

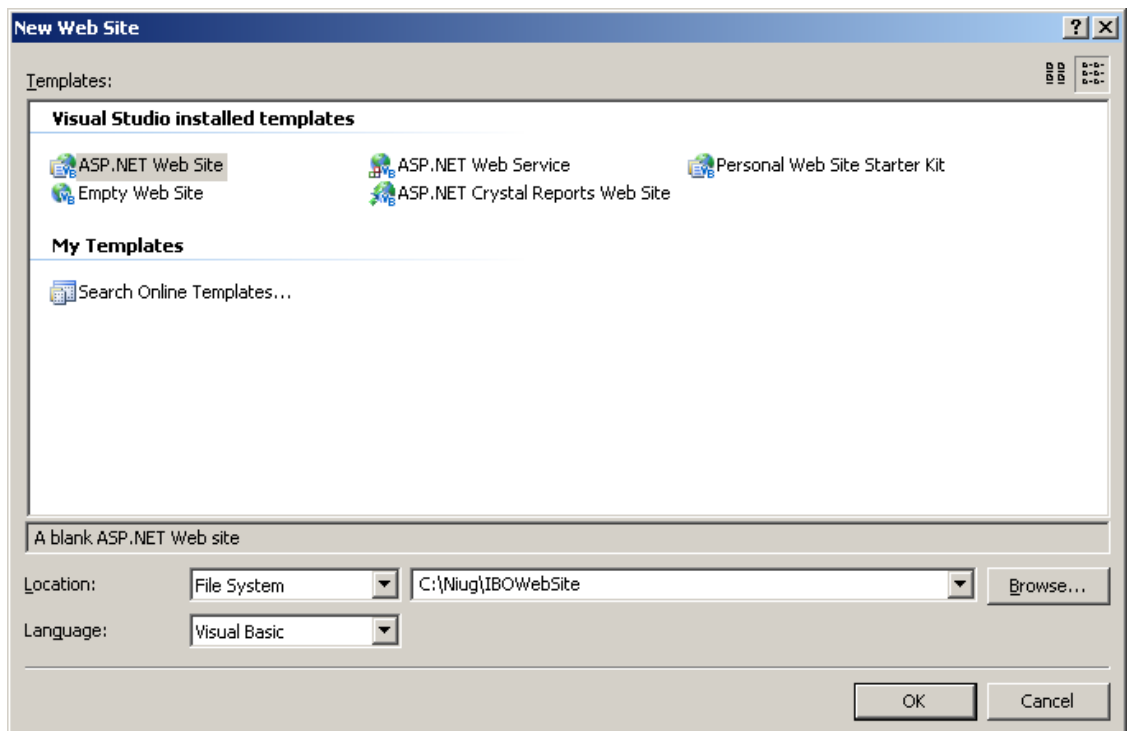
iBO Developer Session Workshop

NiUG Asia Pacific Discovery Conference 2009

1. Create a New Web Site

File -> New -> Web Site

The Location should be: File System C:\Niug\IBOWebSite
The Language should be: Visual Basic

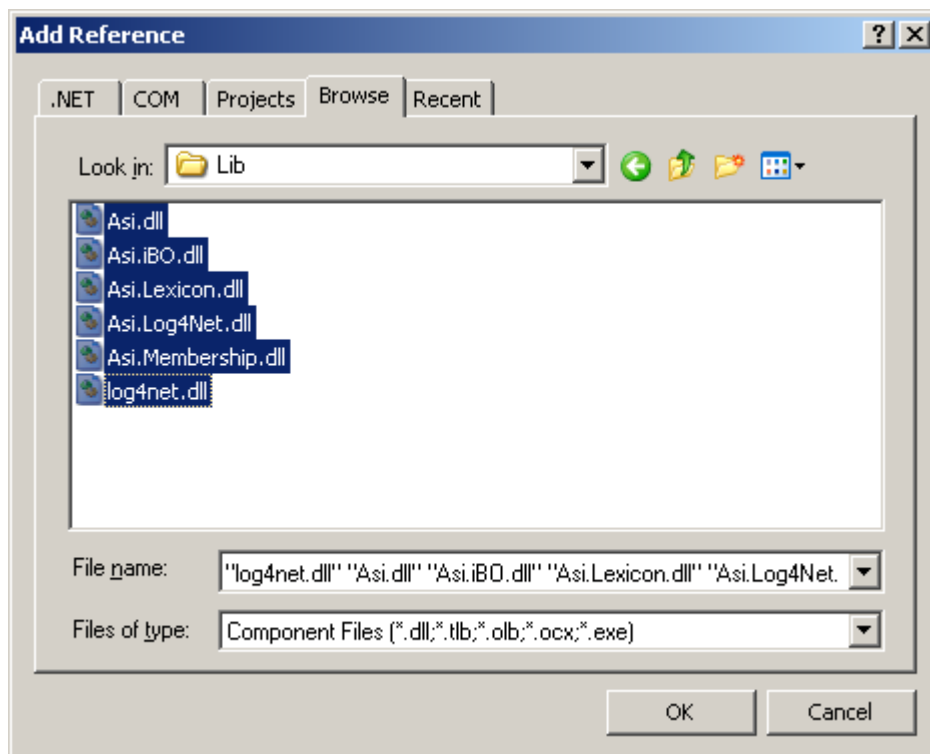


2. iBO Installation and Configuration

- a. Create a Lib folder in the root of the web site
- b. Copy the following dlls from C:\Program Files\ASI\iMIS15\Net\bin in the Lib folder
 - Asi.iBO.dll
 - Asi.dll
 - Asi.Lexicon.dll
 - Asi.Log4Net.dll
 - Asi.Membership.dll
 - log4net.dll



c. Add a reference to the required dlls in the website



d. Add the following entry in the web.config

- Add the URL to the iMIS15 application server in the web.config

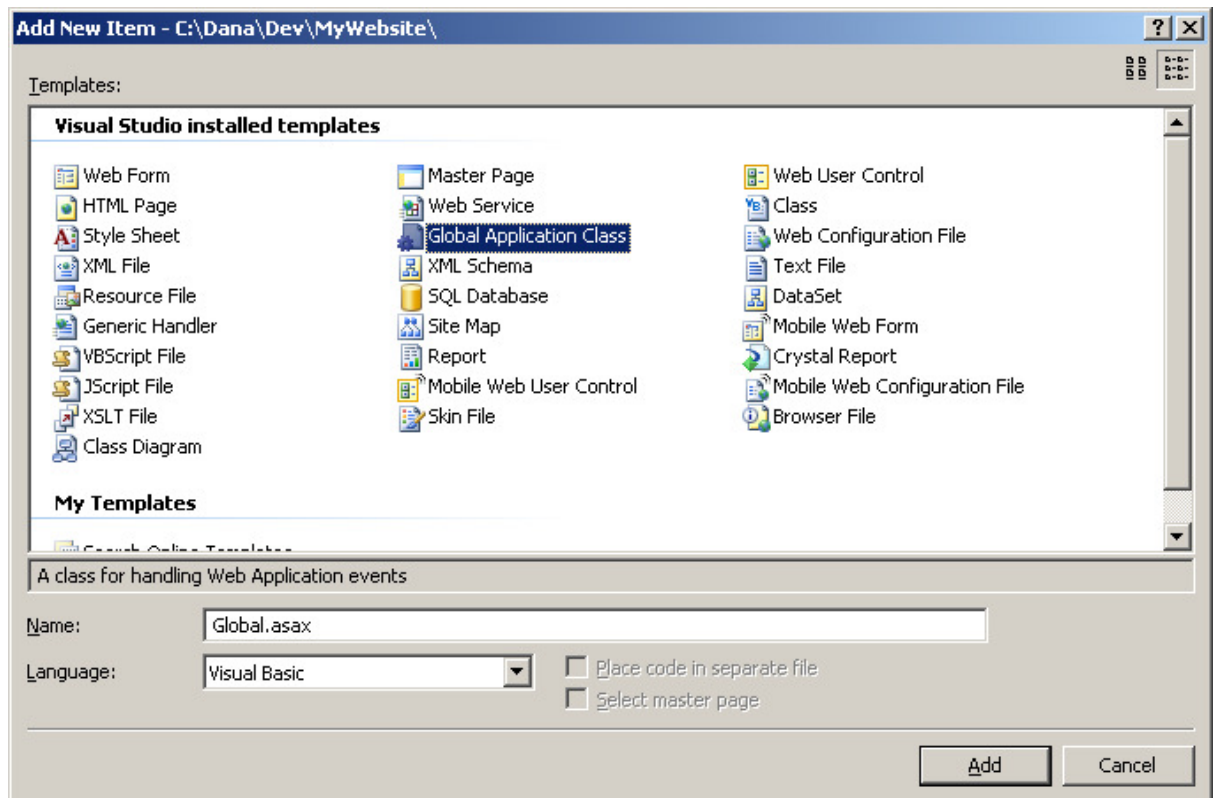
```
<appSettings>
  <add key="ImisWebServerUrl" value="http://localhost/imis15"/>
</appSettings>
```

- e. Add the database connection string in the web.config

```
<connectionStrings>
<add name="DataSource.iMIS.Connection"
      connectionString="Data Source=(local); User ID=sa;
      Password=password;Initial Catalog=iMIS" />
</connectionStrings>
```

3. iBO Initialisation

- a. Create a Global.asax file



- b. In Global.asax file import the Asi.iBO namespace

```
<%@ Import Namespace="Asi.iBO" %>
```

- c. Initialise the iBO in the Application_Start method

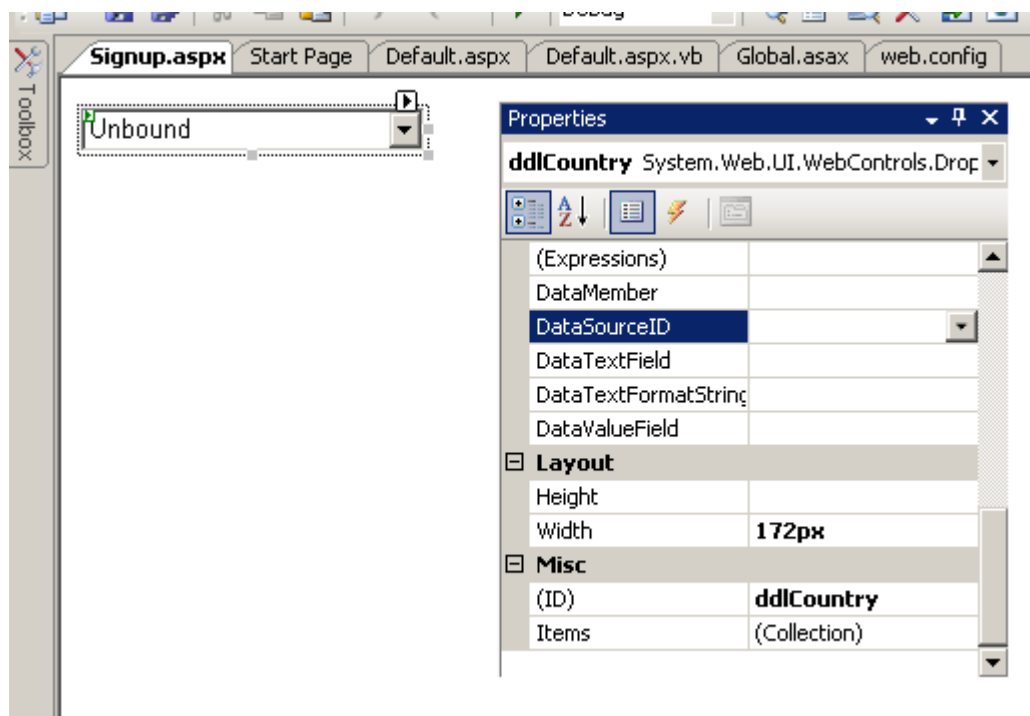
```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    iBoAdmin.InitializeSystem()
End Sub
```

4. Reference Data

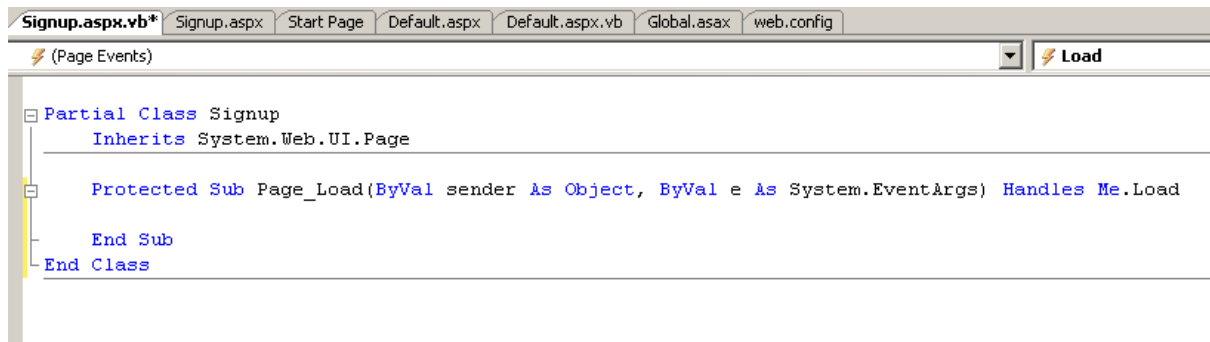
- a. Create a new Web Form, and name it Signup.aspx
- b. Import the Asi.iBO namespace at the top of the Signup.aspx.vb file

[Imports](#) Asi.iBO

- c. From the Toolbox drag a Drop Down List control, and set its ID to ddlCountry



- d. In Signup.aspx.vb create the stub for the Page_Load method by selecting (Page Events) and (Load)



- e. Populate the country drop down list with the list of countries retrieved via the iBO

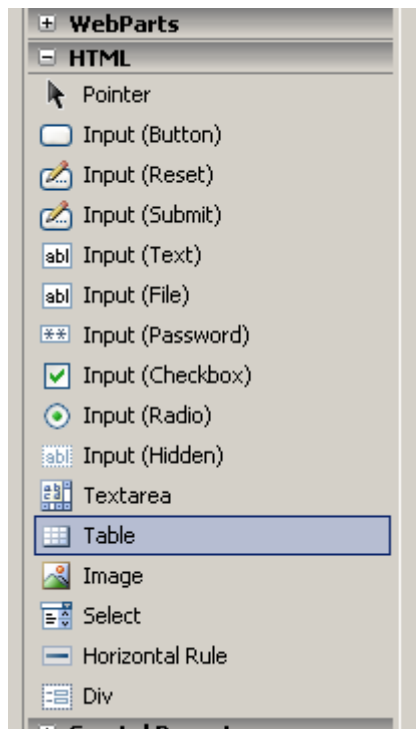
```

' Populate the Countries
Dim lookupCountries As SystemConfig.CCountry()
lookupCountries = iBOAdmin.ReferenceData.Countries
ddlCountry.DataSource = lookupCountries
ddlCountry.DataTextField = "CountryName"
ddlCountry.DataValueField = "CountryName"
ddlCountry.DataBind()

```

5. Signup Application

- a. Drag an HTML table from the toolbox



Adjust the table so it has 2 columns and 12 rows.

- b. Drag the following Label, TextBox, and Drop Down List controls and set the corresponding IDs

		ControlID
First Name:	<input type="text"/>	txtFirstName
Last Name:	<input type="text"/>	txtLastName
Address 1:	<input type="text"/>	txtAddress1
Address 2:	<input type="text"/>	txtAddress2
City:	<input type="text"/>	txtCity
Post Code:	<input type="text"/>	txtPostCode
State:	Unbound <input type="button" value="v"/>	ddlState
Country:	Unbound <input type="button" value="v"/>	ddlCountry
	<input type="button" value="Info"/>	lblInfo
	<input type="button" value="Submit"/>	

- c. Populate the list of states in the Page_Load method

```
Populate the States
Dim lookupStates As String()
lookupStates = iboAdmin.ReferenceData.StatesProvinces
ddlState.DataSource = lookupStates
ddlState.DataBind()
```

- d. Double click on the Submit button to create the btnSubmit_Click method
- e. Set the admin user on behalf of which we are going to singup users

- Add the following keys to the web.config

```
<add key="adminUsername" value="MANAGER" />
<add key="adminPassword" value="manager" />
```

- Add the following code to the Signup class

```
Partial Class Signup
    Inherits System.Web.UI.Page

    Dim adminUser As CStaffUser
```

- Add the following code to the Page_Load method

```
Set the admin user
Dim adminUsername, adminPassword As String
adminUsername = ConfigurationManager.AppSettings("adminUsername")
adminPassword = ConfigurationManager.AppSettings("adminPassword")
adminUser = CStaffUser.Login(adminUsername, adminPassword)

' Once the staff user is logged in, the level of exception handling
can be set

adminUser.ThrowExceptionOnError = True
```

f. Create a new contact

- Add the following code to the btnSubmit_Click method

```
' ccreate a new contact
Dim contact As ContactManagement.CContact
contact = New ContactManagement.CContact(adminUser)
```

g. Populate the contact from the UI controls

- Add the following key to the web.config

```
<add key="defaultCustomerTypeCode" value="M"/>
```

- Add the following code to the btnSubmit_Click method

```
' Populate the contact
Dim customerTypeCode As String
customerTypeCode =
ConfigurationManager.AppSettings("defaultCustomerTypeCode")
contact.FirstName = txtFirstName.Text
contact.LastName = txtFirstLast.Text
contact.CustomerTypeCode = customerTypeCode
```

h. Populate the default address

- Add the following code to the btnSubmit_Click method

```
' populate the default address
contact.DefaultAddress.Address1 = txtAddress1.Text
contact.DefaultAddress.Address2 = txtAddress2.Text
contact.DefaultAddress.City = txtCity.Text
contact.DefaultAddress.PostalCode = txtPostCode.Text
contact.DefaultAddress.StateProvince = ddlState.Text
contact.DefaultAddress.Country = ddlCountry.Text
```

i. Insert the new contact in the database

- Add the following code to the btnSubmit_Click method

```
' insert the contact
contact.Save()
```

```
lblInfo.Text = "Sign Up successful ContactId = " + contact.ContactId
```

j. Handle Errors

- Wrap the code in the btnSubmit_Click in a Try/Catch block

```
Try
```

```
...
```

```
Catch ex As Exception
```

```
lblInfo.Text = "Sign Up unsuccessful " + ex.Message
```

```
End Try
```

6. Raise Dues

```
' sample code
```

```
Dim contactJoinDate As DateTime
```

```
Dim Now As DateTime
```

```
contactJoinDate = contact.JoinDate
```

```
contact.JoinDate = Now
```

```
contact.Save()
```

```
contact.BillDuesSubscriptions(Now)
```

```
contact.JoinDate = contactJoinDate
```

```
contact.Save()
```

7. Pay Dues

```
Dim contact As ContactManagement.CContact
```

```
contact = New ContactManagement.CContact(adminUser,  
txtContactId.Text)
```

```
' get the outstanding subs
```

```
Dim subscriptions As Commerce.CSubscription()
```

```
subscriptions = contact.Subscriptions
```

```
Dim outstandingSubscriptions As List(Of Commerce.CSubscription)
```

```
outstandingSubscriptions = New List(Of Commerce.CSubscription)
```

```
' construct payment
```

```
Dim payment As Financials.CPayment
```

```
payment = New Financials.CPayment(adminUser)
```

```
Dim totalToPay As Decimal
```

```
Dim subscription As Commerce.CSubscription

For Each subscription In subscriptions
If (subscription.Balance > 0) Then
subscription.AmountPaid = subscription.Balance
outstandingSubscriptions.Add(subscription)
totalToPay += subscription.Balance
End If
Next

payment.Amount = totalToPay
payment.CreditDebitCardAuthorizationCode = "TEST000"
payment.CashAccountCode = "VISA"
payment.CreditDebitCardHoldersName = "Dana Visescu"
payment.CreditCardNumber = "4111111111111111"
payment.CreditCardExpiration = "01/10"

Dim batch As Financials.CBatch
batch = New Financials.CBatch(adminUser)

Commerce.CSubscription.PayDues(adminUser,
outstandingSubscriptions.ToArray(), payment, batch)
```

Appendix 1: Source Code

```
Imports Asi.iBO
Imports System.Data.SqlClient
Imports System.Collections.Generic

Partial Class _Default
Inherits System.Web.UI.Page

Dim adminUser As CStaffUser

Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

' Set the admin user
Dim adminUsername, adminPassword As String
adminUsername = ConfigurationManager.AppSettings("adminUsername")
adminPassword = ConfigurationManager.AppSettings("adminPassword")

adminUser = CStaffUser.Login(adminUsername, adminPassword)

' Once the staff user is logged in, the level of exception handling
can be set
adminUser.ThrowExceptionOnError = True

If (Not Page.IsPostBack()) Then

' Populate the Countries
```

```
Dim lookupCountries As SystemConfig.CCountry()
lookupCountries = iboAdmin.ReferenceData.Countries
ddlCountry.DataSource = lookupCountries
ddlCountry.DataTextField = "CountryName"
ddlCountry.DataValueField = "CountryName"
ddlCountry.DataBind()

' Populate the States
Dim lookupStates As String()
lookupStates = iboAdmin.ReferenceData.StatesProvinces
ddlState.DataSource = lookupStates
ddlState.DataBind()

End If

End Sub

Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click

Try

' construct a new contact
Dim contact As ContactManagement.CContact
contact = New ContactManagement.CContact(adminUser)

' populate the contact
Dim customerTypeCode As String
customerTypeCode =
ConfigurationManager.AppSettings("defaultCustomerTypeCode")

contact.FirstName = txtFirstName.Text
contact.LastName = txtFirstLastName.Text
contact.CustomerTypeCode = customerTypeCode

' populate the default address
contact.DefaultAddress.Address1 = txtAddress1.Text
contact.DefaultAddress.Address2 = txtAddress2.Text
contact.DefaultAddress.City = txtCity.Text
contact.DefaultAddress.PostalCode = txtPostCode.Text
contact.DefaultAddress.StateProvince = ddlState.Text
contact.DefaultAddress.Country = ddlCountry.Text

' insert the contact
contact.Save()

lblInfo.Text = "Sign Up successful ContactId = " + contact.ContactId

Catch ex As Exception

lblInfo.Text = "Sign Up unsuccessful " + ex.Message

End Try

End Sub
```

```
Protected Sub btnRaiseDues_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnRaiseDues.Click
```

```
Dim contact As ContactManagement.CContact  
contact = New ContactManagement.CContact(adminUser,  
txtContactId.Text)
```

```
Dim contactJoinDate As DateTime  
Dim Now As DateTime
```

```
contactJoinDate = contact.JoinDate  
contact.JoinDate = Now  
contact.Save()
```

```
contact.BillDuesSubscriptions(Now)
```

```
contact.JoinDate = contactJoinDate  
contact.Save()
```

```
End Sub
```

```
Protected Sub btnPayDues_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnPayDues.Click
```

```
Dim contact As ContactManagement.CContact  
contact = New ContactManagement.CContact(adminUser,  
txtContactId.Text)
```

```
' get the outstanding subs  
Dim subscriptions As Commerce.CSubscription()  
subscriptions = contact.Subscriptions
```

```
Dim outstandingSubscriptions As List(Of Commerce.CSubscription)  
outstandingSubscriptions = New List(Of Commerce.CSubscription)
```

```
' construct payment  
Dim payment As Financials.CPayment  
payment = New Financials.CPayment(adminUser)
```

```
Dim totalToPay As Decimal  
Dim subscription As Commerce.CSubscription
```

```
For Each subscription In subscriptions  
If (subscription.Balance > 0) Then  
subscription.AmountPaid = subscription.Balance  
outstandingSubscriptions.Add(subscription)  
totalToPay += subscription.Balance  
End If  
Next
```

```
payment.Amount = totalToPay  
payment.CreditDebitCardAuthorizationCode = "TEST000"  
payment.CashAccountCode = "VISA"  
payment.CreditDebitCardHoldersName = "Dana Visescu"  
payment.CreditCardNumber = "4111111111111111"  
payment.CreditCardExpiration = "01/10"
```

```
Dim batch As Financials.CBatch
```

```
batch = New Financials.CBatch(adminUser)

Commerce.CSubscription.PayDues(adminUser,
outstandingSubscriptions.ToArray(), payment, batch)

    End Sub
End Class
```