

The future of app development.

18 + 19 May | Dusseldorf

The logo for the Omnis Developer Conference 2022. It features a large black circle with a pattern of smaller grey dots inside. The text 'ODC' is written in large white letters, and '2022' is written in smaller orange letters to the right of 'ODC'. Below this, the words 'Omnis Developer Conference' are written in white. The logo is surrounded by several overlapping circles in orange and blue, and a small blue dot is located below the logo.

ODC 2022
Omnis Developer Conference



What's New In The JS Client

10.2 Onwards

Browser-based Design View

True WYSIWYG



Browser-based Design View

Remote Form design windows render using the JS Client in oBrowser (Chromium).

- Design view matches **exactly** how it will look at runtime.
 - *Including custom CSS.*
- Third-party components no longer need C++ drawing code.
- JSON-defined components now render in design mode.



Visual Redesign

Inspired by Material Design guidelines



Component Appearance Updates

- Almost all comps have been redesigned to give a more modern appearance.
- Default sizes increased to better fit modern web design guidelines (e.g. to work as touch targets).
- Subtle animations & effects added.
 - *Better feedback to user actions.*

The Old

The New



Rest Of Client Updates

Other parts of the JS Client UI updated.

- Dialogs & SFSs have a new Material-like appearance.
- Shadows used to indicate depth for menus etc.
 - *Several comps have \$hasshadow property.*
- Ripple effects added when clicking many elements.
- Automatic tinting of colours in certain situations.



Color Themes

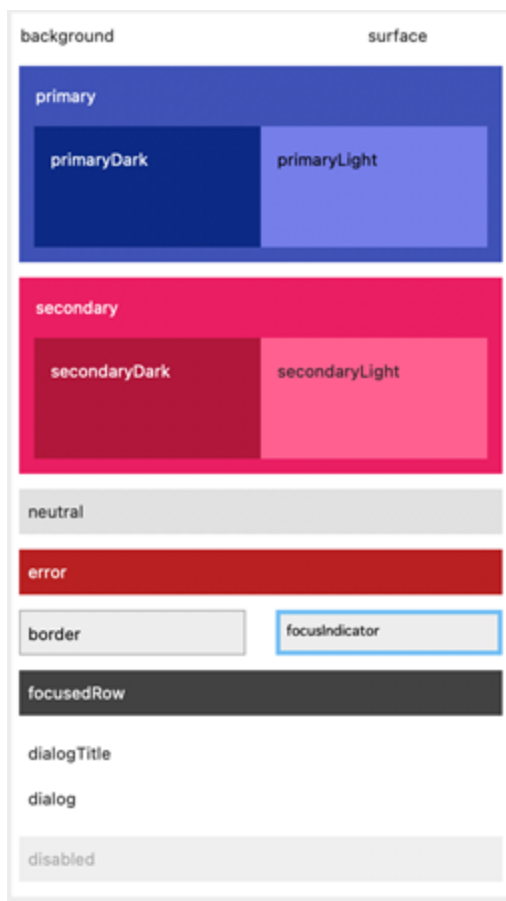


Color Themes

10.2 adds color themes to the JS Client.

Defining a palette of general-purpose color concepts.

- Primary, Secondary, Background, Error, etc...
 - *Light & Dark variants.*
- Color of text when on top of these colors.
- *kColorDefault* uses theme colors.

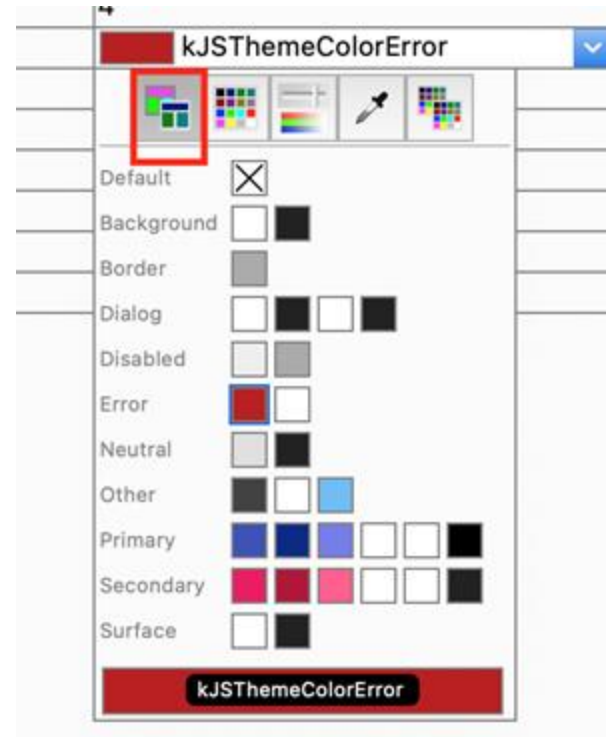


Using Theme Colors

- Property Manager color picker has new theme colors tab.
- New **kJSThemeColor...** constants to use in code.

Use wherever possible:

- Gives a clean and consistent look.
- Allows easy tweaks to colors across your app.
- Setting text colors to *kColorDefault* uses appropriate text color for theme color behind it.



Creating & Setting Themes

Create custom themes with the theme editor:

Tools > Add-Ons > Web Client Tools > JavaScript Theme Editor

Default theme is `$root.$prefs.$javascripttheme`.

View > JavaScript Theme provides UI to switch default theme.



Changing client Theme

Client's **initial** theme defined in the initial .htm file used as app entry point.

To change theme dynamically, use the '**settheme**' \$clientcommand:

- Requires a reload of the web page to take effect.
- Only affects **this** client's browser.
 - *Saved to localStorage.*



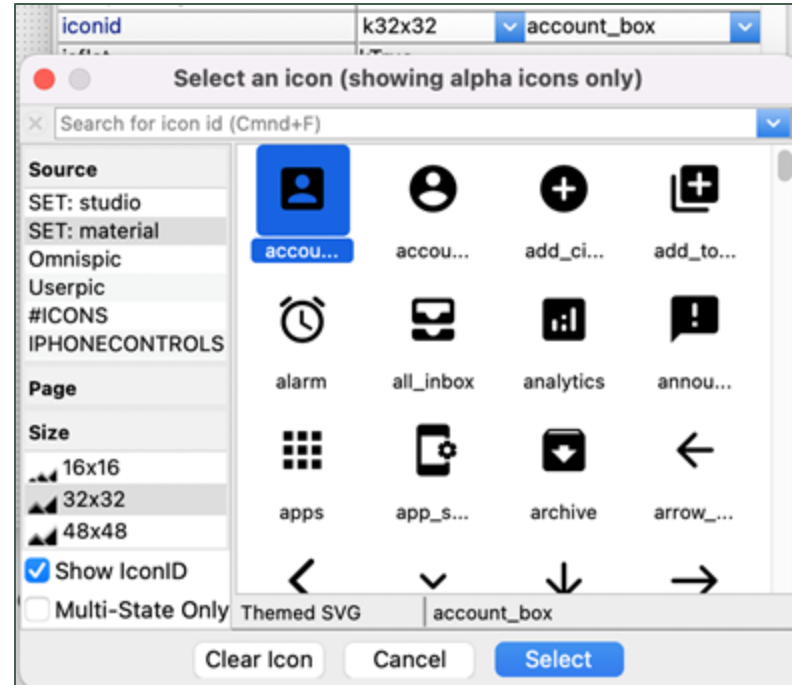
SVG Icons



SVG Icons

- SVG images are vector-based so scale to **any size**.
 - *Replace multiple PNG images with one SVG.*
- Supported in Omnis' **Icon Sets** on the JS Client.
 - Use a **name** rather than integer to refer to their \$iconid.
 - Simpler to manage.

Can be tinted with colours.



'Themeable' SVGs

Omnis has a tool to convert SVGs to a custom format.

Tools > Add-Ons > Web Client Tools > SVG Theme

- Allows us to apply a custom color to the image.
- May not work automatically for all SVGs.
 - *Allows the **fill** and/or **stroke** color to be overridden.*
 - *Allows manual edits to fine-tune.*

When used in a control, typically takes on the control's **text color**.



Tab order



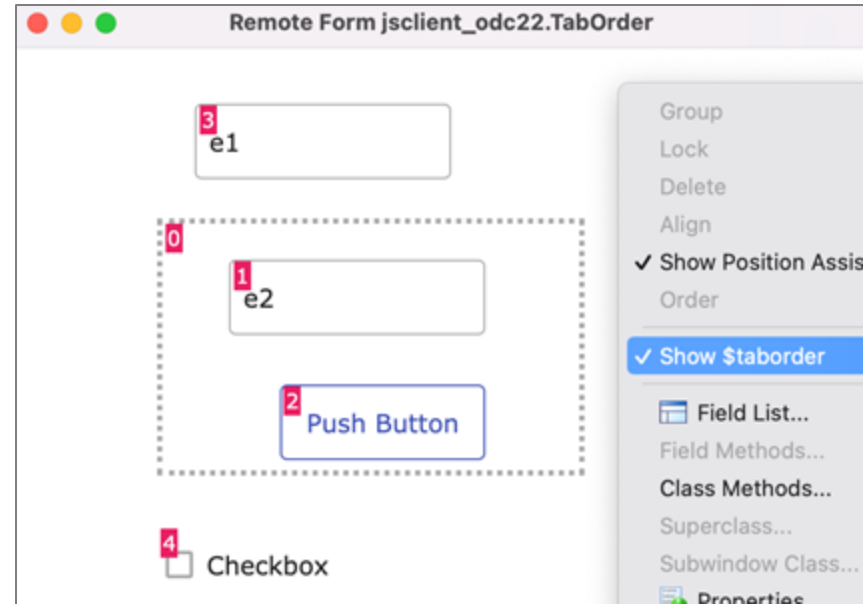
\$taborder

Runtime tab order in the JS Client is governed by the DOM.

- Omnis' **\$order** can only apply locally within the **current container**.
- Can be difficult to work out what the tab order will *actually* be.
 - (Affected by, \$order, \$active, containers, focusable etc)

JS Comps now have read-only **\$taborder** property at design time.

- Use context menu's "**Show \$taborder**" option to see all fields' \$taborder at a glance.



\$nexttabobject

All JS comps have a **\$nexttabobject** property.

- Allows fine-grain control of tab order.
- Lets you specify the name of the control that should be tabbed to after this one.
- Overrides the tab order set by \$order.
- Don't overuse this property, as it does incur some overhead by setting up additional event listeners.



Split Button Control

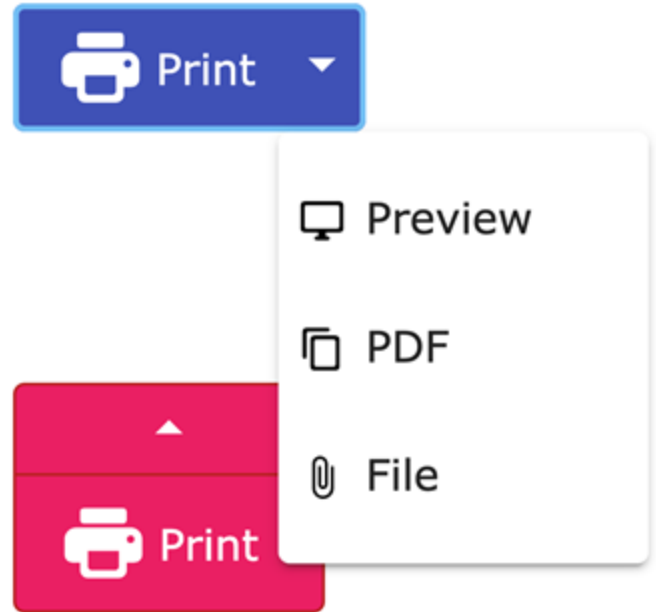


Split Button

Combines a button with a dropdown menu.

- The larger 'button' area for a **primary** action.
- A smaller dropdown section opens a menu of **secondary** actions.
 - *Defined by RemoteMenu class.*
- Dropdown section can be positioned at any orientation.

(Also available in the Fat Client)



Edit Field Input Masks



Input Masks

Gives control over what a user can enter into Edit fields, and its format.

- Specify an **\$inputmask** - e.g. "AA/AA/####"
- Read **\$maskvaluevalid** to check if current mask is filled.
- Client-executed **\$processmask** method
 - *Allows you to dynamically change the mask as the input changes.*
 - *Useful for more complex masks.*



Subform Sets



Subform Sets

New Convenience Methods

The notation required to open Subform Sets has always been complicated (to give the full flexibility they can offer).

But often you just want to open a single modal dialog...

New \$clientcommands:

- **subformdialogshow**
- **subformdialogclose**



subformdialogshow

Parameter row(classname, params, title, width, height, [closeButton, resizable, maxButton, openMax]).

```
Do $cinst.$clientcommand("subformset_add",row("SubformSet","kSFSflagCloseButton+kSFSflagResize","",IFormList))
Do $cinst.$clientcommand("subformset_formadd",row("SubformSet",1001,"jsSub","0","My SFS",kSFScenter,kSFScenter,300,250,1))
```

Becomes:

```
Do $cinst.$clientcommand("subformdialogclose",row("jsSub","1","My SFS",300,250))
```

To close top dialog:

```
Do $cinst.$clientcommand("subformdialoghide")
```



Per-breakpoint Dimensions

For Responsive forms, when adding a form to a Subform Set, you can specify position and dimensions for **multiple breakpoints**.

The **subformset_formadd** (& **subformset_add**'s form list) \$clientcommand accepts a list:

Breakpoint	Left	Top	Width	Height
320	kSFSCenter	kSFSCenter	300	250
700	100	100	500	400

The row with the largest Breakpoint \leq the container form's **current breakpoint** will be used.



Datagrid Improvements

Studio 10.22



Footer Row

Datagrid has a new **\$hasfooter** property.

- Adds always-visible footer row with configurable values for each column.
- Automatically show *total*, *average* or *custom* values.
- **\$footercolumnlabel** allows a text prefix before the value.
- **\$updatefooterrow** method allows complete control over the footer value.
 - *Called whenever data changes, passing **pColumnNumber** - You return footer value.*
- **evFooterClicked** & **evFooterUpdated** events added.
- **\$footercolumnvalue** can be directly assigned/read.

Product	Quantity	Price (each)	Total
Apple	5	€ 0.39	€ 1.95
Banana	7	€ 0.25	€ 1.75
Grapes	2	€ 1.79	€ 3.58
Orange	3	€ 0.35	€ 1.05
Totals	24	(Avg) € 0.83	€ 14.78



Dynamic Filters

Datagrid filters can now be assigned/read from code, using the **\$::filters** property.

- A single-column List of **Rows** with format:

colNumber	colName	filterType	value
1	“username”	kJSDataGridFilterContains	“andr”
2	“active”	kJSDataGridFilterEquals	kTrue

- **evFilterUpdated**: Fired when filters change.
 - **pFilters** - List of rows, matching above format.

If \$multifilters is false, works the same, and the list has one row.



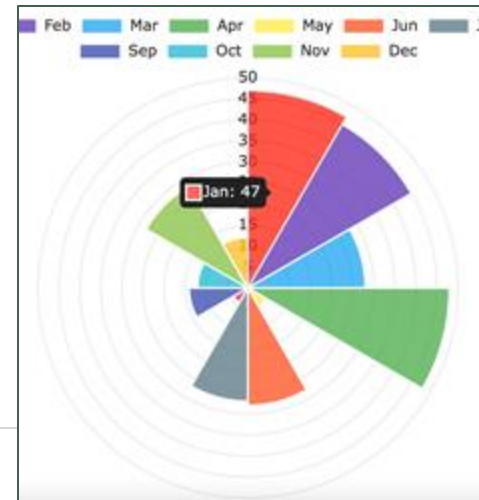
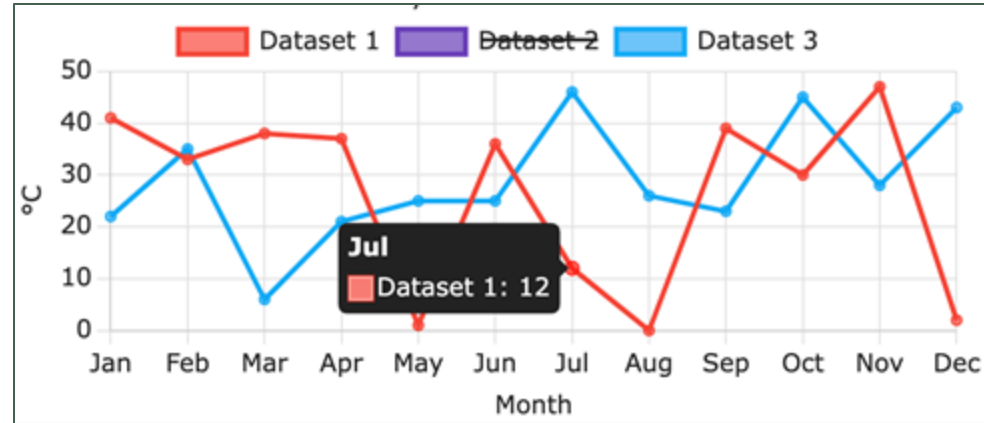
Coming In Studio 11

New Controls

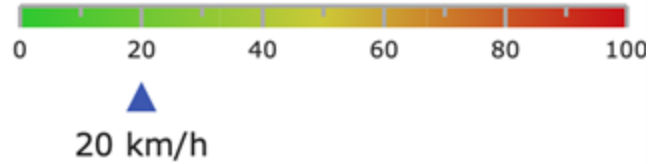


Charts Control

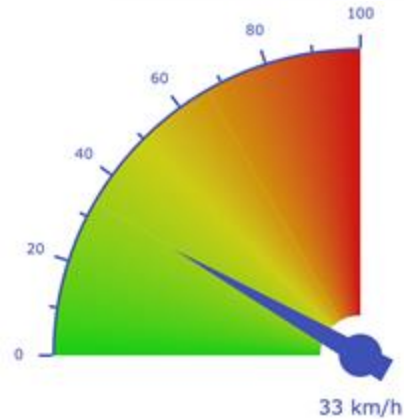
- New chart control based on open source [ChartJS](#) library.
- **Bar, Bubble, Doughnut, Line, Pie, Polar Area, Radar, Scatter** types.
 - *Deprecates old Bar & Pie chart controls.*
- Animations and more visually appealing.
- Simpler list data structures for \$dataname.



Gauge Control



- Way of visualising numerical data.
- Circular or horizontal/vertical appearance.
- Able to define appearance **ranges** with blended colors & thickness.
- User can click/drag needle to change value.
- Value changes can be animated.



Field Labels

Edit field, Droplist & Combobox have **\$label** property.

- **\$labeliscontenttip:** Label shown as content tip when empty.
 - *Animates to new position when filled/focused.*
 - *Color matched to focused border color.*
- **\$labelposition:**
 - **Border:** *Label inset inside the top border.*
 - **Above / Left:** *Label is outside field, above or to left.*
- Can be edited by double-clicking label area in design mode.



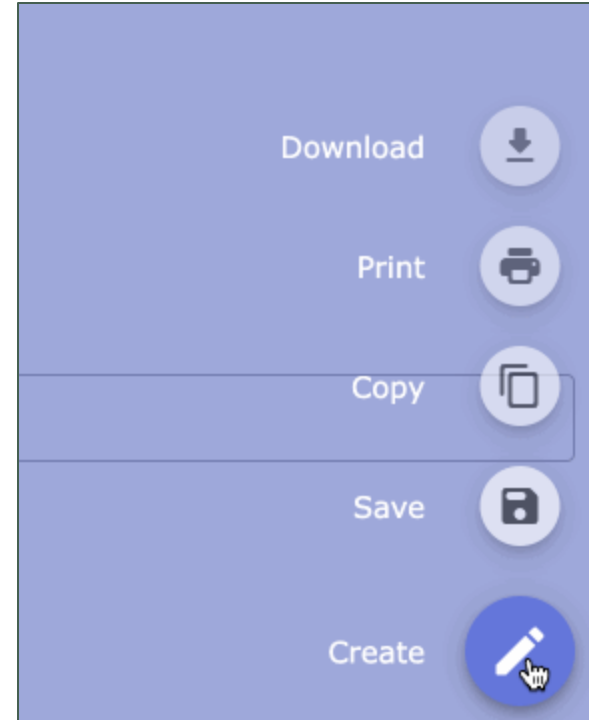
Speeds & simplifies UI Design.

- No need for separate label fields.
- Used as ARIA label for accessibility automatically.



Floating Action Button (FAB)

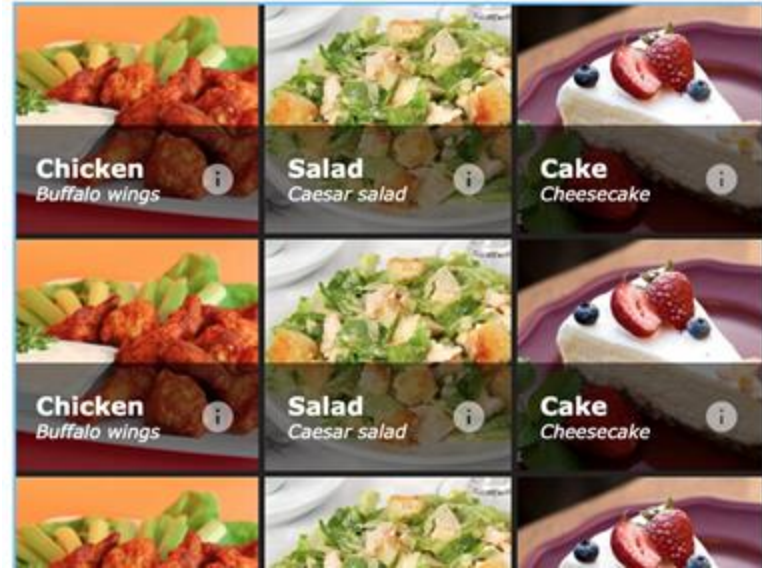
- Concept from Material design.
 - Represents a screen's **primary action**.
- Can open group of **secondary actions** on mouse over or click.
- Animated transitions.
- Familiar especially to mobile app users.



Tile Grid

Scrollable grid of configurable tiles.

- Background image (or block color).
- 'Title bar' with primary, secondary text & button.
 - *Button has separate click event.*
- Number of cols defined **or** auto-fit based on (min) tile widths.
- Tiles are loaded dynamically as the grid is scrolled, to improve performance.



Side Panels

Already present in 10.2 Fat Client.

- Most controls can become a side panel - **\$sidepanel** property.
 - *Especially useful for container-type controls.*
- Dynamically slide in & out from side of container.
 - *Triggered by code or user mouse-over edge of container.*
 - Can **Push** or **Cover** main page content.



Scroll Box

New container control.

- More lightweight & efficient than Paged Pane.
- Can be scrollable.
- Only a single page.
- Can also be made into a 'Group box'.



Single-Layout Breakpoint Forms

Responsive Remote Forms can now have a single layout breakpoint.

- Makes responsive forms suitable for almost all situations.
- Aids the encouraged design of only adding breakpoints as the **content requires**.





Thank You!

Any Questions?