

NGClient Compatibility

Differences with Smart and Web Client, compatibility wise:

- onRender is not longer possible, webcomponents are outside servoy's control we cannot enforce behavior onto them but servoy 8 ships in the package 'servoyextra' an 'onrenderlabel' which mimics onrender, see below
- webdav feature for WebClient is lost due to incompatibility with Tomcat 8 server used inside Servoy
- printing a form from scripting is not possible, browser printing needs to be used or jasper reports for reports
- client design is not present (yet)
- Client side Drag'n'Drop is not supported (yet)
- beans depending on technologies such as Java Swing or Apache Wicket will not function (no image will be shown for swing beans in the browser)
- A much used plugin called webClientUtils from ServoyForge is not supported, since it depending on Apache Wicket'
- tabIndex of tabpanel is changed after onShow of the new tab form, so when switching tabs, in the onShow of the new form component still has old index

New Designer:

- Drag'n'Drop from the Solution Explorer tree into the designer (like dropping forms on a tabpanel) is not supported
- Template support is not yet supported

Small tablefilter behaviour change

The behaviour of `databaseManager.getDataSetByQuery(QBSelect, int)` changes behaviour. Before release 8.0 this call will not take table filters into account, but in v8 release the filters will be applied. The method `databaseManager.createDataSourceByQuery(QBSelect)` will use table filters, both in older versions and in the new release. Both methods have a new optional boolean argument to explicitly use or skip table filters.

Changes to StyleSheets usage

For the NGClient the limitation of only being able to use CSS properties that Servoy explicitly supports in StyleSheets has been lifted, any CSS property can now be used. Instead of preprocessing the stylesheets authored by the developer before sending it's content to the browser, the only preprocessing that now takes place is expanding syntax to browser specific syntax.

This change has the following behavior changes:

1. Stylesheets can't be set anymore on Forms, but only on Solution level
2. Only one stylesheet can be assigned on Solution level
3. Additional stylesheets can be included using the `@import` statement in the main stylesheet
4. Inherited CSS properties (see <http://stackoverflow.com/questions/5612302/which-css-styles-are-inherited>) set on a Form will also affect any Forms contained in containers (TabPanel, SplitPane, Accordion etc.)
5. When different stylesheets are included that utilize the same classnames, the CSS properties set in the classes are merges and in case of duplicates, the last one wins

For checkbox component, in order to horizontal align it from css, we no longer support text-align but have to use justify-content (flexbox system).

Onrender implementation

Because we can't have an onrender callback for all the components, Servoy introduced another way of doing an onrender, which will be most helpful in a portal/tableview like view. In the 'servoyextra' package there is a 'onrenderlabel' which can be used to change its appearance.

See: [OnrenderLabel source](#) for the implementation.

This label can display any dataprowider and have format support for that dataprowider. But besides that it has a `styleClassExpression` property which is also a dataprowider. This dataprowider can be a calculation which returns a string that is can be 1 or more class names. Those class names are defined in the solutions css (so a class that sets the background-color to red) this way you can change the appearance of a single cell in a tableview based on the record data or any relation.