8.2 Whats new

What's new 8.2

Developer

Web Package Manager now supports Servoy Modules installation (Like svyUtils)

Web Package Manager now supports a 3rd party index so that other companies can have there own index file with components/layouts/services/modules

Export solution in the developer now have options to directly export to a server using the new solution REST api.

Place field wizard enhancements, now you can place directly specific components or templates and define how they are placed together (label on top, spacing between label and fields and between fields)

Form Hierarchy viewer, so you can see the full super/sub structure of a form and where is what implemented and defined.

No-Resource project, modules/solutions can now be defined without a resource project, especially easy for utility modules

Export .servoy file now has the components inside it, this is purely for developer to developer, is not used at the server.

Import .servoy file in the developer now gives the option to directly create the missing database

inmem table definitions now support all settings of normal columns (column properties support like title text or certain pk types)

Logical Forms (forms with no ui) can be extended now to become a absolute or a responsive form, so now you can have shared code between an absolute or responsive form.

Runtime

In Memory datasources have an onload event (in the table events tab) where you can attach a method (entity or global) that is called when the in mem is touched so that you can fill the datasource
so it is now more a pull stuff then a push that you needed to do at some point (like on solution load)

Named Foundset support, In the designer you can give a datasource of a form a name (besides 'separate'). This creates at runtime a foundset with that name that forms with the same name will reuse
also datasources.db.table.getFoundset("name") will return that specific instance. You can see this as a named (shared) separate

Client Manager plugin: A user plugin, has methods that the the maintenance plugin has but now usable in a normal solution.
Also has broadcaster functionality so clients can chat/talk to each other

Component attributes, for ngclient you can add html dom attributes to any component so that you can have your own directives that enhances the standard component (like a key listener or mouse hover)

Angular and all the support libs upgraded to 1.6.3, now a 3th party angular support lib will be overwritten by that version.

ngutilsplugin: support for adding/removing style classes for the form itself at runtime

A locale/language change at runtime will at once change all the i18n labels in the client, if the component that displays that i18n tag does listen to server side changes for that property.

i18n keys can be used in the global properties like the dateformat

lazy loading of global method valuelist (for only display valuelist without a real) if both the component and the globalmethod valuelist are configured correctly.

TinyMCE for NGClient is updated from 4.0.20 to 4.5.1

ServerSide api for webcomponents have to be no in its own category called "internalApi"
https://wiki.servoy.com/pages/viewpage.action?pageId=1869552 and https://wiki.servoy.com/pages/viewpage.action?pageId=8061131

The session timeout (https://wiki.servoy.com/display/DOCS/NG+Client+Settings) is now also configurable per client through an application.putClientProperty (APP_NG_PROPERTY.WINDOW_TIMEOUT,3600) (only after 1 hour)

The "this" object inside a calculation now points to the record.

Server

The server does have a REST api for uploading solutions: https://wiki.servoy.com/display/DOCS/Deploying+using+external+tools and https://wiki.servoy.com/display/DOCS/Exporting+a+Solution+from+Servoy+Developer

Server scaling plugin basted on AMQP (https://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol).
This is used for queuing the data broadcast messages of one application server to another so the know each other database changes.

You can now download the performance data from the admin page to a CSV file

Servoy Properties:
1> enhanced support for environment variables, so you can substitute system properties/environment variables by using tags in the servoy.properties file that are then replaced.
2> War export has 2 extra options where you can say how an existing servoy.properties file at the tomcat server should be used: Use the server one (default), override the database properties, overwrite completely.
3> Added a servoy.user.home dir property when creating the WAR, so you can define where it can store and get the servoy.properties file of the server itself.
4> For the developer you can now link a servoy.properties file to a workspace, Preferences -> Servoy - >Startup.