



## **Exercise Solution: Create Menu Entries**

User Interface API

**PUBLIC**

## INTRODUCTION

In this exercise, you will perform the following tasks:

1. Use DB Data Source
2. Use User Data Source
3. Use Choose from List
4. Create Item Event for Choose from List
5. Use Data Tables
6. Insert values into the User Defined Table

## PREREQUISITE:

- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Use the demo database for SAP Business One, version for SAP HANA or SAP Business One
- Credentials: User code: **manager**

## GUIDELINES:

The screenshots provided here are for your reference only and may differ from the actual screenshots in your system.

## 1. TASK – USE DB DATA SOURCE

Declare a `DBDataSource` object. Link the data source to the form you created in the Creating Forms exercise by using a separated function called **DataBinding**.

```
private static void DataBinding(SAPboui COM. Form oForm)
{
    oForm.DataSources.DBDataSources.Add("@TB1_VIDS");
}
```

## 2. TASK – USE USER DATA SOURCE

When you choose the menu entry, your form should be displayed.

- 2.1. Extend the **DataBinding** function with `UserDataSource` object for . For the *U\_Rented* field the `UserData` source is needed to convert the boolean field. For the *U\_CardCode* field we will define a *Choose From List* object later.

```
oForm.DataSources.UserDataSources.Add("ds_Rented", SAPboui COM. BoDataType. dt_SHORT_TEXT);
oForm.DataSources.UserDataSources.Add("ds_RentTo", SAPboui COM. BoDataType. dt_SHORT_TEXT);
```

- 2.2. Bind the remaining form's items with the corresponding data from the User Defined table created in the DI exercises (**TB1\_VIDS**). For each Edit Text field bind the field to its corresponding column in the User defined table. You can please this code to the **DataBinding** function as well.

```
oDBDataSource.Query(null);

oItem = oForm.Items.Item("tx_Name");
oEditText = ((SAPboui COM. EditText)(oItem.Specification));
oEditText.DataBind.SetBound(true, "@TB1_VIDS", "Name");

oItem = oForm.Items.Item("tx_Aisle");
oEditText = ((SAPboui COM. EditText)(oItem.Specification));
oEditText.DataBind.SetBound(true, "@TB1_VIDS", "U_Aisle");

oItem = oForm.Items.Item("tx_Section");
oEditText = ((SAPboui COM. EditText)(oItem.Specification));
oEditText.DataBind.SetBound(true, "@TB1_VIDS", "U_Section");

oItem = oForm.Items.Item("tx_RentTo");
oEditText = ((SAPboui COM. EditText)(oItem.Specification));
oEditText.DataBind.SetBound(true, "", "ds_RentTo");

oItem = oForm.Items.Item("tx_Rented");
oEditText = ((SAPboui COM. EditText)(oItem.Specification));
oEditText.DataBind.SetBound(true, "", "ds_Rented");
var rentedValue = oDBDataSource.GetValue("U_Rented", 0);

oForm.DataSources.UserDataSources.Item("ds_Rented").ValueEx = rentedValue;
```

- 2.3. Call the **DataBinding** function in **LoadFormFromXML** function by passing the *Form* object.

```
DataBinding(oForm);
```

### 3. TASK – USE CHOOSE FROM LIST

Create a choose from list object for Business Partner Master data, which will be assigned to the **tx\_Rented** Edit Text object.

- 3.1. Create new function **CreateChooseFromList**, where the *ChooseFromList* object needs to be defined.

```
private static void CreateChooseFromList(SAPboui.COM. Form oForm)
{
    SAPboui.COM. ChooseFromListCollection oCFLs;
    oCFLs = oForm. ChooseFromLists;
    SAPboui.COM. ChooseFromList oCFL;
    SAPboui.COM. ChooseFromListCreationParams oCFLCreationParams;
    oCFLCreationParams =
    ((SAPboui.COM. ChooseFromListCreationParams) (SBO_Application. CreateObject(SAPboui.COM. BoCreatableObjectType, cot_ChooseFromListCreationParams)));
    oCFLCreationParams.ObjectType = "2";
    oCFLCreationParams.UniqueID = "CardCodeCFL";
    oCFL = oCFLs.Add(oCFLCreationParams);
}
```

- 3.2. Assign the choose from list *CardCodeCFL* to the *tx\_Rented*. You can please this feature to the same **CreateChooseFromList** function.

```
SAPboui.COM. Item oItem;
SAPboui.COM. EditText oEditText;

oItem = oForm.Items.Item("tx_RentTo");
oEditText = ((SAPboui.COM. EditText) (oItem. Specific));
oEditText.ChooseFromListUID = "CardCodeCFL";
```

- 3.3. Call the **CreateChooseFromList** function in the **DataBinding** function.

```
CreateChooseFromList(oForm);
```

### 4. TASK – CREATE ITEM EVENT FOR CHOOSE FROM LIST

- 4.1. Extend the **SetEventFilters** function with Choose From List event type.

```
oFilter = oFilters.Add(SAPboui.COM. BoEventTypes.et_CHOOSE_FROM_LIST);
```

- 4.2. Extend the **SBO\_Application\_ItemEvent** function to handle the Choose From List event type.

```

if (pVal.FormUID == "TB1_DVDAvailability" & pVal.BeforeAction == false &
pVal.ItemUID == "tx_RentTo" & pVal.EventType == SAPboui.COM.ChooseFromListEventTypes.et_CHOOSE_FROM_LIST)
{
    SAPboui.COM.ChooseFromListEvent oCFLEvent;
    SAPboui.COM.Form oForm;
    oCFLEvent = (SAPboui.COM.ChooseFromListEvent)pVal;
    oForm = SBO_Application.Forms.Item(FormUID);
    SBO_Application.MessageBox("ChooseFromListEvent picked");
}

```

## 5. TASK – USE DATA TABLES

We would like to return to our form the selected value from the *choose from list* object. For this purpose, we will use the *DataTable* object.

- 5.1. Define the DataTable object in the function **SBO\_Application\_ItemEvent**.

```

SAPboui.COM.DBDataSource oDBDataSource;
SAPboui.COM.DataTable oDataTable;
oDataTable = oCFLEvent.SelectedObjects;

var selectedCardCode = oDataTable.GetValue(0, 0);
oForm.DataSources.UserDataSources.Item("ds_RentTo").ValueEx =
selectedCardCode.ToString();
oDBDataSource = oForm.DataSources.DBDataSources.Item("@TB1_VIDS");

```

- 5.2. Comment the MessageBox related line in **SBO\_Application\_ItemEvent**.

```
//SBO_Application.MessageBox("ChooseFromListEvent picked");
```

## 6. TASK – INSERT VALUES TO USER DEFINED TABLE

The form needs to be switched to the adding mode and the newly inserted records needs to be inserted in the User Defined Table.

- 6.1. Capture the form switch to the inserting mode and clean all values from the form. This might be achieved by adding some code to the **SBO\_Application\_MenuEvent** function.

```

if (pVal.BeforeAction && pVal.MenuUID == "1282")
{
    SAPboui.COM.Form oForm = SBO_Application.Forms.Item("TB1_DVDAvailability");

    SAPboui.COM.Item oItemName = oForm.Items.Item("tx_Name");
    SAPboui.COM.EditText oEditText = ((SAPboui.COM.EditText)(oItemName.Specific));
    oEditText.Value = "";
}

```

```

        SAPboui COM. Item oItemAisle = oForm.Items.Item("tx_Aisle");
        SAPboui COM. EditText oEditTextAisle =
((SAPboui COM. EditText)(oItemAisle.Specific));
        oEditTextAisle.Value = "";

        SAPboui COM. Item oItemSection = oForm.Items.Item("tx_Section");
        SAPboui COM. EditText oEditTextSection =
((SAPboui COM. EditText)(oItemSection.Specific));
        oEditTextSection.Value = "";

        SAPboui COM. Item oItemRented = oForm.Items.Item("tx_Rented");
        SAPboui COM. EditText oEditTextRented =
((SAPboui COM. EditText)(oItemRented.Specific));
        oEditTextRented.Value = "";

        SAPboui COM. Item oItemRentTo = oForm.Items.Item("tx_RentTo");
        SAPboui COM. EditText oEditTextRentTo =
((SAPboui COM. EditText)(oItemRentTo.Specific));
        oEditTextRentTo.Value = "";

        oForm.ActiveItem = "tx_Name";
    }

```

6.2. Insert your code into the **SBO\_Application\_ItemEven**, which will insert the newly entered values from the form to the User Defined Table.

```

    if (pVal.FormUID == "TB1_DVDAvailability" & pVal.BeforeAction == false &
pVal.ItemUID == "1"
        & pVal.EventType == SAPboui COM. BoEventTypes.et_ITEM_PRESSED)
    {
        SAPboui COM. Form oForm = SBO_Application.Forms.Item(FormUID);
        SAPboui COM. Item oItem = oForm.Items.Item(pVal.ItemUID);
        SAPboui COM. Button oButton = ((SAPboui COM. Button)(oItem.Specific));
        if (oButton.Caption == "Add")
        {
            SAPboui COM. Item oItemName = oForm.Items.Item("tx_Name");
            SAPboui COM. EditText oTextName =
((SAPboui COM. EditText)(oItemName.Specific));

            SAPboui COM. Item oItemAisle = oForm.Items.Item("tx_Aisle");
            SAPboui COM. EditText oTextAisle =
((SAPboui COM. EditText)(oItemAisle.Specific));

            SAPboui COM. Item oItemSection = oForm.Items.Item("tx_Section");
            SAPboui COM. EditText oTextSection =
((SAPboui COM. EditText)(oItemSection.Specific));

            SAPboui COM. Item oItemRented = oForm.Items.Item("tx_Rented");
            SAPboui COM. EditText oTextRented =
((SAPboui COM. EditText)(oItemRented.Specific));

            SAPboui COM. Item oItemRentTo = oForm.Items.Item("tx_RentTo");
            SAPboui COM. EditText oTextRentTo =
((SAPboui COM. EditText)(oItemRentTo.Specific));

            SAPbobsCOM. UserTable oUDT;
            oUDT = diCompany.UserTables.Item("TB1_VIDS");

```

```
oUDT.Code = oTextName.Value;
oUDT.Name = oTextName.Value;
oUDT.UserFields.Fields.Item("U_AISLE").Value = oTextAisle.Value;
oUDT.UserFields.Fields.Item("U_SECTION").Value = oTextSection.Value;
oUDT.UserFields.Fields.Item("U_RENTED").Value = oTextRented.Value;
oUDT.UserFields.Fields.Item("U_CARDCODE").Value = oTextRentTo.Value;

int ret = oUDT.Add();
if (ret != 0)
    SBO_Application.MessageBox("Entry cannot be added: " +
di Company.GetLastErrorDescription());
else
    SBO_Application.MessageBox("Entry " + oTextName.Value + " added.");

System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
GC.Collect();
}
```

*A similar solution can be found in the SDK UI samples (in the SDK Folder – see Appendix “SDK Installations” for more information), COM UI/03.SimpleForm and, 06.MatrixAndDataSources*



© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see

<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.