

Exercise Solution: Additional Event Handling

User Interface API

PUBLIC

INTRODUCTION

In this exercise, you will perform the following tasks:

1. Catch *FormLoad* event for Sales Order form
2. Catch the click event on the “Rent DVD” button
3. Create an event filter
4. Create object level event

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 - CATCH FORMLOAD EVENT FOR SALES ORDER FORM

- 1.1. Create a function which will handle the mandatory *ItemEvent*.

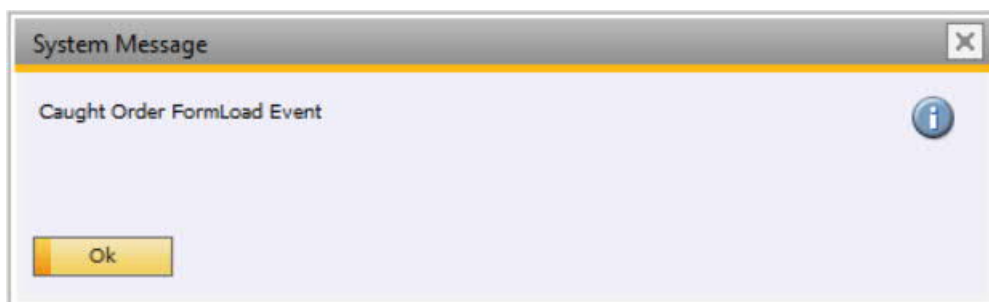
```
public static void SBO_Application_ItemEvent(string FormUID, ref
SAPboui COM. ItemEvent pVal, out bool BubbleEvent)
{
    BubbleEvent = true;
    if (pVal.FormTypeEx == "139" & pVal.BeforeAction == false &
pVal.EventType == SAPboui COM. BoEventTypes.et_FORM_LOAD)
    {}
}
```

- 1.2. Create event subscriber for the previously create function before the application run will be called in the **Main** function.

```
SBO_Application.ItemEvent += new
SAPboui COM. _IApplicationEvents_ItemEventEventHandler(SBO_Application_ItemEvent);
```

- 1.3. Display the message "Caught Order FormLoad Event" when the FormLoad event for the Order form arrives (use the Application.Message method).

```
SBO_Application.MessageBox("Caught Order FormLoad Event");
```



2. TASK - CATCH THE CLICK EVENT ON THE "RENT DVD" BUTTON

- 2.1. Catch the click event on the "Rent DVD" button you've added in the previous exercise. Again, display a message once the event is caught.

```
if (pVal.FormUID == "TB1_DVDAvailability" & pVal.BeforeAction == false &
pVal.ItemUID == "Rent" & pVal.EventType == SAPboui COM. BoEventTypes.et_ITEM_PRESSED)
{
    SBO_Application.MessageBox("Caught click on Rent DVD button");
} if (pVal.FormUID == "TB1_DVDAvailability" & pVal.ItemUID == "Rent" &
pVal.EventType == SAPboui COM. BoEventTypes.et_ITEM_PRESSED)
{
    SBO_Application.MessageBox("Caught click on Rent DVD button");
}
```



3. TASK - CREATE AN EVENT FILTER

3.1. Create a filter to only receive the events we are interested in.

```
public static void SetEventFilters()
{
    SAPboui COM. EventFilters oFilters;
    SAPboui COM. EventFilter oFilter;

    oFilters = new SAPboui COM. EventFilters();
    oFilter = oFilters.Add(SAPboui COM. BoEventTypes.et_ITEM_PRESSED);
    oFilter = oFilters.Add(SAPboui COM. BoEventTypes.et_FORM_LOAD);

    oFilter.AddEx("139"); //Orders Form
    oFilter.AddEx("TB1_DVDAvailability"); //DVD Availability Form
    SBO_Application.SetFilter(oFilters);
}
```

4. TASK - CREATE OBJECT LEVEL EVENT

4.1. Create a new function, which will extend the Cancel button with an ItemEvent. First the Form and the Item objects needs to be instantiated to the existing form.

```
SAPboui COM. Form oForm;
SAPboui COM. Button oButton;
oForm = SBO_Application.Forms.GetForm("TB1_DVDAvailability", 0);
SAPboui COM. Item oItem;
oItem = oForm.Items.Item("2");
oButton = (SAPboui COM. Button)oItem.Specific;
```

4.2. Assign to the oButton object the *ClickBefore* method. After typing the "**oButton.ClickBefore +=**" text you can press the TAB key to generate the event subscriber function automatically.

```
oButton.ClickBefore +=
```

OButton_ClickBefore; (Press TAB to insert)

```
oButton.ClickBefore += OButton_ClickBefore;
```

4.3. Display a message once the *Cancel* button is clicked in the event subscriber.

```
private static void OnButton_ClickBefore(object sboObject, SAPboui COM. SBObjectEventArg
pVal, out bool BubbleEvent)
{
    BubbleEvent = true;
    SBO_Application.MessageBox("Cancel button pressed.");
}
```

4.4. Extend the **SetEventFilters** function with the *Click* event type.

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

A solution implementing events catching can be found in the SDK UI samples (in the SDK Folder – see Appendix “SDK Installations” for more information), COM UI/ 02.CatchingEvents.



© 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.