

## IVizService Core Infrastructure

### *Related Packages*

- bundle: org.eclipse.ice.client.widgets
  - org.eclipse.ice.client.widgets.viz.service
    - Includes the IPlot, IVizService, and IVizServiceFactory interfaces.
- bundle: org.eclipse.ice.viz.service
  - org.eclipse.ice.viz.service
    - The default IVizServiceFactory implementation (BasicVizServiceFactory).
    - Convenient abstract classes for viz services and plots.
    - Includes the Visualization preference page/node.
  - org.eclipse.ice.viz.service.connections
    - Convenient abstract classes for viz services and plots that use remote connections.
    - A TableComponent for connection preferences and custom Entries to make configuring the table easier.
  - org.eclipse.ice.viz.service.preferences
    - Classes that support JFace preference pages and drawing the connection preference TableComponent inside a JFace TableViewer.
- bundle: org.eclipse.ice.viz.service.test
  - org.eclipse.ice.viz.service.connections.test
  - org.eclipse.ice.viz.service.test

### *Current Functionality*

- The ICEResourceView listens for the currently active ICEFormEditor (e.g., a Model Builder or Launcher) with an ICEResourcePage. Double-clicking VizResources in the ICEResourceView draws them in the ICEResourcePage with the first IVizService that can create an IPlot from the resource.
- There is a built-in ToolBar for managing the ICEResourcePage's plot grid.
- There is a built-in context Menu for each IPlot's Composite. This context menu can be extended in the IPlot implementation.
- The packages listed above include many helpful classes for drawing the same IPlot in multiple parent Composites, for IVizServices that use local/remote connections, and for setting up a custom preference page/node.

### *Bugs*

- 457388 - Preference enhancements for connection-based viz service (e.g., VisIt)
  - Only one connection at a time is supported. Ideally, multiple connections should be configured, and any given plot should use whatever connection the user chooses and should default to a reasonable connection (e.g., one that is already running on the same server).
  - Connection preferences should rely on PTP as much as possible.

- This behavior should be implemented in ConnectionManager (org.eclipse.ice.viz.service.connections).
- The ICEResourcePage needs to handle plots more creatively. For instance, a VisItPlot cannot be created until a VisItVizService with a valid connection is available. Ideally, the following scenario would produce the following results:
  - There is a VizResource for a mesh file...
  - The VisItVizService and ParaViewVizService are running, but have no established connections available...
  - The ICEResourcePage should print a message with links to the VisItVizService and ParaViewVizService preferences.
- 465743 - Needs more/full tests.
  - The preference-related classes lack tests. This should be addressed if possible (Eclipse/JFace preferences are used, which may complicate things).
  - Many of the classes for shared connection-based infrastructure lack tests.
  - SWTBot tests should be created where possible.

### *Feature Requests*

- When rendered in the ICEResourcePage, all available plots are enumerated in the context menu (see Set Plot Type sub-menus). Ideally, we would have a wizard or dialog akin to Eclipse's working set wizard (available plots on the left in a tree, added plots on the right in a list). such a wizard should be customizable by specific viz services. For instance, VisIt may only allow one plot type at a time, while the CSV plotter may allow multiple series (including duplicates) at the same time.
- We need to make the IVizService and IPlot implementations easier for developers not well-versed in JFace to pick up. For instance, we may want to update the abstract plot classes to:
  - handle creating the context Menus based on a provided tree of keys/action strings
  - manage a time slider widget either in the plot or a ToolBar
  - decorate the shared ToolBar in the ICEResourcePage
- There is no ability to plot remote files. Ideally, remote ICEResources should be passed to the appropriate IVizService as usual, while the connection-based IVizService should try to find the appropriate host for the remote file.
- A good way to associate file extensions with specific IVizServices.

## VisItVizService

### Related Packages

- gov.llnl.visit.swt
  - The VisIt Java client used to interface with local/remote VisIt instances.
- org.eclipse.ice.viz.service.visit
  - org.eclipse.ice.viz.service.connections.visit
    - The VisIt extension for the ConnectionAdapter (it handles dealing with the VisItSwtConnection client code).
    - VisIt-specific extensions to the connection preference TableComponent.
  - org.eclipse.ice.viz.service.visit
    - The core VisItVizService implementation.
    - VisItVizService preferences.
    - Some utilities required by either the IVizService implementation or the embedded VisIt Java client.
- org.eclipse.ice.viz.service.visit.test

### Current Functionality

- A default local or remote connection must be specified in the Visualization > VisIt preferences.
- Can open local files for view in an embedded VisItPlot. It is hooked up to the ICEResourceView and ICEResourcePage.
- Can change what is plotted through the context Menu.
- Can change the representation of the plotted features through the context Menu.

### Bugs

- 457388 - Preference enhancements:
  - We currently only support one concurrent VisIt connection.
  - As much connection info as possible should be moved to PTP.
- 457389 - The VisItVizService does not provide widgets or preferences for any of the following:
  - Timestep controls (next, previous, play, pause, first, last, etc.)
  - Plot preferences (color palette, title, etc.)
  - A Python console accessible from the ICEResourcePage.
- 464170 - Multiple VisIt “windows” with unique IDs do not work properly, so any two VisIt renders will always show the same thing.
- 459717 - No automation of port detection and limited security key generation in VisIt client. Updates will need to be synced with the VisIt components in ICE.
- 465742 - Needs more/full tests.
  - The VisItPlotTester and VisItVizService tester need to be fully implemented.
  - There are currently no SWTBot tests.
  - Other classes that might be testable: VisItPlotRender, VisItMouseManager, VisItConnectionAdapter.

- There is no mechanism by which errors in the VisIt client can be pushed through standard ICE/Eclipse error reporting mechanisms. It just silently breaks.

#### *Feature Requests*

- Widgets to control the current time of the VisIt plot. Ideally, this could be inherited from an abstract IPlot class with minimal code required. See the general IVizService feature requests.

## ParaViewVizService

### *Related Packages*

- org.eclipse.ice.dependencies
  - Must be imported from the branch org.eclipse.ice.dependencies.paraview
- bundle: com.kitware.paraview.web
  - The ParaViewWeb Java client and remote Python server(s).
- bundle: org.eclipse.ice.viz.service.paraview
  - org.eclipse.ice.viz.service.paraview.connections
    - The ParaView extension for the ConnectionAdapter (it handles dealing with the VtkWebClient code).
  - org.eclipse.ice.viz.service.paraview
    - The core ParaViewVizService implementation.
    - ParaViewVizService preferences.
    - Some utilities required by the IVizService implementation.
- bundle: org.eclipse.ice.viz.service.paraview.test

### *Current Functionality*

- A default local or remote connection must be specified in the Visualization > ParaView preferences.
- A local/remote Python server (provided in com.kitware.paraview.web) must be running.
- A file on the Python server's host can be opened and rendered (with some caveats) from the ParaViewVizService code. See the third bug point below.

### *Bugs*

- The ParaView client (and subsequently its associated VizService) requires a different JSON library. It should instead use the GSON library used elsewhere in ICE. This would eliminate the need for the org.eclipse.ice.dependencies.paraview branch.
- The ParaView Python web server script must be started manually. It should be loaded and started automatically. The user should instead need to point to the ParaView installation directory on the local/remote machine.
- Only one file type can be plotted at a time, and it is hard-coded. This is because ParaViewWeb exposes proxies for the loaded file, and each proxy's structure depends on the file type. For instance, the variables stored in an Exodus file cannot be retrieved using the same method used to retrieve variables stored in a Silo file. A viable solution must be determined.
- Needs more/full tests. So far, very little has been tested. This may have to wait until the main ParaViewVizService infrastructure can be fleshed out.

### *Feature Requests*

N/A



## CSVVizService (and CSV plotting in general)

### Related Packages

- bundle: org.eclipse.ice.viz.service
  - org.eclipse.ice.viz.service.csv
    - The core CSV IVizService implementation.
- bundle: org.eclipse.ice.viz.service.test
  - org.eclipse.ice.viz.service.test
- bundle: org.eclipse.ice.viz
  - org.eclipse.ice.viz
    - Includes some classes utilized by the CSVPlotViewer.
  - org.eclipse.ice.viz.plotviewer
    - Most of the classes required to create and use a CSVPlotViewer or CSVPlotEditor.
- bundle: org.eclipse.ice.viz.test
  - org.eclipse.ice.viz.plotviewer.test

### Current Functionality

- CSVVizService
  - CSVPlots (which load and render using the classes in org.eclipse.ice.viz.plotviewer) can be added to the plot grid inside the ICEResourcePage by double-clicking on VizResources in the ICEResourceView.
  - Each CSVPlot's context menu additionally provides the ability to add more series to the clicked plot, to remove a specific plotted series, or to clear the series entirely.
  - Of course, this includes all of the built-in functionality provided by the CSVPlotEditor (customizing titles, axes, gridlines, series, etc. via the SWTXYGraph toolbar).
- org.eclipse.ice.viz.plotviewer (The core CSV classes used in the viz perspective)
  - CSV files with the appropriate format can be loaded (this is a bug, as CSV files do not really have a fixed "standard" format... see the first bug point below).
  - Customization of the titles, axes, grids, etc. available through SWTXYGraph toolbar.
  - File sets can be selected to provide time series data to the plot. Timesteps can be cycled via a slider or spinner at the bottom of the rendered plot.

### Bugs

- The format of each file is expected to match the following:
  - The file must have a consistent number of columns in each row, e.g., the delimited cells should form an  $m \times n$  matrix.
  - Delimiters are restricted to commas, when in practice other delimiters (colons, semicolons, tabs) can be used (this may be a feature request rather than a bug).

- Colons, semicolons, and forward slashes are replaced with commas when the line format matches the regex `#\s*\w+\s*([:;/.]+)` (e.g., `#label1:label2:label3`)
- There are lots of special formatting cases in `CSVDataLoader.load(File)`.
- Adding/removing plots causes a complete redraw. This resets all plot customizations, including the plot title, axes, gridlines, annotations, etc. This may also prove inefficient, slow, or buggy when data updated in real time is streamed through a `CSVPlot`.
- Updating what is shown in the `CSVPlotEditor` uses three consecutive dialogs. This is simply too much. Ideally, the solution to this bug would be the same one used for the feature request for a plot series selection wizard/dialog.
- 457764 - Adding a “fileset” causes the `VizFileViewer` (package and bundle: `org.eclipse.ice.viz`) to throw an NPE.
- 465741 - Needs more/full tests.
  - The `CSVPlot` and `CSVVizService` tests should be reviewed and updated, as they are somewhat limited in scope. This includes testing that a plot can correctly load and draw a CSV file. `SWTBot` tests need to be as complete as possible.

### *Feature Requests*

- When rendered in the `ICEResourcePage`, all available combinations of plots are enumerated in the context menu (see `Set Plot Type` and `Add Series` sub-menus). The list will be of  $n \times n$  size for  $n$  data variables. There is a feature request to address this more abstractly in the viz service infrastructure.
- It should be easy to feed a given `CSVPlot` pre-defined properties (what is plotted, plot customizations, etc.). It should also be easy to update the data and refresh the plot.
- It might be useful to have a way to open remote files for `CSVPlots`.
- File sets are a little confusing. It would be cool if the file set could be configured via a wizard or dialog so that the order of the file set can be customized. An ill-formatted file in the set also shouldn't break anything.
- If a developer wants to have a CSV file with a really funky structure, we may want to provide them with an API to handle loading the CSV file into an `IDataProvider`.