

Introduction

Learn how to create a custom command using Add Feature to Legend example.

Software

- Microsoft Visual Basic.
- GeoMedia Professional

Data

- AddFeatureDryRunCmd_lc.bmp
- AddFeatureDryRunCmd_sc.bmp

Transcript

0:09

Thank you for watching this Hexagon Geospatial eTraining module: Creating a Custom Command in GeoMedia.

In this module you will see how to create a custom command using Microsoft Visual Studio.

You must install Visual Studio before installing the product in order for the Command Wizard and its components to be registered. If Visual Studio is installed after the product is installed, you must run regwiznet.exe to register the Command Wizard and its components.

Start with Visual Studio open.

0:40

We will begin by enabling the GeoMedia Command Wizard in Visual Studio.

1. From the **Tools** drop down, click **Add-in Manager**.
2. In the **Add-in Manager**, select the check box for **GeoMedia Command Wizard** and click **OK**.
3. Now go back to the **Tools** drop down and select the newly enabled **GeoMedia Command Wizard**.

1:05

4. In the first dialog, click **Next**.
5. In the Second dialog change the **Project Name** to AddFeatureToLegend.
6. For the **Project Location**, select **C:**
7. In the **New Subfolder Name** textbox type AddFeatureToLegend.
8. Click **Next**.

9. In the **Command Name** type `AddFeatureCommand`.
10. For **Command Description** type `Add Feature to Legend`.
11. For **Command Tool Tip** type `Add feature to map legend. \nAdd Feature to Legend`.
12. Click **Next**
13. In the **Add Toolbar Bitmap** dialog select **Add custom bitmap**.
14. Click **Next**.

1:51

Now you will choose the Custom Bitmap for your command resource.

15. For Large Color Bitmap select **Browse**.
16. Select `AddFeatureDryRunCmd_lc.bmp` and click **Open**.
17. Next click **Browse** for Small Color Bitmap.
18. Select `AddFeatureDryRunCmd_sc.bmp` and click **Open**.
19. Click **Next**.

2:24

A warning appears saying "Default bitmaps will be used for the ones that are not specified."

20. Click **OK**.
21. For the **Command Classification**, leave it set to **Command is Modal** and click **Next**.
22. In the **Choose Form Name** dialog select **Form is centered in the application window**.
23. Click **Next**.
24. For the **Command Enabling Conditions** do not check any of the boxes.
25. Click **Next**.
26. In the **Complete Dialog** scroll through all of the specifications.
27. Click **Finish** when satisfied with the specifications.

A dialog appears confirming the command was added to the command set.

28. Click **Yes** to open the project.

3:03

29. On the right side, from the **Solution Explorer**, double click `frmAddFeatureCommand.cs`
30. Click all of the features in the dialog that appears and delete them.
31. Using the Tool box panel on the left side create the layout of the dialog.
32. Then using the Properties Panel in the bottom right, rename and set the properties for the added tools.

3:30

Now go to the Add Feature to Legend Form file.

33. From the **Solution Explorer**, Open the **AddFeatureCommand** file.
34. Expand **Class Fields**.
35. Scroll down and cut the definition for **AddFeatureToLegend**.
36. Scroll down a little further and expand the **Activate** method.
37. Paste the definition cut from **AddFeatureToLegend** into the **Activate** method.

The constructor will take in a GeoMedia application as its parameters.

38. Set the parameter to **gobjGeoApp**.

3:56

Next, we will add some methods to the Add Feature to Legend Form file.

39. From the **Solution Explorer** panel, Right **Click frmAddFeatureCommand.cs** and click **View Code**.

We are going to use a sample code to replace the using statements to correspond with the GeoMedia components.

40. Select all of the **Using** statements and copy and paste in the new statements from the sample codes.

4:21

41. Next we will add all the private variables used in these methods.
42. As mentioned earlier, the constructor will take in an application. Then it sets the private variable current application and initializes the component.
43. The **AddFeaturetoLegend Load** method loads the combo box with a list of the current open connections.
44. Now we will paste sample codes in the next five methods to handle button click events from our forms.

5:05

Next, we will create a couple of new methods.

45. The Add Feature to Map method takes in the name of the feature the user wants to add to the active map window.
46. The Display Record Set method takes in the originating pipe and map view of the record set that is to be displayed.
47. The GetGeometryFieldName method takes in the record set of geometry to return its field name.
48. The GetStyleObject method takes in a geometry type to return the geometry's "Style Definition" object.
49. The GetRGB method takes in interger values for red, blue, and green color values and converts them into a long value needed by Feature Symbolology.
50. The GetNextBoundaryStyleColor method checks to see what index to use for the "Random Color" integer array.

51. The `ReleaseComReference` method is needed to release any objects left by the method calling it.

6:41

Next, we will build the project and see that it builds successfully.

52. Click **Build > Build Solution**.

You can view the build being executed in the Output Panel.

53. Once the build returns successful, open **GeoMedia**.

In GeoMedia, we'll start with `USSampleData.gws` opened. You can now see My Workflow appears as a new tab.

54. Click **My Workflow**.

Notice that the bitmap icon we added in the Command Wizard. When the mouse is hovered over the icon, it displays the tool tip's title and information that we created earlier. Note how the text we entered after the newline character appears as the command name.

7:21

We'll now run the command and see if it functions correctly.

55. Click **Add Feature to Legend**.

56. Scroll through the **Features** field and select **USSampleImage**.

57. Click **OK**.

58. Select **Add New Feature to Legend** again and add the **States** feature.

Both of the features should be added to your viewer.

7:44

Now that we have successfully run the command, let's see how to uninstall a command from GeoMedia.

59. Close GeoMedia.

We start with the command prompt to the GeoMedia Program folder.

60. In the prompt type the command you see here:

```
InstallAppCmd /r /prod "GeoMedia Professional" / cmdset  
AddFeatureCommand / loc C:\AddFeatureToLegend\bin\dll  
C:\AddFeatureToLegend\bin /u.
```

7:58

Let's break down that command and look at it piece by piece.

After `InstallAppCmd`, the `R` indicates this is a ribbon command. Next, we insert `prod` and the product name, which in this case is `GeoMedia Professional`. Then we typed `cmdset` and the command name, `AddFeature Command`. After that, we typed `loc` and the path to the XML file. Then we put `dll` and the path to the `.dll`. Finally, we put a `U` to indicate that this command is to uninstall.

61. Click **Enter** to execute the command.

A notification should appear indicating that the application command folder successfully unregistered.

62. Click **OK**.

63. Return to GeoMedia to make sure the command has been removed.

8:46

Note that the My Workflow tab is no longer on the ribbon since you have uninstalled all of the custom commands.

64. To reinstall the **Add Feature to Legend** command, go back to the command prompt and enter the same command and arguments as before, this time without /u at the end.

A notification should appear indicating the command successfully registered.

65. Click **OK**.

66. Return to GeoMedia and ensure the My Workflow tab has returned.

In this eTraining module, we have seen how to create, install, uninstall, and reinstall custom commands in GeoMedia. You should now be able to create your own custom commands.

9:34

Thank you for watching this eTraining module from Hexagon Geospatial. For more eTraining, visit hexagongeospatial.com/eTraining.