ODC 2025

Omnis Developer Conference

Did you know?

Omnis Hidden Features & Tips

Götz Krija

Launch Omnis from a browser

Use **Custom URL Scheme** Feature to create links that can be added to PDF reports or HTML files.

Custom URL is made up of the scheme **name** (e.g. studio112) and **parameters** passed to Omnis: studio112://lib=Features&param1=Omnis&param2=started

Click on the link:

- Omnis starts up (if necessary)
- runs **$urlinvoked** method in Startup_Task of the specified library
- receives parameters specified in URL scheme

URL scheme name defined via “customURLSchemes” item in config.json

URL scheme registered in Windows Registry when Omnis is started for the first time

- Key added to: HKEY_CLASSES_ROOT\studio<version>\shell\open\command
- Points to: **omnisopenurl.exe** (starts Omnis)

Accessing the Windows Registry

In older Omnis versions there was a component called **RegAdmin** (non-visual object).

- Allowed you to manage Keys and Values within the Windows Registry
- No longer supported & documented, but dll is kept in the tree (xcomp) for compatibility
- **Reading keys and values still works** but creating and deleting not reliable

Creating RegAdmin object:

Object variable; subtype: External Objects > RegAdmin

Opening a Key:

$openkey(kRegAdminKeyConstant,PathToKey,bReadOnly)

Reading Values:

$getvalue(ValueName) Returns Value

Mouse events in JS Client

Use JavaScript mouse events and register a mouse event handler in RemoteForm's client executed **$init** method:

1. Get a reference to the desired field
2. Select the HTML element by using the getClientElem() method
3. Register the mouse event using addEventLister() method:

```
Calculate lBtn as $cinst.$objs.myButton
JavaScript: lBtn.getClientElem().addEventListener("mouseenter", () => {
Do lBtn.$onmouseenter()
JavaScript:});
```

4. Add client executed **$onmouseenter** method to field in RemoteForm

Triggering events with code

Thick Client

Window fields: **$sendevent(iEvent[,eventParameters...])**

```
Do $cinst.$objs.myButton.$sendevent(evClick)
```

JS Client

Remoteform fields do **not** have a $sendevent method…

…but JavaScript Component SDK has an API method: **sendEvent(pEventCode)**

```
Calculate lCtrl as $cinst.$objs.myButton
JavaScript: lCtrl.sendEvent("evClick")
```

$objlink Property

Hidden property that contains information about the container of a field in a Window or RemoteForm **class**:

- Container‘s tab/pane number a field is located on
- Container‘s $order number a field is located

Can be used to:

- get container‘s pane number in RemoteForms ($panenumber only available in Window instances)
- move a field into a container in the same class **using code**

$objlink is Integer, but information needs to be extracted from Binary:

```
Calculate lObjLink as $cclass.$objs.pagedPane.$objs.myButton.$objlink
Calculate lObjLinkBin as binfromint32(lObjLink)
```

Show any Property within Fields in Design Mode

Window, RemoteForm and Report Editor (design mode):

- By default: $name is displayed in each field
- Alternatively: ‘**Show $order**’ via context menu (or **$ident** in Reports)
- Can be changed by **Right-clicking on a property** in the Property Manager and selecting **'Show property using editor context menu**’
- Allows e.g. displaying the $top property within all fields

Import/export CSV or TSV files to/from database

SQL Browser allows to import CSV or TSV files into an open SQL Browser session:

- Use SQL Browser‘s **Import File** option or **drag&drop** file from desktop
- Table is created automatically
- Column headings in file can be used as column headings in the table
- Optional sequence column can be added (adds a unique ID to each row in the table)

Use **Export Results** function in Interactive SQL window to export data to CSV/TSV file.

Alternating list line colors

Thick Client (Window)

List controls (Check Lists, Headed Lists and List Boxes):

- Change "alternatelinecolorplatforms" setting in $prefs. $appearance (appearance.json)

Data Grid:

- Set datagrid's Property **$studioide** to kTrue

Complex Grid:

- Use background control (e.g. Label) and change $visible in odd or even rows:

```
Do   iList.$sendall($cinst.$objs.CompGrid.$bobjs.1026.[iList.$line].$visible.$assign(kFalse),isodd(iList.$line))
```

Alternating list line colors

JS Client (RemoteForm)

List Control and Data Grid:

- Set Property **$evenrowcolor** to desired color

Complex Grid:

- Use background control (e.g. Label) and change $visible in odd or even rows:

```
Do
iList.$sendall($cinst.$objs.CompGrid.$objs.backlabel.[iList.$line].$visible.$
assign(kFalse),isodd(iList.$line))
```

Many thanks