

## Table of Content

<b>Loading XML elements to System Forms</b> .....	<b>1</b>
<b>Applies To</b> .....	<b>2</b>
<b>General</b> .....	<b>2</b>
<b>References</b> .....	<b>2</b>
<b>Case description</b> .....	<b>3</b>
What we want to achieve. ....	3
1. Sales Order Form before modification .....	3
2. Sales Order Form with the loaded elements .....	4
3. Detail of Tab1 .....	5
4. Detail of Tab2.....	6
How to achieve it.....	6
Diagram.....	7
To do's.....	7
<b>Process</b> .....	<b>8</b>
1. Hooking into SAP Business One UI process flow .....	8
2. Manipulate B1 UI.....	9
3. Data used to change the UI. ....	10
<b>Disclaimer &amp; Liability Notice</b> .....	<b>13</b>

## **Applies To**

SAP Business One / SAP Business One SDK

**By:** Felipe Ortega Aguilar

**Title:** Solution Architect

**Date:** 23 April 2005

## **General**

The following document will describe how to load GUI elements into a system form, when these elements (Tabs, buttons, fields, etc) are contained in an Xml file.

We'll load two folder or tabs elements into the sales order form. The first folder contains two check boxes and the second one three radio buttons.

Please note that the provided source code was developed in VB.NET.

## **References**

Additional information regarding Namespaces can be found in note 647987 or under the following link: <http://service.sap.com/namespaces>

## Case description.

As the Form UID of system forms is automatically assigned each time a form is opened (to support multiple instances), it is needed to change the Form UID in the Xml file each time the sales order form is loaded.

### **What we want to achieve.**

We want to modify a system form by adding two new Tabs into it. These Tabs will contain additional elements such as checkboxes and radio buttons. Why Tabs? Because in this way your own GUI elements can be easily identified and you assure an area in which standard fields will not be placed.

The next screen shots will show how the Sales Order form will look like after executing all steps contained in this document.

### **1. Sales Order Form before modification**

This screen shows the system form before adding the tab. As you can see it consists in three Tabs.

The screenshot displays the SAP Sales Order form. The left sidebar shows the 'Main Menu' with 'Sales - A/R' selected, and 'Sales Order' highlighted. The main form area contains fields for Customer, Name, Contact Person, and Customer Ref. No. on the left, and No., Primary, Status, Open, Posting Date, Delivery Date, and Local Currency on the right. Below these are three tabs: 'Contents', 'Logistics', and 'Accounting'. The 'Contents' tab is active, showing a table with columns: Item No., Item Description, Quantity, Price, Tax Code, and Total (L). The table has one row with '1' in the Item No. column. Below the table are fields for Sales Employee (set to '-No Sales Employee-'), Owner, and Remarks. To the right of the Remarks field are summary fields: Total Before Discount, Discount (%), Add. Expenses (USD 5.00), Rounding (USD 0.00), Tax, and Total (USD 5.00). At the bottom of the form are 'Add', 'Cancel', and 'Sales Quotations' buttons. The status bar at the bottom of the SAP GUI shows '[Form=169 Item=6 Column=7 Row=6]' and the date/time '04/21/05 12:21'.

## 2. Sales Order Form with the loaded elements

After executing the Add-On Two new Tabs will be added ("Tab1 Add-On" and "Tab2 Add-On")

The screenshot shows the SAP Sales Order form interface. On the left is the Main Menu with categories: Administration, Financials, Sales Opportunities, and Sales - A/R. The Sales Order menu item is highlighted. The main form area is titled 'Sales Order' and contains the following fields and sections:

- Customer: [Field]
- Name: [Field]
- Contact Person: [Field]
- Customer Ref. No.: [Field]
- No.: Primary 14
- Status: Open
- Posting Date: 04/21/05
- Delivery Date: [Field]
- Local Currency: [Field]

Below these fields are tabs for Contents, Logistics, Accounting, Tab1 Addon, and Tab2 Addon. The Contents tab is active, showing a table with columns: #, Item No., Item Description, Quantity, Price, Tax Code, and Total (LC). The table contains one row with item number 1.

At the bottom of the form, there are fields for Sales Employee (-No Sales Employees-), Owner, and Remarks. To the right, a summary section shows:

- Total Before Discount: [Field]
- Discount: [Field] %
- Add. Expenses: USD 5.00
- Rounding: USD 0.00
- Tax: [Field]
- Total: USD 5.00

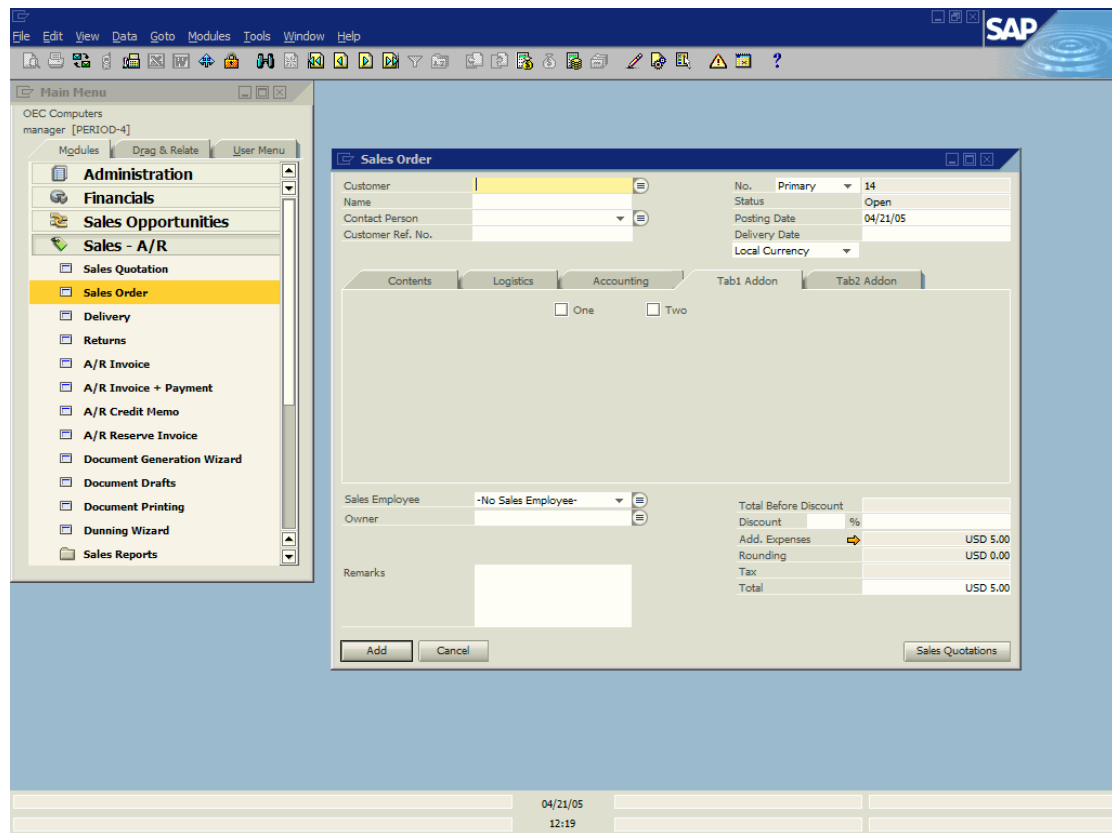
Buttons for Add, Cancel, and Sales Quotations are located at the bottom of the form.

[Form=169 Item=6 Column=7 Row=6]

04/21/05 12:17

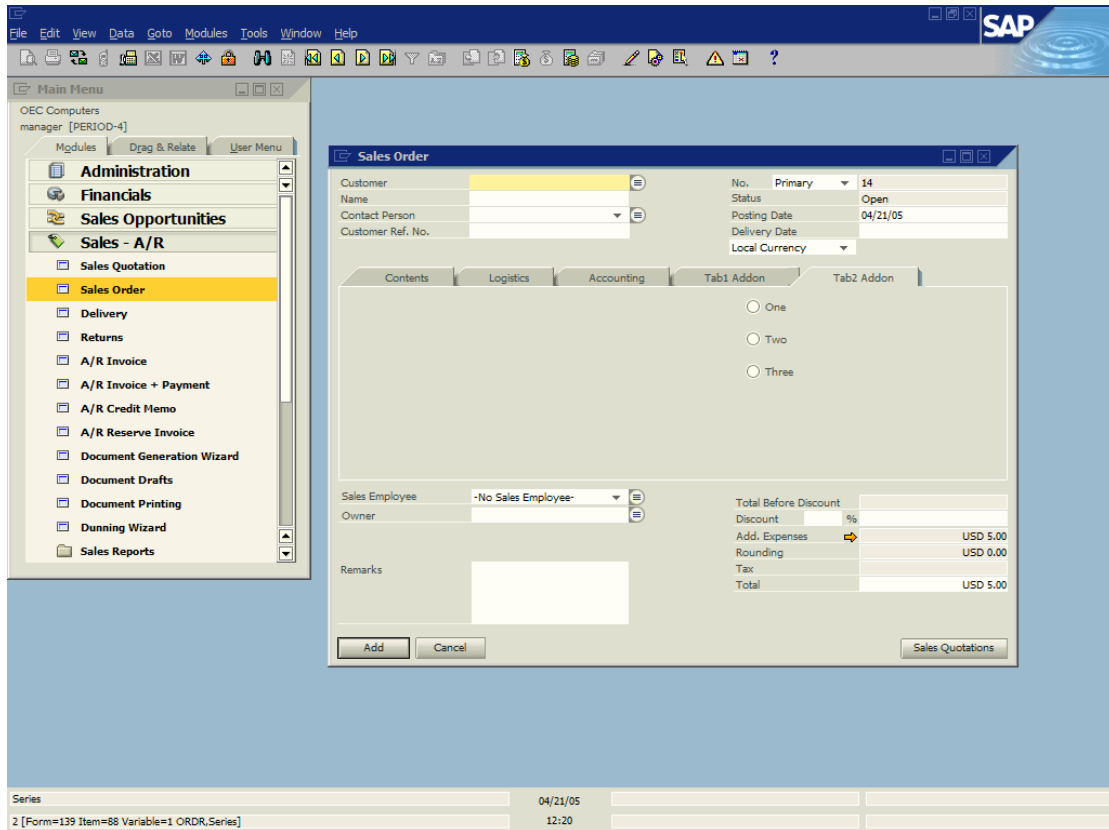
### 3. Detail of Tab1

Tab1 Add-On contains two check boxes



## 4. Detail of Tab2

Tab2 Add-On contains three radio buttons.



### How to achieve it.

You will have to create a new project in .Net or you can include the code in an existing project. Just after establishing a connection to B1 using UI API you will be ready to start using the provided code.

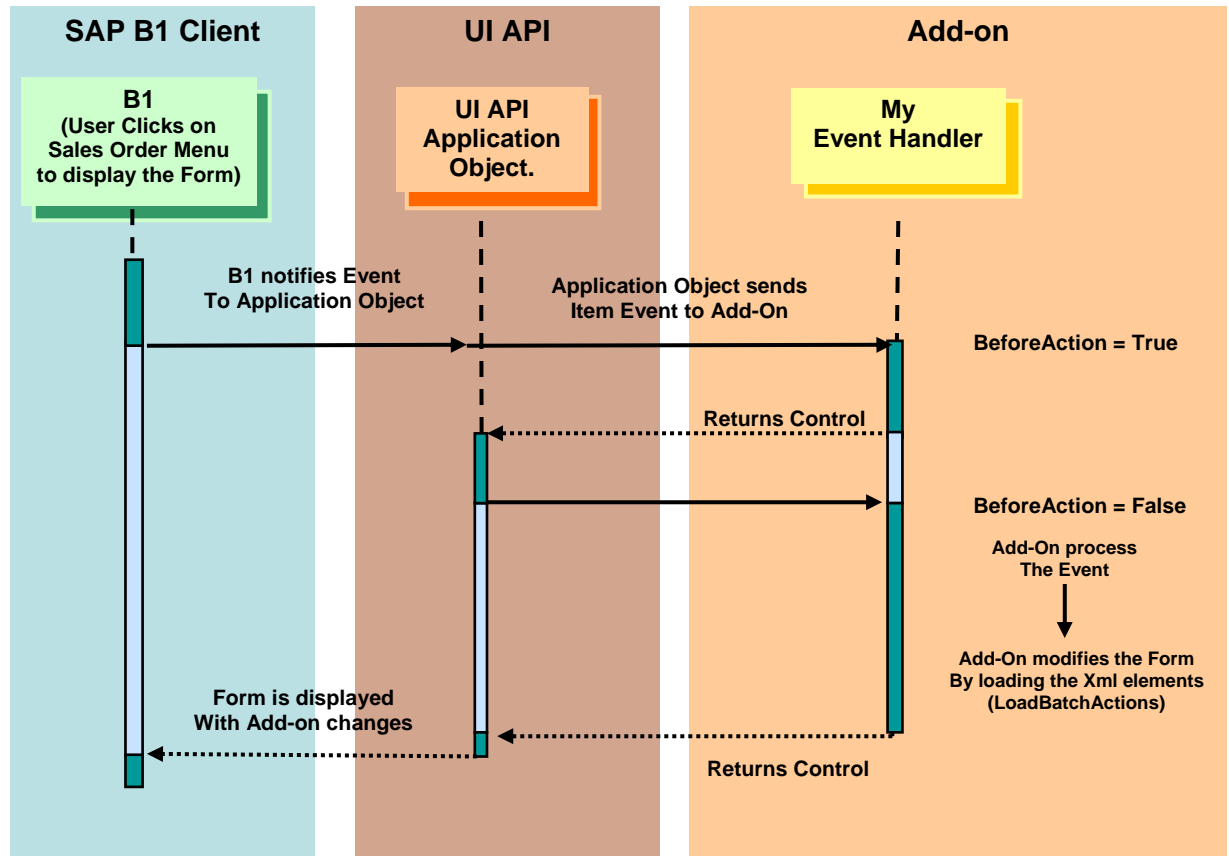
As you can see in the Diagram below this section, SAP B1 application notifies the UI API Application object on events by connecting to the UI API and registering with the Application object, add-ons receive all the events notified on this object.

For our specific case, we need to catch the Item Event because we want to know when the Sales Order Form is loaded. This event provides us the event type "et\_FORM\_LOAD" that will notify that the form will be loaded. (refer to "Hooking into SAP Business One UI process flow" section to see the corresponding code)

Now that the Add-On was notified about this event, and that we know the UID of the specific Form (provided by the event), we can proceed adding the (UID) into our Xml file (refer to "Manipulate B1 UI" section to see code). After this step we load the xml into the form that will be presented in B1.

At this point the Add-on will return the control to the Application Object that will send the processed event to SAP B1 application and the form will finally be displayed.

**Diagram.**



**To do's**

- Copy the sample code into your project (Steps 1 and 2)
- Copy the xml code, described in step 3 and save it as xml type.

## Process.

### 1. Hooking into SAP Business One UI process flow

First we will catch the event when the Sales order is loaded. The process of replacing the Form UID will take place in the "GetandReplaceUID" subroutine, which will be described in the next step.

```
Private Sub B1_Application_ItemEvent(ByVal FormUID As String, ByRef pVal As SAPbouiCOM.ItemEvent, ByRef BubbleEvent As Boolean) Handles B1_Application.ItemEvent
```

```
If pVal.FormType = "139" And pVal.BeforeAction = False Then
```

```
    Dim oItem As SAPbouiCOM.Item  
    Dim oFolder As SAPbouiCOM.Folder
```

```
    Select Case pVal.EventType
```

```
        Case SAPbouiCOM.BoEventTypes.et_FORM_LOAD  
            oForm = B1_Application.Forms.Item(FormUID)  
            ReplaceUIDandLoadToB1("TabElements.xml", pVal.FormUID)  
            ' // Set the first Tab as default  
            oItem = oForm.Items.Item("112")  
            oFolder = oItem.Specific  
            oFolder.Select()  
            oForm.PaneLevel = 1
```

```
    ' *
```

```
        Case SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED  
            ' // Set the correct pane level depending on the folder selection.  
            ' You can use different range of numbers (not the first 50 at least) to avoid collisions when a  
            ' system tab is added to the form.
```

```
            oForm = B1_Application.Forms.Item(FormUID)
```

```
            If pVal.ItemUID = "FOA_FOL1" Then  
                oForm.PaneLevel = 90  
            End If
```

```
            If pVal.ItemUID = "FOA_FOL2" Then  
                oForm.PaneLevel = 91  
            End If
```

```
    End Select
```

```
End If
```

```
End Sub
```



## **2. Manipulate B1 UI.**

This routine will load the Xml elements to the form.

```
Public Sub ReplaceUIDandLoadToB1(ByRef FileName As String, ByVal a_uid As String)
```

```
Const XML_FormUIDPath As String = "Application/forms/action/form/@uid"
```

```
Dim oXmlDoc As MSXML2.DOMDocument
```

```
Dim oNode As MSXML2.IXMLDOMNode
```

```
Try
```

```
oXmlDoc = New MSXML2.DOMDocument
```

```
'// load the content of the XML File
```

```
Dim sPath As String
```

```
sPath = IO.Directory.GetParent(Application.StartupPath).ToString
```

```
oXmlDoc.load(sPath & "\" & " TabElements.xml")
```

```
'// Find and Change the UID dynamically
```

```
oNode = oXmlDoc.documentElement.selectSingleNode(XML_FormUIDPath)
```

```
oXmlDoc.selectSingleNode(XML_FormUIDPath).nodeValue = pVal.FormUID
```

```
'Save the changed xml data to verify the new Form UID
```

```
oXmlDoc.save(sPath & "\" & "xmlChanged.xml")
```

```
'Load the xml into de Form
```

```
B1_Application.LoadBatchActions(oXmlDoc.xml)
```

```
Catch ex As Exception
```

```
B1_Application.MessageBox(ex.Message)
```

```
End Try
```

```
End Sub
```

### 3. Data used to change the UI.

For detailed information about each Element and Attribute used in this xml file, please refer to Sap Business One Help under the path: UI API → How To... → Add Forms to Xml or follow this link [Add Forms with XML](#)

To create the Xml file please copy and save the following data as xml. In our case you should name it "TabElements.xml". Please note that the Form UID is equal "", as we will insert it dynamically when each Sales order form is loaded.

```
<?xml version="1.0" encoding="UTF-16" ?>
<Application>
  <forms>
    <action type="update">
      <form FormType="139" uid="">
        <datasources>
          <dbdatasources>
            <action type="add" />
          </dbdatasources>
          <userdatasources>
            <action type="add">
              <datasource size="1" type="9"
uid="FOLDER" />
              <datasource size="1" type="9"
uid="Radio1" />
              <datasource size="1" type="9"
uid="Check1" />
              <datasource size="1" type="9"
uid="Check2" />
            </action>
          </userdatasources>
        </datasources>
        <Menus />
        <items>
          <action type="add">
            <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
forecolor="-1" from_pane="0"
height="20" left="360" linkto="" right_just="0" supp_zeros="0"
tab_order="0" text_style="0"
to_pane="0" top="88" type="99" uid="FOA_FOL1" visible="1" width="93">
              <AutoManagedAttribute />
              <specific AffectsFormMode="1"
caption="Tab1 Addon" val_off="" val_on="FOLDER">
                <databind alias="FOLDER"
databound="1" table="" />
              </specific>
            </item>
            <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
forecolor="-1" from_pane="0"
height="20" left="455" linkto="" right_just="0" supp_zeros="0"
tab_order="0" text_style="0"
to_pane="0" top="88" type="99" uid="FOA_FOL2" visible="1" width="93">
              <AutoManagedAttribute />
              <specific AffectsFormMode="1"
caption="Tab2 Addon" val_off="" val_on="FOLDER">
                <databind alias="FOLDER"
databound="1" table="" />
              </specific>
            </item>
          </action>
        </items>
      </form>
    </action>
  </forms>
</Application>
```

```

        </item>
        <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
        forecolor="-1" from_pane="90"
height="14" left="200" linkto="" right_just="0" supp_zeros="0"
        tab_order="0" text_style="0"
to_pane="90" top="120" type="121" uid="FOA_CHK1" visible="1"
        width="80">
        <AutoManagedAttribute />
        <specific AffectsFormMode="1"
caption="One" val_off="N" val_on="Y">
        <databind alias="Check1"
databound="1" table="" />
        </specific>
    </item>
    <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
        forecolor="-1" from_pane="90"
height="14" left="285" linkto="" right_just="0" supp_zeros="0"
        tab_order="0" text_style="0"
to_pane="90" top="120" type="121" uid="FOA_CHK2" visible="1"
        width="80">
        <AutoManagedAttribute />
        <specific AffectsFormMode="1"
caption="Two" val_off="N" val_on="Y">
        <databind alias="Check2"
databound="1" table="" />
        </specific>
    </item>
    <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
        forecolor="-1" from_pane="91"
height="14" left="320" linkto="" right_just="0" supp_zeros="0"
        tab_order="0" text_style="0"
to_pane="91" top="120" type="122" uid="FOA_RD1" visible="1"
        width="80">
        <AutoManagedAttribute />
        <specific AffectsFormMode="1"
caption="One" val_off="N" val_on="1">
        <databind alias="Radio1"
databound="1" table="" />
        </specific>
    </item>
    <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
        forecolor="-1" from_pane="91"
height="14" left="320" linkto="" right_just="0" supp_zeros="0"
        tab_order="0" text_style="0"
to_pane="91" top="150" type="122" uid="FOA_RD2" visible="1"
        width="80">
        <AutoManagedAttribute />
        <specific AffectsFormMode="1"
caption="Two" val_off="N" val_on="2">
        <databind alias="Radio1"
databound="1" table="" />
        </specific>
    </item>
    <item AffectsFormMode="1" bgcolor="-1"
description="" disp_desc="0" enabled="1" font_size="0"
        forecolor="-1" from_pane="91"
height="14" left="320" linkto="" right_just="0" supp_zeros="0"
        tab_order="0" text_style="0"
to_pane="91" top="180" type="122" uid="FOA_RD3" visible="1"

```

```

caption="Three" val_off="N" val_on="3">
databound="1" table="" />
width="80">
<AutoManagedAttribute />
<specific AffectsFormMode="1"
    <databind alias="Radio1"
        </specific>
    </item>
    </action>
</items>
<items>
    <action type="group">
        <item uid="FOA_FOL1" />
        <item uid="FOA_FOL2" />
        <item uid="112" />
        <item uid="114" />
        <item uid="138" />
    </action>
</items>
<items>
    <action type="group">
        <item uid="FOA_RD1" />
        <item uid="FOA_RD2" />
        <item uid="FOA_RD3" />
    </action>
</items>
<DataBrowser />
</form>
</action>
</forms>
</Application>

```

## **Disclaimer & Liability Notice**

This document may discuss sample coding, which does not include official interfaces and therefore is not supported. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing of the code and methods suggested here, and anyone using these methods, is doing it under his/her own responsibility.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of the technical article, including any liability resulting from incompatibility between the content of the technical article and the materials and services offered by SAP. You agree that you will not hold SAP responsible or liable with respect to the content of the Technical Article or seek to do so.

Copyright © 2004 SAP AG, Inc. All Rights Reserved. SAP, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product, service names, trademarks and registered trademarks mentioned are the trademarks of their respective owners.

