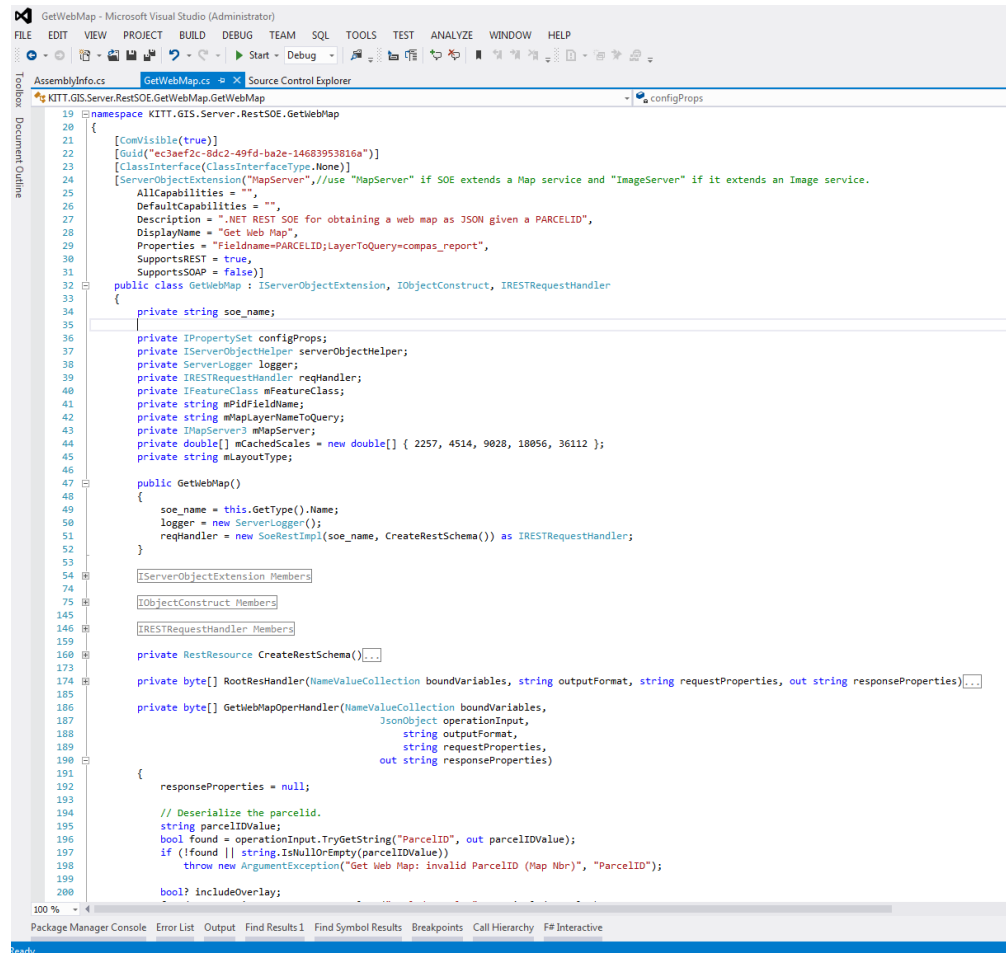


Creating the SOE in Visual Studio...



```
19 namespace KITT.GIS.Server.RestSOE.GetWebMap
20 {
21     [ComVisible(true)]
22     [Guid("ec3ae2c-8dc2-49fd-ba2e-14683953816a")]
23     [ClassInterface(ClassInterfaceType.None)]
24     [ServerObjectExtension("MapServer", //use "MapServer" if SOE extends a Map service and "ImageServer" if it extends an Image service.
25         AllCapabilities = "",
26         DefaultCapabilities = "",
27         Description = ".NET REST SOE for obtaining a web map as JSON given a PARCELID",
28         DisplayName = "Get Web Map",
29         Properties = "FieldName=PARCELID;LayerToQuery=compas_report",
30         SupportsREST = true,
31         SupportsSOAP = false)]
32     public class GetWebMap : IServerObjectExtension, IObjectConstruct, IRESTRequestHandler
33     {
34         private string soe_name;
35
36         private IPropertySet configProps;
37         private IServerObjectHelper serverObjectHelper;
38         private ServerLogger logger;
39         private IRESTRequestHandler reqHandler;
40         private IFeatureClass mFeatureClass;
41         private string mPidFieldName;
42         private string mMapLayerNameToQuery;
43         private IMapServer mMapServer;
44         private double[] mCachedScales = new double[] { 2257, 4514, 9028, 18056, 36112 };
45         private string mLayoutType;
46
47         public GetWebMap()
48         {
49             soe_name = this.GetType().Name;
50             logger = new ServerLogger();
51             reqHandler = new SoeRestImpl(soe_name, CreateRestSchema()) as IRESTRequestHandler;
52         }
53
54         [ServerObjectExtension Members]
55
56         [IObjectConstruct Members]
57
58         [IRESTRequestHandler Members]
59
60         private RestResource CreateRestSchema()...
61
62         private byte[] RootResHandler(NameValueCollection boundVariables, string outputFormat, string requestProperties, out string responseProperties)...)
63
64         private byte[] GetWebMapOperHandler(NameValueCollection boundVariables,
65             JsonObject operationInput,
66             string outputFormat,
67             string requestProperties,
68             out string responseProperties)
69         {
70             responseProperties = null;
71
72             // Deserialize the parcelid.
73             string parcelIDValue;
74             bool found = operationInput.TryGetValue("ParcelID", out parcelIDValue);
75             if (!found || string.IsNullOrEmpty(parcelIDValue))
76                 throw new ArgumentException("Get Web Map: invalid ParcelID (Map Nbr)", "ParcelID");
77
78             bool? includeOverlay;
```

Register File.soe With ArcGIS Server...

The screenshot shows the ArcGIS Server Manager web interface. The browser address bar displays 'gis-01:6080/arcgis/manager/site.html'. The interface includes a navigation menu with 'Services', 'Site', 'Security', and 'Logs'. The 'Site' tab is active, showing 'GIS Server', 'Web Adaptor', and 'Software Authorization' sub-tabs. The 'Extensions' section is highlighted in the left sidebar. The main content area displays the 'Extensions' page, which includes a description of SOEs and a table of existing extensions. An 'Add Extension' dialog box is open in the foreground, prompting the user to browse for a file to deploy.

ArcGIS Server Manager Services **Site** Security Logs

GIS Server Web Adaptor Software Authorization

Directories
Configuration Store
Clusters
Machines
Data Store
Extensions

Extensions [Help](#)

ArcGIS Server supports extensions such as Server Object Extensions (SOEs) and Geoprocessing tools. SOEs allow you to add additional features to map services, while Geoprocessing tools allow you to create web services for performing analysis.

Click Add Extension to deploy your extension to ArcGIS Server.

Name	Extension	Description	Type
GetWebMap.soe	Get Web Map	.NET REST SOE for obtaining a web map as JSON given a PARCELID	REST <input type="button" value="edit"/> <input type="button" value="delete"/>

Browse to the extension you want to deploy.

Extension: No file chosen

Add Capability to the Service...

The screenshot displays the ArcGIS Server Manager web interface. The browser address bar shows the URL: `gis-01:6080/arcgis/manager/service.html?name=TaxParcelQuery.MapServer&folder=COMPAS`. The interface includes a navigation menu with tabs for 'Services', 'Site', 'Security', and 'Logs'. Under the 'Services' tab, there are sub-tabs for 'Manage Services', 'OGC Services', 'KML Network Links', and 'Sharing'. The main content area is titled 'Editing: Site (root) > COMPAS > TaxParcelQuery' and includes 'Help', 'Save and Restart', and 'Cancel' buttons. A left-hand sidebar contains a menu with options: 'General', 'Parameters', 'Capabilities' (selected), 'Pooling', 'Processes', 'Caching', and 'Item Description'. The 'Capabilities' section is titled 'Select and configure capabilities' and contains several checkboxes: 'Mapping (always enabled)' (checked), 'WCS', 'WMS', 'Schematics', 'Network Analysis', 'WFS', 'Feature Access', 'Mobile Data Access', 'KML' (checked), and 'Get Web Map' (checked). Below this is the 'Mapping Configuration' section, which includes 'URLs' (REST and SOAP), 'Operations Allowed' (Map, Query, Data), and 'Dynamic Workspaces' (Allow per request modification of layer order and symbology). At the bottom, there is a table with columns for 'Workspace ID', 'Type', and 'Connection String', and an 'Add' button.

Calling the REST Endpoint of the SOE...

```
1870
1871 /// <summary>
1872 /// Get map option json
1873 /// </summary>
1874 /// <param name="mMapNumber">Map number</param>
1875 /// <param name="mOverlay">Overlay to get</param>
1876 /// <param name="mMapTitle">Map title</param>
1877 /// <returns></returns>
1878 private string GetMapOptions(string mMapNumber, string mOverlay, string mMapTitle)
1879 {
1880     WebClient webClient = new WebClient();
1881
1882     /// Web service
1883     string webServiceURL = "http://gis.co.kittitas.wa.us/kcgis/rest/services/COMPAS/TaxParcelQuery/MapServer/exts/GetMapOptions?f=json&IncludeOverlay=true";
1884
1885     string url = String.Format("{0}&ParcelID={1}&OverlayToInclude={2}&MapTitle={3}&f=json&IncludeOverlay=true",
1886         webServiceURL, mMapNumber, mOverlay, mMapTitle);
1887
1888     /// Get JSON
1889     string json = webClient.DownloadString(url);
1890
1891     return json;
1892 }
```



Site Analysis Application...

Site Analysis

Review | TEST

● Application details

Application #:

Map #: [19-16-03000-0024](#) [↗]

Parcel #: [955396](#) [↗]

Site address: LAMBERT RD CLE ELUM

Proposed use: TESTING

Contact info: BUGNI, AGNES M FAMILY (Owner)
1370 LAMBERT RD
CLE ELUM, WA 98922-9154

Jason Eklund (Contractor)
205
Ellesburg 98926
jason.eklund@co.kittitas.wa.us

Land Use & Zoning | Districts | **Critical Areas** | Design Criteria | Access | Flood | EH | Code | Dirt

Shoreline: Yes No

DNR Water Type: Yes No

Wetlands: Yes No

Hazardous Slope:

Comments:

Landslide area: Yes No

Priority Habitat: Yes No

Coal Mine Shaft: Yes No

Critical Areas review complete?

COMPAS info:

Shoreline: Rural

DNR: Type 4

Wetlands district: PEMA

Hazardous Slope: N/A

Landslide: N/A

Priority Habitat: TEANAWAY RIVER RIPARIAN AREA

Map Back to Applicant as part of Report

