2021.03 Whats new

The developer is based on Eclipse 4.17 (2020.09) which requires Java 11, we ship with Java 15.0.2

NGClient2

This is the first release that also has NGClient2 shipped with it. This is a first beta release of NGClient2 that is now fully based on Angular 11 (not AngularJS).

This change has effect on all the components (and services) because these needs to be rewritten based on the Angular framework. We ported our servoy-default, servoy-bootstrapcomponents, most components of servoy-extra and some components of bootstrap-extra. This wil be an ongoing thing throughout the this release (until he final) and in other releases done this year (2021.06 and later)

Currently those Angular packages are backed in, but you do need to install the NG1 packages that you want to use of those 3 packages. Because the designer and also the script editor (code completion) is still build around those. The Servoy Component or Service Projects to make components are not suited yet for making NGClient2/Angular packages. This will be looked at in a later release (the same goes for porting also the designer over).

Some services like ngutils,dialogs and the keylistener are also ported. Fontawesome is shipped by default because the core now also uses icons from that library, this is because NGClient2 is build on Bootstrap 4.6 and Bootstrap 4 removed glyphicons.

Styling/theming can be a bit different in NGClient2 compared to NGClient1 because we are now build on top of Bootstrap 4 but also because some components are now a bit differently build up in the browser so existing selectors could not match on them. Some components are also now quite different because of the underlying library change, like the Datepicker. Also the Servoy Theme is still work in progress. (still is for a bit)

With **RC2** we added a special Servoy theme for NGClient2, the best way to start using this for testing NGClient2 but not disrupting also NGClient1 is to make a copy of your solution stylesheet, if you stylesheet is named "mysolution.less", make a copy and name that copy "mysolution_ng2.less". If you startup a NGClient2 then when serving out the stylesheet, Servoy will check if there is a "_ng2" variant of the given solution stylesheets name. In that "mysolution_ng2.less" stylesheet you can then point to another copy: @import 'custom_servoy_theme_properties_ng2.less'; you can make, and that properties less file then points to the new "2021.3.0_ng2" Servoy theme file. This way if you start the same solution in NG1 you will just be served out the normal less/css file including our ng1 servoy stylesheet through 'custom_servoy_theme_properties.less' and starting in in NG2 we will serve out another set of stylesheets and a different Servoy theme file. After this you can then start tweaking the NG2 look in the "_ng2" solution stylesheet.

You can switch to test NGClient2 by changing the URL in the browser after you started an NGClient1 /solutions/[solutionname]/index.html /solution/[solutionname]/index.html. So remove the last "s" from "solutions". You can also change what the client should launch by default in the preferences. then you can go back by adding an "s" to "solution".

The WAR Exporter in the developer and command line also do have an option to export the NGClient2 runtime inside the WAR so you can also test this in a test or staging server.

Some default stuff are not ported over at all, like the old table and listview form views. Its very likely those will never be ported over to NGClient2

We like to have feedback, especially on components or services that are already ported but are behaving differently or are not looking the same.

Other changes:

Upgraded for NGClient a few core javascript packages to the latest releases: (https://support.servoy.com/browse/SVY-15881)
1> Angular to 1.8.2
2> JQuery to 3.5.1
3> Angular translate to 2.18.4

and the html area (default and servoy extra package) upgraded TinyMCE to 5.7.0

Added support for the Content Security Policy header 'frame-ancestors' (https://support.servoy.com/browse/SVY-15870)
The default value is 'self' so it only allows embedding of NGClient on the same host. This can be changed on the admin page.
This property could only be set by using headers (not <meta> tags in the html page itself)
So we moved all CSP headers to use header tags instead of a meta tag.

Streaming files api on the File Plugin (https://support.servoy.com/browse/SVY-15658)
plugins.file.streamFile(...) is added so no need anymore to read files fully in to memory.
It can stream any file on disk that you want, you only need to explicitly say that in servoy code (give a file path)

SQL Case When support in the query builder. (https://support.servoy.com/browse/SVY-3358)

SQL keywords are now more relaxed, stuff that needed to be quoted are now defined per database (https://support.servoy.com/browse/SVY-15676)
But your are still allowed to use them. Only keywords that are also illegal to use in scripting (like "function") are still not allowed.
Besides this if a specific database needs some other names to be quoted you can provide your own now through a servoy property:
server.1.quoteList=name1,name2,name3

We changed the way what you can test against WebComponents a bit in jsunit tests (https://support.servoy.com/browse/SVY-15816)
this only really applies to the solution model part, for testing actual ui components a E2E test is needed (a browser)
This could have a side effect that the jsunit test runner now expects to have sablo and ngclient jars it its classpath.
We adjusted the software factory build script: https://github.com/Servoy/software_factory_build_scripts/commit/90853bd0e2b77f367366bb0881cfbc9e1f5d475c
So it could be that if you have your own or copied it from there you need to change this also on your end.