.
- Committer Marcel Austenfeld must **create** (https://dev.eclipse.org/site_login/createaccount.php) an Eclipse Foundation Account.

Eclipse Advanced Visualization Project

Overview (</proposals/eclipse-advanced-visualization-project>)

Edit (</proposals/eclipse-advanced-visualization-project/edit>)

BASICS

This proposal is in the Project Proposal Phase (as defined in the **Eclipse Development Process** (http://www.eclipse.org/projects/dev_process/development_process.php)) and is written to declare its intent and scope. We solicit additional participation and input from the community. Please add your feedback in the comments section.

Parent Project:
Technology Project (</projects/technology>)

Background:

The Eclipse Community continues to grow in new ways in interesting areas, including through the formation of a suite of Working Groups that focus on everything from embedded systems to the advanced location aware technologies. Talks at both FOSS4GNA and the EclipseCons in recent years have revealed a startling amount of visualization technologies as part of this growth. Most recently, cross-working group discussions at EclipseCon NA uncovered a desire to formalize development around some of these activities into a new Eclipse project dedicated to Visualization. Further discussions across the Science, LocationTech and Internet of Things (IoT) Working Groups have helped refine the scope and identify the multi-institutional team that will work on the project.

Scope:

The scope of this project will include, but may not be limited to, the following areas:

- 1D & 2D Plotting

- Advanced 2D and 3D visualization for data analysis and post-processing
- Time series visualizations in multiple dimensions
- Constructive modeling tools for building 3D geometries and meshes
- Constructive modeling tools for building molecules and materials models
- Domain-specific scientific visualizations
- Imaging

Many in the community use SWT-XY-GRAPH from Eclipse Nebula for 1D & 2D plotting, but several teams have extensions and other features that either enhance or replace it and there is dedicated development in low-dimension plotting on several existing Eclipse RCP-based projects.

Description:

Visualization is a critical part of science and engineering projects and has roles in both setting up problems and post-processing results. The input or "construction" side can include things like constructing 3D geometries or volume meshes of physical space and the post-processing side can include everything from visualizing those geometries and meshes to plotting results to analyzing images to visualizing real data to almost everything else imaginable. There are numerous technologies for performing these tasks and most of them, with the exception of SWT-XY-GRAPH, are unavailable natively in the Eclipse ecosystem.

This work proposes to develop new visualization technologies based on the needs of projects in the working groups and to provide a framework for integrating these and third-party visualization technologies seamlessly with the workbench. The integration framework, which is part of the initial contribution that will be moved from the Eclipse Integrated Computational Environment (ICE), uses pluggable OSGi services to standardize the way that all integrated visualization products interact with the platform. Several abstractions are made for common elements such as plots and construction canvases and editable properties are provided by the implementations. Each visualization service realizes a standard interface (IVizService) that provides factory methods for constructing the widgets.

This work also proposes to develop a standalone, executable Eclipse product (flavor), complete with its own perspective and views, that can be downloaded by users and in which all of the integrated capabilities may be used in a pure visualization context. This insures that the project will be able to stand as its own product without being a kind of hidden project that is integrated into other projects.

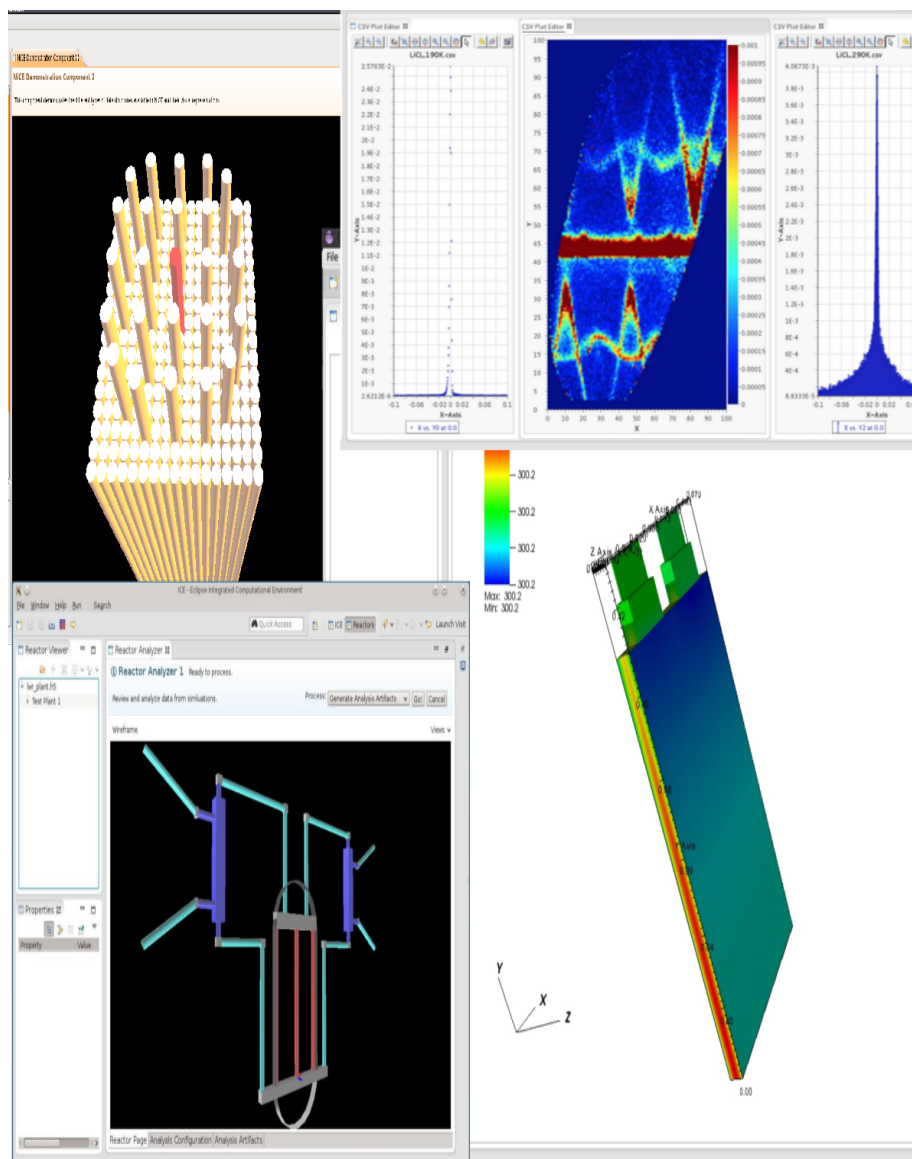
Development of Swing/AWT Widget

The development of Swing/AWT widgets is beyond the scope of this work, although there is nothing in the design that fundamentally prevents such a development. Contributions from the community that extend the platform to support Swing/AWT widgets are welcome but will not be pursued.

Examples

The picture below shows some of the proposed capabilities that would be part of this project. The collage shows a 3D constructive solid geometry created interactively in the top left, a domain-specific view of a nuclear reactor plant in the bottom left, 2D plots of neutron scattering data in the top right and a 3D view of a prismatic cell battery, commonly used in phones and laptops, with its temperature as a function of time color mapped onto it in the bottom right.

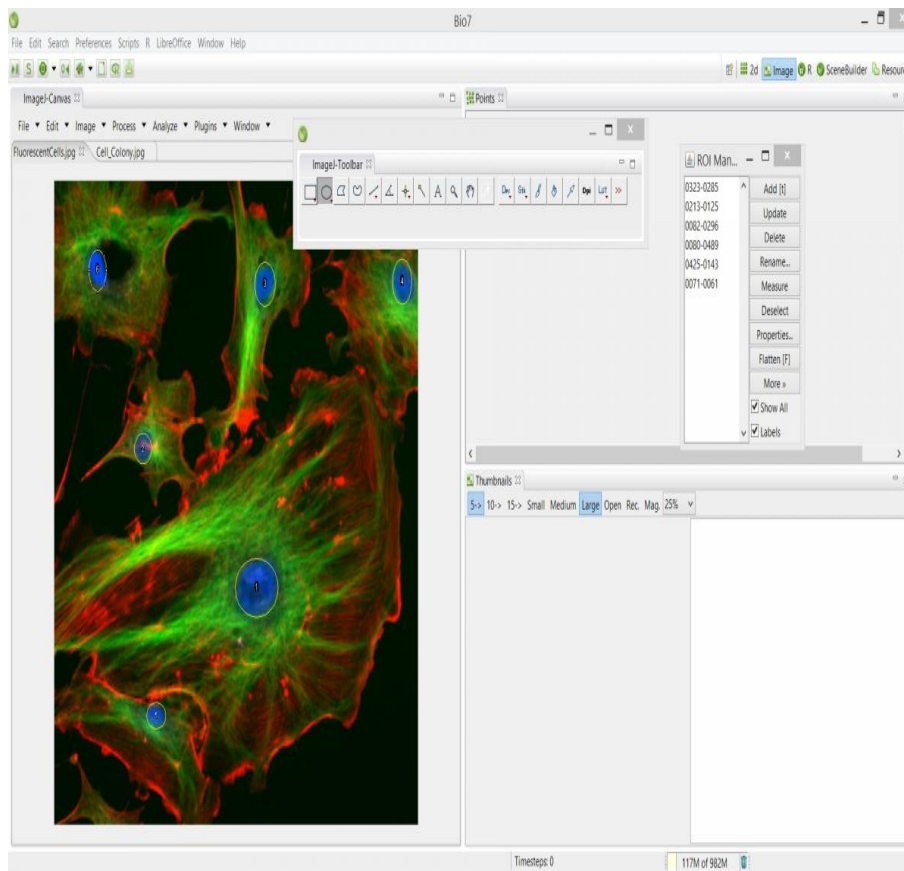
The plot of neutron scattering data is rendered using SWT-XY-GRAPH, but all of the other images are provided by third party services that are integrated with the visualization service framework. The view of the battery is actually rendered by VisIt, a third party tool for 3D visualization and post-processing that is written in C/C++ and executes as a separate process.



The next picture shows some of the proposed imaging capabilities in the initial contribution, courtesy of Marcel Austenfeld from his Bio7 project (<http://bio7.org/>). It shows a standard United States National Institutes of Health sample image for ImageJ, specifically, according to their website:

"This image is made from a Molecular Probes demo slide:

*Cells: bovine pulmonary artery endothelial cells
 Blue: nucleus stained with DAPI
 Green: Tubulin stained with Bodipy FL goat anti-mouse IgG
 Red: F-Actin stained with Texas Red X-Phalloidin"*



Why Here?:

This project was identified by members of the Eclipse Community, specifically the working groups, as a project that would give us a common area to work and allow us the opportunity to benefit from significant re-use and shared development of visualization technologies.

This project will also expand Eclipse support into a new area: Scientific Visualization.

Licenses:

Eclipse Distribution License 1.0 (BSD) (/content/eclipse-distribution-license-1.0-bsd)

Eclipse Public License 1.0 (/content/eclipse-public-license-1.0)

Legal Issues:

As far as the proposers know, no one owns the trademark to the project name.

Many, if not all, of the third-party dependencies in the initial contribution have already received preliminary approval in CQs filed by existing Eclipse projects.

We have selected both EDL and EPL for licenses because technology such as this will most likely be in demand by many scientific institutions, some of which may not want to use it under the terms of the EPL. The project lead is willing to provide a justification letter for the IP-Team, Executive Director and the Board if required. For its part of the initial contribution, ORNL approves of the dual licensing.

Project Scheduling:

Since most of the third-party libraries on which this project depends have already been approved, piggyback CQs can be used and only a minimal amount of additional IP work is necessary. The initial contribution is already working in two existing projects, suggesting that only a minimal amount of code development will be required. Thus we expect that a relatively quick incubation of six months will be required for the first release. The proposed project schedule is as follows:

- 6 Months - 1.0 release to Eclipse project site

- 6-9 Months - First talk or demo at EclipseCon
- 1 Year - First look at and prototypes of 2.0 release from requirements based on community feedback
- 1 Year - First peer-reviewed publication

Future Work:

The first twelve months will focus on deployment of the existing technologies and prototyping of new capabilities based on community feedback. New functionality includes:

- ImageJ2 support
- 3D mesh editing
- 3D molecule builder/editor
- Additional implementations of visualization services (IVizServices) for existing functionality
- Functionality required by LocationTech and IoT (since most of the proposed is based on the Science Working Group)
- Improved User Experience
- Integration of IDataset from DAWNSci where appropriate

This project will have broad applications in the general community, so in addition to proposing tutorials and talks at the EclipseCons we will build community by reaching out to the traditional visualization community and conferences as well as submit a peer-reviewed journal article.

PEOPLE**Project Leads:**

Jay Billings (/users/jay-billings-oak-ridge-national-laboratory)

Committers:

Alex McCaskey (/users/alex-mccaskey)

Robert Smith (/users/robert-smith)

Matthew Gerring

Joel Odgen

Marcel Austenfeld

Hari Krishnan (/users/hkrishnanh10)

Interested Parties:

This project is of interest to the Science Working Group, LocationTech and IoT and may be of interest to others in the Eclipse community, either users or developers.

The United States Department of Energy (DOE) will have a strong interest in the development of this project as the ORNL portion of the initial contribution is directly used and funded by several DOE Offices.

SOURCE CODE**Initial Contribution:**

The initial contribution will be primarily based on the existing source code in the Eclipse ICE project in the org.eclipse.ice.viz.* and org.eclipse.ice.clients.widgets.rcp bundles. These bundles provide the core capabilities required for 3D constructive solid geometry visualization, mesh visualization, post-processing visualization and the infrastructure described above for managing the services. They include initial service interface implementations for CSV plots, VisIt and ParaView. The copyright for this code is owned by UT-Battelle, LLC, which is a Solutions Member of the Eclipse Foundation, and released under the EPL as part of Eclipse ICE, which is still in incubation.

Marcel Austenfeld is also planning to contribute his ImageJ plugins from Bio7 to provide imaging support.

The initial contribution is already used by several hundred people around the world and includes plugins for popular visualization tools that will continue to increase its adoption over time.

Third-party libraries and their licenses:

Visit (Works With) - New BSD

Paraview (Works With) - New BSD

Visit Java Client (Requires) - New BSD

JMonkeyEngine3 (Requires) - BSD - The org.eclipse.ice.client.widgets.rcp bundle currently depends on JMonkeyEngine3, which was rejected in the CQ process. The team is working to replace the capability with JavaFX-based technologies and by the time of the initial contribution this requirement will no longer apply.

ImageJ - 2 Clause (Simplified) BSD

"Works-With" Dependencies will be filed as required.

Source Repository Type:

GitHub (/source-repository-type/github)

 Add new comment

Your name Jay Billings [O... (/users/jay-billings-oak-ridge-national-laboratory)]

Subject

Comment *

[Switch to plain text editor](#)

More information about text formats (/filter/tips)

Text format

Full HTML

- Lines and paragraphs break automatically.

SAVE

PREVIEW

Copyright © 2015 The Eclipse Foundation. All Rights Reserved.