Chapter2: Introduction to MapGuide APIs

MapGuide Server API Overview and Map Viewer API

Chapter Overview

- Introduction to MapGuide APIs
- Overview of server API
  - Site service
  - Resource service
  - Mapping and rendering service
  - Feature service
  - Tile service
- View API
  - Viewer frame set structure
  - Common API usage
    - Map manipulation
    - Information retrieval
    - Digitization

MapGuide APIs

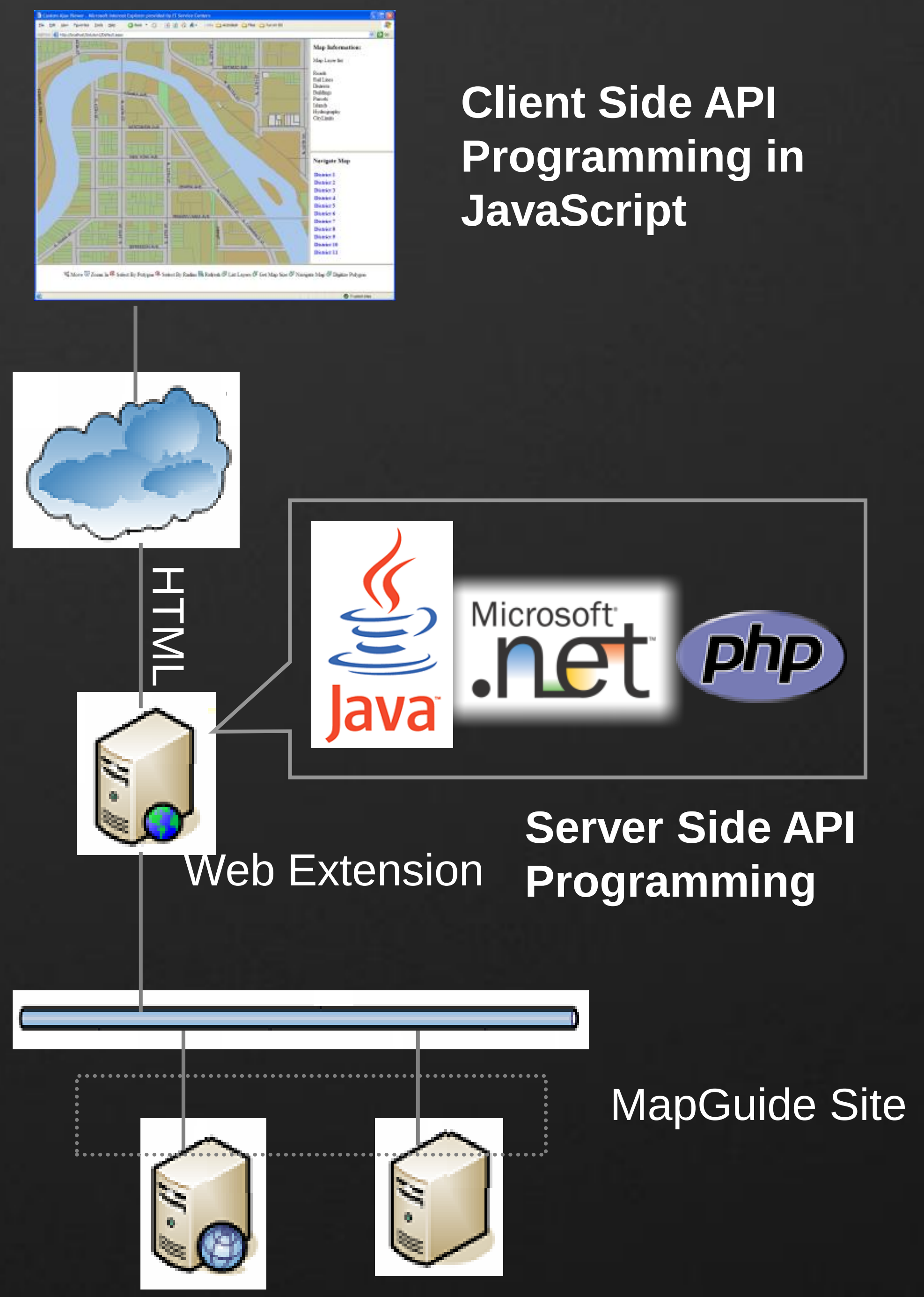
MapGuide is a server-based system. Programming is primarily on the server side.

Server APIs

- Create maps
- Control map display
- Access map data
- Perform queries and selections.
- Manage web connection and user authentication

Viewer APIs

- Interact with users
- By default, no programming required with map viewers



Client Side API Programming in JavaScript

HTML

Web Extension

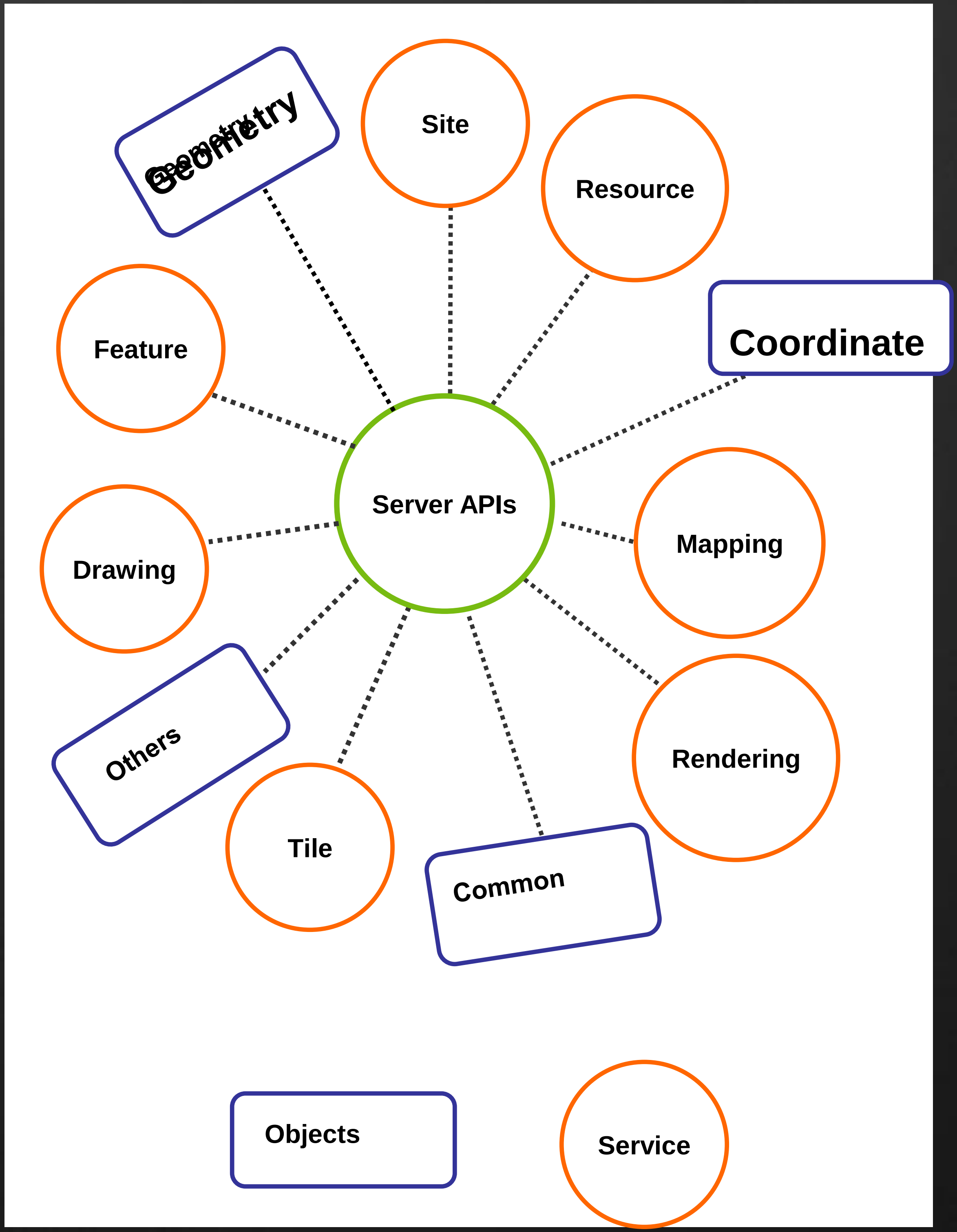
Server Side API Programming

MapGuide Site

Server Side API Overview

- Server side API is organized by service types
  - Site service
  - Resource service
  - Mapping service
  - Rendering service
  - Tile service
  - Drawing service
  - Feature service

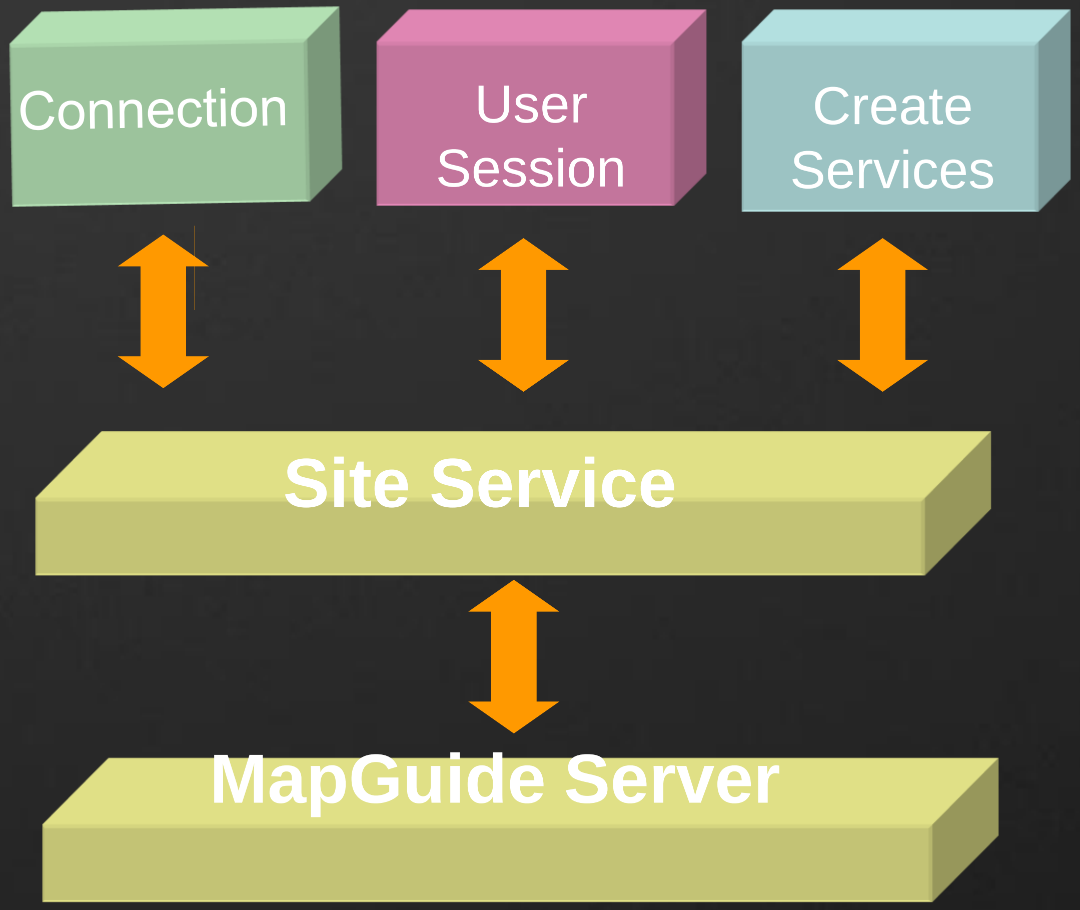
- Accompanied by auxiliary objects
  - Geometry
  - Common
  - Coordinate



Site Service

Site service

- Manages connections to MapGuide site.
- Lists site users and groups
- Controls user sessions
  - Create
  - Destroy
- Create other services by MgSiteConnection object
- Run by site server



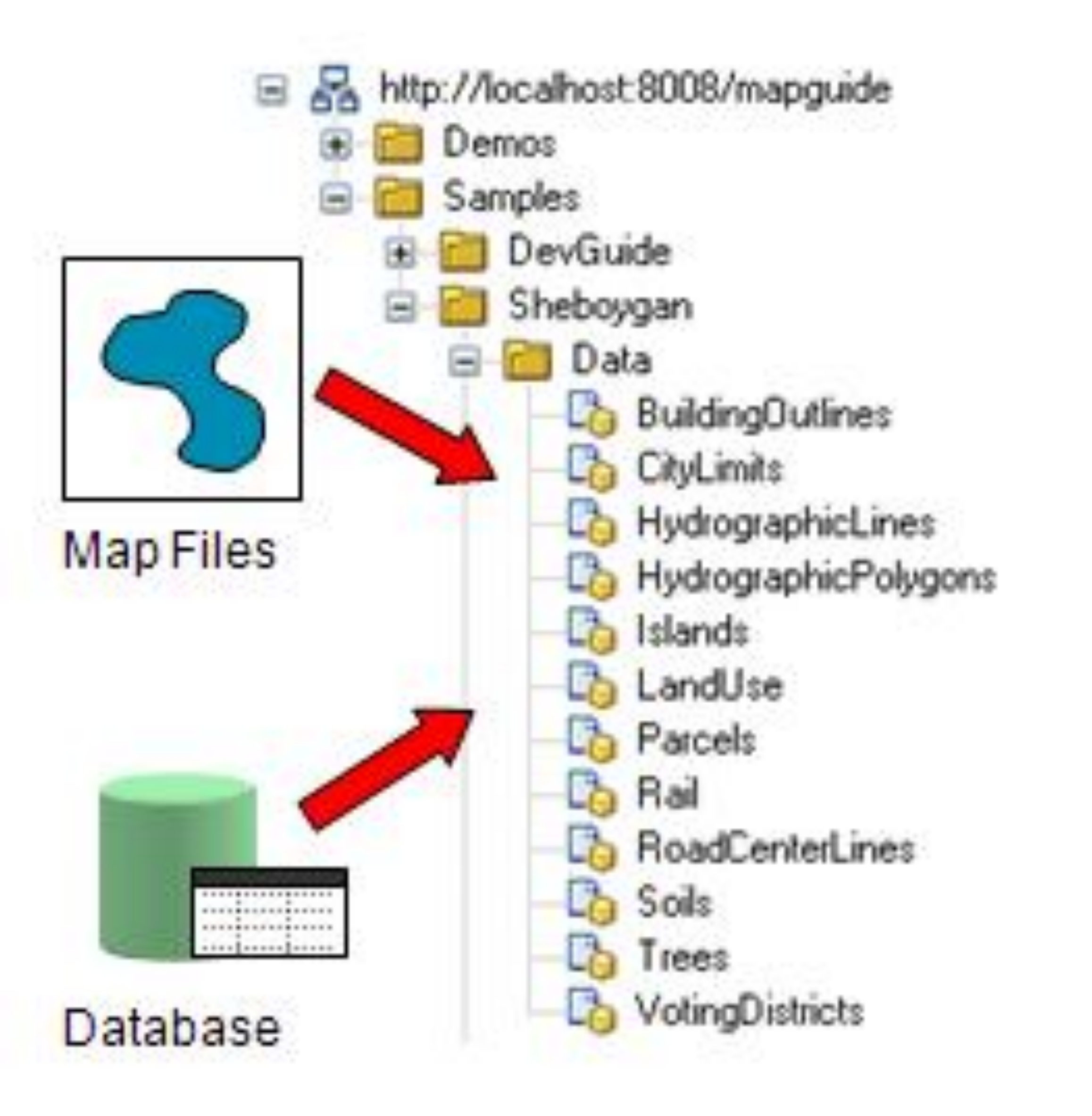
Resource Service

Resource service

- Manipulate data repository
  - Site library
  - Session repository
- Manage map data
  - Load
  - Copy
  - Move
  - Rename
  - Delete
- Run by site server

Resource identifier

- A string that uniquely identifies map resources on MapGuide site

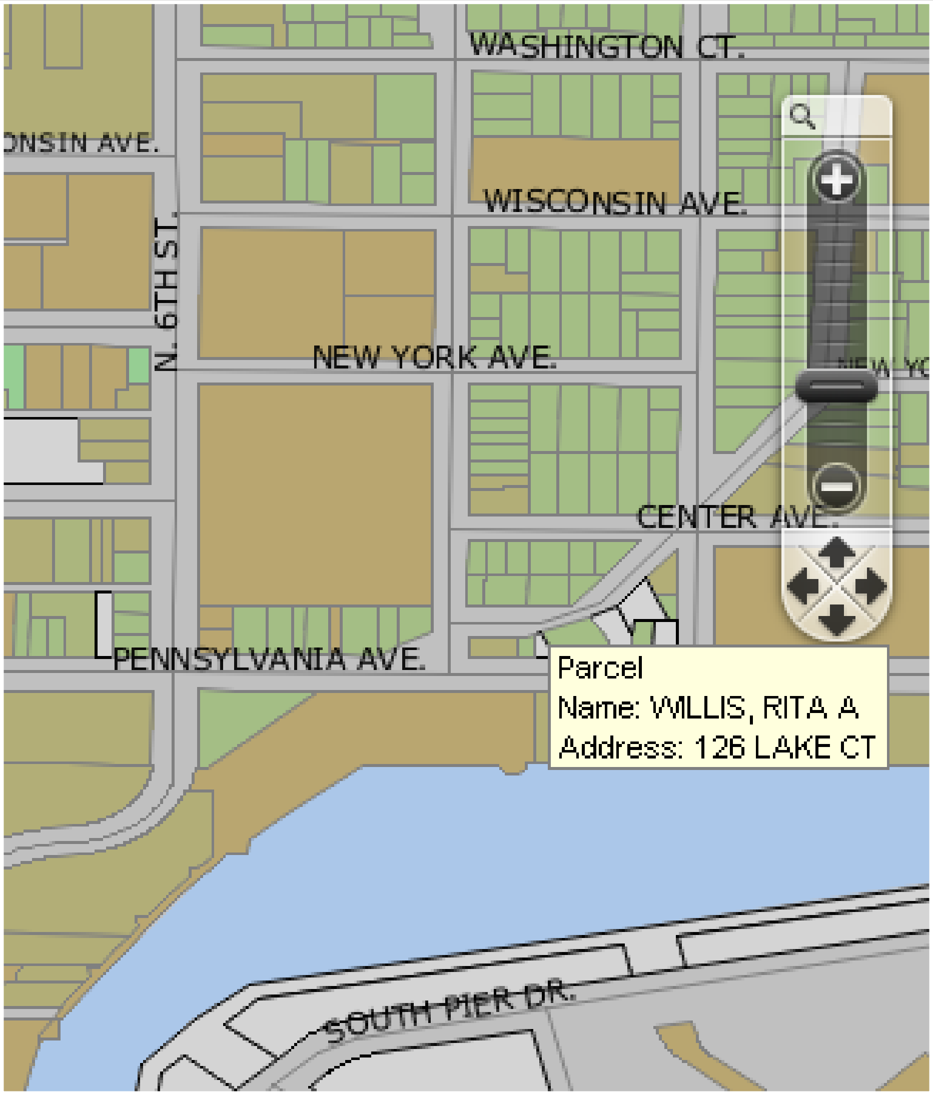


Library://Demos/GoogleEarth/Layers/SheboyganParcels.LayerDefinition

Session:a421c694-ffff-ffff-8000-005056c00008_eng//tempLayer.LayerDefinition

Mapping and Rendering Service

- Display and plot maps
- Manage map layers and layer groups
- Create map legends



Parcel
Name: WILLIS, RITA A
Address: 126 LAKE CT

Mapping service

- Create DWFs in eMaps format for map display
- Create DWFs in ePlots format for map plot

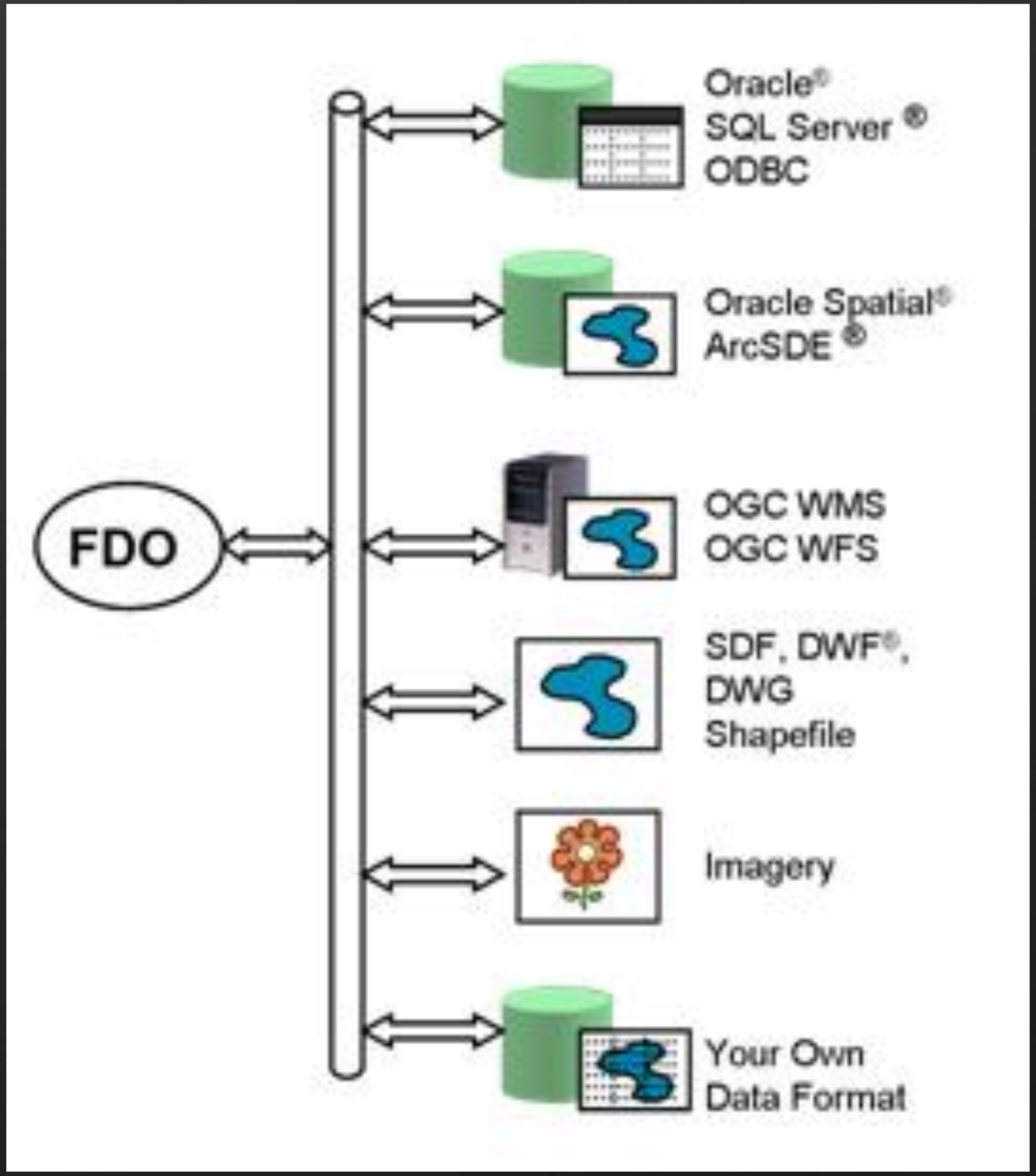
Rendering service

- Create maps in plain images

Feature Service

Feature service

- provides an abstraction layer for saving and retrieving feature data independent of data type.
- FDO is used to access map data in different formats.
- Performs spatial and tabular queries against map data sources.
- Edits map features.
  - Create
  - Update
  - Delete

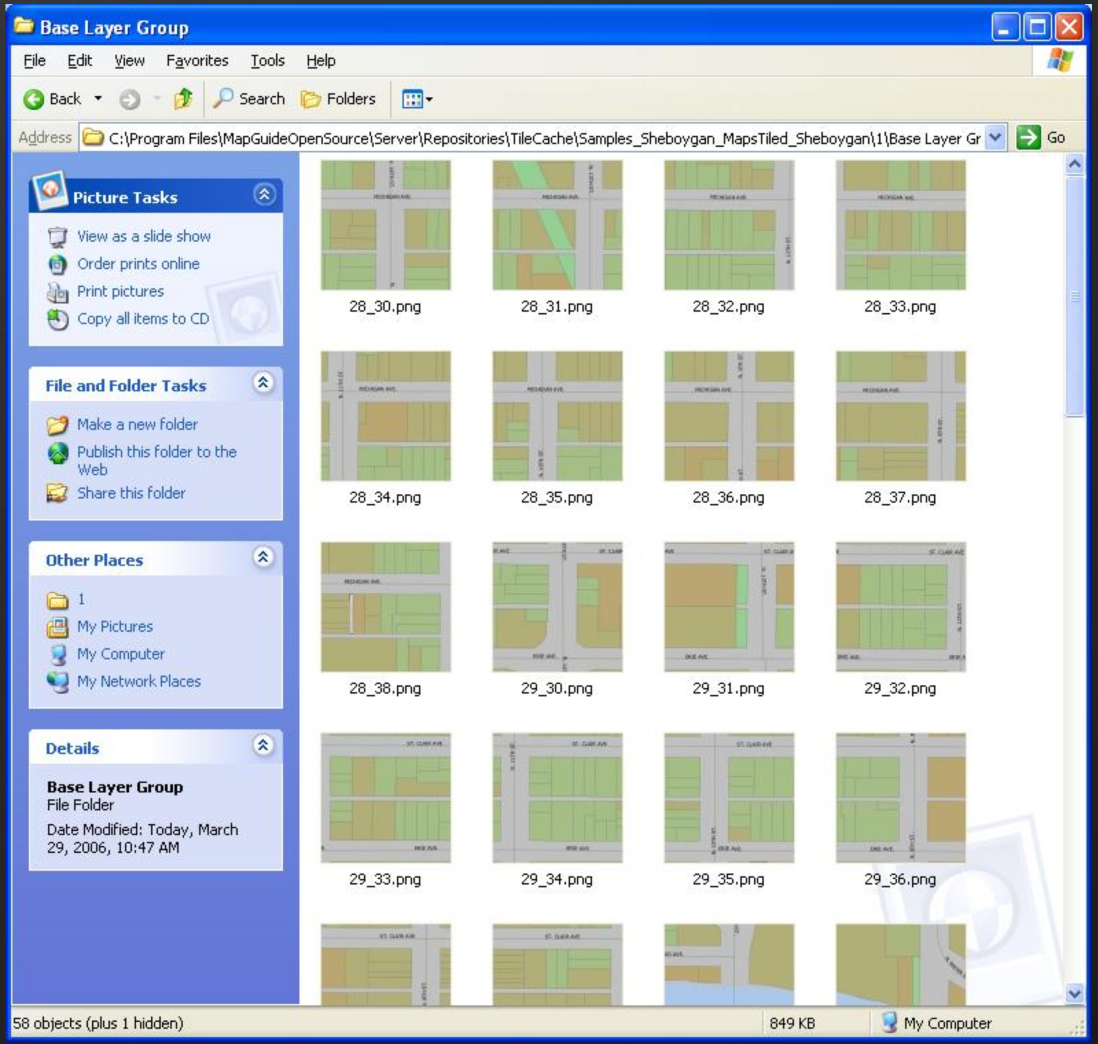


Tile Service

Tile service creates map tiles

Map Tiles –

- Enable you to smoothly and continuously move map display like Google Map.
- Created on demand.
- Cached for future use.



Address C:\Program Files\MapGuideOpenSource\Server\Repositories\TileCache\Samples_Sheboygan_MapsTiled_Sheboygan\1\Base Layer Gr

Details

Base Layer Group
File Folder

Date Modified: Today, March 29, 2006, 10:47 AM

58 objects (plus 1 hidden)

849 KB

Get Started with Server API Programming

Certain steps need to be followed in .NET

1. Initialize web tier with MapGuide web configuration file.
2. Connect to the site, get session ID and web layout path.
3. Reference the proper viewer with web layout and session id

```
MapGuideApi.MgInitializeWebTier(@"C:\MapGuide\webconfig.ini");
MgUserInformation userInfo = new MgUserInformation(
    "Anonymous", "");
MgSite site = new MgSite();
site.Open(userInfo);
String sessionId = site.CreateSession();
String webLayout =
    "Library://Exercise/Layouts/SheboyganAsp.WebLayout";
```

```
<frame src="/mapserver/mapviewernet/ajaxviewer.aspx?SESSION=
<%= sessionId %>&WEBLAYOUT=<%= webLayout %>" />
```

```
<frame src="/mapserver/mapviewernet/dwfviewer.aspx?SESSION=
<%= sessionId %>&WEBLAYOUT=<%= webLayout %>" />
```

Build AIMS Applications

Build AIMS Applications

Integrated Development Enviroment

AIMS 2012:

- Visual Studio 2010
- .net framework 3.5
- **NOT** .net framework 4.0
- Visual Studio 2008

AIMS 2013

- Visual Studio 2010/2012
- .net framwork 4.0

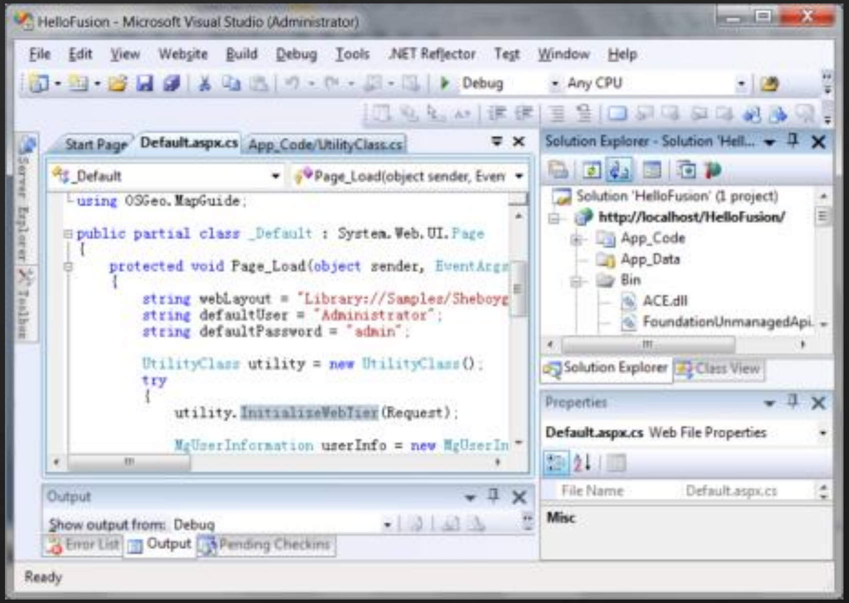
Build AIMS Applications

For 32bit – Asp.net Website or WebApplication

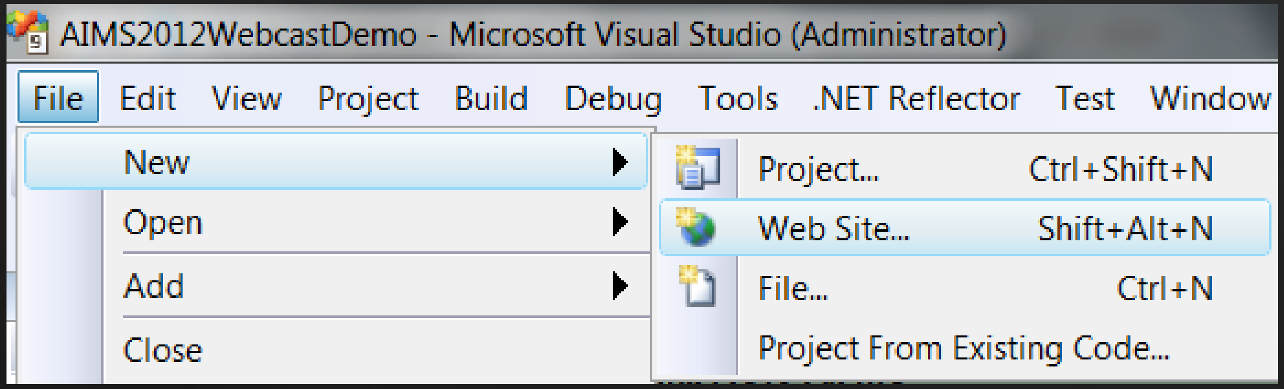
Create an ASP.Net Website in Visual Studio on local IIS

Add MapGuide references

- Copy all MapGuide .net assemblies to bin folder
- Initialize MapGuide web tier
- Connect to MapGuide Server
- Embed MapGuide viewer in webpage


HelloFusion - Microsoft Visual Studio (Administrator)
File Edit View Website Build Debug Tools .NET Reflector Test Window Help
Debug
Any CPU
Start Page
Default.aspx.cs
App_Code/UtilityClass.cs
_Default
Page_Load(object sender, Even
Solution Explorer - Solution 'Hell...
Solution 'HelloFusion' (1 project)
http://localhost/HelloFusion/
App_Code
App_Data
Bin
ACE.dll
FoundationUnmanagedApi.
Solution Explorer
Class View
Properties
Default.aspx.cs Web File Properties
File Name
Default.aspx.cs
Misc
Output
Show output from: Debug
Error List
Output
Pending Checkins
Ready

```
using OSGeo.MapGuide;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs
    {
        string webLayout = "Library://Samples/Sheboyg
        string defaultUser = "Administrator";
        string defaultPassword = "admin";

        UtilityClass utility = new UtilityClass();
        try
        {
            utility.InitializeWebTier(Request);

            MgUserInformation userInfo = new MgUserIn
```


AIMS2012WebcastDemo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Tools .NET Reflector Test Window
New
Open
Add
Close
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
File... Ctrl+N
Project From Existing Code...

No difference with building MGE 2011 applications

Build AIMS Applications

For 64bit -- a little tricky, use Asp.net WebApplication

Issue:

Could not load file or assembly 'OSGeo.MapGuide.Foundation, Version=2.3.0.4202, Culture=neutral, PublicKeyToken=null' or one of its dependencies. An attempt was made to load a program with an incorrect format.

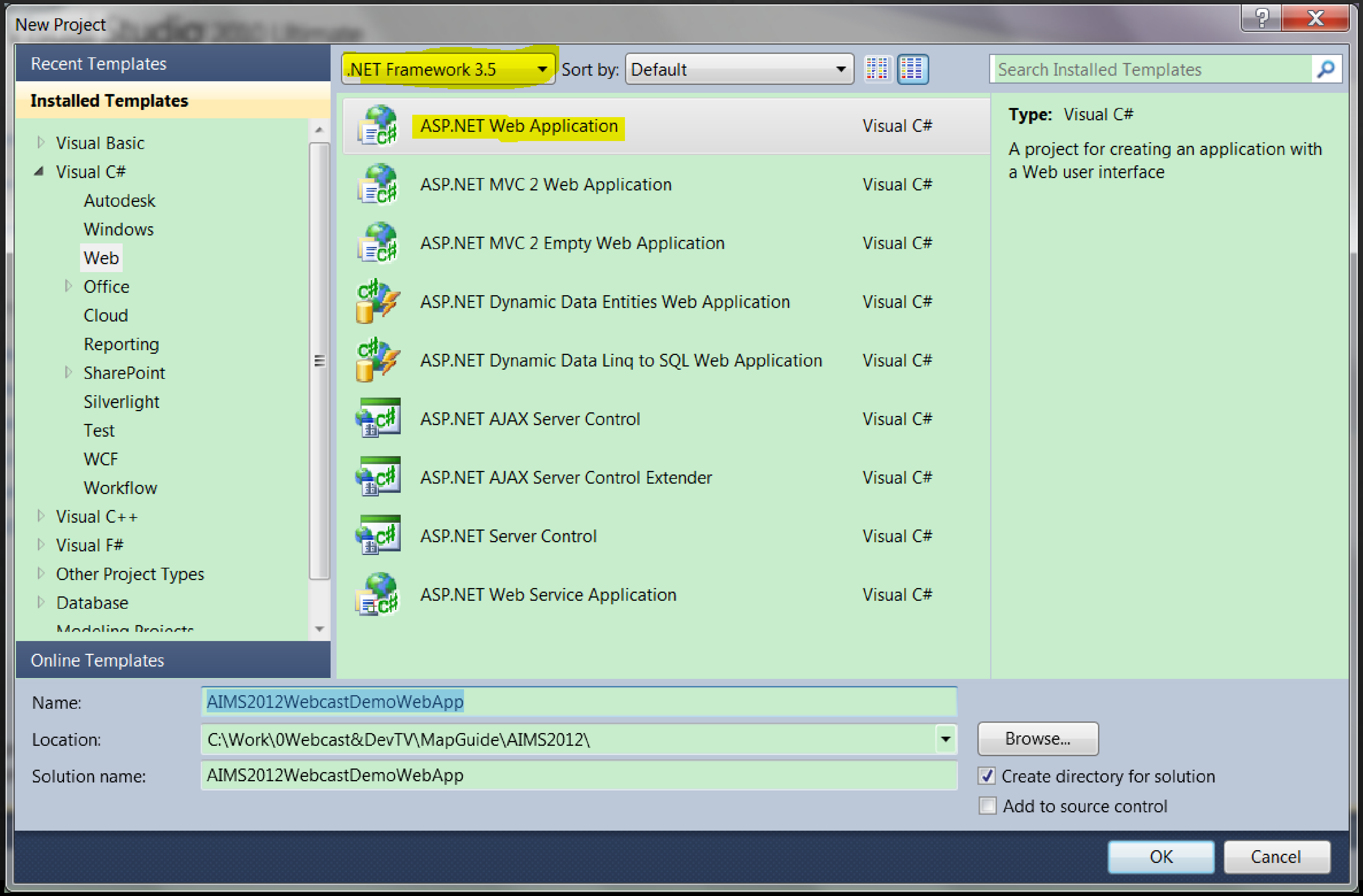
Solution:

using WebApplication instead of Website in Visual Studio to hide this problem.

Build AIMS 2012 Applications

For 64bit -- a little tricky, use Asp.net WebApplication

1. New Project → ASP.NET Web Application, select .NET Framework 3.5



.NET Framework 3.5 Sort by: Default

ASP.NET Web Application	Visual C#
ASP.NET MVC 2 Web Application	Visual C#
ASP.NET MVC 2 Empty Web Application	Visual C#
ASP.NET Dynamic Data Entities Web Application	Visual C#
ASP.NET Dynamic Data Linq to SQL Web Application	Visual C#
ASP.NET AJAX Server Control	Visual C#
ASP.NET AJAX Server Control Extender	Visual C#
ASP.NET Server Control	Visual C#
ASP.NET Web Service Application	Visual C#

Type: Visual C#

A project for creating an application with a Web user interface

Name: AIMS2012WebcastDemoWebApp

Location: C:\Work\0Webcast&DevTV\MapGuide\AIMS2012\

Solution name: AIMS2012WebcastDemoWebApp

☑ Create directory for solution

☐ Add to source control

Build AIMS 2012 Applications

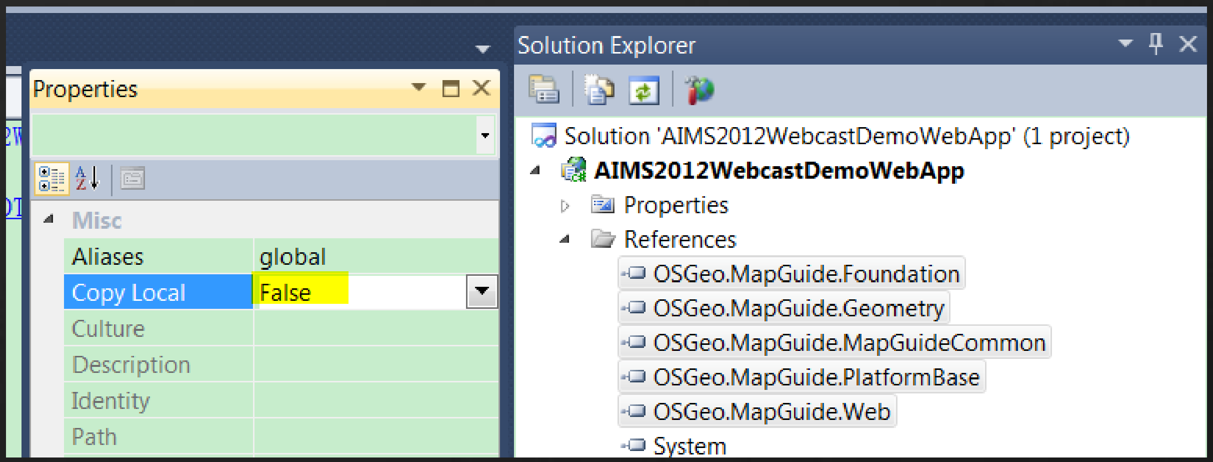
For 64bit -- a little tricky, use Asp.net WebApplication

2. Copy all dlls from AIMS web extension to <YourWebApp>\bin

<Infrastructure Web Server Extension >\www\mapviewernet\bin
→
<AIMSWebcastDemoWebApp>\bin

3. Add references to following assemblies and set “copy local” to False

OSGeo.MapGuide.Foundation.dll
OSGeo.MapGuide.Geometry.dll
OSGeo.MapGuide.MapGuideCommon.dll
OSGeo.MapGuide.PlatformBase.dll
OSGeo.MapGuide.Web.dll



Misc

Aliases	global
Copy Local	False
Culture	
Description	
Identity	
Path	

Solution 'AIMS2012WebcastDemoWebApp' (1 project)
- **AIMS2012WebcastDemoWebApp**
  - Properties
  - References
    - OSGeo.MapGuide.Foundation
    - OSGeo.MapGuide.Geometry
    - OSGeo.MapGuide.MapGuideCommon
    - OSGeo.MapGuide.PlatformBase
    - OSGeo.MapGuide.Web
    - System

Build AIMS 2012 Applications

For both 32bit and 64bit

4. InitializeWebTier
5. Connect to MapGuide Server
6. Generate Session String

```
public string sessionId, weblayout;

MapGuideApi.MgInitializeWebTier(@”webconfig.ini");
MgUserInformation userInfo = new MgUserInformation(defaultUser,
  defaultPassword);
MgSite site = new MgSite();
site.Open(userInfo);
sessionId = site.CreateSession();

Weblayout = “Library://Exercise/WebCast/Hello.weblayout”;
```

Build AIMS 2012 Applications

For both 32bit and 64bit

7. Pass session string to Ajax Viewer

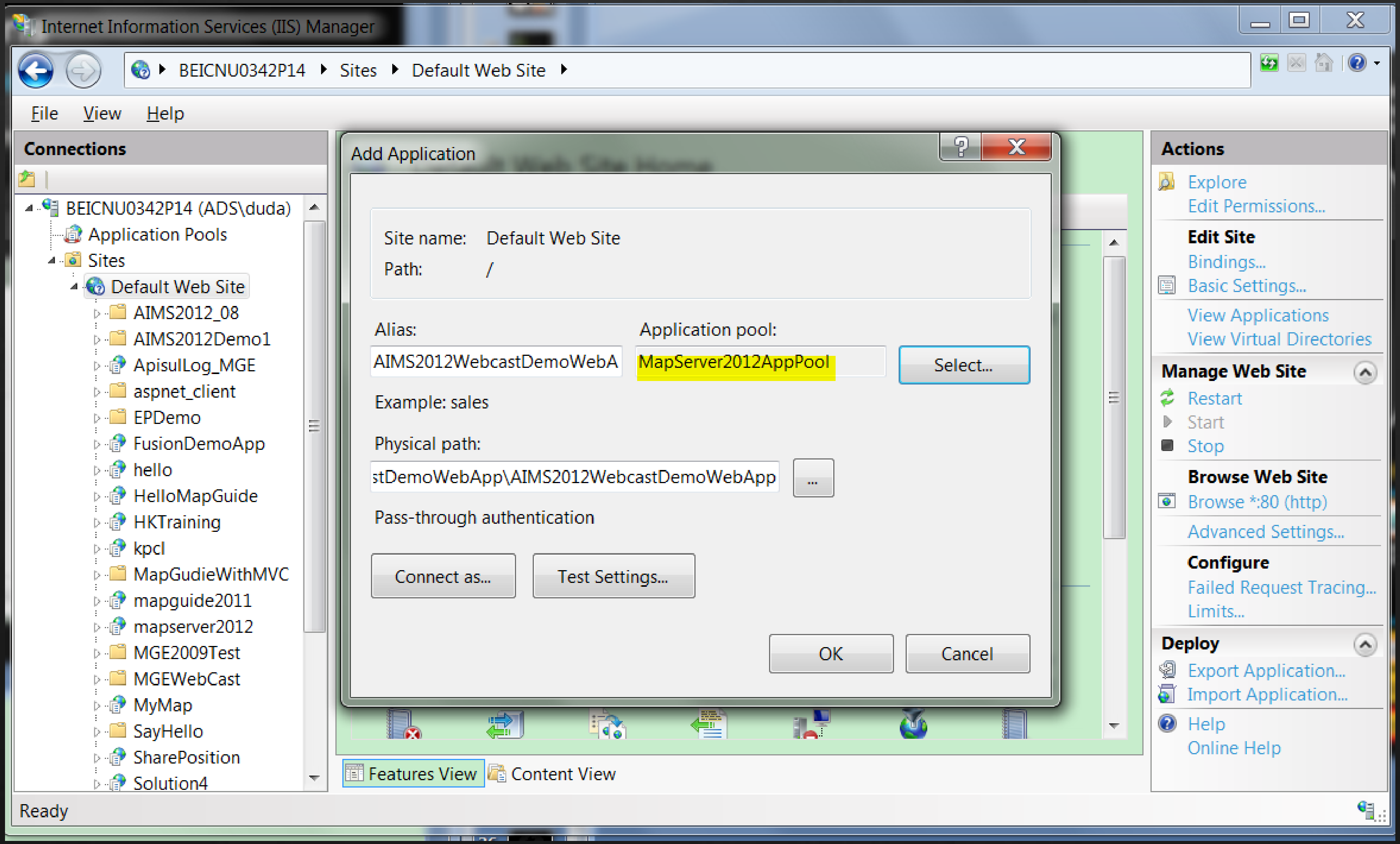
```
  <html xmlns="http://www.w3.org/1999/xhtml" >
<head>
   <title>Exercise 4</title>
</head>
<frameset rows="100,*" border="0" framespacing="0">
  <frame src = “Banner.htm" />
  <frame
   src="/mapserver201x/mapviewernet/ajaxviewer.aspx?SESSION=
   <%= sessionId %>&WEBLAYOUT=<%= webLayout %>"
   name="ViewerFrame" />
</frameset>
</html>
```

Build AIMS 2012 Applications

For 64bit -- a little tricky, only Asp.net WebApplication

8. Create Web Application in IIS manager, direct to your web app

“MapServer AppPool” is recommended



Add Application

Site name: Default Web Site
Path: /

Alias: AIMS2012WebcastDemoWebA
Application pool: MapServer2012AppPool
Example: sales

Physical path: stDemoWebApp\AIMS2012WebcastDemoWebApp

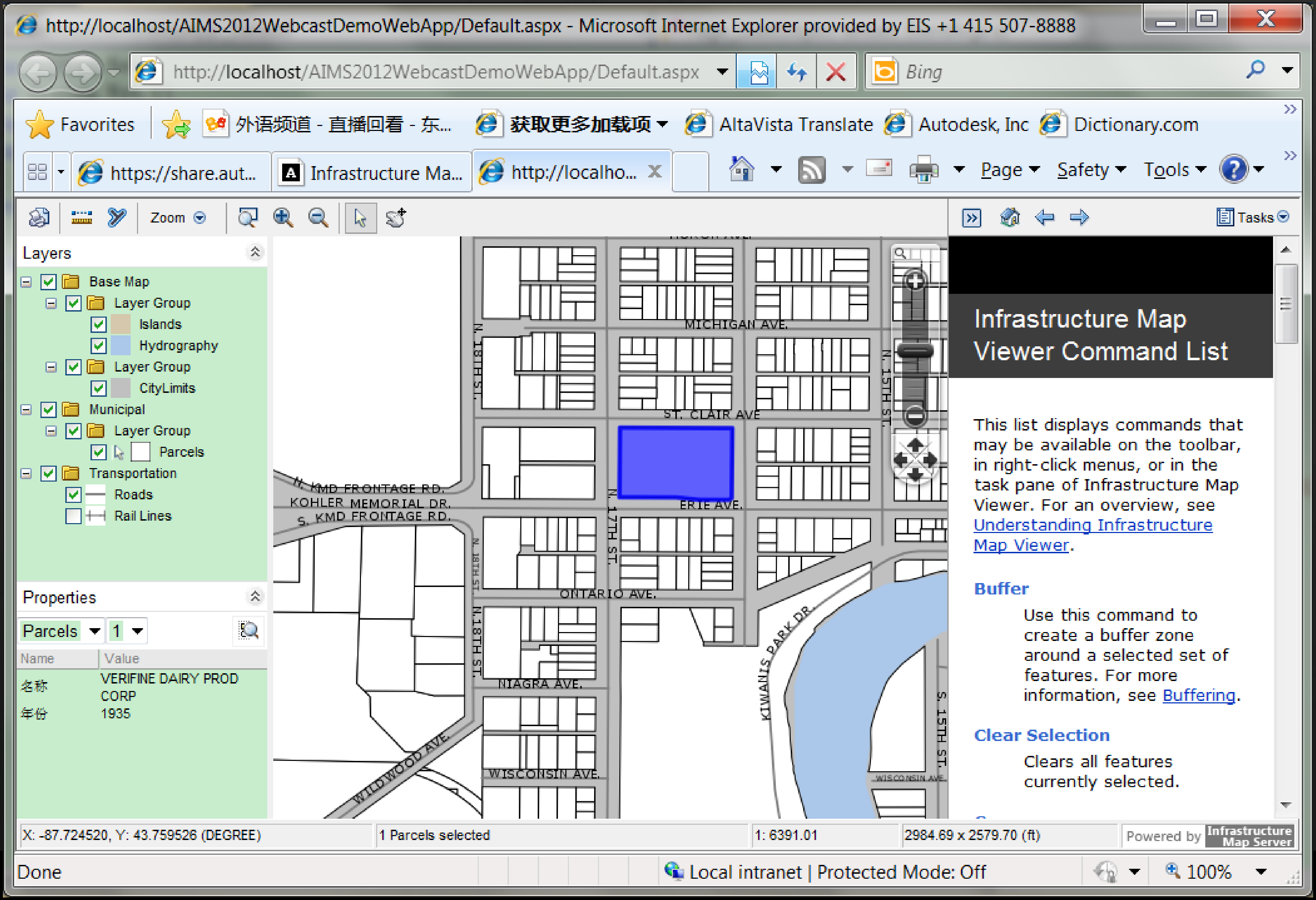
Pass-through authentication

Autodesk

Build AIMS 2012 Applications

For 64bit -- a little tricky, only Asp.net WebApplication

9. Build you application and browse your web page in browser, it should be working



http://localhost/AIMS2012WebcastDemoWebApp/Default.aspx - Microsoft Internet Explorer provided by EIS +1 415 507-8888

http://localhost/AIMS2012WebcastDemoWebApp/Default.aspx

Infrastructure Map Viewer Command List

This list displays commands that may be available on the toolbar, in right-click menus, or in the task pane of Infrastructure Map Viewer. For an overview, see Understanding Infrastructure Map Viewer.

Buffer

Use this command to create a buffer zone around a selected set of features. For more information, see Buffering.

Clear Selection

Clears all features currently selected.

Properties: Parcels, 1

Name	Value
名称	VERIFINE DAIRY PROD CORP
年份	1935

X: -87.724520, Y: 43.759526 (DEGREE) | 1 Parcels selected | 1: 6391.01 | 2984.69 x 2579.70 (ft)

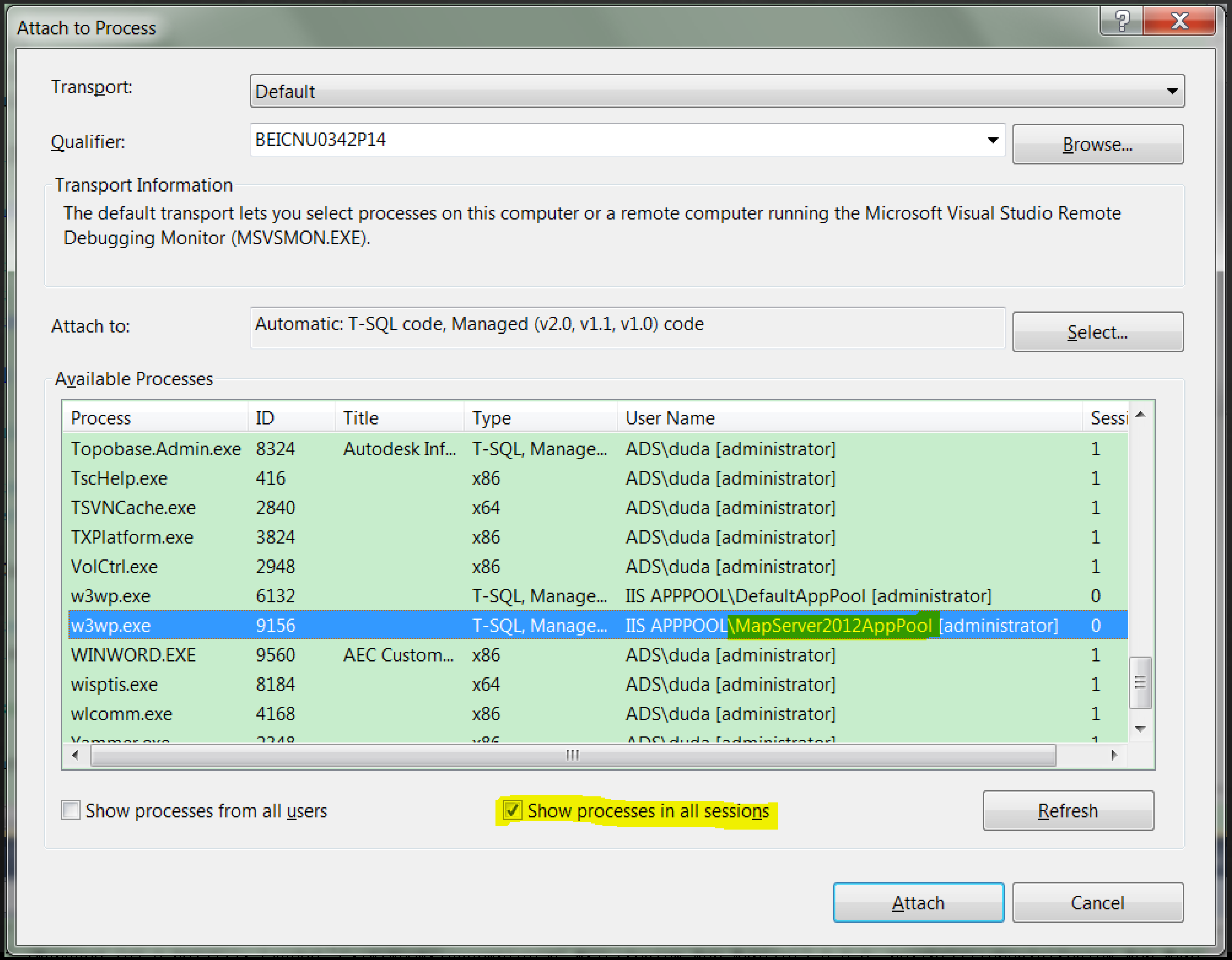
Done | Local intranet \| Protected Mode: Off | 100%

Build AIMS 2012 Applications

For 64bit -- a little tricky, only Asp.net WebApplication

10. DEBUG: Attach process to w3wp.exe, do not use F5

pay attention to the AppPool



Attach to Process

Transport: Default

Qualifier: BEICNU0342P14

Transport Information

The default transport lets you select processes on this computer or a remote computer running the Microsoft Visual Studio Remote Debugging Monitor (MSVSMON.EXE).

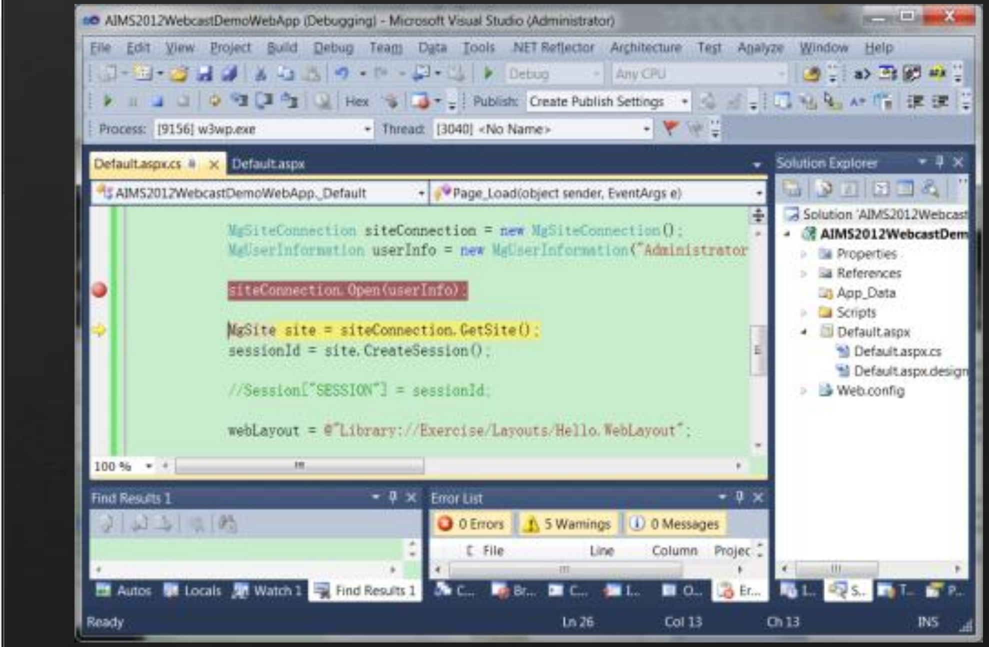
Attach to: Automatic: T-SQL code, Managed (v2.0, v1.1, v1.0) code

Available Processes

Process	ID	Title	Type	User Name	Sessi
Topobase.Admin.exe	8324	Autodesk Inf...	T-SQL, Manage...	ADS\duda [administrator]	1
TscHelp.exe	416		x86	ADS\duda [administrator]	1
TSVNCache.exe	2840		x64	ADS\duda [administrator]	1
TXPlatform.exe	3824		x86	ADS\duda [administrator]	1
VolCtrl.exe	2948		x86	ADS\duda [administrator]	1
w3wp.exe	6132		T-SQL, Manage...	IIS APPPOOL\DefaultAppPool [administrator]	0
w3wp.exe	9156		T-SQL, Manage...	IIS APPPOOL\MapServer2012AppPool [administrator]	0
WINWORD.EXE	9560	AEC Custom...	x86	ADS\duda [administrator]	1
wisptis.exe	8184		x64	ADS\duda [administrator]	1
wlcomm.exe	4168		x86	ADS\duda [administrator]	1
Yammer.exe	2348		x86	ADS\duda [administrator]	1

☐ Show processes from all users

☑ Show processes in all sessions



AIMS2012WebcastDemoWebApp (Debugging) - Microsoft Visual Studio (Administrator)

Process: [9156] w3wp.exe Thread: [3040] <No Name>

AIMS2012WebcastDemoWebApp._Default Page_Load(object sender, EventArgs e)

```
MgSiteConnection siteConnection = new MgSiteConnection();
MgUserInformation userInfo = new MgUserInformation("Administrator

siteConnection.Open(userInfo);

MgSite site = siteConnection.GetSite();
sessionId = site.CreateSession();

//Session["SESSION"] = sessionId;

webLayout = @"Library://Exercise/Layouts/Hello.WebLayout";
```

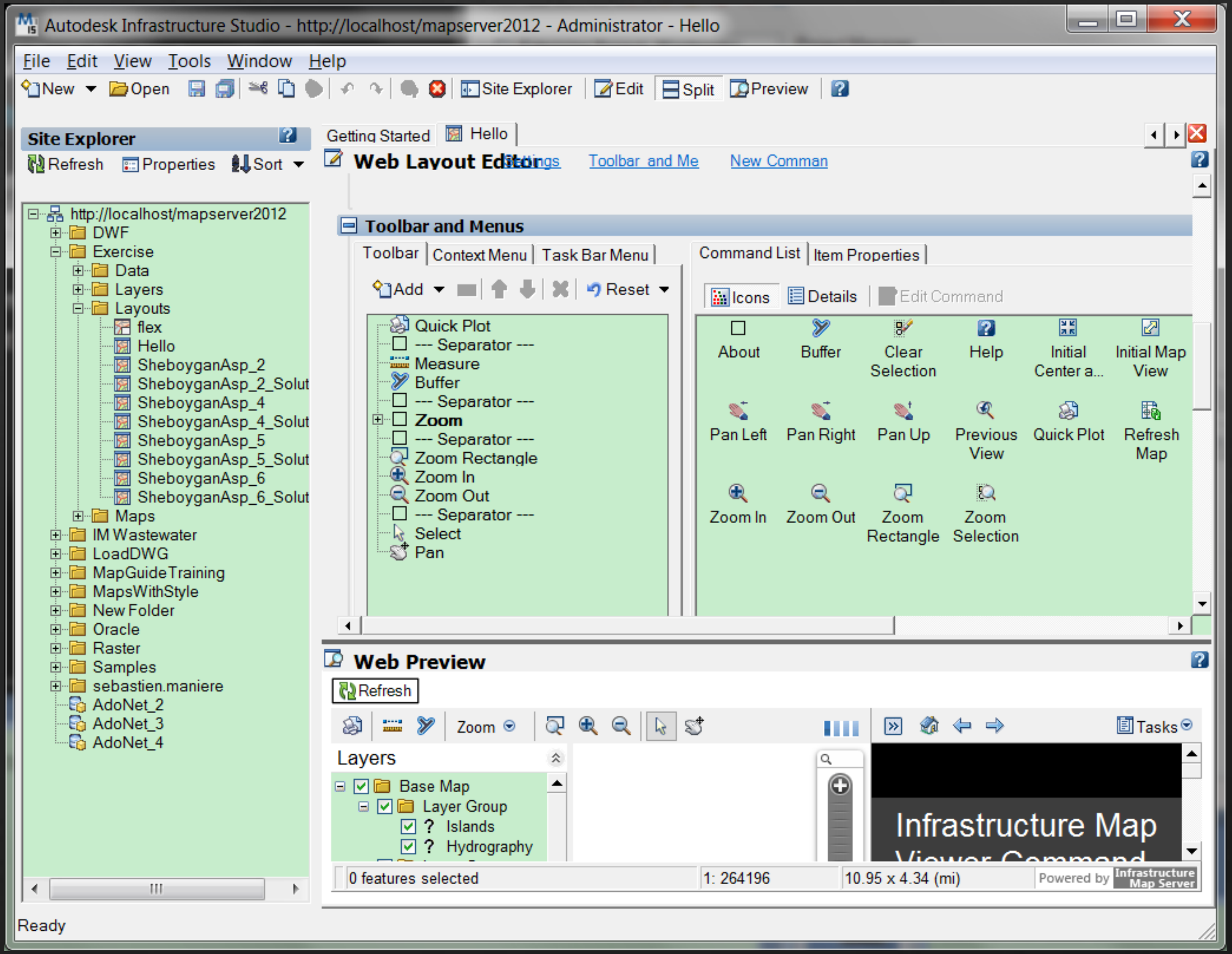
0 Errors 5 Warnings 0 Messages

Build AIMS 2012 Applications

Extend custom functionalities

11. New Command
    - Invoke Script
    - Invoke URL
12. Add Command to layout
13. Implement the command(JavaScript/Web page)

Infrastructure Studio



For AIMS 2013/2014

Use .NET Framework 4.0/ASP.NET 4.0

Use correct version of dlls in <webExtension>\bin folder

Use correct version of webconfig.ini

Steps are almost the same with AIMS 2012.

Demo – Build AIMS custom applications

1. Create web application
2. Copy dlls/ Add Reference
3. Embed Ajaxviewer in webpage
4. How to debug
5. Create custom command
6. Implement the custom command

Check out this video:

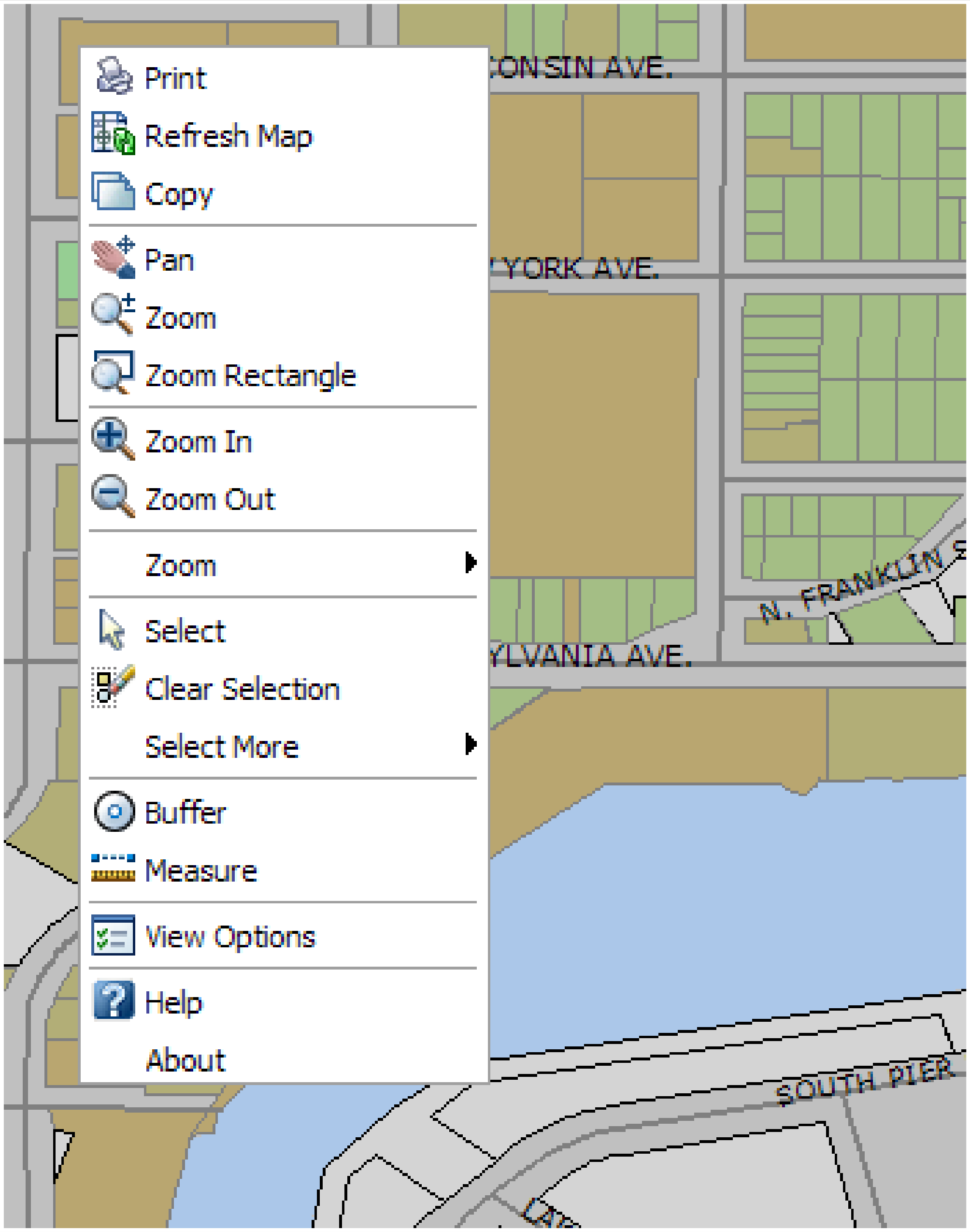
Video : Autodesk® Infrastructure Map Server 2012 API Webcast
Recorded version of the Autodesk® Infrastructure Map Server 2012 API webcast
View Online | Download

Exercise

- Create first MapGuide Web Application

Map Viewer API

- A set of JavaScript functions in browsers
- A uniform abstraction layer that works with both DWF and AJAX viewer
- Provide functions to interact with users in browser.
  - Zoom, pan, scale, refresh
  - Select and clear
  - Digitize
- In most cases, if default web layouts are used, there is no need to write program against viewer API.

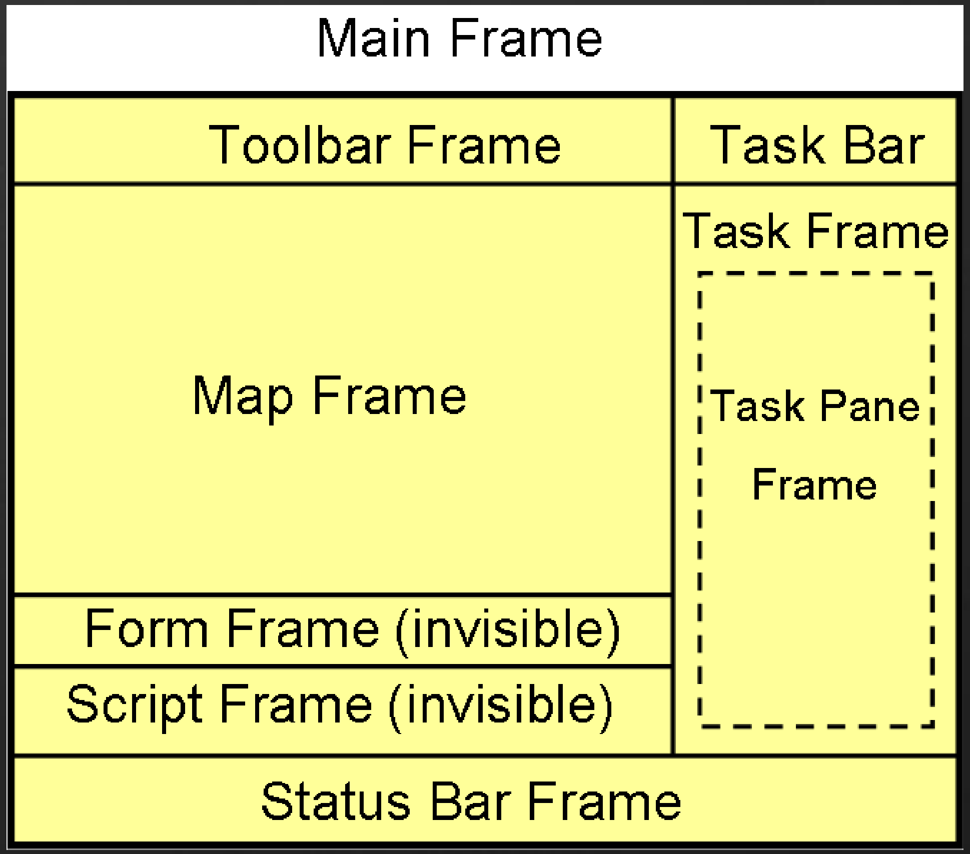


MapGuide Viewer API

- Map viewer has a set of frames.
- Frame set is in hierarchical relationship.
- Each frame has its own API methods.
- You need to invoke the method in the proper frame.

```
//From taskPaneFrame to refresh map
parent.parent.mapFrame.Refresh();
//From sbFrame to refresh map
parent.mapFrame.Refresh();
```

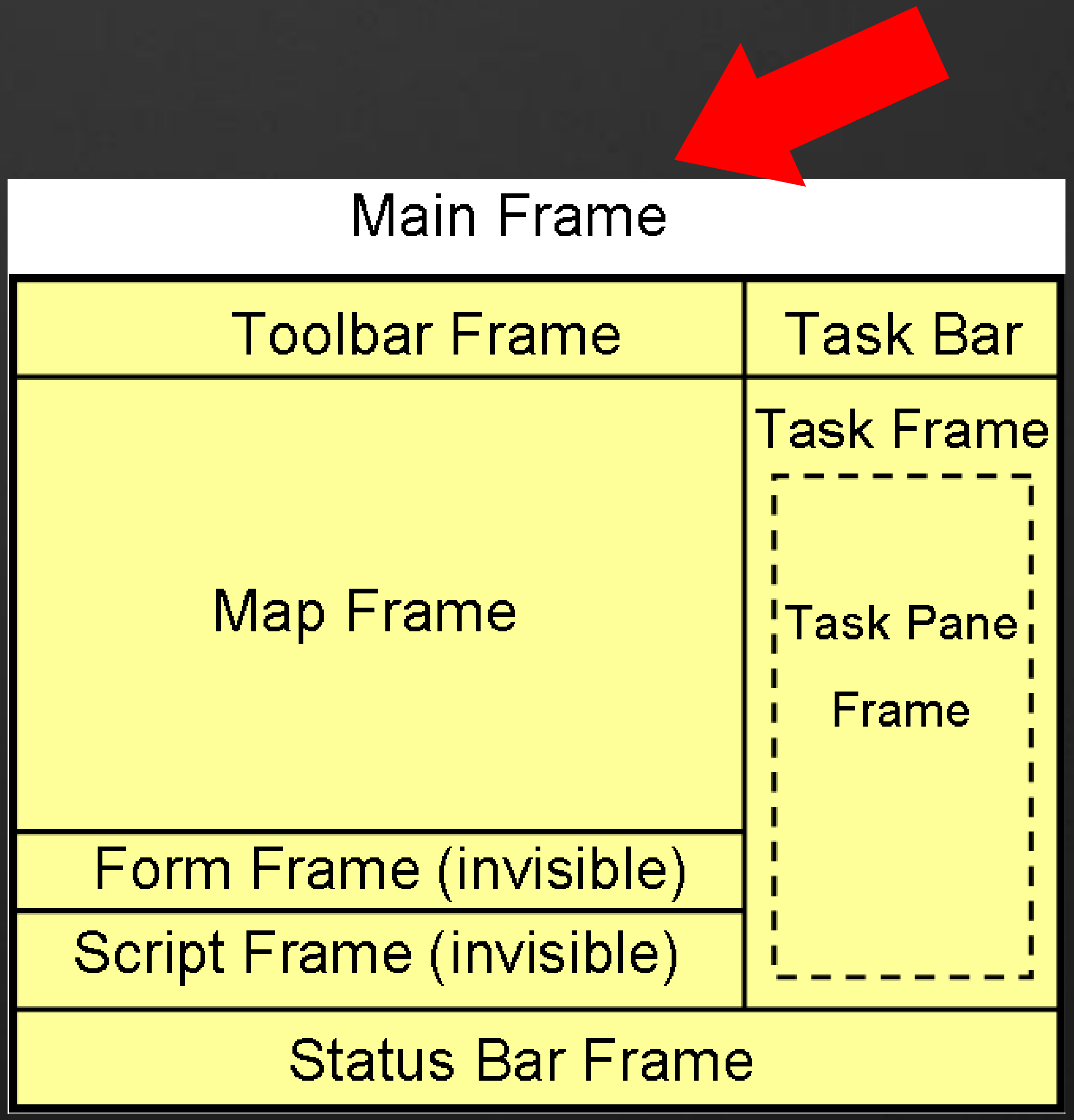
Viewer Frame Structure



Main Frame APIs

Main frame has methods

- Execute map actions
  - Pan
  - Zoom
  - Select
  - Refresh
- Get other frames
  - Map frame
  - Form frame
  - Scrip frame
  - Task frame
- Get other items
  - Toolbar items
  - Context menu items



Execute Map Actions

ExecuteMapAction

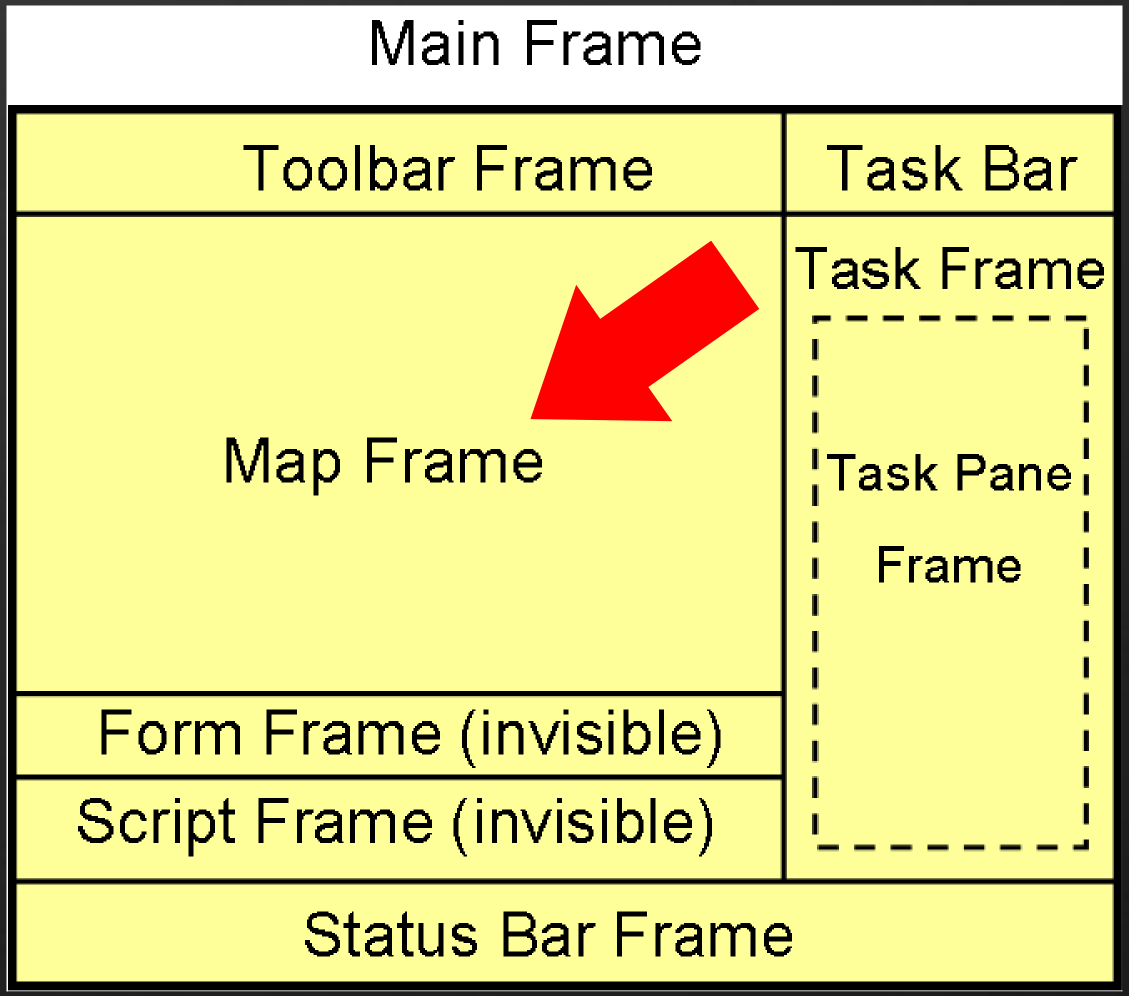
- Method for map manipulations
- Has 22 actions
- Coded by number
  - Enter pan mode: 1
  - Zoom by rectangle: 9
  - Select: 15
  - Refresh: 20

```
//Pan map
parent.mainFrame.ExecuteMapAction(1);

//Zoom by rectangle
parent.mainFrame.ExecuteMapAction(9);

//Refresh
parent.mainFrame.ExecuteMapAction(15);
```

Map Frame APIs

Map frame has methods

- Digitize new geometries
  - Polygon
  - Polyline
  - Circle
  - Rectangle
  - Line
  - Point
- Retrieve map information
  - Layers
  - Map center/scale
  - Map size
  - Selection
- Convert units
  - Screen to map



Map Information Retrieval

- Use viewer API to get map status
  - Map size (in map units)
  - Map layers

```
//Get map size
function GetMapSize() {
    var height = parent.ViewerFrame.GetMapFrame().GetMapHeight();
    var width = parent.ViewerFrame.GetMapFrame().GetMapWidth();
}
```

```
//Get map layers
function ListLayers() {
    var layers = parent.ViewerFrame.GetMapFrame().
        GetLayers(false, false);

    var outString = "";
    for (var i = 0; i < layers.length; i++) {
        var layer = layers[i];
        outString = outString + layer.name + "<br>";}
    return
}
```

Zoom to View

- Use map frame API to zoom to a certain view
- Methods take these arguments
  - X coordinate of view center
  - Y coordinate of view center
  - Scale
  - If refresh the whole map (Ajax Viewer always refreshes)

```
//Zoom to map view
function GoToView(x, y, scale, refresh) {
    parent.ViewerFrame.GetMapFrame().ZoomToView
         (x, y, scale, refresh);

}
```

```
<p onclick="GoToView(-87.7237299526527, 43.7691299333886,
        25000, true)">District 1</p>
```

Digitize Geometries

- Map frame APIs provide methods to digitize new geometries.
- To digitize polygons,
  1. Write an event handler to take the finished polygon.
  2. Pass the event handler as the parameter to invoke DigitizePolygon().

```
//Event handler
function OnPolygonDigitized(poly) {
    ShowResults(FormatLineResults(poly));
}
function DigitizePolygon() {
    parent.ViewerFrame.GetMapFrame().DigitizePolygon(
        OnPolygonDigitized);
}
```

```
<input type="button" value="Digitize"
         onclick="DigitizePolygon();" />
```

Flexible web layout/Fusion viewer API

- **DevTV: MapGuide Fusion Technology Introduction**
  A self-paced video tutorial on how to build a web-based mapping application using the Fusion Viewer Technology and MapGuide Enterprise.
  View Online | Download

- Document of Flexible web layout in Developer's Guide
  - http://wikihelp.autodesk.com/Infr._Map_Server/enu/2013/Help/0005-Develope0/0094-Flexible94

- All source code:
  - C:\Program Files\Autodesk\Autodesk Infrastructure Web Server Extension 201x\www\fusion

Mobile Viewer API

- Mobile Viewer Reference Guide
  - http://wikihelp.autodesk.com/Infr._Map_Server/enu/2013/Help/0006-Mobile_V0

- **Blog posts on Infrastructure Modeling DevBlog**
  - **Creating custom control for Mobile Viewer of AIMS**
  - Debugging Fusion Viewer or **Mobile** Viewer of AIMS in Firebug
  - Speed up the scrolling of **Mobile** Viewer panel – Infrastructure
  - Search for more …

- All source code:
  - C:\Program Files\Autodesk\Autodesk Infrastructure Web Server Extension 201x\www\MobileViewer

Questions

Questions ?

Exercise

- Solution 2 – viewer API