

The CEDRA Corporation's COMMAND OF THE MONTH

A monthly information bulletin

August 2009

FEATURED COMMAND

ArcGIS Server

Identify w/ Related Data



Application Description

ArcGIS users know that over the last few years, ESRI has devoted a great deal of attention to developing and promoting ArcGIS Server.

For those unfamiliar with ArcGIS Server, this software allows users to create server based, focused Web Applications and Web Services. Unlike the desktop version of ArcGIS, ArcGIS Server utilizes the user's Web browser as the interface between the application and the user.

So that, rather than using ArcMap's standard interface, an ArcGIS Server Web Application will appear in an Internet Explorer or Mozilla Firefox environment.

As such, a Web based application is invoked by connecting to a URL address. After doing so, the Web browser will display the application's commands and tools. The user then interacts with the application within the context of the Web browser with much of the processing being done on the Server, rather than the client's PC.

A common tool often included in an ArcGIS Server Web based application is the Identify tool. The native Identify tool, , included with ArcGIS Server, however, does not display the related data that may be attached to a feature.

The question becomes, how can an Identify tool, which displays related data associated with a feature, be included in an ArcGIS Server Web application .

The CEDRA Solution

To address this application, The CEDRA Corporation has developed a custom Identify tool, , that displays the related data associated with a feature. This custom tool is written in VB.NET and can be included in any ArcGIS Server Web Application that utilizes VB as its application code language.

If the ArcGIS Server application utilizes C# (CSharp) as its application code language, this code will not work. It is not possible to mix application code of different type in an ArcGIS Server Web Application.

Command Of The Month bulletin

This month's issue discusses how a custom tool that displays related data can be added to an ArcGIS Server Web based application.

Note that there are free utility programs that will convert VB.NET code to C# and vice versa. The following site is an example of a free on-line code conversion tool:

<http://www.developerfusion.com/tools/convert/vb-to-csharp/>.

Creating an ArcGIS Server Web Application

The most common method of creating an ArcGIS Server Web Application is to use ArcGIS Server Manager. Included with the ArcGIS Server software, ArcGIS Server Manager is a pro-

gram that provides a user, who does not have programming expertise, the ability to easily create a Web based application. For simplicity, the ArcGIS Server Manager program is referred to as Manager.

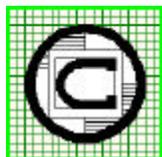
Those users who do have programming expertise with VB.NET or C# can use Microsoft Visual Studio, rather than ArcGIS Server Manager, to create a Web based application. Additionally, it is possible to create a Web based application with Manager and then modify the application within Visual Studio. The modified Web Application can then be imported back into Manager.

Presented below is brief description of how to create a .NET Web application, rather than a JavaScript Web application using Manager. JavaScript Web applications are not as robust as .NET applications and do not allow developers to leverage ASP.NET within the application.

Once the Web Application has been created, we will modify the application to incorporate the custom Identify tool created by The CEDRA Corporation. It should be pointed out that the modification of the Web application will be done using a text editor, Notepad, and not the Visual Studio software.

1. Creating the Map Document

The first step in creating a Web Application with Manager is to create the map document that will be published. This is accomplished by using ArcMap



The CEDRA Corporation

**Total CADD for Engineers™
Bridging Engineering and GIS™**

Copyright © 2009 The CEDRA Corporation. The CEDRA logos, CEDRA-AVseries, CEDRA-AVcad, CEDRA-AVcogo, CEDRA-AVparcel, CEDRA-AVland, CEDRA-AVsand, CEDRA-AVwater, CEDRA-DataEditor and CEDRA-DxfExport are trademarks of The CEDRA Corporation.
ArcView® GIS, 3D Analyst, and ArcGIS are registered trademarks of Environmental Systems Research Institute, Inc. All World Rights Reserved.

Phone: 585-232-6998
Fax: 585-262-2042

151 Sully's Trail - Suite 6
Pittsford, New York 14534

E-mail: cedra@cedra.com
URL: <http://www.cedra.com>

to create the map document. The following topic in the ArcGIS Server Help should be read for a guide as to items to consider when creating a map document that will be published: [Map authoring considerations for ArcGIS Server](#).

Generally speaking most any ArcMap document can be published, however, there are certain items that need to be taken into consideration when creating the map document to be published. As such, it is a good idea to review the above help topic to get an idea of what may or may not be allowed in publishing an ArcMap document.

2. Creating the Map Service

The second step in creating a Web application with Manager is to create a map service. The following topic in the ArcGIS Server Help should be read for a description of what a map service is: [Map services](#), while the topic [Publishing a GIS resource to the server](#) offers a description of how to create a map service utilizing the map document created in the previous step.

Summarizing, to create a map service, the user will:

- a. Invoke ArcGIS Server Manager and log in, see Figure 1.
- b. Click the Services tab in Manager, see Figure 2.
- c. Click the Publish GIS Resource, menu item see Figure 3.
- d. Browse to the resource you want to publish from the Resource drop-down list, or type its path in the data field to the right of the Resource: label, see Figure 4.



Figure 3 Available Services Tasks

If the Browse icon, , is selected in Figure 4, the File Browser dialog box of Figure 5 will appear. Note the Laptop icon that appears to the left of the \\CEDRA-PC label at the top of the list. This denotes a local resource, while the other icon denotes an Internet resource.

Note that the resource that is selected, in this case a .mxd file, must be shared. The user can use



Figure 1 Log In Dialog Box

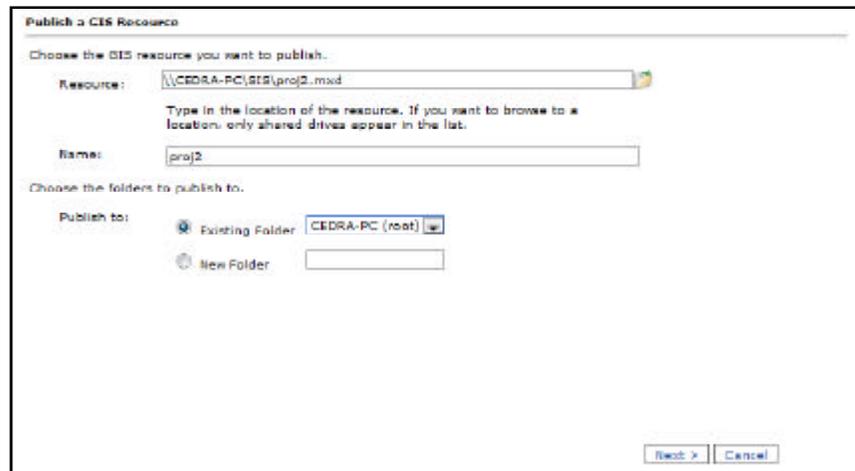


Figure 4 Log In Dialog Box

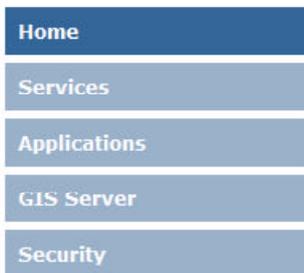


Figure 2 Available Service Manager Tabs

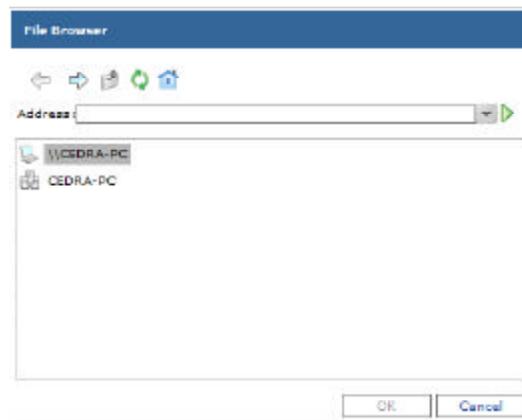


Figure 5 Resource File Browse Dialog Box

the operating systems's tools for sharing the appropriate file(s).

Windows users (XP and Vista) should review the topic [Share files from any folder on your computer](#) in the Windows Help and Support for more information regarding sharing files.

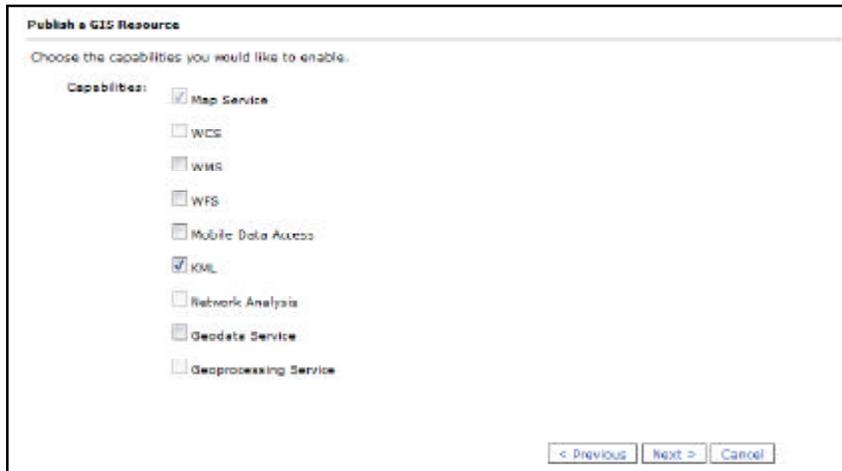


Figure 6 - GIS Resource Capabilities Dialog Box

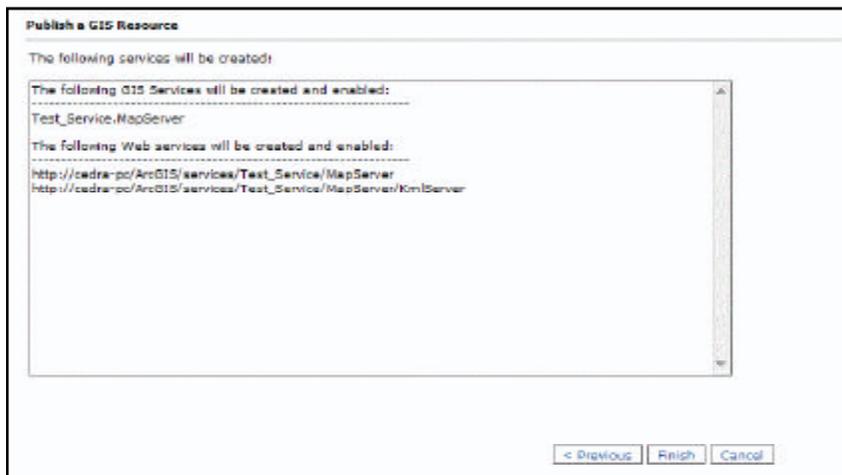


Figure 7 - GIS Resource Summary Dialog Box

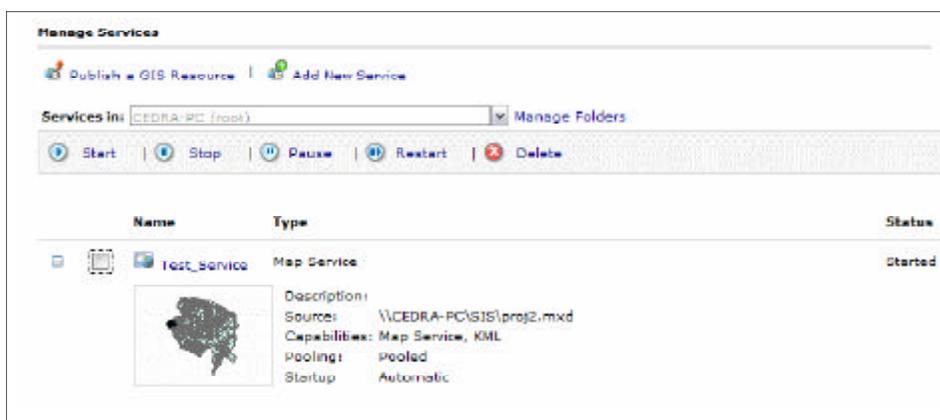


Figure 8 - Published Map Service

- e. Optionally, within Figure 4, change the default name of the service. In this example, we will change the name proj2 to Test_Service.

Note that the name can only contain alphanumeric characters and underscores. No spaces or special characters are allowed. Additionally, the name cannot be more than 120 characters in length.

- f. Choose which folder the service will be published to, see Figure 4. You can either specify an existing folder or create a new folder. Click Next.

- g. Choose the capabilities that you would like to enable, see Figure 6. Depending on the type of the resource and the information it contains, you will see more or fewer available capabilities. Click Next.

- h. Review the information about the service that will be created, see Figure 7, after which, click Finish.

If the service is started and working correctly, you should be able to see a thumbnail image on the Services tab in Manager when you click the plus (+) button next to the service name, see Figure 8.

If, for some reason, your service is not working as expected, you can review the log files for errors. Note that log files refer to services as server objects.

3. Creating a Web Application

Once the Map Service has been created, it is now possible to create a Web Application that utilizes the Map Service.

Prior to creating the Web Application, the user should define the application code setting. That is, should the Web Application support C# or VB.NET.

To define this setting the **Settings** menu item within the **Applications** tab can be employed, see Figure 9. Specifically, the drop-down list to the right of the Language label can be used.

Note that the default program language setting is C#, not VB. So that if a VB based Web Application is desired, the language setting must be defined prior to creating the Web Application.

The following topic in the **ArcGIS Server Help** should be read for a description of how to create a Web Application: [Tutorial: Creating a Web Application](#).

Summarizing the process described in the above help topic, the user will:

- Log in to Manager or, if you're already logged in, click the **Applications** tab, see Figure 10.

The account you used to log in must be in the agsadmin group, and it must be an administrator on the machine in order to create the Web Application. If you're not sure if your account is an administrator on the machine, consult your system administrator or use the operating system tools to view the Administrators group.

- Click the blue text labeled [Create Web Application](#), see Figure 10.
- In the Name text box, type a name for your application, see Figure 11. This is the name by which you will see your application listed in

Figure 9 - Available Settings Parameters

Name	URL
test	http://cedra-pc/test/
Test2	http://cedra-pc/Test2/
Test3	http://cedra-pc/Test3/

Figure 10 - Available Applications Tasks

Figure 11 - Create Application Initial Dialog Box

ArcGIS Server Manager. Optionally, type a description. Then click Next.

- d. Now you will choose the map service that your application will display. First, you need to establish a connection with a GIS server. Click the blue text labeled **Add Layer**, see Figure 12.
- e. **Double-click** on the appropriate server from the list similar to that shown in Figure 13.

Note that in order to display related data a local map service must be selected. It is not possible to display related data for an Internet Map Service.

- f. From the list of map services, from the server you just added, click on the desired Map Service, after which, click **Add**, see Figure 14.

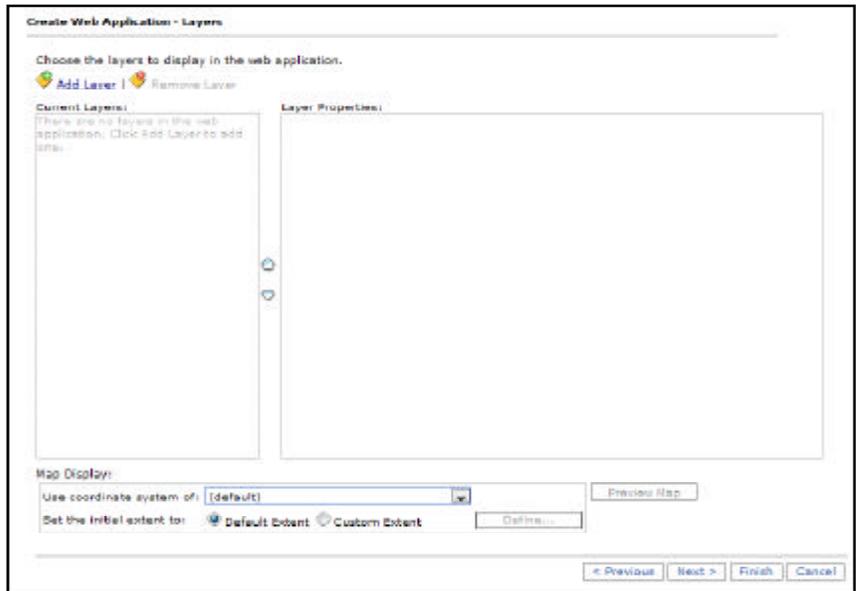


Figure 12 - Add Layer Dialog Box

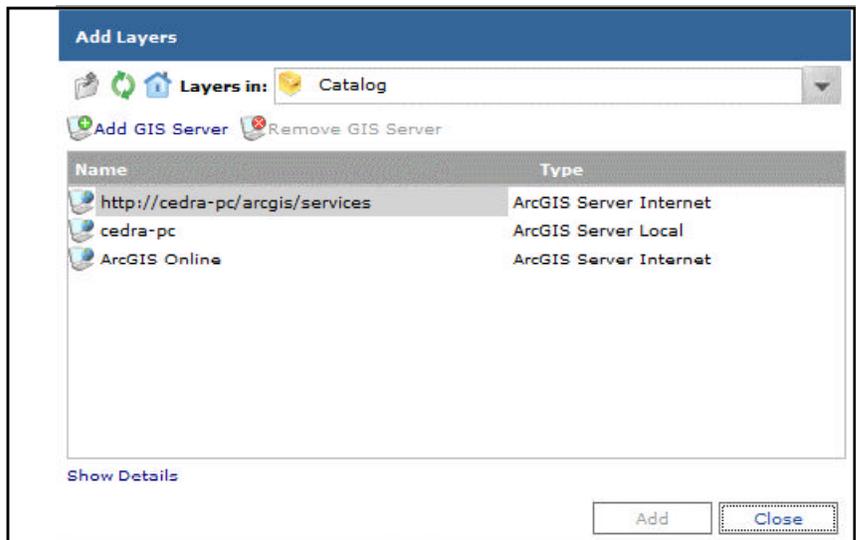


Figure 13 - Available Servers Dialog Box

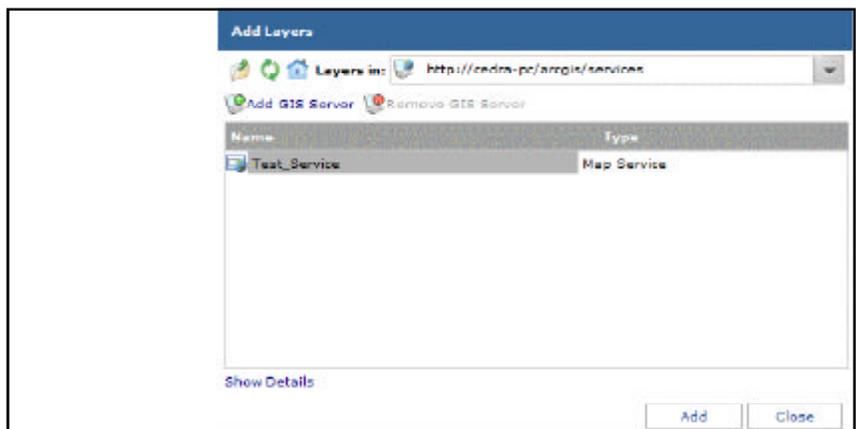


Figure 14 - Log In Dialog Box

- g. Once the Map Service has been added, the user is able to set certain properties for each of the layers within the Map Service, see Figure 15. Once the properties have been defined for the appropriate layers, click **Next**.
- h. The next screen that appears allows the user to add tasks to the application, see Figure 16. Configuring tasks is beyond the scope of this publication, but you can find information in the [ArcGIS Server Help](#) under the topic [Configuring tasks](#). Shown in Figure 17 are the available tasks. Tasks are optional, so for now, click **Next** to move to the next screen.

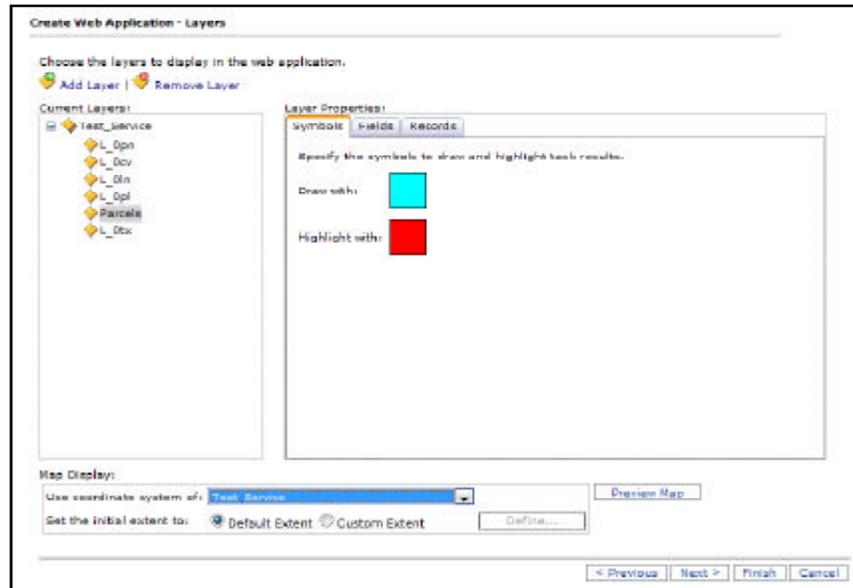


Figure 15 - Layer Properties Dialog Box

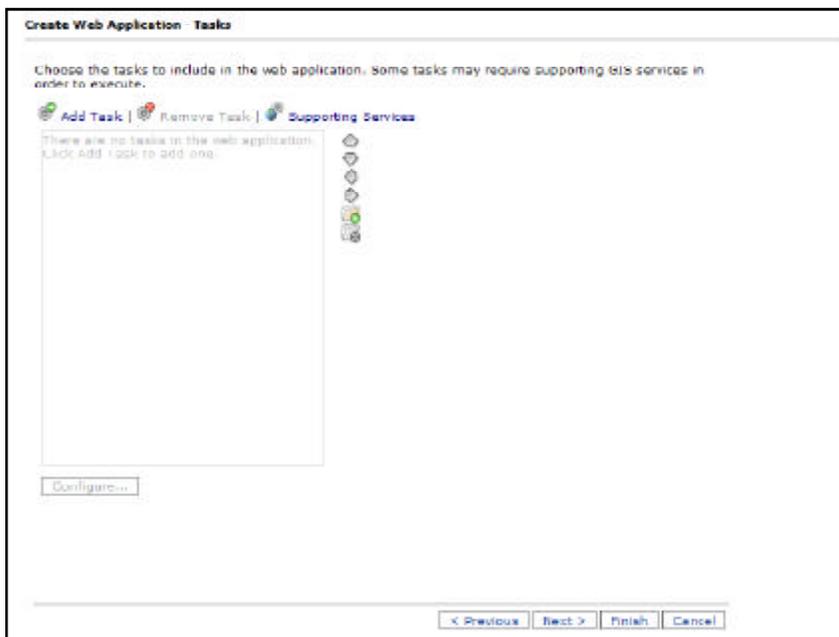
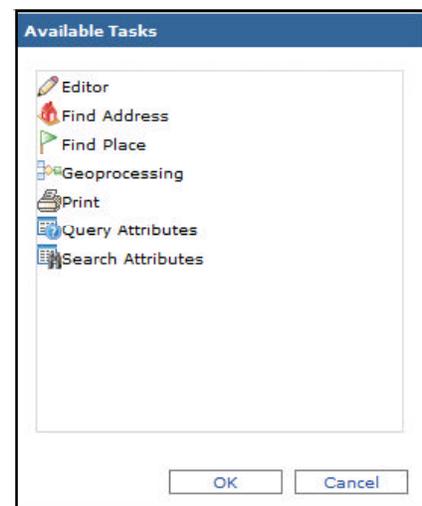


Figure 16 - Tasks Dialog Box

Figure 17
Available Tasks Dialog Box

- i. If a local server was selected in Step e, a dialog box will appear prompting for the user that the Web Application will run as. When the web application you are building contains one or more ArcGIS Server Local connections, it is necessary to specify the user that will be used when the Web Application connects to the local server. So that, accept the default user name or enter the appropriate user name, after which, click **Next** to move to the next screen.

If an internet server is selected in Step e, this dialog box will not appear.

- j. Set the Title text, Theme, and Web page links, see Figure 18.
 - The Title text will appear across the top banner of your application and in the title bar of the browser window.
 - The Theme specifies the top banner graphic and the color scheme of your application's menus.
 - The Web page links will appear in the top corner of your application. You can remove or edit the default links or add your own.

When you finish setting these properties, click **Next**.

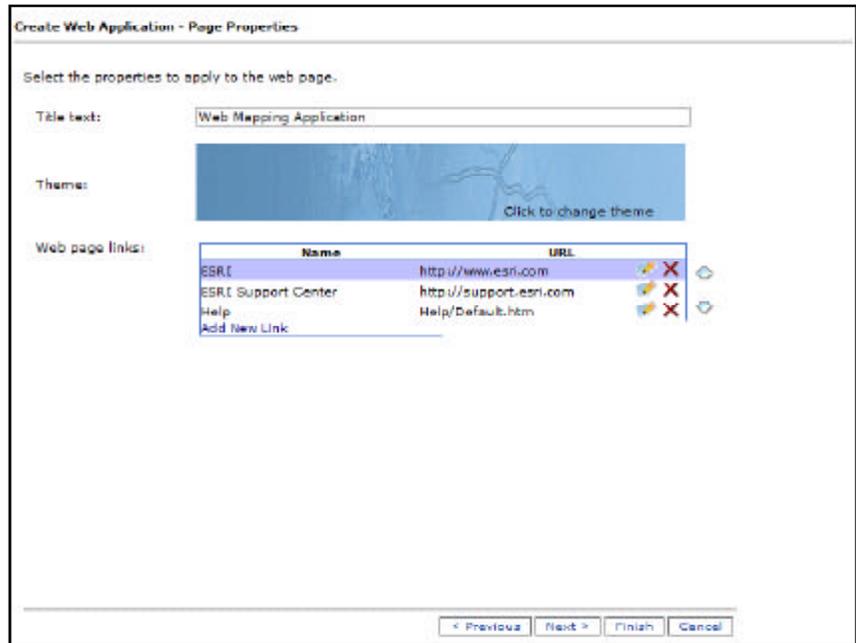


Figure 18 - Web Application Page Properties Dialog Box

- k. Choose which map elements will be enabled in your application, see Figure 19. Available elements are:
 - Table of Contents,
 - Overview Map,
 - Toolbar,
 - Navigation,
 - Scale Bar,
 - Zoom Level, and
 - Map Copyright Text.

Each element has a Properties window where you can adjust how the element will look and behave in your Web Application. For example, you can choose an icon or image that will be used for the Navigation control. When finished selecting the desired map elements, click **Next**.

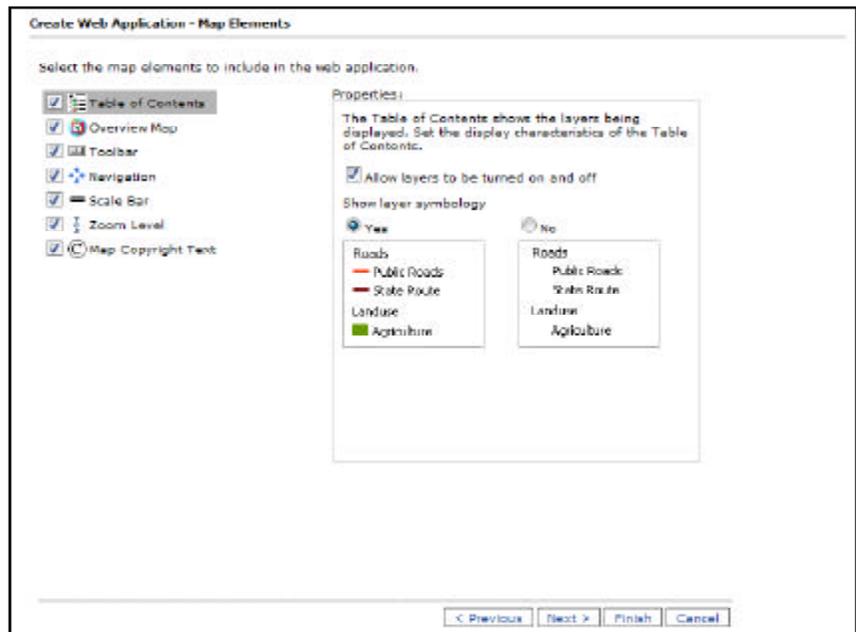


Figure 19 - Available Map Elements Dialog Box

- l. Review the information describing the application that will be created. Take note of the URL so that you can access it later. You'll also be able to see this URL on the **Applications** tab of Manager.
- m. Click **Finish** to create the Web Application. By default, the newly created Web Application will open in a new browser window. To dismiss the application simply close the browser. If you want to change

any of the settings you selected, you can now access the application from the **Applications** tab in Manager. Once the browser has been closed the name of the new Web Application will appear in the list of created Web Applications, see Figure 20.

To make more advanced edits to your application, you can use an IDE, such as Microsoft Visual Studio. The reader is referred to the topic [Customizing the Web Mapping Application](#) within the **ArcGIS Server Help** for a discussion on how a Web Application can be modified.



Figure 20
Web Application Created

4. Modifying the Web Application

At this point the Web Application has been created and we are now ready to modify the application by incorporating the custom Identify tool, , created by The CEDRA Corporation.

As mentioned previously, it is possible to use an IDE, such as Microsoft Visual Studio, to modify a Web Application. However, it is also possible to use a text editor, such as Notepad, to make certain modifications.

Now this is not to say that Notepad is as powerful as Visual Studio, but if the user has customized software already developed, it is possible to incorporate the customized software, in an existing Web Application, by simply using a text editor, such as Notepad.

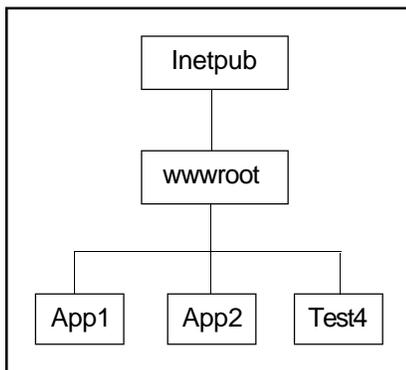


Figure 21
Published Web Applications

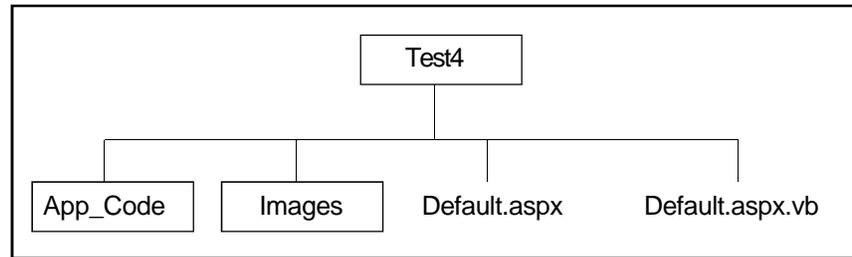


Figure 22
Web Application Directory Structure

This is precisely the scenario discussed in this publication. In this case, CEDRA has created customized ArcGIS Server software which an end user can incorporate into their own application.

Prior to describing how the modifications will be performed, let us describe the directory structure of a Web Application. Shown in Figure 21 is a diagram illustrating where published Web Applications are stored on disk. In this figure it is shown that our sample Web Application, Test4, is stored in a sub-folder called wwwroot within a top level folder called Inetpub. Note that folders are denoted by an enclosing rectangle.

Shown in Figure 22 is a partial directory structure of an ArcGIS Server Web Application. That is to say, there will be more folders and files within a Web Application folder. A file is denoted by the absence of an enclosing rectangle. In Figure 22, Default.aspx and Default.aspx.vb are files, not folders.

To add our custom Identify tool to our sample Web Application, Test4, we will be modifying the files Default.aspx and Default.aspx.vb and adding files to the App_Code and Images folders. Notepad will be used to modify the files, after which, Manager will be re-invoked to republish the Web Application. As such, we are able to modify a Web Application without using Visual Studio or another IDE.

CEDRA Identify Tool Overview

The custom Identify tool developed by The CEDRA Corporation operates on the premise that an active layer must be

defined. Therefore, the feature that is identified must reside in the active layer. By utilizing an active layer, the user is able to control which feature is to be processed.

As such, the CEDRA Custom Identify tool is actually comprised of two tools. The first is the [Set Active Layer] tool, , while the second is the [Identify] tool, .

Furthermore, to use the [Identify] tool, , the user must first activate the [Set Active Layer] tool, , to define the active layer, prior to activating the [Identify] tool.

Implementing the CEDRA Identify Tool

The following steps describe how custom ArcGIS Server code can be incorporated into an existing Web Application.

1. Copy the following files into the App_Code folder.
 - ActiveLayerPicker.vb
 - CustomTool.vb
 - Utility.vb

The App_Code folder contains all of the custom classes within a Web Application. That is to say, the above three files contain the code that does the actual work to set the active layer and display the attributes, including the related data, of a selected feature. This code was developed using Visual Studio 2008, VB.NET, .NET Framework 3.5 and the ArcGIS Web ADF.

- Copy the following files into the Images folder.
AVLAYERS.GIF
IDENTIFY.GIF

The above two files are the icons for the two tools. As the names indicate, the avlayers.gif file is the icon for the [Set Active Layer] tool, while the identify.gif file is the icon for the [Identify] tool.

- Modify the Default.aspx.vb file, using Notepad, by adding Code Block A above the line:

```
End Sub 'Page_Load
```

This is the last line of the Page_Load subroutine and appears at the top of the Default.aspx.vb file.

- Modify the Default.aspx.vb file, using Notepad, by adding Code Block B above the line:

```
End Class
```

This is the last line in the Default.aspx.vb file.

- Select the {File} [Save] command to save the modifications and then the {File} [Exit] command to exit Notepad.

- Modify the Default.aspx file, using Notepad, as shown in Code Blocks C through H.

- Invoke ArcGIS Server Manager and log in.

- Click on the Applications tab and select the Web Applications menu item. A list of the published Web Applications should appear similar to what is shown in Figure 10.

- Right-click on the Test4 application and select the Edit menu item from the pop-up menu list, see Figure 23.

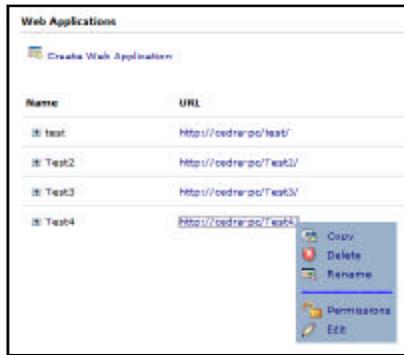


Figure 23
Edit Modified Web Application

- Click the **Finish** button to rebuild the application. Once the application has been rebuilt it should appear in the Web browser.

5. Using the Custom Identify Tool

Once the appropriate modifications have been made to the Web Application, the user will notice the addition of the [Set Active Layer] tool, , and the [Identify] tool, , to the Web Application's toolbar.

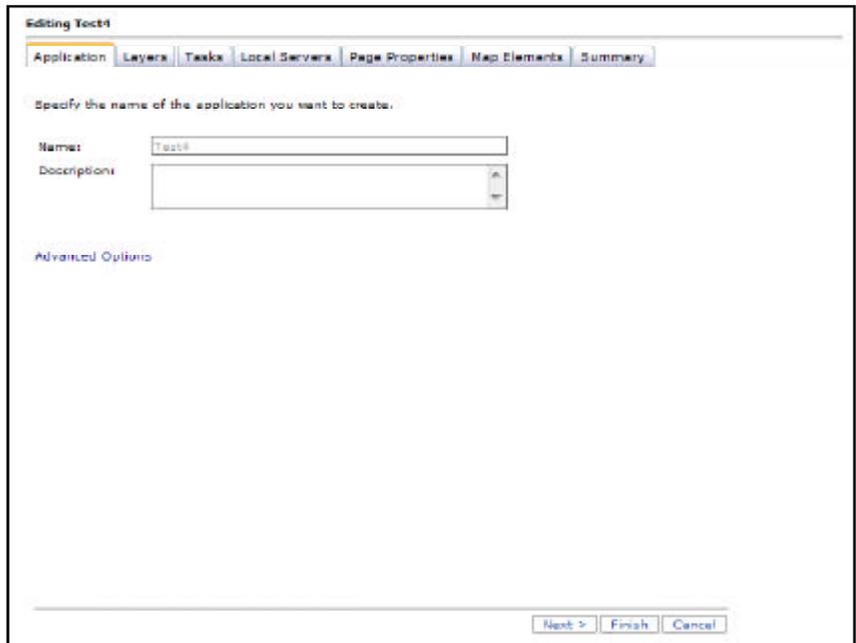


Figure 24
Edit Web Application Dialog Box

- A message box stating that the application has been modified external to ArcGIS Server Manager will appear. Click the **Yes** button to import the modified Web Application. If all goes well the Edit Web Application dialog box of Figure 24 should appear.

If not, an error was detected in the Web Application and a message to that effect will be displayed. At this point exit Manager and check the modifications that were made to the files: Default.aspx and Default.aspx.vb.

As stated earlier, the [Set Active Layer] tool needs to be selected prior to selecting the [Identify] tool. If not, the error message shown in Figure 25 will be displayed.



Figure 25
Active Layer not Defined Error

Upon selection of the [Set Active Layer] tool, the dialog box of Figure 26 will be displayed prompting the user to select



Figure 26
Set Active Layer Dialog Box

the layer that is to serve as the Active Layer. Once a layer is selected from the drop-down list, the name of the layer will appear to the right of the Active Layer label. The next time that the [Set Active Layer] tool is selected, the name of the current active layer will be displayed to the right of the Active Layer label, in so doing the user is able to tell what layer is actually the active layer.

The layers that appear in the drop-down shown in Figure 26 represent the layers in the Web Application which support map query. If a layer does not support map query, it will not appear in the drop-down list.

Once the Active Layer has been defined, the [Identify] tool can be employed. To use the [Identify] tool, the user:

1. Select the [Identify] tool, .
2. Make a pick in the map area to select the feature to be processed.

Once the pick has been made, the tool searches for a feature within proximity of the pick.

The proximity value is a function of the width of the current display. That is to say the larger the width of the map area, the larger the proximity value. Like-

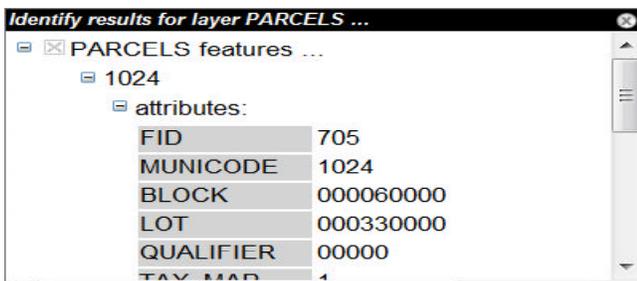


Figure 27
Feature Attributes with No Related Data

wise, the smaller the width of the map area, the smaller the proximity value.

If a feature is found a dialog box similar to that of Figure 27 is displayed. In this figure note that there is no related data associated with the feature. Shown in Figure 28 is an example of how the dialog box would look if the feature did contain related data. Note that there is no limit to the number of relates and that the relates are listed below the main attributes (attributes) in the Identify dialog box.

Notes

- a. In order to display related data associated with a feature, a local Map Service must be used in the Web Application. It is not possible to display related data for an Internet Map Service.
- b. The Web Browser that is used is extremely important when working with a Web Application. Depending upon the browser how a tool performs will vary. The custom tools described here were tested using Mozilla Firefox 3.5.2.

Summary

As users migrate to Web based applications, the ability to incorporate custom tools becomes more and more important. Therefore, the ability to incorporate custom tools, without having to add staff that is programming knowledgeable, is beneficial. As such, the approach discussed in this month's issue of Command of the Month should be helpful in this regard.

For those who are interested in acquiring the source code for the [Set Active Layer] and [Identify] tools, as well as code blocks A through H, contact: Lisa Stone at lstone@cedra.com.

As always, should the reader have any suggestions on functionality that should be featured in Command of the Month, please feel free to forward them on to us.

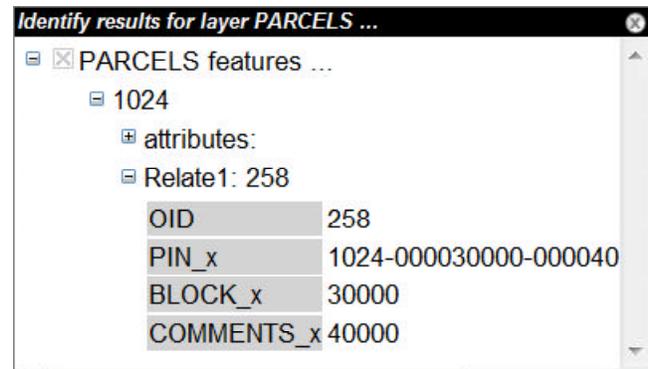


Figure 28
Feature Attributes with Related Data

If you have a request for Command Of The Month, feel free to phone, fax or e-mail your request to The CEDRA Corporation.

```

'
' ---Note this event is fired every time the page refreshes so care should
' ---be taken in what is initialized here, that is to say, if you have
' ---constants (such as rII below) this is a good spot to initialize them
'
' ---Define the map resource index value, this value indicates the position
' ---of the map in the MapResourceManager control (index values begin with 0,
' ---not 1 so that a value of 1 denotes that the map is the second entry in
' ---the MapResourceManager control). This may need to be changed by the user
Dim rII As Short = 0
'
' ---Place the variable in global memory
System.Web.HttpContext.Current.Session("mapRIValue") = rII
'
' ---IsPostBack is True if the page is being loaded in response to a client postback,
' ---if not it is being loaded and accessed for the first time
If Not Page.IsPostBack Then
'
' ---Begin a Try/Catch/Finally/End block to handle any potential errors
Try
' ---Populate the Active Layer drop-down list which contains the names
' ---of the layers in the map
ActiveLayerPicker.AddMapServiceLayerNamesAndIndicesToDropDown(Map1, rII, ddlActiveLayer)
Catch ex As Exception
'
Finally
'
End Try
End If

```

Code Block A - Default.aspx.vb Modification to Sub Page_Load

```

'
' ---Utility code for the Set the Active Layer Button
'
' ---This routine will update the name of the Active Layer in the
' ---Active Layer dialog box
Protected Sub ddlActiveLayer_SelectedIndexChanged(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles ddlActiveLayer.SelectedIndexChanged
'
' ---Check if the default drop down list item ("- select layer -") has been selected
If (CType(sender, DropDownList)).SelectedIndex = 0 Then
    lblActiveLayer.Text = ""
    System.Web.HttpContext.Current.Session("ActiveLayerIndex") = -1
    Return
End If
'
' ---Display the name of the Active Layer that has been selected in the
' ---Active Layer dialog box
lblActiveLayer.Text = (CType(sender, DropDownList)).SelectedItem.Text
'
' ---Store in global memory the index value of the Active layer, as well as,
' ---the name of the Active Layer
System.Web.HttpContext.Current.Session("ActiveLayerIndex") = (CType(sender, _
    DropDownList)).SelectedItem.Value
System.Web.HttpContext.Current.Session("ActiveLayerName") = (CType(sender, _
    DropDownList)).SelectedItem.Text
'
End Sub

```

Code Block B - Default.aspx.vb Addition of a new Sub

Change the line:

```
<body style="margin:
```

to look like:

```
<body onmouseup="stopMoving()" style="margin:
```

Code Block C - Default.aspx Modification, Add onmouseup="stopMoving()" code to <body> tag

Below the line:

```
<asp:ScriptManager ID="ScriptManager1" runat="server" EnablePageMethods="true" />
```

add the following code:

```
<script type="text/javascript" language="javascript">

    Sys.WebForms.PageRequestManager.getInstance().add_pageLoading(PageLoadingHandler);

    function PageLoadingHandler(sender, args) {
        var dataItems = args.get_dataItems();
        if (dataItems['Map1'] != null)
            processCallbackResult(dataItems['Map1'], 'Map1');
    }
</script>
```

Code Block D - Default.aspx Modification, Addition of ScriptManager Javascript code

Below the line:

```
<ToolBarItems>
```

add the following code:

```
<esri:Command DefaultImage="~/Images/AVLAYERS.GIF" Name="SetActiveLayer"
BuddyItem="ChooseActiveLayer" ToolTip="Set the Active Layer" ServerActionAssembly="App_Code"
ServerActionClass="ActiveLayerPicker" ClientAction="" JavaScriptFile="" />

<esri:Tool DefaultImage="~/Images/IDENTIFY.GIF" Name="IdentifyFeature" ToolTip="Identify
Feature" ServerActionAssembly="App_Code" ServerActionClass="CustomTools.CustomIdentifyTool"
ClientAction="hideShowActiveLayerDialog(); MapPoint('Map1', '%toolbarItem%', true,
'crosshair');" JavaScriptFile="" />
```

Code Block E - Default.aspx Modification, Addition of Two Tools to the Application's Toolbar

Change the line:

```
<div id="Map_Panel" style="width: 512px; height: 512px; position: absolute; top: 0px; over-
flow: hidden; background-color: White;" class="mapPosition">
```

to look like:

```
<div id="Map_Panel" style="width: 512px; height: 512px; position: absolute; top: 0px; over-
flow: hidden; background-color: White;" class="mapPosition" onclick="return
Map_Panel_onclick()" onmousemove="return Map_Panel_onmousemove()">
```

Code Block F - Default.aspx Modification, Addition of onclick and onmousemove events

Below the line:

```
<uc1:MapIdentify ID="MapIdentify1" runat="server" MapBuddyId="Map1" />
```

and above the line:

```
</form>
```

add the following code:

```
<!-- -->
<!-- This DIV is the Set the Active Layer dialog -->
<!-- -->
<div id="ActiveLayerPanel" style="border-style: solid; border-bottom-width:1px; border-
right-width:1px; border-left-width:1px; border-top-width:1px; z-index:500; position:absolute;
background-color:White; border-color:Gray; width:400px; height:100px; top:30%; left:30%;
display:block; visibility:hidden">
  <!-- -->
  <!-- This DIV is the Active Layer dialog's title bar -->
  <div style="position:absolute; color:White; font-family:Arial; font-style:italic;
background-color:Black; width:100%; height:18px; font-weight:bold; font-size:small; top:0%;
left:0%; " onmousedown="setMoving('ActiveLayerPanel')">
    <!-- This is the title bar's caption -->
    <asp:Label ID="captionActiveLayer" runat="server" Text="Select the active layer
... " />
    <!-- This is the title bar's dismiss button, the X icon -->
    <asp:Image ID="ImageActiveLayer" ImageUrl="images/dismiss.png" runat="server"
AlternateText="Close dialog" onclick="hideShowActiveLayerDialog();" style="right: 0px; font-
weight:bold; font-family:Arial; font-size:9pt; position:absolute; top: 0px" />
  </div>

  <!-- This is the Layers parameter label (left side of the dialog box) -->
  <asp:Label ID="lblSelectActiveLayer" runat="server" style=" font-style:italic; font-
weight:bold; font-family:Arial; font-size:10pt; text-align:right; color:Navy;
position:absolute; left:11px; top:35px; width: 123px;" Text="Layers:" />

  <!-- This is an invisible update panel that shortcuts updating page elements
in the browser from server-side code results: the items in the update panel
(in this case the layers dropdownlist and the active layer text on a label)
will update automatically when their values are changed on the server -->

  <!-- This is the list of AJAX-enabled objects and their events that can
cause requests to the web server that send results back to the update
panel's ContentTemplate items. Refer to the Page_Load event of the
Default.aspx.vb code to see how the list of layers gets populated -->
  <asp:UpdatePanel ID="UpdatePanel2" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
      <asp:DropDownList style="left: 150px; position: absolute; top: 35px"
ID="ddlActiveLayer" runat="server" Width="225px" AutoPostBack="true"
OnSelectedIndexChanged="ddlActiveLayer_SelectedIndexChanged" />
      <asp:Label ID="lblActiveLayer" runat="server" style=" font-style:italic;
font-weight:bold; font-family:Arial; font-size:10pt; text-align:left; color:Navy;
position:absolute; left:150px; top:70px; width: 123px;" Text="" />
    </ContentTemplate>
    <Triggers>
      <asp:AsyncPostBackTrigger ControlID="ddlActiveLayer"
EventName="SelectedIndexChanged" />
    </Triggers>
  </asp:UpdatePanel>

  <!-- This is the Active Layer parameter label (left side of the dialog box) -->
  <asp:Label ID="lblLayerLabel" runat="server" style=" font-style:italic; font-
weight:bold; font-family:Arial; font-size:10pt; text-align:right; color:Navy;
position:absolute; left:11px; top:70px; width: 123px;" Text="Active Layer:" />
```

Code Block G - Default.aspx Modification, Addition of Dialog Boxes via the DIV tag

```

<!-- Define the client-side Javascript code -->
<script type="text/javascript" language="javascript">

    // This is client-side code that is used to hide and show the dialog and clear
any error messages
    function hideShowActiveLayerDialog(opt) {
        var vis = 'hidden';
        if (opt == 'show') vis = 'visible';
        if (opt == 'Show') vis = 'visible';
        if (opt == 'SHOW') vis = 'visible';
        //
        document.getElementById('ActiveLayerPanel').style.visibility = vis;
    }
</script>
</div>

<!-- -->
<!-- This DIV is the Identify Results dialog -->
<!-- -->
<div id="IdentifyResults" style="border-style: solid; border-bottom-width:1px; border-
right-width:1px; border-left-width:1px; border-top-width:1px; z-index:500; position:absolute;
background-color:White; border-color:Gray; width:400px; height:230px; top:30%; left:30%;
overflow:hidden; display:block; visibility:hidden" onmouseup="return
IdentifyResults_onmouseup()" onclick="return IdentifyResults_onclick()" onmousedown="return
IdentifyResults_onmousedown()">
    <!-- -->
    <!-- This DIV is the Identify Results dialog's title bar -->
    <div style="position:absolute; color:White; font-family:Arial; font-style:italic;
background-color:Black; width:100%; height:18px; font-weight:bold; font-size:small; top:0px;
left:0px;" onmousedown="setMoving('IdentifyResults')">
        <!-- This is the title bar's caption -->
        <asp:Label ID="lblIdentifyTitle" runat="server" style=" font-weight:bold; font-
family:Arial; font-size:small; text-align:left; color:White; position:absolute; left:2px;
top:0px; width: 376px;" Text="Identify Results" />
        <!-- This is the title bar's dismiss button, the X icon -->
        <asp:Image ID="imgCloseIdentify" ImageUrl="images/dismiss.png" runat="server"
AlternateText="Close dialog" onclick="hideIdentifyResults();" style="right: 0px; font-
weight:bold; font-family:Arial; font-size:9pt; position:absolute; top: 0px" />
    </div>

    <div style="position:absolute; background-color:White; width:100%; height:18px;
overflow:scroll; height:99%; top:19px; left:0px;">
        <esri:TreeViewPlus ID="tvpIdentify" runat="server" BackColor="White" Font-
Names="Arial" Font-Size="10pt" ForeColor="Black" Height="200px" Width="90%"
style="position:absolute; left:1px; top:2px" />
    </div>

    <asp:Label ID="LabelIdentify" runat="server" style=" font-style:normal; font-
weight:bold; font-family:Arial; font-size:10pt; text-align:left; color:Red;
position:absolute; left:11px; top:36px; width: 380px;" Text="Processing..." />

<!-- Define the client-side Javascript code -->
<script type="text/javascript" language="javascript">

    function showIdentifyResults(title, opmode) {
        if (title == null) title = "Identify Results";
        if (title.length == 0) title = "Identify Results";
        document.getElementById('lblIdentifyTitle').innerHTML = title;
        document.getElementById('IdentifyResults').style.visibility = 'visible';
        if (opmode == "NO") document.getElementById('LabelIdentify').style.visibility
= 'hidden';
        if (opmode == "NO") document.getElementById('IdentifyResults').style.height =
'230px';
    }

```

Code Block G - Default.aspx Modification, Addition of Dialog Boxes via the DIV tag

```

        if (opmode == "NO") document.getElementById('tvpIdentify').style.visibility =
'visible';
        if (opmode == "YES")
document.getElementById('LabelIdentify').style.visibility = 'visible';
        if (opmode == "YES") document.getElementById('IdentifyResults').style.height
= '120px';
        if (opmode == "YES") document.getElementById('tvpIdentify').style.visibility
= 'hidden';
    }
    function hideIdentifyResults() {
        document.getElementById('IdentifyResults').style.visibility = 'hidden';
        document.getElementById('LabelIdentify').style.visibility = 'hidden';
    }
</script>
</div>

<!-- -->
<!-- This DIV is the Custom Message box -->
<!-- -->
<!-- The code overflow:auto results in vertical and horizontal -->
<!-- scroll bars appearing when the text exceeds the height or -->
<!-- width of the DIV -->
<!-- -->
<div id="processMessage" style="border-style: solid; border-bottom-width:1px; border-
right-width:1px; border-left-width:1px; border-top-width:1px; z-index:500; position:absolute;
background-color:White; border-color:Gray; width:450px; height:120px; top:30%; left:30%;
display:block; overflow:auto; visibility:hidden">
    <!-- -->
    <!-- This DIV is the Custom Message dialog's title bar -->
    <div style="position:absolute; color:White; font-family:Arial; font-style:italic;
background-color:Black; width:100%; height:18px; font-weight:bold; font-size:small; top:0%;
left:0%;" onmousedown="setMoving('processMessage')">
        <!-- This is the title bar's caption -->
        <asp:Label ID="lblProcessTitle" runat="server" style=" font-weight:bold; font-
family:Arial; font-size:small; text-align:left; color:White; position:absolute; left:2px;
top:0px; width: 376px;" Text="Process Message" />
        <!-- This is the title bar's dismiss button, the X icon -->
        <asp:Image ID="imgCloseMessage" ImageUrl="images/dismiss.png" runat="server"
AlternateText="Close dialog" onclick="hideProcessMessage();" style="right: 0px; font-
weight:bold; font-family:Arial; font-size:9pt; position:absolute; top: 0px" />
    </div>

    <asp:Label ID="lblProcessMessage" runat="server" style=" font-style:normal; font-
weight:bold; font-family:Arial; font-size:10pt; text-align:left; color:Red;
position:absolute; left:11px; top:36px; width: 380px;" Text="Processing message" />

    <!-- Define the client-side Javascript code -->
    <script type="text/javascript" language="javascript">
        function showProcessMessage(title, message, lblColor) {
            if (title == null) title = "Process Message";
            if (title.length == 0) title = "Process Message";
            if (message == null) title = "Unknown error has occurred.";
            if (message.length == 0) title = "Unknown error has occurred.";
            if (lblColor == null) lblColor = "Red";
            if (lblColor.length == 0) lblColor = "Red";
            document.getElementById('lblProcessTitle').innerHTML = title;
            document.getElementById('lblProcessMessage').innerHTML = message;
            document.getElementById('lblProcessMessage').style.color = lblColor;
            document.getElementById('processMessage').style.visibility = 'visible';
        }
        function hideProcessMessage() {
            document.getElementById('processMessage').style.visibility = 'hidden';
        }
    </script>
</div>

```

Code Block G - Default.aspx Modification, Addition of Dialog Boxes via the DIV tag

Below the line:

```
Sys.Application.add_init(startUp);
```

add the following code:

```
//
// ---Declare the public variables
//
var m_bMoving = false
var m_sItemID = ""
var m_MouseDownX
var m_MouseDownY
var m_MouseUpX
var m_MouseUpY
var m_Left
var m_Top
//
function Map_Panel_onclick() {
    //
    // ---Check if a pick has been made in the Map
    //
    //
    // ---Determine the active tool in the Map
    //
    // ---Define the name of the toolbar to be processed
    //
    var toolbarName = "Toolbar1";
    //
    // ---Define the name of the form containing the toolbar
    //
    var f = document.forms["form1"];
    //
    // ---Check if the Identify Features Tool has been selected
    //
    if ((f.elements[Toolbars[toolbarName].currentToolField].value) == 'IdentifyFeature')
    {
        //
        // ---Dismiss the Set the Active Layer and Identify Results dialog boxes
        //
        hideShowActiveLayerDialog();
        hideIdentifyResults();
        //
        // ---Display the processing message in the map panel
        //
        showProcessMessage("Identify Feature Tool", "Processing. Be right back...");
    }
}
function Map_Panel_onmousemove() {
    //
    // In the JavaScript folder there is a file called WebMapApp.js that contains
    // a function called MapCoordsMouseMove that displays the coordinates of the
    // cursor as it moves across the map.
    //
    //
    // Note that most (newer) major browsers disable status bar messages by default. If
    // your status bar doesn't change when you move the cursor, it's probably because
    // of this.
    //
    //
    // To enable status bar messages to appear, you may need to change your browser
    // settings.
    //
    //
    // For example, in Firefox:
    //
    //
    // 1. Go to Tools > Options
    // 2. Click the Content tab
```

Code Block H - Default.aspx Modification, Addition of Code to handle user moving of Dialog Boxes

```
// 3. Ensure that the JavaScript option is checked
// 4. Click Advanced (next to the Enable JavaScript option)
// 5. Check the Change status bar text option
// 6. Click OK to save this screen
// 7. Click OK again
//
// In Internet Explorer:
//
// 1. Go to Tools > Internet Options
// 2. Click the Security tab
// 3. Ensure that the Internet option is selected/highlighted
// 4. Click Custom Level... (this launches the security settings for the Internet
//    zone)
// 5. Scroll down until you see Allow status bar updates via script (under the
//    Scripting option). Click Enable
// 6. Click OK to save this screen
// 7. Click OK again
//
}
//
// ---Function to handle the ending of moving a DIV
//
function stopMoving() {
    // ---Make sure we have an element to process
    if (!m_sItemID) return;
    if (m_sItemID == null) return;
    if (m_sItemID.length == 0) return;
    // ---Get the element to be moved
    var moveElem = document.getElementById(m_sItemID);
    // ---Check if the element was not found
    if (!moveElem) return;
    // ---Set the upper left corner of the element
    if(m_Left == -1)
    {
        moveElem.style.left = m_MouseUpX + "px";
        moveElem.style.top = m_MouseUpY + "px";
    }
    else
    {
        moveElem.style.left = (m_MouseUpX + (m_Left - m_MouseDownX)) + "px";
        moveElem.style.top = (m_MouseUpY + (m_Top - m_MouseDownY)) + "px";
    }
    // ---Initialize the move flag
    m_bMoving = false;
    // ---Initialize the ID of the element to be processed
    m_sItemID = "";
}
//
// ---Function to handle the start of moving a DIV
//
function setMoving(sID) {
    // ---Validity check of the element ID passed in
    if (!sID) return;
    if (sID == null) return;
    if (sID.length == 0) return;
    // ---Get the element to be moved
    var moveElem = document.getElementById(sID);
    // ---Check if the element was not found
    if (!moveElem) return;
    // ---Get the last character in the string
    var lasChr = Right(moveElem.style.left, 1);
    // ---Preserve the upper left coordinates of the element making
    // ---sure to strip off the px characters at the end of the string
    if(lasChr == '%')
    {
```

Code Block H - Default.aspx Modification, Addition of Code to handle user moving of Dialog Boxes

```

        // ---Handle case when position is defined in terms of a %
        m_Left = -1;
        m_Top = -1;
    }
    else
    {
        // ---Handle case when position is defined in terms of pixels
        var nChrs;
        // ---Determine the number of characters in the string
        nChrs = moveElem.style.left.length;
        m_Left = parseInt(Left(moveElem.style.left, nChrs-2));
        nChrs = moveElem.style.top.length;
        m_Top = parseInt(Left(moveElem.style.top, nChrs-2));
    }
    // ---Set flag we are moving an element
    m_bMoving = true;
    // ---Preserve the ID of the element being moved
    m_sItemID = sID;
}
//
// ---Function to preserve the mouse coordinates on a Mouse Down Event
//
document.onmousedown = function (event) {
    if (!event) {
        // ---Handle case when the event passed in is equal to none
        event = window.event;
    }
    m_MouseDownX = event.clientX;
    m_MouseDownY = event.clientY;
}
//
// ---Function to preserve the mouse coordinates on a Mouse Up Event
//
document.onmouseup = function (event) {
    if (!event) {
        // ---Handle case when the event passed in is equal to none
        event = window.event;
    }
    m_MouseUpX = event.clientX;
    m_MouseUpY = event.clientY;
}
//
// ---Function to return the left n characters in a string
//
function Left(str, n) {
    if (n <= 0)
        return "";
    else if (n > String(str).length)
        return str;
    else
        return String(str).substring(0,n);
}
//
// ---Function to return the right n characters in a string
//
function Right(str, n) {
    if (n <= 0)
        return "";
    else if (n > String(str).length)
        return str;
    else {
        var iLen = String(str).length;
        return String(str).substring(iLen, iLen - n);
    }
}
}

```

Code Block H - Default.aspx Modification, Addition of Code to handle user moving of Dialog Boxes