Browsing the source, where to start

Project overview

The following projects are located in the SVN repository:

Project name	Description
com.servoy.eclipse.appserver	Eclipse plugin to start the Servoy Application Server in Servoy Developer
com.servoy.eclipse.core	Core eclipse classes that together make up Servoy Developer
com.servoy.eclipse.debug	Classes that revolve around debugging, connectors to DLTK
com.servoy.eclipse.designer	All Form designer code
com.servoy.eclipse.exporter	The workspace exporter
com.servoy.eclipse.feature	Holds the definition to makeup the Eclipse Servoy Developer
com.servoy.eclipse.jsunit	JSunit <-> Junit bridge and command line JS unit test/suite runner
com.servoy.eclipse.model	The workspace model
com.servoy.eclipse.profiler	The profiler (view) which is present in Servoy Developer
com.servoy.eclipse.team	The Servoy Team Provider, used against the server/repository interfaces
com.servoy.eclipse.ui	GUI helper classes, some default dialogs, abstract GUI elements
com.servoy.extensions	Contains the default shipped client plugins and beans, shipped in installer
servoy_debug	client side debug classes and connectors
servoy_headless_client	The Headless and Web Client code
servoy_smart_client	The Smart Client (webstart)
servoy_shared	The shared code / libs between Web, Headless and Smart Client (and server interfaces)
org.eclipse.dltk.javascript.rhino	The Rhino engine, contains couple of small modifications for DLTK/Debug hooks and some improvements
eclipse_target	Holds all the needed eclipse dependency jars

High level overview of client code

Type Hierarchy view

By selecting a class in the Package Explorer in Eclipse, the Type Hierachy of that class can be shown by pressing F4.

Locating Types

By pressing Control-Shift-T, a dialog pops up that allows quick lookup of classes based on their name

Call Hierargy

By pressing Control-Alt-H on a function shows where in the code the function is used

The entry point for all Clients is the ClientState class. The ClientState class is in fact the most top level class containing logic. From it, all other "client" applications are derived. It holds many object managers (all implementing the IManager interface), for example:

- FoundsetManager: class managing all IFoundset instances (datahandling layer). IFoundSet has multiple implementations from which the FoundSet class is the main one. It contains records (IRecord class) and records contains rows (Row class). The FoundSetManager also keeps track of the RowManagers for each database table to handle Row objects.
- PluginManager: class managing all plugins (a plugin is an external piece of code which has to satisfy to the IPlugin.java interface and can be used in scripting)
- FormManager: class managing all FormController instances. Subclasses are WebFormManger and SwingFormManager
- ModeManager: class to control the mode the application is in, for example edit, find and print
- BeanManager: class to create/control the beans and their lifecycle
- A reference to the script engine (Rhino)

The MVC pattern is used in Servoy when it comes to Forms:

- Model: IFoundSet class - defines the db data in form, by the select SQL it contains

- View: SwingForm class for Smart Client or WebForm class for Web Client - handles the UI
- Controller: FormController class - binds model and view and controls them

Smart Client

The J2DBClient class is the Java Webstart client. It generates the Swing user interface through the ComponentFactory and ItemFactory classes.

- Contains a Java main method in the J2DBClient class
- LAFManager: class mainly used in Smart Client only to control appearance
- CmdManager: class for binding all GUI actions

Headless Client

The SessionClient class is the Headless Client

Web Client

The WebClient class is the apache-wicket browser-based client. It extends SessionClient. It generates the user interface through the ComponentFactory and ItemFactory classes, coupled via TemplateGenerator to produces plain html templates.

- Is created via the SolutionLoader class