Knowledge Base
Projektbeispiel “Zimmer mit Aussicht”

Elvis Beispielbeschreibung zu Projekt „Zimmer mit Aussicht"

INF - Beschreibung (.pdf)

Produkt: Elvis 3
Version: ab 3.2.134
Stand: 2015-03-20
Autor: Manuela Rameil

Ziel des Dokuments

Dieses Dokument soll als Beschreibung für das Beispielprojekt „ZimmerMitAussicht“ dienen und die darin enthaltenen Projektschritte verdeutlichen.

Grundlagen

Wir haben ein Hotel mit 3 Etagen und jeweils 12 Zimmern. Es gibt zwei Arten von Zimmern, Einzelzimmer und Doppelzimmer. Alle Zimmer sind prinzipiell gleich ausgestattet. Es soll pro Etage eine Seite projektiert werden. Außerdem wird die Schnittstelle Fidelio genutzt, die CheckIn und CheckOut Daten liefert.

KNXDatenpunkte

Hier betrachten wir die Datei KNX.elvissc bzw. die Datei, in der unsere KNX Datenpunkte liegen. Es sind alle projektrelevanten Datenpunkte für den KNX Anschluss zu finden. In unserem Beispiel haben wir die Funktionsstruktur angelegt, die die Gewerke Beleuchtung Jalousie und Präsenz abbildet.

Fidelio

Nun wird ein weiterer Prozessanschluss hinzugefügt: „Fidelio“.

Hierzu geht man mit der Maus auf das Server Projekt „ZimmerMitAussicht“ und fügt per rechten Mausklick über „Hinzufügen“ -> „Serverdaten hinzufügen“ -> „Anschluss mit Datenpunkte“ hinzu.

Dieses Serverprojekt haben wir „Fidelio.elvissc“ genannt.

Es wird nun als Treibertyp „Fidelio“ ausgewählt, jedoch zum Testen auf „Null“ umgestellt wie bei den KNX Anschlüssen.

Den Ordner „Funktionen“ haben wir gelöscht, den Ordner „Datenpunkte“ haben wir in „Fidelio-Datenpunkte“ umbenannt (Ordner markieren und in den Eigenschaften den Namen ändern). Hier haben wir die Datenpunkte für die Zimmer angelegt, um diese als CheckIn/Out für den Anschluss Fidelio zu benutzen.

IT GmbH
An der Kaufleite 12
D-90562 Kalchreuth

Knowledge Base
Projektbeispiel “Zimmer mit Aussicht”

Funktionstypen und Funktionen

Neben dem schon bestehenden Ordner „Funktionstypen“ in der Serverdatei funktionstypen.elvissc haben wir einen weiteren Ordner „Zimmer“ angelegt. Je nach Zimmerart (EZ/DZ) werden in der Spalte „Rollen“ die Funktionen angelegt, die dafür in Frage kommen, wie z.B. Deckenlicht, Licht Nachttisch, Jalousien Fahren usw.

In der Datei KNX.elvissc unter dem Ordner „Funktionen“ werden nun die gewünschten Funktionstypen für die Zimmer mit den entsprechenden Datenpunkten gebunden. Wenn Sie mit dem „grünen Plus“ eine Funktion hinzu fügen, können Sie sich in der Zeile „Funktionstyp“ die verschiedenen Funktionen anzeigen lassen und dementsprechend auswählen. Hier werden dann die Datenpunkte per Drag & Drop gebunden. In unserem Projekt haben wie nur im Ordner „EG“ die Funktionen angelegt und gebunden.

Nutzung von Diagramm vs. ServerElement .elvissc

Gebunden werden die Datenpunkte entweder in einem Diagramm oder in einem Server Element. Wir haben beide Beispiele hier einmal aufgeführt, damit man den Unterschied klar erkennen und für sich die ideale Darstellung finden kann.

Das Diagramm finden Sie im Ordner „CheckInEG.elviswr“. Verwendet wurde hier ein logisches Und mit vielen Eingängen, das in dieser Darstellung schnell unübersichtlich werden kann. Zum Vergleich haben wir die gleiche Funktionalität dann noch einmal in einem Server Element dargestellt unter „CheckIn.elvissc“.

ServerElement .elvissc

Das Server Element bekommen Sie, indem Sie über „Hinzufügen“ -> „Serverdaten hinzufügen“ -> „Benutzerdefinierte Elemente“ auswählen und ein beliebiges hinzufügen.

Wir haben diese Mappe mit dem Namen „CheckInLogik“ benannt. Im Eigenschaftfeld wählen Sie unter Punkt „Ordner“ -> „Erlaubter Elemententyp“ -> „LogicalOp“ aus und stellen hier auch den passenden Namen ein.

Nun haben wir aus „Bindbare Elemente“ die unter „Fidelio-Datenpunkte“ angelegten ZimmerDatenpunkte mit Drag & Drop auf den Input mit ActualValue gezogen. Angelegt wurde hier für CheckIn OG1 und CheckIn OG2.

Diagramm

Das Diagramm bekommen Sie ebenfalls über „Hinzufügen“ -> Diagramm hinzufügen“. Das Diagramm haben wir „CheckIn EG“ genannt und folgendermaßen angelegt:

Auch hier ziehen wir wieder die Datenpunkte aus „Bindbare Elemente“ -> „Fidelio-Datenpunkte“ mit Drag & Drop hinein.

Aus dem Werkzeugkasten wählen wir aus dem Ordner „Arithmetik/Logik“ -> „Logische Op“ aus und platzieren diese innerhalb des Diagrammes.

IT GmbH
An der Kaufleite 12
D-90562 Kalchreuth

Nun werden alle Datenpunkte mit der Funktion „Verbindung“ aus dem Werkzeugkasten unter Ordner „Elvis“ mit der Funktion „LogischeOp“ verbunden (durch das Anklicken des „grünen Plus“ an der „LogischeOp“ erhält man mehr Eingänge).

Vergleich

Alle verwendeten logischen Operationen arbeiten auf der gleichen Grundlage und somit exakt gleich, es soll hier nur gezeigt werden, dass es für jede Bedienart immer zwei Wege für die Umsetzung gibt – das Diagramm und das Server Element.

In unserem Fall möchten wir darauf hinweisen, dass die schwierig zu erfassende Darstellung im Diagramm sich im ServerElement sehr viel einfacher verstehen lässt.

Da beide Elemente das gleiche Ergebnis liefern, bleibt es dem Nutzer der Software überlassen, mit welchem Element er arbeitet.

Navigation mit Parts

Wir erstellen einen Part (über rechten Mausklick „hinzufügen“ -> „Part hinzufügen“) und nennen ihn Seitennavigation.

Dann werden alle Seiten aus dem Ordner Pages hineingezogen. Dadurch wird ein CommandButton erzeugt, der einen Befehl GoToPage mit dem Parameter zur entsprechenden Seite erhält.

Um den Part verwenden zu können müssen Sie die Projektmappe erstellen (Menü Erstellen -> Projektmappe erstellen).

Nun kann der Part in jeder Seite bereits eingesetzt werden.

Wie bekommen wir die Anzeige der aktiven Seite? Ganz einfach! Überlagern Sie das genutzte Steuerelement auf der entsprechenden Seite mit einem Rechteck oder anderen Elementen Ihrer Wahl. Beachten Sie dabei, dass Sie den ZIndex in der Eigenschaftskategorie Layout auf einen Wert einstellen, der höher ist als der ZIndex des Navigationsparts.

Funktionen und Parts

Räume mit immer gleichem Aufbau sollen nur ein Mal projektieren werden.

Das Vorgehen dazu ist ganz einfach.

a. Funktionstypen anlegen (siehe oben)
b. Funktion erstellen (siehe oben)
c. Part vorbereiten: Hintergrundbild über Drag & Drop einfügen, diverse Steuerelemente
d. Bindungen erstellen (siehe Punkt 2) (unter KNX.elvissc – Ordner „Funktionen“ – Unterordner diverse Geschossebenen – „Funktionstyp auswählen“ –) Rollen mit den verschiedenen Bezeichnungen erscheinen bei der jeweiligen Auswahl, da unter Funktionstypen angelegt. Hier werden nun die Steuerelemente mit Drag & Drop gebunden.
e. Anschliessend ist der gesamte Part noch an den Funktionstyp selbst zu binden - Prototypebinding.

IT GmbH
An der Kaufleite 12
D-90562 Kalchreuth

f. Part erstellen (über Menu Punkt „Erstellen“ -> „Projektmappe erstellen“)
g. Nun kann der Part in der Bedienseite eingefügt werden.
h. Binden Sie den eingefügten Part anschliessend an die Funktion, in unserem Beispiel also an das Zimmer, das der Part bedienbar machen soll.

So projektieren Sie im Handumdrehen ganze Etagen.

Seitendarstellung mit einem TabControl übersichtlich gestalten

Beim WPF TabControl ist zu beachten:

- Ein TabControl enthält TabItems.
- Ein TabItem besitzt einen TabItem.Header.

Bemerkung: Diese Struktur ist sehr schön über die Dokumentgliederung sichtbar zu machen: Menü ANSICHT -> Weitere Fenster -> Dokumentgliederung.

Wenn man einem TabControl ein weiteres TabItem hinzufügen möchte, geht man mit dem Cursor auf das TabControl, klickt mit der rechen Maustaste und wählt „TabItem hinzufügen“. Danach muss hier noch ein Canvas angelegt werden, um eventuell Steuerelemente zu platzieren. Man klickt also wieder mit der rechten Maustaste in das graue Feld (Grid) und wählt hier „Layout ändern“ -> „Canvas“.

Nun kann man wie gewohnt seine Steuerelemente einfügen.

Möchte man in den TabItems die Beschriftung mit Label und Checkbox wie im Beispiel haben, muss im Skript folgendes einfügen (leider ist dies über Drag&Drop nicht möglich):

Nach der Zeile: <TabControl Height =“509“ Width=“597“> (die Zahlen können bei Ihnen variieren)

```
<TabItem Height="46" VerticalAlignment="Top" >
            <TabItem.Header>
               <Canvas Width="40" Height="40">
                  <CheckBox IsChecked="TRUE" Canvas.Left="11" Canvas.Top="3" IsEnabled="False" />
                  <Label Content="EG" Canvas.Left="6" Canvas.Top="17"/>
               </Canvas>
            </TabItem.Header>
```

Der dargestellte Text innerhalb des TabItems lässt sich aber wie gewohnt auch über die Eigenschaften ändern.

IT GmbH
An der Kaufleite 12
D-90562 Kalchreuth