Customize Web Application Template

Minimal Effort

Use existing ArcGIS Server samples and scripts to add more functionality

By Kuang Yao Lee, Geo-Objects, LLC

This article proposes a simple approach for customizing the basic ESRI Web application template that is created using ArcGIS Server Manager. It uses the existing ESRI sample code provided with the ArcGIS Server installation media and from ESRI online support information such as ArcScripts (arcscripts.esri.com) and the ESRI Developer Network (edn.esri.com).

The default Web application template created using ArcGIS Server Manager performs standard mapping operations such as Zoom In, Zoom Out, Pan, Full Extent, Identify, and Measure. Often additional functionality is needed and time-consuming coding is required to add advanced mapping functions.

This article describes how three tools can be added into the Web application toolbar by programmatically modifying the template. These tool controls are SelectByRectangle, SelectByShape, and Clear Selection. The function/subroutine code required for these custom controls can be found from either the ESRI Support Center (support.esri.com) or the sample codes in the ArcGIS Server Installation.

Assumptions and Requirements

This exercise assumes that the development environment in the host machine contains proper licenses for ArcGIS Server 9.2 (.NET) , ArcGIS Desktop 9.2, and Microsoft Visual Studio 2005. You can download the listings referenced in this article from *ArcUser Online* (www.esri.com/news/arcuser). The data served in the Web application can be stored in shapefiles, file geodatabases, personal geodatabases, or enterprise geodatabases, but this demonstration imports all the layers into ArcSDE using Microsoft SQL Server 2005. This exercise was written for information-sharing purposes. Similarities to other ESRI Web-based applications and datasets may occur as a result of using ESRI's tutorial template and datasets.

What You Will Need
- ArcGIS 9.2 Server
- ArcGIS 9.2 ArcSDE
- Microsoft SQL Server 2005
- Visual Studio 2005
- Listings downloaded from *ArcUser Online*

Create a Map Service and a Web Application Using ArcGIS Server Manager

1. Go to the location of the ArcTutor sample data (<drive>:\ArcGIS\ArcTutor\Map) and inspect airport.mxd and airport.gdb.
2. Examine the layers in airport.mxd. Import these layers, originally stored in the file geodatabase (airport.gdb), into a SQL Server geodatabase through a database connection in ArcCatalog.
3. In airport.mxd, update the data sources for all the layers in airport.mxd to those layers now in the SQL Server geodatabase.
4. Use ArcGIS Server Manager to publish airport.mxd as a MapService named AirportService.
5. Create a Web Application named AirportApp.
6. Add AirportService as a layer in the Web application through a local connection to your GIS Server.
7. Accept all defaults when creating the Web application.

Add Two Tool Items and One Command Item

1. Use Visual Studio 2005 to open the Web Site application, AirportApp, by browsing to the directory <drive>:\Inetpub\wwwroot\AirportApp, which was just created.
2. Once the Web project is loaded into Visual Studio, open the ASP.NET page, Default.aspx, in Design mode.
3. Right-click on ESRI Toolbar1 (i.e., ESRI.ArcGIS.ADF.Web.UI.WebControls.Toolbar) on Default.aspx to open the property table for the toolbar.
4. Click on (Collection) of Toolbar items and add three custom buttons named SelectByRectangle, SelectByShape, and ClearSelection. SelectByRectangle and SelectByShape are tool items that require interacting with the map (or Map Control) while ClearSelection is a command item that does not interact with the map.

Add a Floating Panel to Show Selected Features

1. Add a Floating Panel into Default.aspx from ArcGIS Web Control in the Visual Studio ToolBox. This panel will serve as a container that will hold a GridView control to show a features attributes.
2. Add an unbound DropDownList from Visual Studio ToolBox into Default.aspx in the upper-right panel where SiteMapDataSource is located. This will provide the ability to set a selectable layer.
3. Add the two statements in Listings 1 and 2 (refer to downloaded listings) to clear and populate the DropDownList in the Page_Load event of Default.aspx. This ensures that activation of the event will refresh the selectable layers when the page is loading.
4. Create a check box (Name: ViewTableCheckBox1; TEXT: ViewTable) on Default.aspx. This control will display or hide the attribute table for selected features. Add an event handler for the controls selected index changed event (2).

Copy an Existing Class for Server-Side Action from a Sample

1. Copy CustomTools.vb from one of the ESRI ArcGIS Server .NET sample codes to youíre applications app_code folder. This sample can be found at <drive>:\ProgramFiles\ArcGIS\DeveloperKit\SamplesNET\Server\Web_Applications\ArcGIS_SelectBufferToolVBNet.zip. Modify the object variables so they will respond properly to the Web control names.
2. Find the SelectToolNew class in CustomTools.vb.
3. Rename the class to SelectTool and modify the code in response to the Map Action.
4. Click on (Collection) of Toolbaritems.
5. Type CustomTools.SelectTool in the input box for SelectbyRectangle next to the column of ServerActionClass.
6. Type App_code in the input box next to the column of ServerActionAssembly.

SUMMARY

Add advanced functionality to the basic ArcGIS Server Web Application template with minimal coding using readily available resources from ESRI. This article assumes the reader is familiar with ArcGIS Server, ArcGIS Desktop, and Microsoft Visual Basic Studio 2005.

7. Set ClientAction to DragRectangle
8. Adjust other properties based on the application design or preferences.
9. Copy, paste, and rename the SelectTool class to SelectByShape in CustomTools.vb.
10. Modify the code in response to the polygon drawing on the map.
11. For SelectByShape, set ClientAction to Polygon.
12. Follow the same procedure for the ClearSelection command. Note that the ClearSelection code is not shown due to limited space.

Test the Application

1. Press F5 in Visual Studio 2005 to run the application.
2. **Check the check box ViewTable so that the attribute table shows in the GridView in the Floating Panel.**
3. Click on the SelectByRectangle tool item on the Toolbar.
4. Draw a rectangle on the map to select some features.

Conclusion

A GIS Web developer can readily develop custom applications by modifying existing ESRI ArcGIS Server templates. Coding can be minimized by using resources such as the ArcGIS Server code examples found in the DeveloperKit folder in the ArcGIS installation directory, samples from the ESRI Developer Network (EDN), and suggestions and code obtained from members of the ESRI Support Centerís user forums.

For more information, contact

Kuang Yao Lee, GIS Manager
Geo-Objects, LLC
1780 Corsica Drive
Wellington, FL 33414
E-mail: geoobjects@yahoo.com

About the Author

Kuang Yao Lee earned a doctorate in environmental science with a focus on GIS and watershed modeling from the MEES Program at the University of Maryland. Since 1995, he has been involved in development of GIS applications including desktop, Web-based, and mobile applications.

Acknowledgments

This article was reviewed by Dr. Thomas R. Fisher, professor at the University of Maryland Center for Environmental Science.

Listing 1

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
EventArgs) _
            Handles MyBase.Load

        If Not Page.IsCallback And Not Page.IsPostBack Then
            '...The original ESRI code is omitted in this
            section

            '...Clear the Dropdownlist each time when page
is loading
            Me.ActiveLayerList.Items.Clear()

            '...Call function to populate all the layer
            names into an unbound dropdownlist
            PopulateAlltheLayersInDropDownList()

        End If
End Sub
```

'In the event of Page_Load (Default.aspx), two statements are added to clear and refresh layers in Dropdownlist

Listing 2

```
Protected Sub PopulateAlltheLayersInDropDownList()
'...Populate all the selectable feature layers into the
DropDownList
'...There is only one selectable layer allowed during
spatial select
'...Author: Kuang Yao Lee, Oct 2007

 Try
      '...Create a serverContext using the existing Map
Service
      '...Set the string strGISServer to your GIS Server
Host Name

   Dim pGISServerConnection As IGISServerConnection = New _
      GISServerConnection
   pGISServerConnection.Connect(strGISServer)
   Dim pServerObjectManager As IServerObjectManager = _
      pGISServerConnection.ServerObjectManager
   Dim pMapcontext As IServerContext = _
pServerObjectManager.CreateServerContext("AirportService",
"MapServer")
   '...Reference to the Map Document and Layers in the Map
Service
```

```
    Dim ms As ESRI.ArcGIS.Carto.IMapServer = _
            CType(pMapcontext.ServerObject, ESRI.ArcGIS.Carto.IMapServer)
    Dim mso As ESRI.ArcGIS.Carto.IMapServerObjects = _
            CType(ms, ESRI.ArcGIS.Carto.IMapServerObjects)
    Dim map As ESRI.ArcGIS.Carto.IMap = mso.Map(ms.DefaultMapName)
    Dim pLayer As ESRI.ArcGIS.Carto.ILayer

    '...Looping through all the layers including the ones in grouplayers
    Dim pEnumLayer As ESRI.ArcGIS.Carto.IEnumLayer
    pEnumLayer = map.Layers(Nothing, True)
    pEnumLayer.Reset()
    pLayer = pEnumLayer.Next

    '...Create an instance object for FeatureLayer
    Dim pFeautreLayer As ESRI.ArcGIS.Carto.IFeatureLayer = _
            pMapcontext.CreateObject("esriCarto.FeatureLayer")

    '...Populate all the selectable Feature layers from Map Document
    Do While Not pLayer Is Nothing
        If TypeOf pLayer Is ESRI.ArcGIS.Carto.IFeatureLayer Then
                pFeautreLayer = pLayer
                If pFeautreLayer.Selectable = True Then
                    Me.ActiveLayerList.Items.Add(pLayer.Name)
                End If
        End If
        pLayer = pEnumLayer.Next
    Loop

    '...Release pointer to avoid locking issues
    pMapcontext.ReleaseContext()
    pMapcontext = Nothing: ms = Nothing: mso = Nothing: map = Nothing
    pLayer = Nothing: pEnumLayer = Nothing: pFeautreLayer = Nothing

    '...the default value is the first layer
    Me.ActiveLayerList.SelectedValue = 0

    '...Put the Selected Value in the DropDownList into a session variable
    Session("ActiveLayer") = Me.ActiveLayerList.SelectedValue

  Catch
    '...Intentionally not to prompt any messages if there is an error
  End Try

End Sub
'A subroutine is used to populate all the active layers in Dropdownlist from a Map Service

Protected Sub ActiveLayerList_SelectedIndexChanged(ByVal sender As _
      Object, ByVal e As System.EventArgs) Handles _
      ActiveLayerList.SelectedIndexChanged

Session("ActiveLayer") = Me.ActiveLayerList.SelectedValue

End Sub

'A handler for the dropdownlist SelectedIndexChanged event updates the "ActiveLayer" session variable
```

Listing 3

```
'...NameSpace – CustomTools.vb
Namespace CustomTools

'...Select Tool Class in repsonse to Spatial Select Event on the Map
Public Class SelectByRectangle
        Implements IMapServerToolAction

Sub ServerAction(ByVal args As ToolEventArgs) _
            Implements IMapServerToolAction.ServerAction

'...Referencing Object - mapctrol to Map Control on Default.aspx
Dim mapctrl As ESRI.ArcGIS.ADF.Web.UI.WebControls.Map = _
            CType(args.Control, ESRI.ArcGIS.ADF.Web.UI.WebControls.Map)

'...session variable from Default.aspx stores the name of the activelayer for spatial
selection
Dim targetlayername As String = _
            CStr(mapctrl.Page.Session("ActiveLayer"))

'...Referecing rectargs to the rectangle drawn on the map control
Dim rectargs As RectangleEventArgs = _
            CType(args, RectangleEventArgs)

Dim myrect As System.Drawing.Rectangle = rectargs.ScreenExtent

'...Record the lower-left corner point
Dim minpnt As ESRI.ArcGIS.ADF.Web.Geometry.Point = _
              ESRI.ArcGIS.ADF.Web.Geometry.Point.ToMapPoint(myrect.Left,  myrect.Bottom,
              mapctrl.Extent, CInt(mapctrl.Width.Value), CInt(mapctrl.Height.Value))

'...Record the upper-right corner point
Dim maxpnt As ESRI.ArcGIS.ADF.Web.Geometry.Point = _
            ESRI.ArcGIS.ADF.Web.Geometry.Point.ToMapPoint(myrect.Right, myrect.Top,
            mapctrl.Extent, CInt(mapctrl.Width.Value), CInt(mapctrl.Height.Value))

'...Convert rectangle into polygon in ADF
Dim mappoly As New ESRI.ArcGIS.ADF.Web.Geometry.Envelope(minpnt, maxpnt)

'...ESRI.ArcGIS.ADF.Web.DataSources.ArcGISServer
Dim ags_mf As MapFunctionality = Nothing
Dim gisFunctionality As IGISFunctionality = Nothing

'...Loop through all the functionalites; not Graphic Layer or others
For Each gisFunctionality In mapctrl.GetFunctionalities()
      If TypeOf gisFunctionality Is MapFunctionality Then
         ags_mf = CType(gisFunctionality, MapFunctionality)
           Exit For
      End If
Next gisFunctionality

'...Get dataframe from MapService
Dim gisResource As ESRI.ArcGIS.ADF.Web.DataSources.IGISResource = _
gisFunctionality.Resource

Dim supported As Boolean = gisResource.SupportsFunctionality( _
      GetType(ESRI.ArcGIS.ADF.Web.DataSources.IQueryFunctionality))
```

```
If supported = False Then Exit Sub

'...Get all the queryable layers to String arrays of layerIDs and layerNames
'...Using Namespace - ESRI.ArcGIS.ADF.Web.DataSources
Dim layerIDs As String() = Nothin
Dim layerNames As String() = Nothing
Dim queryFunctionality As IQueryFunctionality = CType(gisResource.CreateFunctionality(Get
Type(ESRI.ArcGIS.ADF.Web.DataSources.IQueryFunctionality), Nothing), ESRI.ArcGIS.ADF.Web.
DataSources.IQueryFunctionality)
            queryFunctionality.GetQueryableLayers(Nothing, layerIDs, layerNames)


'...Get Map description
Dim mapdescription As ESRI.ArcGIS.ADF.ArcGISServer.MapDescription = ags_
mf.MapDescription()

'...Get Layer Description
Dim layerdescs() As ESRI.ArcGIS.ADF.ArcGISServer.LayerDescription = mapdescription.
LayerDescriptions

'...Unselect all the selections of all the layers if there is any
Dim i As Integer = 0
Do While i < layerNames.Length
     layerdescs(i).SelectionFeatures = Nothing
     i += 1
Loop

'...Define spatial query using the geometry from the map draw event
Dim spatialFilter As ESRI.ArcGIS.ADF.Web.SpatialFilter = _
      New ESRI.ArcGIS.ADF.Web.SpatialFilter()
spatialFilter.ReturnADFGeometries = True
spatialFilter.MaxRecords = 100
spatialFilter.Geometry = mappoly

Dim activeLayerString As String = targetlayername

i = 0
Dim MyActiveLayerIndex as Long

'...Loop through all the queryable layers in dataframe in MapService
Do While i < layerNames.Length

      '...Get the active layer
      If layerNames(i).Equals(activeLayerString) Then MyActiveLayerIndex = i
      '...Exit Do if the activelayer is found and show the table
           Exit Do
      End If
      i += 1
Loop

'...perform spatial query and list selected features in the table
'...Define QueryTables and queryResultsDataTable as result tables
Dim QueryTable2 As New System.Data.DataTable
Dim queryResultsDataTable As New System.Data.DataTable
queryResultsDataTable = queryFunctionality.Query(gisFunctionality.Name, layerIDs(i),
spatialFilter)

If queryResultsDataTable.Rows.Count = 0 Then Exit Sub
```

```
QueryTable2 = queryResultsDataTable

'...get column of OBJECTID and SHAPE
Dim pObjectIDColum As Integer = 0
Dim dr As Data.DataRow = queryResultsDataTable.Rows(0)
Dim pIndex As Integer = 0


For Each g As System.Data.DataColumn In dr.Table.Columns
    '...Get the Colum Index of OBJECTID or FID
   If g.Caption.ToUpper = "OBJECTID" Or g.Caption.ToUpper = "FID" Then
           pObjectIDColum = pIndex
           Exit For
   End If
   pIndex += 1
Next g

'...get the FIDSet from OBJECTID or FID column
Dim j As Long = 0
Dim dr_Rows As Data.DataRow() = queryResultsDataTable.Select(Nothing)
Dim pFID(dr_Rows.Length - 1) As Integer

'...Store all the OBJECTID or FID into pFID array
For Each dr_Row As System.Data.DataRow In dr_Rows
       Dim temp_val = (Convert.ToInt32(dr_Row.Item(pObjectIDColum)))
       pFID(j) = temp_val
       j += 1
Next

'...Build the array of objectids
Dim fids As ESRI.ArcGIS.ADF.ArcGISServer.FIDSet = _
      New ESRI.ArcGIS.ADF.ArcGISServer.FIDSet()

Dim ids() As Integer = pFID
fids.FIDArray = ids

'...Define the color for selected features
Dim irgbc As ESRI.ArcGIS.ADF.ArcGISServer.RgbColor = _
      New ESRI.ArcGIS.ADF.ArcGISServer.RgbColor()
irgbc.Red = 0
irgbc.Green = 255
irgbc.Blue = 255

Dim pColor As ESRI.ArcGIS.ADF.ArcGISServer.Color
pColor = irgbc

'...Define selected features and the color in layerdescription object
Dim layerDesc As ESRI.ArcGIS.ADF.ArcGISServer.LayerDescription = _
                layerdescs(layerIDs(MyActiveLayerIndex))
layerDesc.SelectionColor = pColor
layerDesc.SelectionFeatures = fids.FIDArray
layerDesc.ShowSelectionBuffer = False

'...Refresh Map to show selected features in highlighted color
mapctrl.Refresh()

'...Create a GridView Object
Dim gdview As New GridView

'...Get reference to Floating Panel in Defaut.aspx
```

```
Dim gvviewPanel As FloatingPanel = _      CType(mapctrl.Page.FindControl("GridViewFloating
Panel"), FloatingPanel)

'...Check = True, then show selected features
   Dim cbx As CheckBox = _
      CType(mapctrl.Page.FindControl("ViewTableCheckBox1"), CheckBox)
   Dim cbxChecked As Boolean = cbx.Checked
If cbxChanged = True Then
   gvviewPanel.Visible = True
   QueryTable2.TableName = "SelectTable"
   gdview.DataSource = QueryTable2
   gdview.DataBind()
   gvviewPanel.Controls.Add(gdview)
   gvviewPanel.Refresh()
End If

'...Enforce Garbage Collector
mapctrl = Nothing
rectargs = Nothing
minpnt = Nothing
maxpnt = Nothing
mappoly = Nothing
ags_mf = Nothing
gisFunctionality = Nothing
queryResultsDataTable = Nothing
GC.Collect()
GC.WaitForPendingFinalizers()
End Sub 'IMapServerToolAction.ServerAction

End Class 'End of SelectTool
```

'The VB code for SelectByRectangle were modified from a ESRI sample –ArcGIS_SelectBufferToolVBNet.zip

Listing 4

```
Public Class SelectToolByShape
    Implements IMapServerToolAction

Sub ServerAction(ByVal args As ToolEventArgs) _
            Implements IMapServerToolAction.ServerAction


    Dim targetlayername As String = CStr(mapctrl.Page.Session("ActiveLayer"))

    '...### This section differs from SelectTool - Select By Retangle
    '...Define a polygon Event Argument from the screendrawing on the Map Control
    Dim polyargs As PolygonEventArgs = args
    Dim screenpoly As System.Drawing.Point() = polyargs.Vectors

    '...Capture point vectors from screen polygon collection
    Dim pc As ESRI.ArcGIS.ADF.Web.Geometry.PointCollection = _
        New ESRI.ArcGIS.ADF.Web.Geometry.PointCollection()
    For Each dpnt As System.Drawing.Point In screenpoly
            pc.Add(ESRI.ArcGIS.ADF.Web.Geometry.Point.ToMapPoint(dpnt, _
            mapctrl.Extent, CInt(mapctrl.Width.Value), _
            CInt(mapctrl.Height.Value)))

    Next
```

```
'...Define Geometry Ring by collecting all the points
Dim ring As ESRI.ArcGIS.ADF.Web.Geometry.Ring = _
    New ESRI.ArcGIS.ADF.Web.Geometry.Ring()
ring.Points = pc

'...Define multiple Geometry Rings by collecting single ring
Dim rings As ESRI.ArcGIS.ADF.Web.Geometry.RingCollection = _
    New ESRI.ArcGIS.ADF.Web.Geometry.RingCollection()
rings.Add(ring)

'...Compose polygon from multiple ring collection
Dim mappoly As ESRI.ArcGIS.ADF.Web.Geometry.Polygon = _
     New ESRI.ArcGIS.ADF.Web.Geometry.Polygon()
mappoly.Rings = rings

'...### Rest of implemention …
```