# Exercises

## Unit: User Interface API

## Topic: Basics

At the conclusion of this exercise, you will be able to:

- Connect to a SAP Business One Application
- Display a MessageBox in SAP Business One
- Use Single Sign-On and the “Multi Add-On” feature
- React to AppEvents

You want to use the SAP Business One User Interface API for actively manipulating Process flow. As a first step you have to connect to the Application actually running.

1-1 Implement a connection to a running SAP Business One application.

1-1-1 Create a new Visual Basic project

1-1-2 Define the variables you need for a connection to a running SAP Business One application.

You will need at least two variables, one for the SboGuiApi object and one for the Application object

1-1-3 Connect to the SAP Business One SboGuiApi and get a handle to the running application.

1-2 Display a MessageBox within SAP Business One.

There is a Method of the Application object to display message boxes within SAP Business One

1-2-1 The method to display a MessageBox has several optional parameters. Check them out.

The lecture will continue after you have implemented this..; the remaining pieces of this exercise will be covered in the next steps.

1-3 Use the Single-Sign-On feature (and/or the Multiple Add-On feature) to connect to DI API as well.

1-4 Define the AppEvent handler – and implement the handling of these events (which are mandatory to be handled).

To define Event Handlers in Microsoft Visual Studio .NET please check the content on the drop-down comboboxes – which are displayed just above the source code…

# Exercises

## Unit: User Interface API

## Topic: Creating Forms

At the conclusion of this exercise, you will be able to:

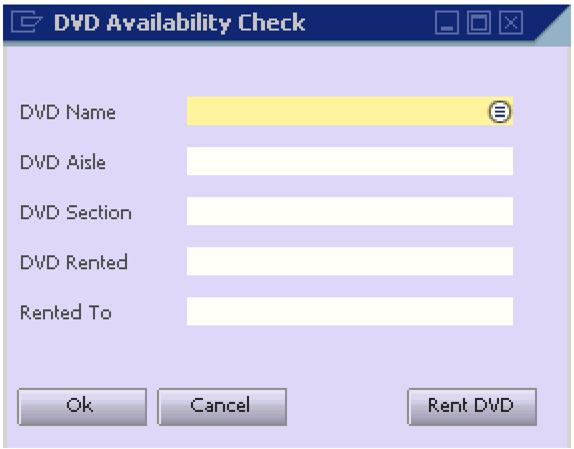
- Create a form within SAP Business One

You want to create a form which is displayed in the SAP Business One main window.

2-1 Create a new form within SAP Business One. The form should contain the following items:

- Input field for DVD Name (will be linked to a Choose from list)
- Input field for DVD Aisle
- Input field for DVD Section
- Input field for DVD Rented
- Input field for Rented To
- OK button
- Cancel button.
- Rent DVD button

This (the screenshot below) is the final goal, but you will only get data when you have gone through the “Databinding” lesson as well; in this exercise we will only focus on the form’s layout…



2-2 Enhance your program so that the form will be saved as an XML file.

2-3 Change your program. The form should now be loaded from the XML file you have created in the last step. Display the form in the SAP Business One window.

2-4 Use the tools from the B1TE toolset (essentially Form Checker) to check whether you have designed your form(s) according to some important UI guidelines…

## Some helpful data for designing Forms

## (See ScreenDesignGuidelines.pdf for more information!)

| | |
|---|---|
| **Form** | |
| Height | 413px |
| Width | 557px |
| **Controls common** | |
| Distance to left edge | 5px |
| Distance to right edge | 5px |
| Distance to top edge | 5px |
| Distance to bottom edge | 5px |
| **Button** | |
| Height | 19px |
| Width | 65px |
| Spacing | 5px |
| **Label Field** | |
| Height | 14px |
| Width | Depends on text |
| Horizontal spacing | >=12px (ungrouped) |
| Vertical spacing | 1px (grouped)<br>>=3px (ungrouped) |
| **Input Field** | |
| Height | 14px |
| Width | Enough to show complete field value |
| Horizontal Spacing | >=12px (ungrouped) |
| Vertical spacing | 1px (grouped)<br>>=3px (ungrouped) |
| **Field Help** | |
| Vertical distance to input field | 1px |
| **Matrix Objects** | |
| Width | Form width minus 2 times 5px to the left and right edge |
| Number of rows | <=7 rows, add scrollbars if more necessary |

# Exercises

## Unit: User Interface API

## Topic: Additional Event Handling

At the conclusion of this exercise, you will be able to:

- Handle SAP Business One events.

You want to actively influence the SAP Business One dialogues. Therefore you need to handle SAP Business One events.

You only want to receive the events you are interested in.

3-1 Catch FormLoad event for Order form

Look for the FormLoad event into the possible events thrown by SAP Business One (Application, Menu or Item events)

Have a look into the Event parameters to find out whether the FormLoad event is coming from the Order form or from another form

3-2 Display the message "Caught Order FormLoad Event" when the FormLoad event for the Order form arrives (use the Application.Message method).

3-3 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.

3-4 Create a filter to only receive the events we are interested in.

To set a filter on the UI events you should use the EventFilter object.

Don’t forget to assign the EventFilter to the application you are connected to.

# Exercises

## Unit: User Interface API

## Topic: New Menu entries in SAP Business One window

At the conclusion of this exercise, you will be able to:

- Add a menu entry in the SAP Business One main menu

The form of your Add-On solution should be displayed when the user chooses your new menu entry.

For that purpose you will create a new entry and handle the menu event.

4-1 Add a menu entry to the "Modules" menu called **DVD Store**. Those are the menu entries which are also displayed in the Main Menu. Add a new sub-menu to that menu entry called **DVD Availability**.

Ideally you use the specification for the menu of the “Course Project” example!

4-2 Handle the menu event: When you choose the menu entry, your form should be displayed.

4-3 Add another menu only visible when your form is open. This menu should appear under the GoTo menu.

Use the Menu collection property into the Form object to link the menus to your form

# Exercises

## Unit: User Interface API

## Topic: Data Binding

At the conclusion of this exercise, you will be able to:

- Bind data to fields of a form within the SAP Business One window.

You have created a new form that is displayed within SAP Business One. Now you want the system to display data on that form

5-1 Declare a DBDataSource and UserDataSource object. Link it to the form you created in the Creating Forms exercise.

5-2 Bind the form's items with the corresponding data from the User Defined table created in the DI exercises (*TB1_DVD)*.

Use the Method "DataBind.SetBound" on each item to assign it

- the corresponding table and field name it is associated to for the DBDataSources

5-3 Get the data from the data sources and display them in the corresponding fields on your form. Note you will first need to read the value selected by the user from the Choose from List (Item Event) and then fill all other fields accordingly.

Use the Query method into the DBDataSource object to filter the information you want to show

5-4 Test your application.