2020.06 Whats new

This release has many changes in our builders inside the developer.
The ScriptBuilder works a lot better now with circular references between js files, this fixes the problem that you did see errors when you saved (build) fileA.js
that then dissappears again when you are saving fileB.js
Also the ScriptBuilder now understands which js files are dependend on other js files.
This means if you touch fileA.js that is used by fileB.js then fileB.js is also checked if you saved fileA.js, so if you renamed a function that is used in another file than that is reported right away.
We tried to avoid making a lot of copies of objects, so the memory consumption should be going down for large workspaces.

All these changes in the ScriptBuilder did result in that we needed to drop the "none shallow parsing" mode that we still had as an option in the preferences. This option is removed and now the your javascript files are always build in shallow parsing mode, this means that having @param and especially @return is quite important to have (besides @constructor and @parse for functions that needs to be fully parsed anyway)

Also our ServoyBuilder is improved so that it doesn't check all the time everything on a build, but looks more what is actual changed and what do i need to check for this.
It also does need to build up a depedency tree for this to work, so 1 full build is always needed for now after a restart.

A lot of improvements are done when installing solutions/modules from the Servoy Package Mananger.
This does include that solutions now more have a version already as a property of the solution, so the exporter will now ask for this.

Improvements in loading of the tables of servers, this is done in a background now, a regression was also fixed that was introduced in 2020.03, that loaded on startup already to many tables/servers.

Palette improvements: Now components are ordered based on the category they belong to instead by package name. So components from different packages can now be shown under 1 category like "Grids"

Servoy Default (legacy) components are now depricated and not shown by default. These components are only shown when the form already has some of these components,
or in the preferences under Default Packages the checkbox "Show always default/legacy servoy components in designer (if enabled above)" is checked.

A **warning** is now generated if you use none valid names for forms or components in the solution model
The same validator that is used in the developer itself is now used to validate the name of your form or component.
So doing something like: solutionModel.newForm("form_" + application.getUUID().toString()); will result in such a warning because a UUID has '-' in its string. Make sure you replace all those chars (like '-') to '_'. Also a name can't start with a number. In certain parts of servoy using illegal names could result in weird behavior.

When showing a (Modal)FormDialog and then the user did a refresh, the dialog disappeared and the state was not fully correct anymore (server side still waits for the dialog)
This is now fixed, Servoy will reshow the dialog(s) that where shown before the refresh.

ViewFoundset/ViewRecord now do have a lot like the same api that JSFoundset/databaseManager have, vfs.getFailedRecords() and record.exception are not yet implemented.

We reintroduced databaseManager.removeDataSource(datasource) to clean up an in mem datasource if you are sure it is not used anymore. This way you can control the memory usage of a client.
If you don't call this then only when the client is killed we will also clean up all the created datasources.

Eclipse is upgraded to 4.15

Internal java is updated to Java 14

HSQLDB driver updated to 2.5.0