ODC 2025

Omnis Developer Conference

Schon gewusst?

Versteckte Features & Tipps in Omnis

Götz Krija

Omnis aus einem Browser starten

Custom URL Scheme Feature zur Erstellung von Links, die in PDF Reports oder HTML Dateien eingefügt werden können.

Custom URL besteht aus **Namen** des URL-Schemes (z.B. studio112) und **Parametern**, die an Omnis übergeben werden:
studio112://lib=Features&param1=Omnis&param2=started

Klick auf Link:

- Omnis wird gestartet (falls nötig)
- **$urlinvoked** Methode im Startup_Task der Library wird ausgeführt
- Parameter werden empfangen

Name des URL-Schemes definiert via "customURLSchemes" Eintrag in config.json

Registry Eintrag des URL-Schemes erfolgt beim ersten Start von Omnis

- Schlüssel zugefügt: HKEY_CLASSES_ROOT\studio<version>\shell\open\command
- Zeigt auf: **omnisopenurl.exe** (startet Omnis)

Zugriff auf die Windows-Registry

In älteren Omnis-Versionen gab es eine **RegAdmin** Komponente (Non-Visual Object).

- Ermöglichte die Verwaltung von **Schlüsseln und Werten** innerhalb der Windows-Registry
- Nicht mehr supported & dokumentiert; DLL ist weiterhin vorhanden (xcomp)
- Schlüssel und Werte **lesen** funktioniert weiterhin; **Erstellen und Löschen** nicht zuverlässig

RegAdmin Object erzeugen:

Object variable; subtype: External Objects > RegAdmin

Schlüssel öffnen:

$openkey(kRegAdminKeyConstant,PathToKey,bReadOnly)

Wert auslesen:

$getvalue(ValueName) Returns Value

Mouse Events im JS Client

JavaScript Mouse Events verwenden und Mouse Event Handler registrieren in RemoteForm's **$init** Methode:

1. Referenz auf das gewünschte Feld setzen
2. Mit getClientElem() Methode das zugehörige HTML-Element auswählen
3. Mouse Event registrieren mit addEventLister() Methode:

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

4. Client Executed Methode **$onmouseenter** dem Feld im Remoteform zufügen

Events mit Code anstoßen

Thick Client

Felder in Fenstern: **$sendevent(iEvent[,eventParameters...])**

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

JS Client

Felder in Remoteforms haben **keine** $sendevent Methode…

…aber JS Component SDK hat eine API-Methode: **sendEvent(pEventCode)**

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

$objlink Property

Versteckte Property, die Informationen über den Container eines Feldes in einer Window- oder RemoteForm-**Klasse** enthält:

- Tab/Pane Nummer des Containers in dem sich das Feld befindet
- $order Nummer des Containers in dem sich das Feld befindet

Kann verwendet werden um:

- in RemoteForms die Pane Nummer des Containers herauszubekommen ($panenumber nur in Fenster Instanzen verfügbar)
- Ein Feld **per Code** in einen Container derselben Klasse zu verschieben

$objlink ist Integer, aber die Information muss aus Binary extrahiert werden:

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

Beliebige Property in Feldern im Designmode anzeigen

Window-, RemoteForm- und Report-Editor (Designmode):

- Standardmäßig wird **$name** in jedem Feld angezeigt
- Alternativ: '**Show $order**' über Kontextmenü (oder **$ident** in Reports)
- Kann geändert werden, indem man mit der **rechten Maustaste** auf eine Property im Property Manager klickt und **'Show property using editor context menu**' auswählt
- Ermöglicht z. B. die Anzeige der **$top**-Property in allen Feldern

CSV- oder TSV-Dateien in/aus Datenbank importieren/exportieren

SQL-Browser ermöglicht Import von CSV- oder TSV-Dateien in eine geöffnete SQL Browser session:

- SQL Browser‘s **Import File** Option verwenden oder Datei per **Drag&drop** vom Desktop hineinziehen
- Tabelle wird automatisch erstellt
- Spaltenüberschriften in Datei können als Spaltenüberschriften in Tabelle verwendet werden
- Optional kann eine Sequenzspalte hinzugefügt werden (fügt jeder Zeile in der Tabelle eine eindeutige ID hinzu)

Export Results Funktion im Interactive SQL Fenster verwenden um Daten in CSV/TSV Datei zu exportieren.

Abwechselnde Listenzeilenfarben

Thick Client (Window)

List Controls (Check Lists, Headed Lists und List Boxes):

- "alternatelinecolorplatforms" Einstellung ändern in $prefs.$appearance (appearance.json)

Data Grid:

- Datagrid‘s Property **$studioide** auf kTrue setzen

Complex Grid:

- Background Control nutzen (z.B. Label) und $visible in geraden oder ungeraden Zeilen ändern:

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

Abwechselnde Listenzeilenfarben

JS Client (RemoteForm)

List Control und Data Grid:

- **$evenrowcolor** Property auf gewünschte Farbe setzen

Complex Grid:

- Background Control nutzen (z.B. Label) und $visible in geraden oder ungeraden Zeilen ändern:

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

Many thanks