Extending Servoy

Extensions

Servoy can be extended in various ways:

1. Plugins
2. Beans
3. Custom Java code
4. Server WAR file deployment
5. Developer eclipse plugins

This allows for use of nearly any existing Java functionality inside Servoy, even integrate with native applications such as web-browsers.

Writing extensions requires understanding of Java.

Plugins

Plugins can be written for the Server, SmartClient and WebClient, see IServerPlugin, ISmartClientPlugin and IClientPlugin resp. in the Public Java API documentation.

A plugin is capable to expose Java methods as JavaScript functions, which can seen/used by Servoy developers via the Solution Navigator.

The plugin packing mechanism allows to specify all needed resources/libs for remote loading by SmartClient in a Java WebStart jnlp file.

Many plugins shipped with servoy are opensource and even shipping with Java source inside, see for example the mail.jar plugin file in default install 'plugins' directory.

See also:

- Creating Client Plugins
- Providing Converters and Validators from Plugins
- Providing UI Converters from Plugins

Beans

Servoy allows usage of regular Java Beans in SmartClient, to utilize some databinding they can implement IServoyAwareBean (or subclasses) see API docs.

By default all Java methods on a bean are exposed as JavaScript functions, which can seen/used by Servoy developers via the Solution Navigator.

For loading additional libs a bean can specify all libs in the manifest.

To make beans for both Web and SmartClient the IServoyBeanFactory can be utilized, see the source of the DBTreeView bean (which is shipping with Servoy)

Public Java API Docs

For information on the usable Servoy API, see the Servoy API documentation online.

External Docs

Checkout the comprehensive overview for building plugins and beans for Servoy.

Existing Extensions

Besides default extensions shipping with Servoy, many Third-party extensions are available at ServoyForge.

Custom Java Code

In Smart and WebClient its possible to call any static Java method, like

```
var mytime = Packages.java.lang.System.currentTimeMillis();
```

Server WAR File Deployment

Since Servoy incorporates Apache Tomcat its possible to deploy .war (Web Application Resource) in application_server/server/webapps.

These .war files can also be uploaded via the Servoy admin page via "Upload Library"

When a war requires database access, it's possible to get a database as datasource via JNDI, all the database servers are exposed in the global JNDI scope as "jdbc/dbservername" and should be accessed in current context via a <resource-ref>, see Tomcat manual for more details.

Developer Eclipse Plugins

Servoy developer is another plugin ontop of the the eclipse.org framework.

Its possible to install many other plugins like SVN, GIT, SQLExplorer etc, via "Help" > "Install new software" > enter an eclipse update site url.