



SAP Business One 6.7 Features

@ Gali Moualem 28.4.04

6.7 Features

- **UI API Enhancements**
- **Permission Tree**
- **Add-On Administration**
- **User Defined Object**
- **DI Server (DIS)**
- **DI API new objects**
- **New SAP Business One And SDK Licensing**
- **Q+A**

UI Enhancements

- **Unique IDs for SAP Business One menus**
- **Menu for User Form**
- **Linked Button in matrix**
- **Delete column in Matrix**
- **New interface for Creating a form**
- **New objects: Progress Bar and Status Bar**
- **New interface for Activating the navigation button**
- **New methods and properties in DBDataSource object**
- **New methods in Matrix object**
- **Addition of Icon to menus**

How it will be

- **Description of the feature**
- **Why it is good enhancement**
- **How to implement it in Code**
- **Compatibility issues**
- **Sample**

Unique ID's for menus

Description of the feature

- Each SAB Business One menu will have unique ID
- The developer will be able add\update menu by unique ID and not position
- The developer will be able to set multiple menus by XML (batch operation)
- The developer will be able to manipulate menus by XML (batch operation)

Why it is good enhancement

- **Compatibility** The add-on menu position will not be affected of menu position changes in SAP Business One in future versions.
- **Performance** Ability to manipulate multiple menus with xml simultaneous
- **Better usability of code**

How To Do

Code for adding new menu

```
Dim oCreationPackage As SAPbouiCOM.MenuCreationParams (new object)
oCreationPackage = New SAPbouiCOM.MenuCreationParams()

oMenuItem = SBO_Application.GetMenuItemByID("43520") modules Unique ID
oCreationPackage.Type = SAPbouiCOM.BoMenuItemType.mt_SEPARATOR
oCreationPackage.UID = "MyMenu0"
oCreationPackage.AfterUID = "43545" Reports Unique ID
oCreationPackage.String = ""
oMenuItem.SubMenus().AddEx(oCreationPackage)

'// Add a pop-up menu item
oCreationPackage.Type = SAPbouiCOM.BoMenuItemType.mt_POPUP
oCreationPackage.UID = "MyMenu1"
oCreationPackage.AfterUID = "MyMenu0"
oCreationPackage.String = "Students Grades"
oMenuItem = oMenuItem.SubMenus().AddEx(oCreationPackage)

'// Get the menu collection of the newly added pop-up item
oCreationPackage.Type = SAPbouiCOM.BoMenuItemType.mt_STRING
oCreationPackage.UID = "MathGrade"
oCreationPackage.AfterUID = Nothing in case this is the first menu below
the popup
oCreationPackage.String = "Student Math Grades"
oMenuItem.SubMenus().AddEx(oCreationPackage)
```

How To Do

- How to get the system Unique ID's

SBO_Application.Menus.GetAsXML()

new method that will return XML with all menus unique ID

XML

- Set menus with XML – very recommend for setting multiple menus simultaneously to use batch menu setting

- Compatibility

- The old way for setting menus will work

Menus for User Form

Description of the feature

- **Add menu dynamically to user form**
- **The menu will be displayed in the application menu bar (under the go to) when the form will be in focus**
- **Will be supported in code & XML**

Why it is good enhancement

- **Give user form the ability to behave like system form – same “feel”**

How To Do

```
Dim oCreationPackage As SAPbouiCOM.MenuCreationParams (new object)  
oCreationPackage = New SAPbouiCOM.MenuCreationParams()
```

```
oCreationPackage.Check = False  
oCreationPackage.Enable = True  
oCreationPackage.Type = SAPbouiCOM.BoMenuType.mt_STRING  
oCreationPackage.UID = "MyFormMenu1"  
oCreationPackage.String = "MyFormMenu1"  
oForm.Menu.AddEx(oCreationPackage)
```

```
oCreationPackage.Check = False  
oCreationPackage.Enable = True  
oCreationPackage.Type = SAPbouiCOM.BoMenuType.mt_STRING  
oCreationPackage.UID = "MyFormMenu2"  
oCreationPackage.String = "MyFormMenu2"  
oForm.Menu.AddEx(oCreationPackage)
```

How To Do

- **Compatibility**
 - **The default will be like in previous versions (no menu for user form)**

Linked Button In matrix

Description of the feature

- **support adding link button column to user matrix**

Why it is good enhancement

- **Link button is a basic functionality in SAP Business One**
- **Give user form the ability to behave like system form – same “feel”**

Code for adding link button column to matrix

```
Set oltem = oForm.Items.Add("Matrix1", it_MATRIX)
```

```
oltem.Left = 5
```

```
oltem.Width = 400
```

```
oltem.Top = 5
```

```
oltem.Height = 130
```

```
Set oMatrix = oltem.Specific
```

```
Set oColumns = oMatrix.Columns
```

```
Set oColumn = oColumns.Add("F", it_LINKED_BUTTON) new type of column
```

```
oColumn.TitleObject.Caption = "Linked to object"
```

```
oColumn.Width = 40
```

```
oColumn.Editable = True
```

```
Set oLink = oColumn.ExtendedObject new property
```

```
oLink.LinkedObject = If_BusinessPartner
```

Delete Column

Description of the feature

- **Enable deletion of column in matrix in run time**

Why it is good enhancement

- **Manipulation of columns in matrix in run time**

Code for deleting columns (ask system & user)

```
Set oltem = oForm.Items.Add("Matrix1", it_MATRIX)
```

```
Set oColumn = oColumns.Add("G", it_LINKED_BUTTON)
```

```
oColumn.TitleObject.Caption = "Linked to none object"
```

```
oColumn.Width = 40
```

```
oColumn.Editable = True
```

```
Call oColumns.Remove("G") also possible with column index
```

New interface to create a form

Description of the feature

- **New interface for adding a form.**
- **This new interface is to support string type for the form type property and for supporting User Defined Object**

Why it is good enhancement

- **To support name space for form type**
- **Filtering events is on form type property**

Adding new form

Dim creationPackage As SAPbouiCOM.FormCreationParams **new
object**

```
creationPackage = New SAPbouiCOM.FormCreationParams()  
creationPackage.FormUid = "MathExamsID"  
creationPackage.Type = "SM_MathExam"
```

**creationPackage.ObjectType = "SM_MATHGRADES" **this property relevant for
UDO and for replacing system form****

```
creationPackage.WindowType = SAPbouiCOM.BoFormTypes.ft_Fixed
```

**oForm = SBO_Application.Forms.AddEx(creationPackage) **new method
AddEX for adding form****

**All objects in the UI API that deal with this property will have additional
property type string and additional method AddEx to support the new type**

Compatibility

- **The old way will work the same**
- **Creating User form for User defined Object will work only with the new interface**

New objects: progress bar & status bar

Description of the feature

- **The add-on will be able to activate the SAP Business One Progress Bar**
- **This add-on will be able to read messages of error and warning from the SAP Business One status bar**
- **New events will be expose to manipulate the 2 new objects**

Why it is good enhancement

- **Give user form the ability to behave like system form – same “feel” in case progress bar is need.**
- **Additional new objects in the User Interface**

Activating the progress bar

Start the progress bar

```
Dim m_oProgBar As SAPbouiCOM.ProgressBar new object
```

```
Set m_oProgBar = m_App.StatusBar.CreateProgressBar("My Progress Bar", 10,  
True)
```

```
m_oProgBar.Position = 0
```

Stop the progress bar

```
m_oProgBar.Stop  
Set m_oProgBar = Nothing
```

Forward\Backward cmds

```
Dim iPos As Integer  
iPos = m_oProgBar.Position  
m_oProgBar.Position = iPos - 1 / +1
```

Activate navigation button easily

Description of the feature

- **The add-on will be able to activate the SAP Business One navigation button easily**

Why it is good enhancement

- **Usability of code**

Activating the progress bar

```
oItem = oForm.Items.Add("BrsBtn",  
SAPbouiCOM.BoFormItemTypes.it_BROWSE_BUTTON) new form item type
```

```
oItem.LinkTo = "NameEdit1" link it to the field that will set what order of  
navigation will be
```

Compatibility

- The old way – enabling the buttons in the tool bar and setting the logic behind it will work.

New properties & methods in DBdataSource

Description of the feature

- **Add Fields()** – return the collection of the DBdataSource Fields
- **GetValue()** – get the value of the fields from the DBdataSource
- **SetValue()** – set the value of the field
- **InsertRecord** – insert new record to the DBdataSource
- **RemoveRecord** – remove a record from the DBdataSource

Why it is good enhancement

- **Usability** expose new methods and properties to ease the work with DBdataSource
- **Performance** – manipulate the values from the data source not the view

New methods in Matrix

Description of the feature

- **GetDataFromDataSource**
- **SetDataToDataSource**

Those methods copy the full matrix from the DataSource -> matrix (view) and via versa

Why it is good enhancement

- **Performance**
- **Usability of code**

Permission Tree in the SDK

Description of the feature

- **Ability Add User permission object to the SAP Business One Permission Tree**
- **Set the users permission for the user permission object**
- **DI – set\ get users permission for system permission objects**

Why it is good enhancement

- **Add-On will be able to add new permission objects to SAP Business One that will manage the permission for the add-on**
- **Add-On will be able to get\set the user permissions**

Create permission in the SAP Business One interface

DI code

```
'// create parent permission object  
Dim UserPermission1 As SAPbobsCOM.UserPermissionTree new object  
Set UserPermission1 = oCompany.GetBusinessObject(oUserPermissionTree)  
UserPermission1.PermissionId = "DI58testID"  
UserPermission1.Name = "DI58test"  
UserPermission1.Options = bou_FullReadNone  
UserPermission1.Levels = 1  
RetVal = UserPermission1.Add  
'// create son  
Dim UserPermission1 As SAPbobsCOM.UserPermissionTree  
Set UserPermission1 = oCompany.GetBusinessObject(oUserPermissionTree)  
UserPermission1.PermissionId = "DI58testIDSon"  
UserPermission1.Name = "DI58testIDSon"  
UserPermission1.Options = bou_FullNone  
UserPermission1.Levels = 2  
UserPermission1.ParentID = "DI58testID"  
UserPermission1.UserPermissionForms.FormType = "GL_LinkedButton"  
RetVal = UserPermission1.Add
```

DI code

```
Dim Users1 As SAPbobsCOM.Users
Set Users1 = oCompany.GetBusinessObject(oUsers)
RetVal = Users1.GetByKey(2)
oCompany.GetLastError RetVal, ErrMsg
Users1.UserPermission.PermissionId = "DI58testIDSon" new object in
user object
Users1.UserPermission.Permission = boper_Full
RetVal = Users1.Update

// 3 new bobs
1. Dim oSBObob As SAPbobsCOM.SBObob
oSBObob.GetObjectPermissions (oBudget)

2. oSBObob.GetSystemPermission(user, permissionID)

3. oSBObob.SetSystemPermission(user, permissionID, bo_Full)
```

Add-On administration

Description of the feature

- The Add-On Administration tool is designed to help administrators to easily deploy and manage add-on applications on end-users workstations.

The Add-On Administration allows you to:

- Register add-on
- Set the company and the user-specific preferences for the add-on.
- Monitor the add-on status

Why it is good enhancement

Enhance the current mechanism:

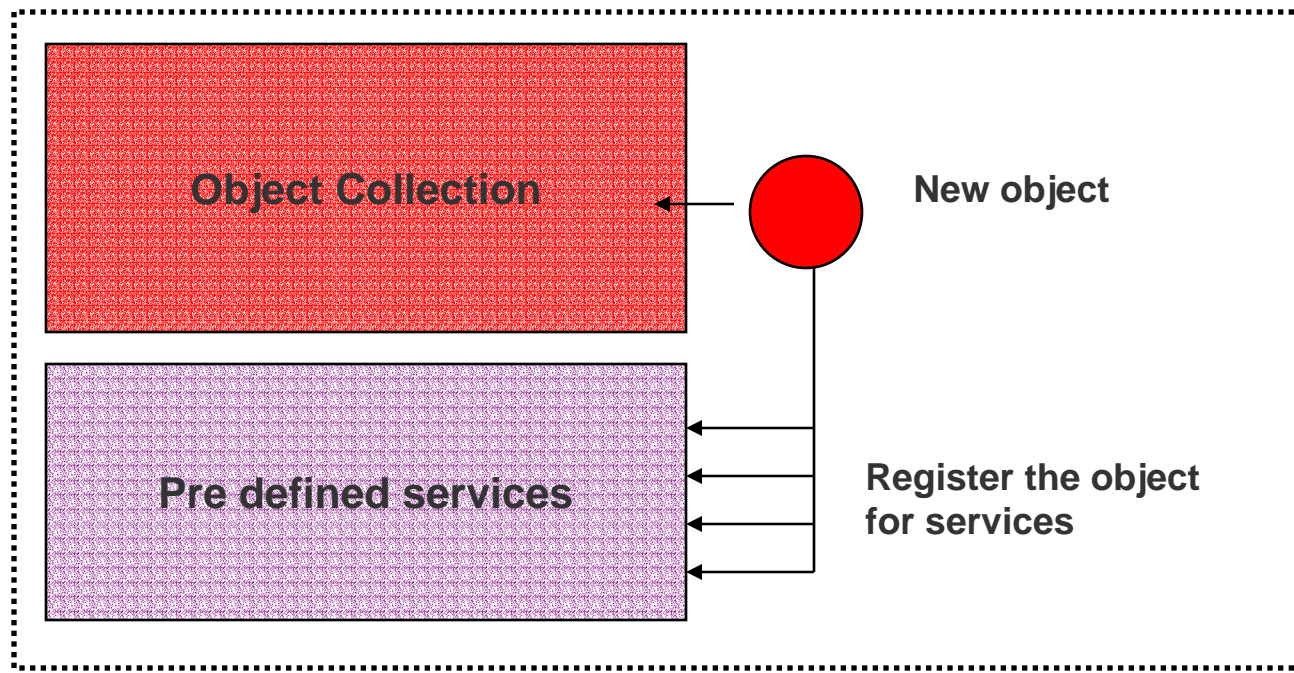
- Deployment in the server
- Monitor of add-ons
- Preference Setting per user & company

How To Do

Add new add-on company

User Defined Object

- Give partners the ability to add their own Business Object into the application object collection
- Give partners the ability to use set of services that application offers its objects for “free”



How to use this feature

Need to follow the below instruction

- Create new table (type: Master Data or Document)
- Write a c++ dll for the object (optional)
- Register your object with the wizard registration
- create a UI form (optional)

Description of the default behavior of the service

Service	Description
Add	Add new record of the object to DB (only fields that are linked to field in the DB)
Update	Update the fields of the object in DB (only fields that are linked to field in the DB)
Find	Support choose from list of the object
Cancel	Change the record status
Delete	MS –deleting record Doc - nothing
Manage Series	Relevant to document object. Adding to object to document numbering form. And managing the series for the object.
History	Create log table for the object and saving the history for this object
Default form	Create a default form for the object and that manages all the services.
Year transfer	Copying the tables and the record in year transfer operation

How to support the UDO in the UI API

- **Connect the form to the object**

creationPackage = New SAPbouiCOM.FormCreationParams() use the new interface for creating a form

creationPackage.FormUID = "MathExamsID"

creationPackage.Type = "SM_MathExam"

creationPackage.ObjectType = "SM_MATHGRADES" need to set the parameter with the object unique id

creationPackage.WindowType = SAPbouiCOM.BoFormTypes.ft_Fixed

oForm = SBO_Application.Forms.AddEx(creationPackage)

How to support the services in the UI API

Service	UI form support
Add\ update\ find (automatic event handling for // the Ok buttons)	oltem = oForm.Items.Add("1", SAPbouiCOM.BoFormItemTypes.it_BUTTON) oltem.Left = 5 oltem.Width = 65 oltem.Top = 325 oltem.Height = 19 oButton = oltem.Specific Do not set the caption of the button
Cancel	oltem = oForm.Items.Add("2", SAPbouiCOM.BoFormItemTypes.it_BUTTON) oltem.Left = 5 oltem.Width = 65 oltem.Top = 325 oltem.Height = 19 oButton = oltem.Specific Do not set the caption of the button

How to support the services in the UI API

Service	UI form support
Manage Series	<pre>// create a combo box for the series oItem=oForm.Items.Add("SeriesName",SAPbouiCOM.BoFormItemTypes.it_COMBO_BOX) // fill the combo with relevant series oComboBox.ValidValues.FillWithSeries(True, False, 0) new method oComboBox.DataBind.SetBound(True, "@MATH", "Series") // edit text the hold the numbering of the series oItem = oForm.Items.Add("SrValue", SAPbouiCOM.BoFormItemTypes.it_EDIT) oEditText.DataBind.SetBound(True, "@MATH", "DocNum") // get the next series number in add mode INum = oForm.BusinessObject.GetNextSeriesNumber(CLng(str)) oEditText = oForm.Items.Item("SrValue").Specific() oEditText.String = INum</pre>

Description of the feature

- **The DI server is an milestone towards migrating DI functionality from desktop client architecture to highly scalable server application.**

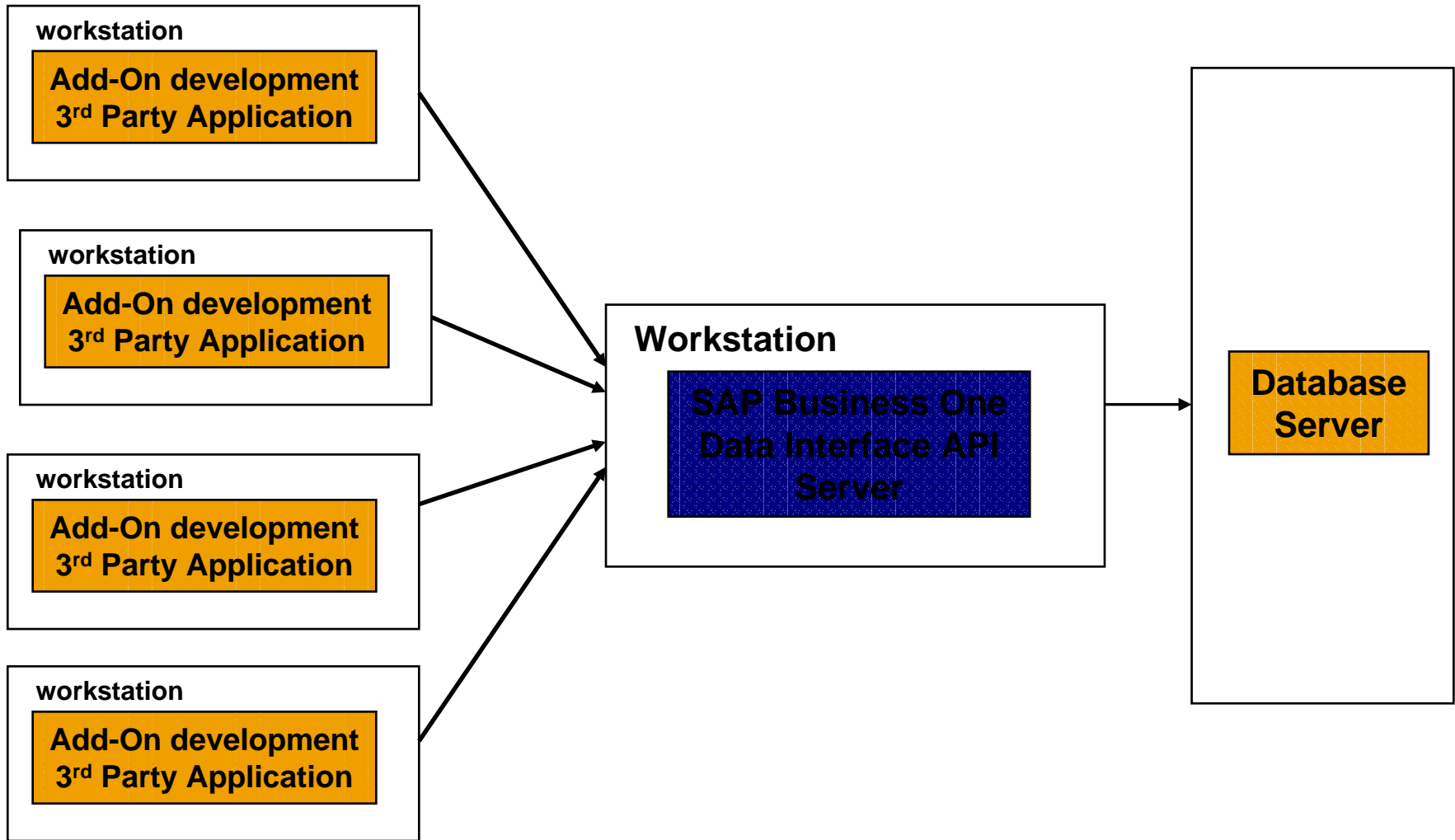
Such application might be:

- **Traditional Web application – ASP\JSP UI backed up by server business logic**
- **Web services – example for using the web services interface**
- **Heavy duty operation – like a batch operation**
- **Large number of Remote clients working at the same time**

The DI server is mostly enhancement of the DI in two main areas:

- **Scalability and Performance**
- **Neutral language interface not obligate to COM clients integration.**

DI Server



Samples

Web services

XML interface

New Objects in the DI API

ActivityStatus	OCLA
BPBankAccounts	OCRB
BPPaymentDates	CRD5
BPPaymentMethods	CRD2
BPWithholdingTax	CRD4
Budget	OBTG
Budget_Lines	BGT1
BudgetDistribution	OBTG
BudgetScenarios	OBTG
DunningLetters	ODUN
EmployeeRolesInfo	HEM6
ExtendUserPermTree	UPT1
Industries	OOND
ItemGroups	OITB
PackagesTyps	OPKG
Queue	OQUE
QueueMembers	QUE1
Relationships	OORL
SalesForecast	OFCT
SalesForecast_Lines	FCT1
SalesOpportunitiesInterests	OPR4
SalesOpportunityPartner	OPR2

New Objects in the DI API

SalesOpportunitiesReasons
SalesTaxAuthorities
SalesTaxAuthoritiesTypes
SalesTaxCodes
SalesTaxCodes_Lines
SpecialPricesDataAreas
SpecialPricesQuantityAreas
Territories
Warehouses
WithholdingTaxData

OPR5
OSTA
OSTT
OSTC
STC1
SPP1
SPP2
OTER
OWHS
INV5