



Autodesk MapGuide® Enterprise 2010

Partha Sarkar
Developer Consultant

Bio

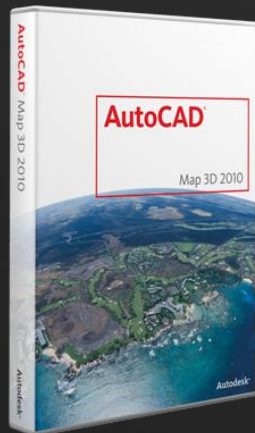


Partha has 10 years of industry experience in the design, analysis, development, and implementation of many civil engineering and geospatial projects and applications. Based in Bangalore, India, he currently works in the Developer Technical Services (DevTech) Division of Autodesk. Before joining Autodesk, he worked for many large-scale infrastructure projects including metro rails, highways, and defense projects. He has a Bachelor's and Master's degree in Civil Engineering from Premier Universities in India

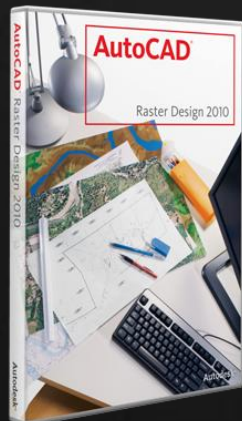
Agenda

- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- MapGuide Enterprise API Overview
- Integration of Fusion & Google Maps
- Data Migration Tool

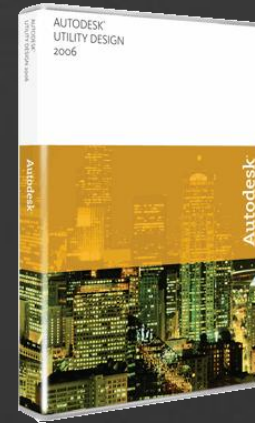
AutoCAD® Map 3D



AutoCAD® Raster Design

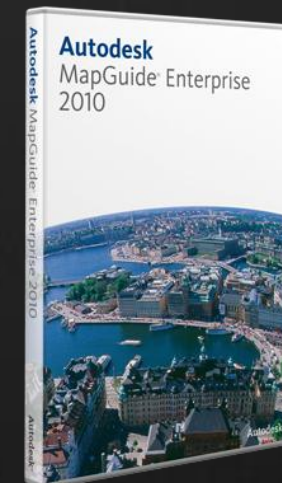
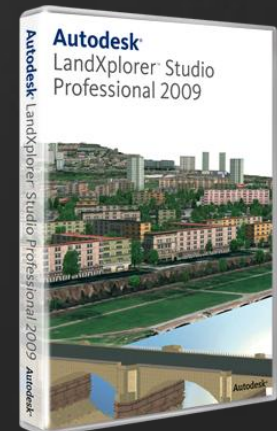


Autodesk® Topobase™ Client & Web

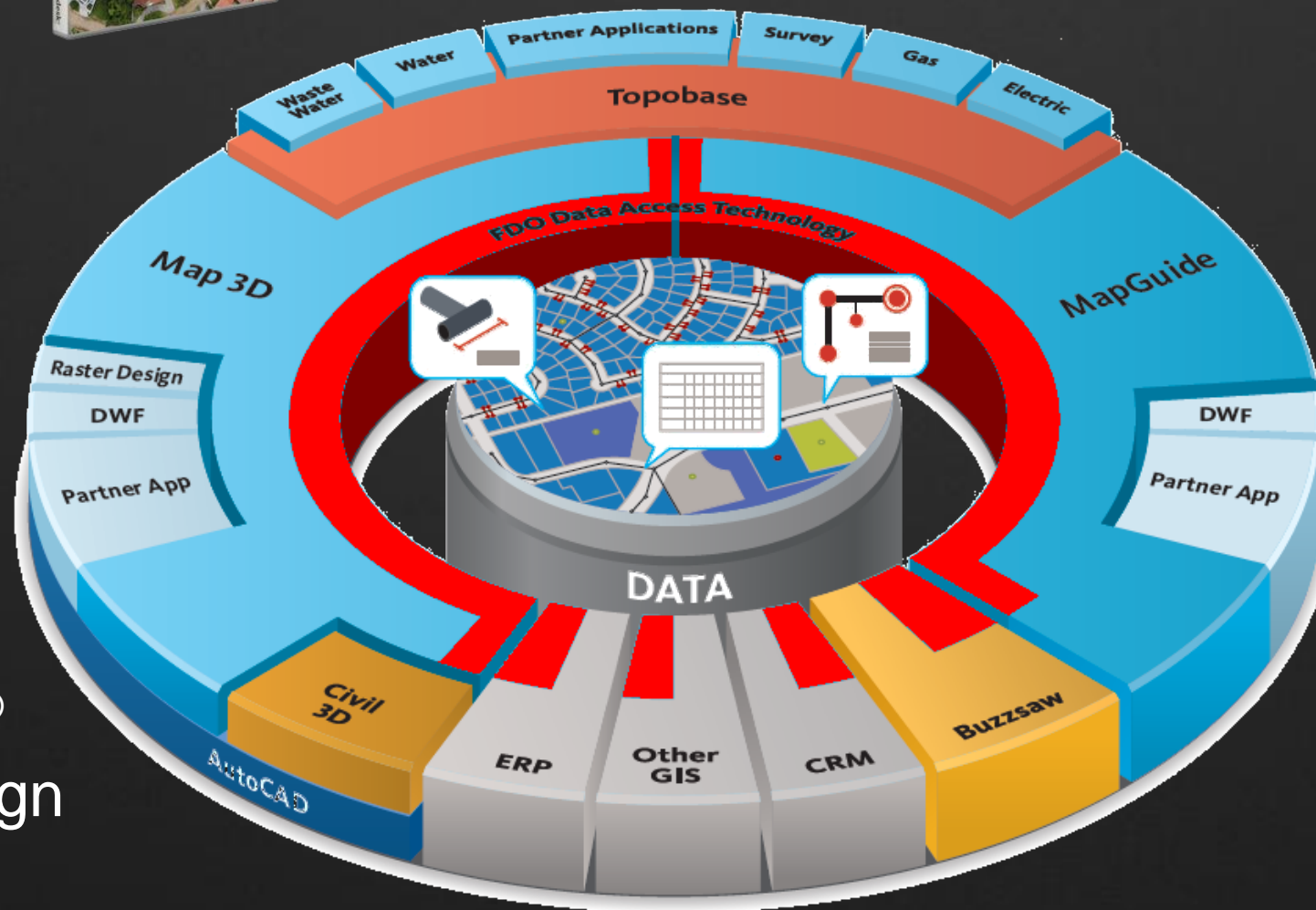


Autodesk® Utility Design

Autodesk® LandXplorer Studio



Autodesk MapGuide® Enterprise and Autodesk MapGuide® Studio



Why Deliver Information on the Web?

- Improve business performance, reduce costs, increase customer satisfaction, meet regulatory requirements.
- Manage GIS and CAD as part of an IT portfolio.
- Get information to where it is needed:
 - Reach the **maximum number of users**.
 - Get the **right** data to the **right** people at the **right** time.
 - Enable **participation** across the organization.
 - Get data off the desktop and out of departmental **data silos**.
 - **Integrate** with other business systems.

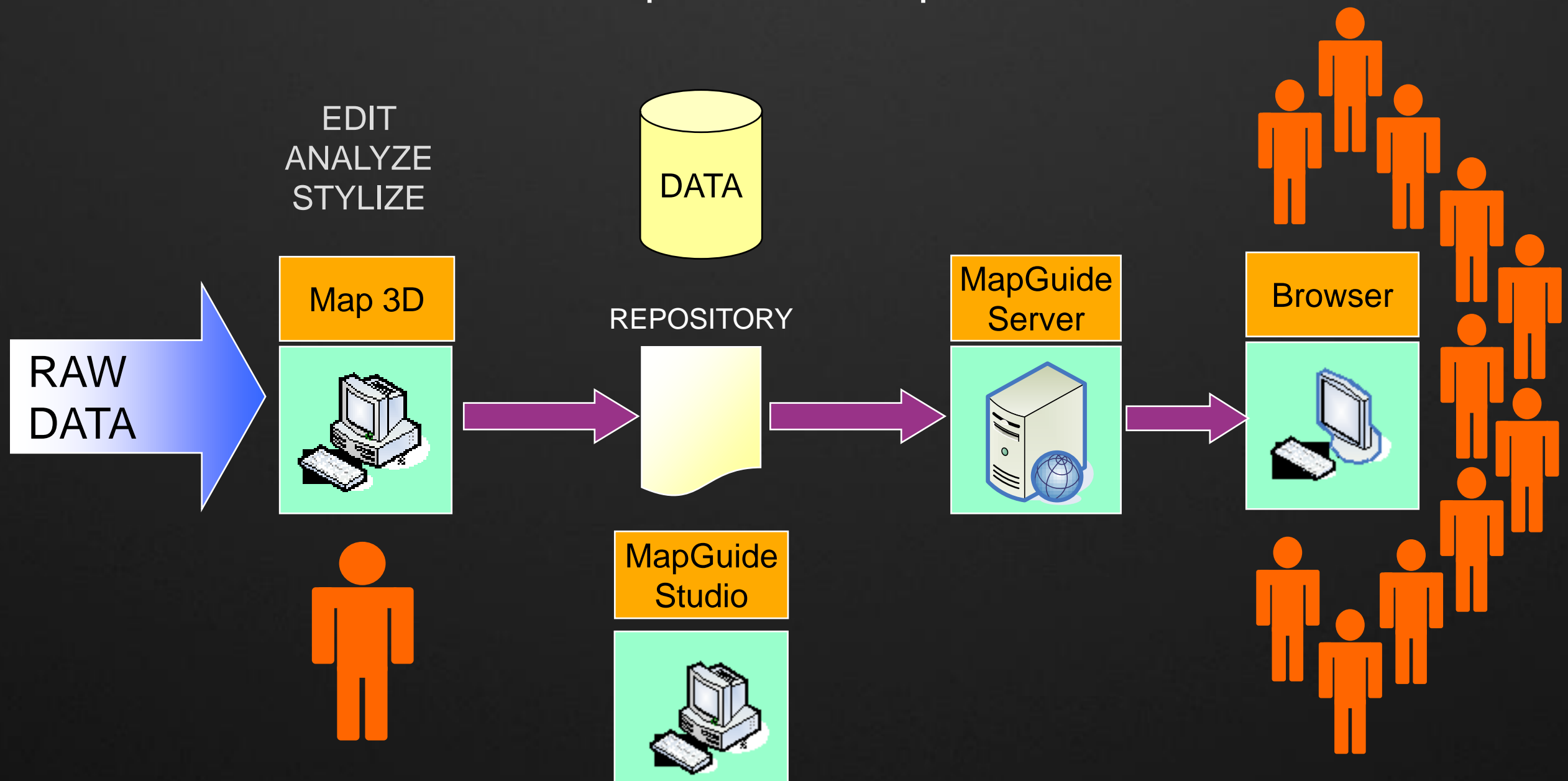
Move Information from Desktop to Web

Get the right data, to the right people, at the right time

Create on the desktop

Publish to Autodesk
MapGuide Enterprise

Deliver to end users

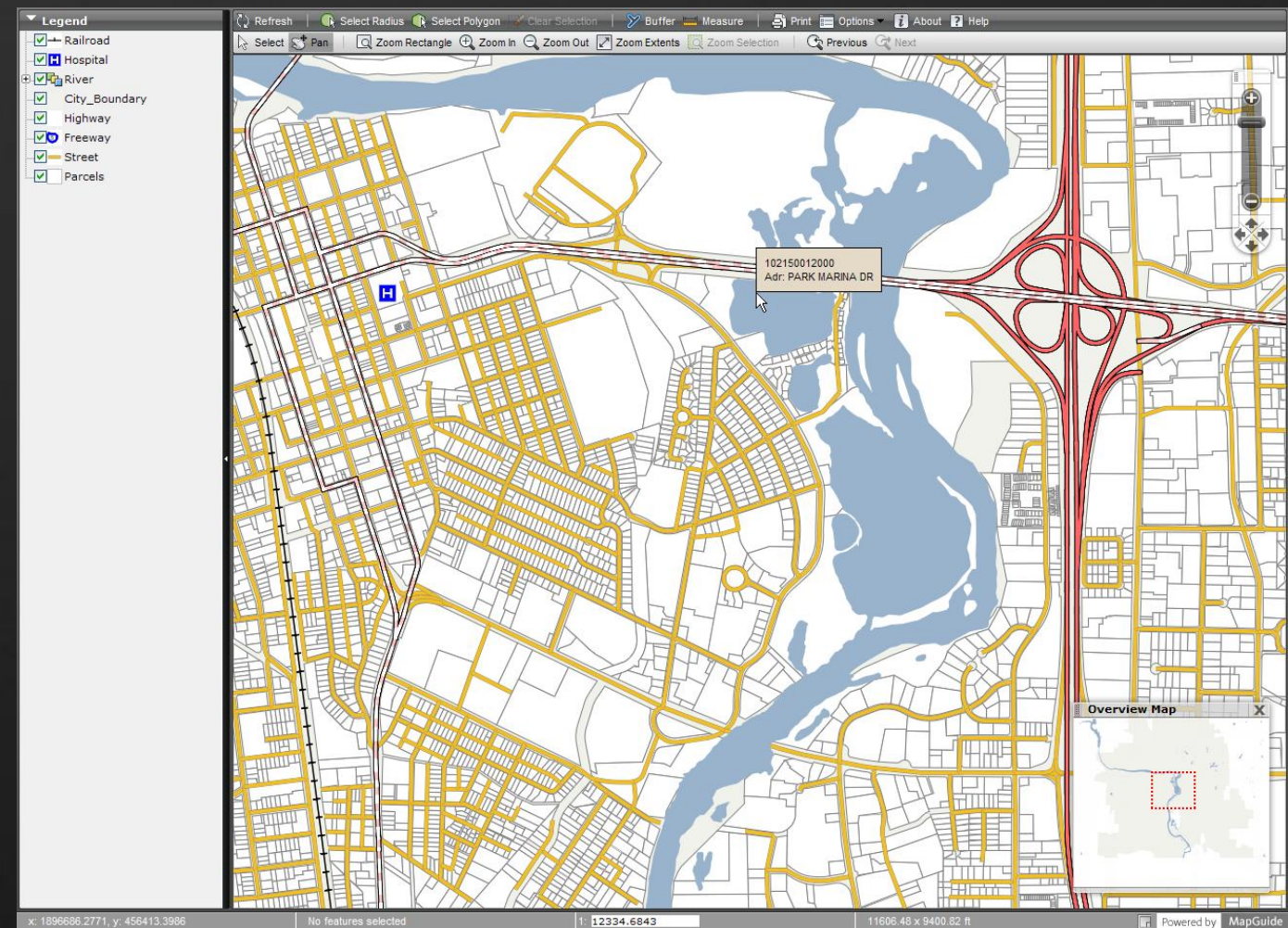


Agenda

- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- MapGuide API Overview
- Integration of Fusion & Google Maps
- Data Migration Tool

MapGuide Enterprise

- Web-based mapping platform that publishes maps and spatial applications on-line
- Server-side API to execute GIS business logic
- Distributed in two ways
 - Open source, LGPL
 - Commercial version



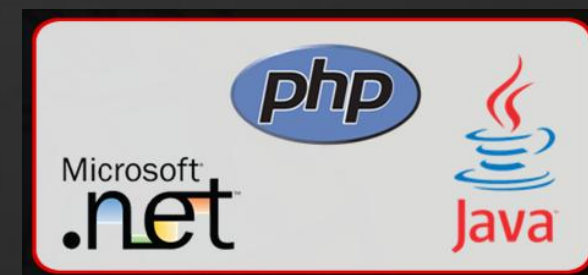
Technical Features

- Native Linux and Windows support
- Three APIs, .NET, Java, and PHP
- Multiple data source access via FDO
- AJAX Viewer / Fusion Viewer / (DWF Viewer)
- Server side business logic execution
- Web-based server administration tool
- Stream-lined authoring tool - MapGuide Studio

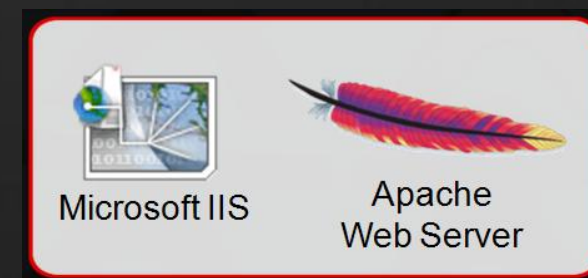
Deploy on



Develop with...



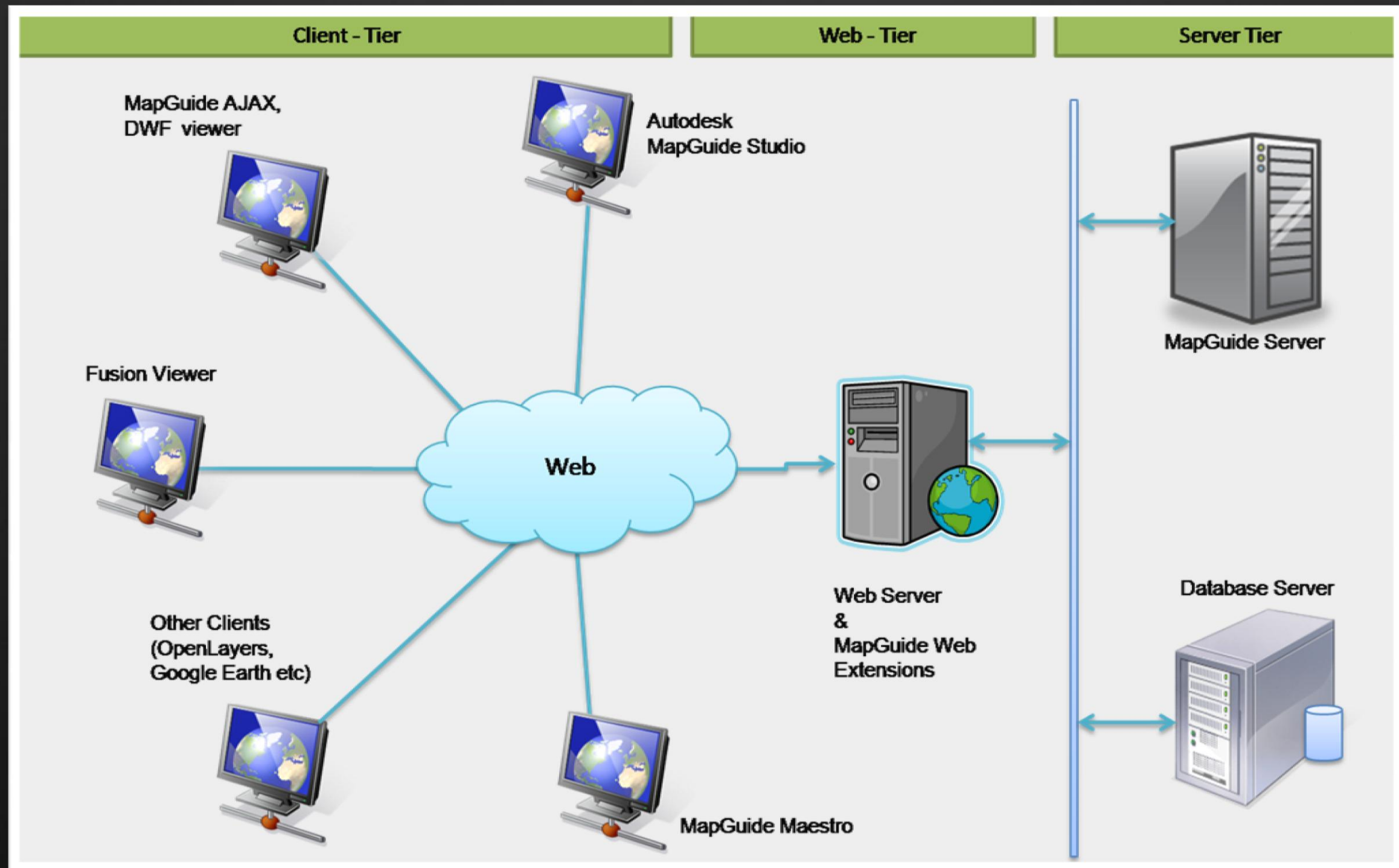
Serve with...



Browse with...

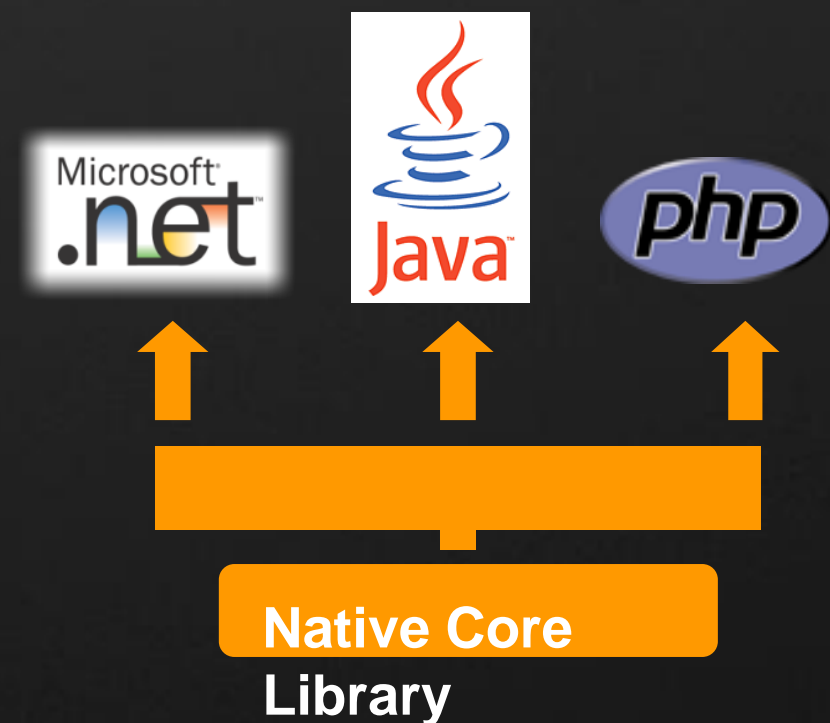
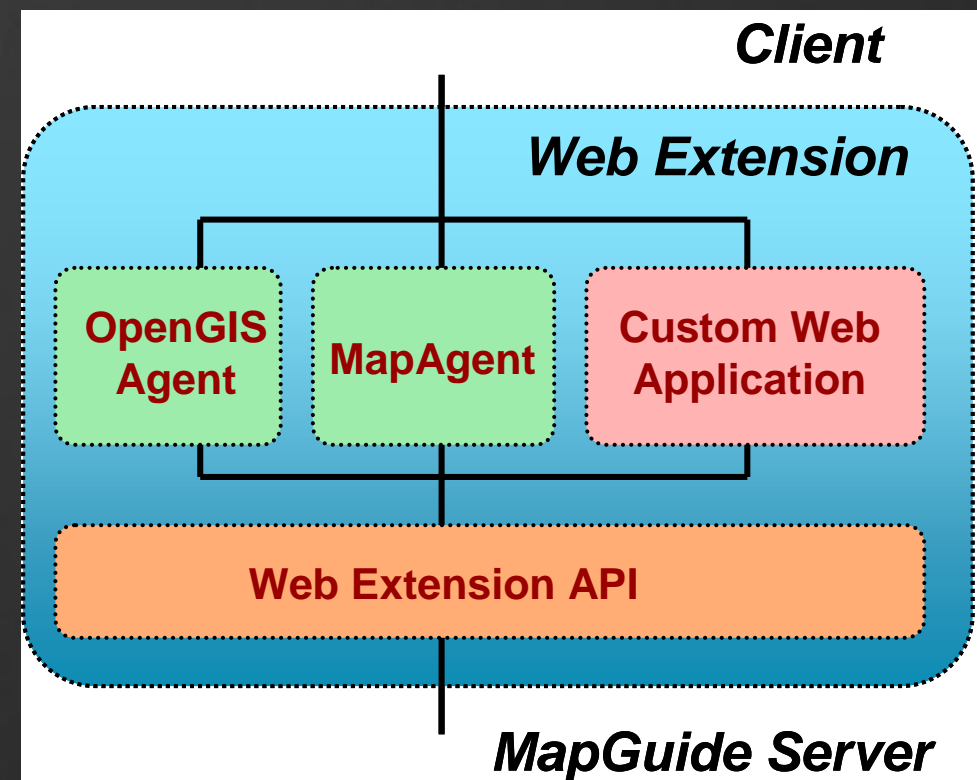


System Architecture



Web Tier

- Web extension is a customizable interface that exposes the services from the MapGuide Server to the web.
- MapAgent is the gateway used by viewers to access the GIS services.
- OpenGIS Agent exposes services in WMS and WFS formats.
- Custom web application incorporates the GIS business logic.
- Web tier runs on top of supported web server or application server such as Apache, IIS, Tomcat, and so on.
- 3 APIs, .NET, Java, and PHP



MapGuide Viewer

- out-of-box functionality.
 - Pan/Zoom/Distance/buffer/Attribution information display...
- No programming is needed by default.
- Multiple options available:

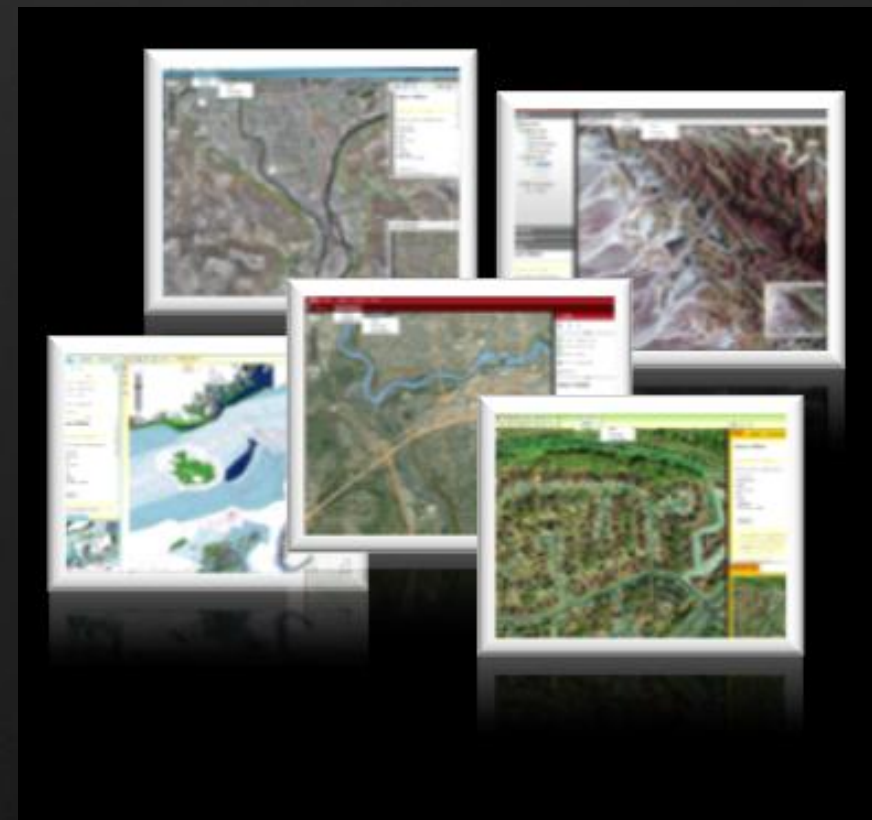
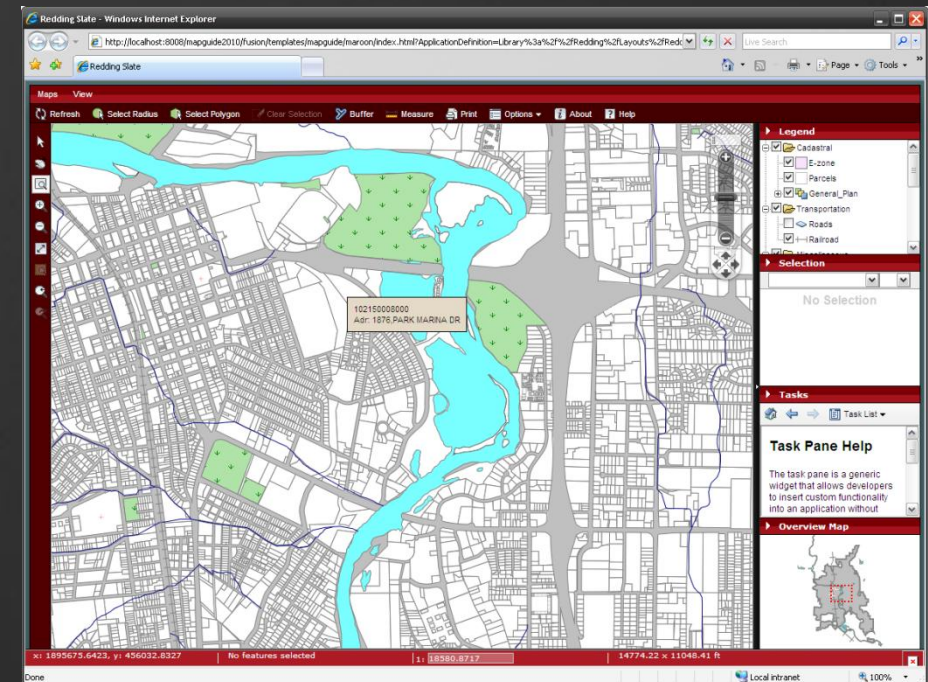
AJAX Viewer

- Based on HTML, JavaScript, and XML
- Not ActiveX control
- No download required

Fusion Viewer

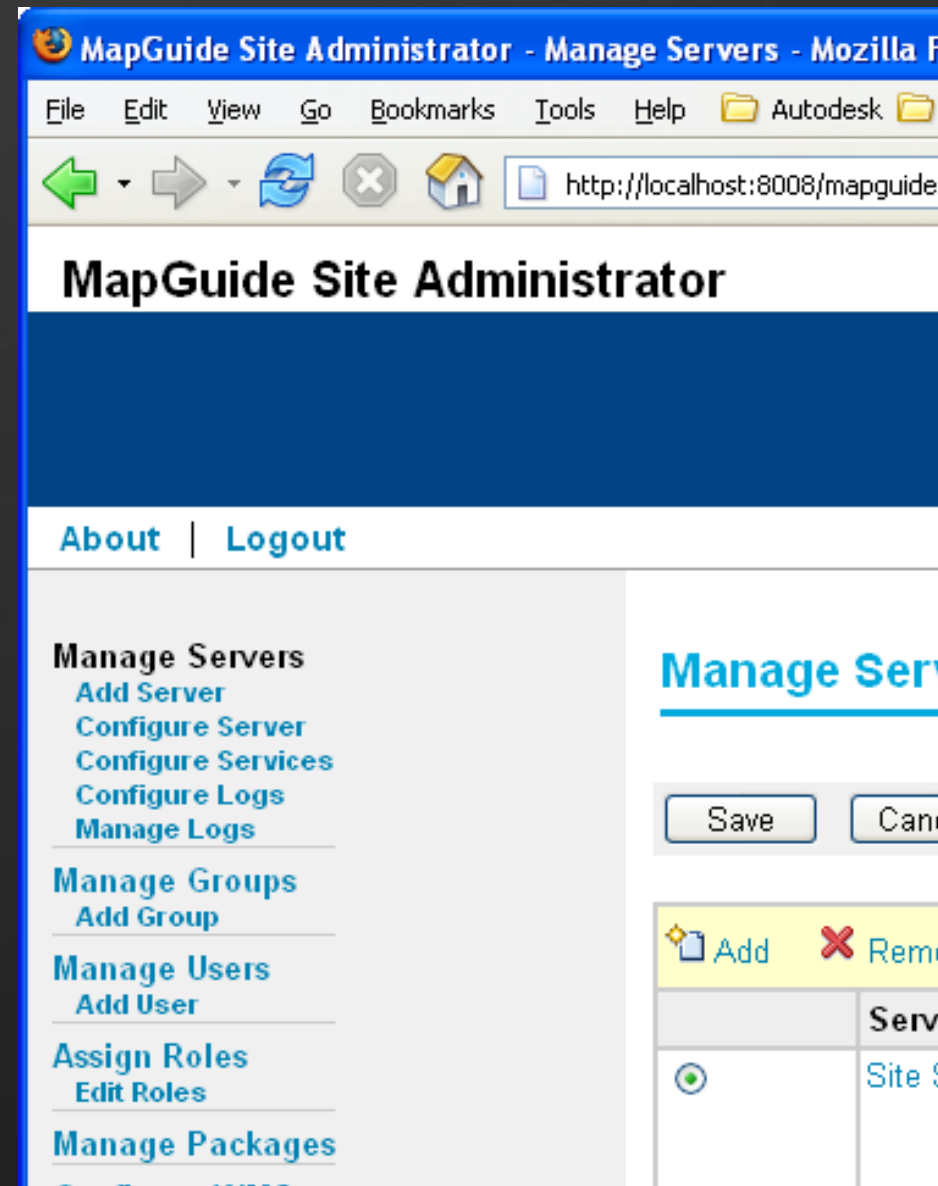
- Based on AJAX technology
- Powerful functionalities from widgets
- Multiple templates
- Out of box experience
- Customizable

DWF Viewer (Deprecated)



MapGuide Site Administrator

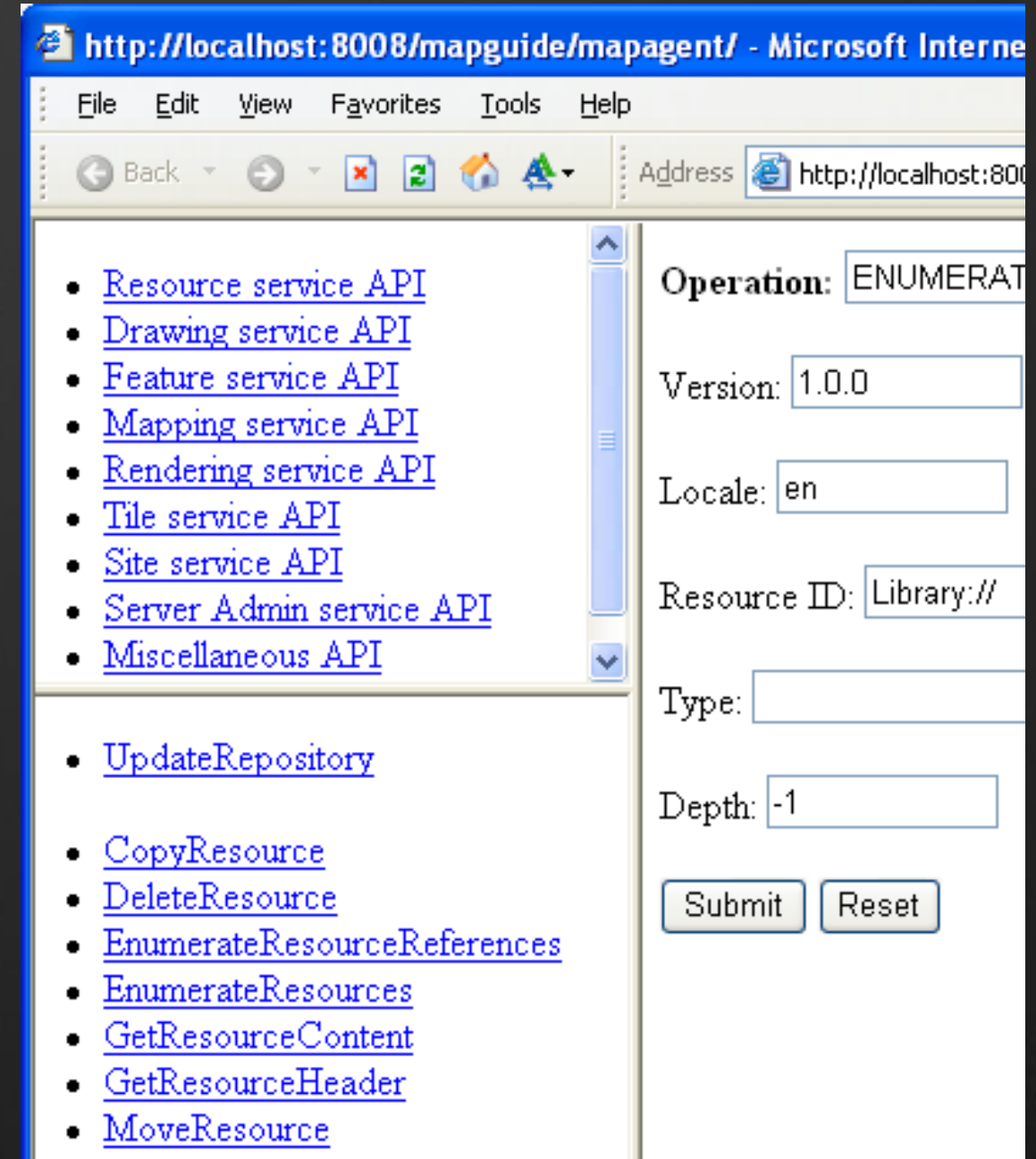
- Web-based administration tool
- Manage the site and servers
- Administrate services.
- Manage users
- Monitor running status
- View server logs



<http://localhost/MapGuide2010/MapAdmin/login.php>

MapAgent Test Page

- MapAgent test page enables you to run quick tests on the server APIs.
- It's organized by service types.
- No programming required.
- It uses mapagent.fcgi which is also used by map viewers.



<http://localhost/mapguide2010/mapagent/index.html>

Package Management

Load Package

- Data package can be loaded to the server
- Copy the *.mgp file to the package directory and click on the Load Package link.

Make Package

- Data repository on the server can be packaged and loaded on another site
- Specify the folder name, i.e., //Exercise/Layouts
- All the data under this path will be packaged **C:\Program Files\Autodesk\MapGuideEnterprise2010\Server\Packages**

Make Package

You can package a section of this site repository and store [Configure Packages](#) directory.


Folder name (e.g. //<root_folder>/<folder1>):

Resulting package name:

Load Package

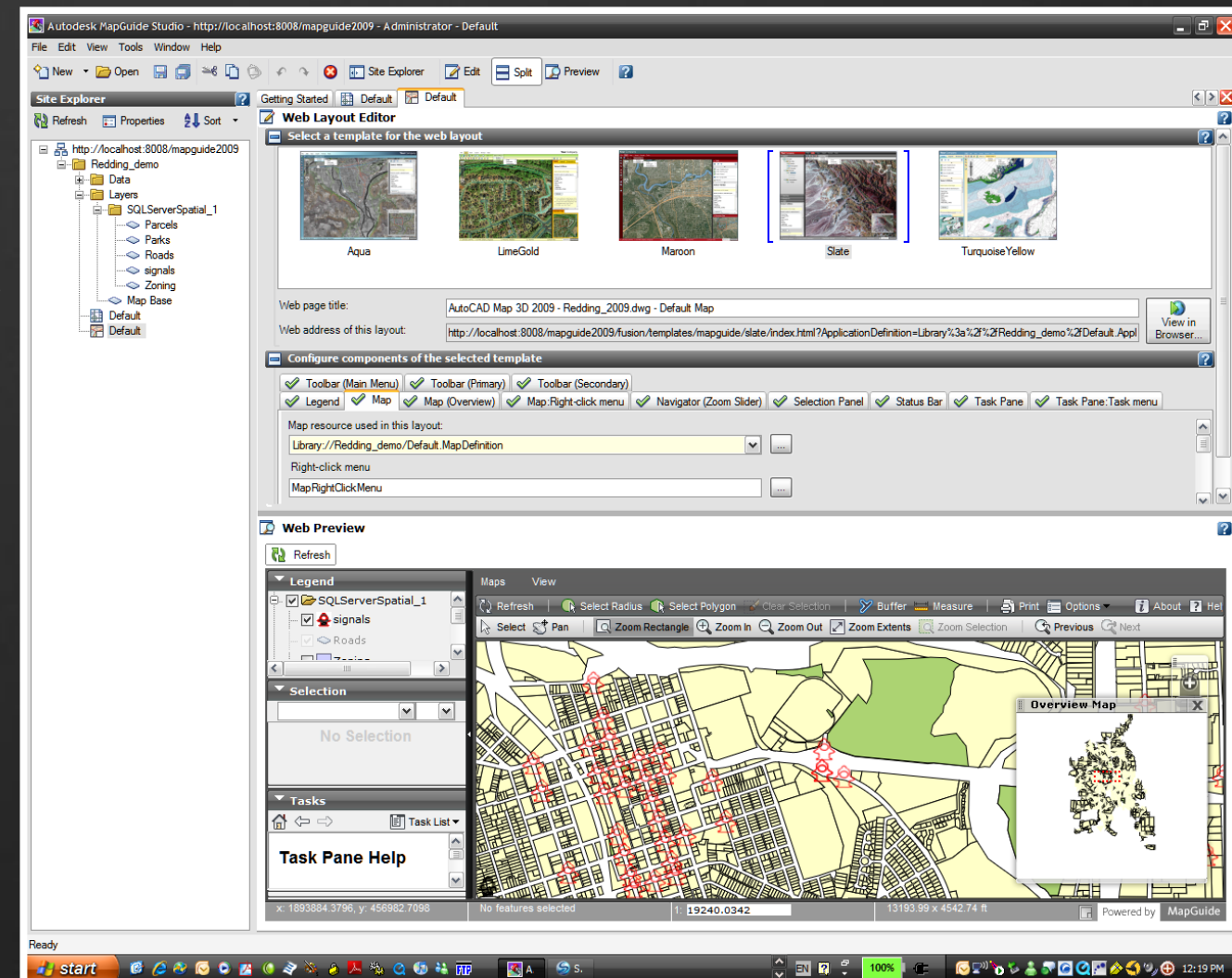
Packaged resources or data in the package folder can be loaded [Configure Packages](#) directory.

 [Load Package](#)  [View Log](#)  [Delete](#)

	Package Name
	Sheboygan.mgp

Autodesk MapGuide Studio 2010

- Map authoring environment for Autodesk MapGuide Enterprise.
- Load Map data resources
- Build feature style and layer theme
- Compile layers into map display
- Manage web layout in browser
- Significantly ease the job of map creation and publication.



Agenda

- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- MapGuide API Introduction
- Integration of Fusion & Google Maps
- Data Migration Tool

What's New in the 2010 Version?

Focus on Performance

Performance

Performance

Performance

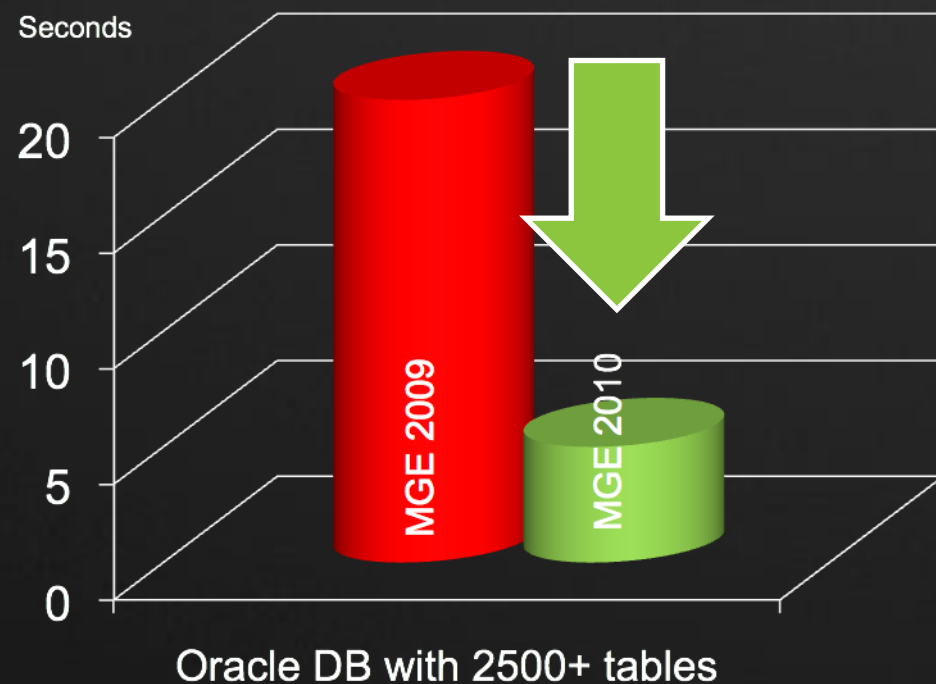
- We made good progress in 2009 Update 1...

Improved Performance

Faster Oracle® Connectivity

- Optimize the describe schema at connect

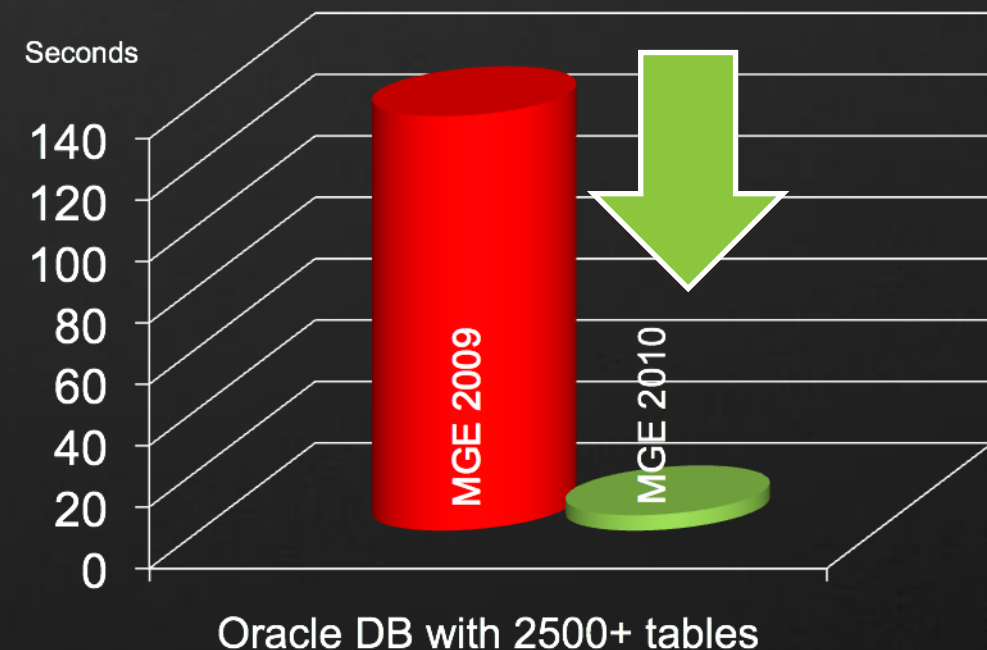
Draw a map that has a layer using one of the tables from the Oracle DB without a cached* connection



*When the connection is cached, MGE 2010 and MGE 2009 consistently display the map in 2 seconds.

- Faster loading of layers and maps

Create a new layer in Autodesk MapGuide® Studio



Please note: All performance numbers are indicative based on preliminary testing during development.

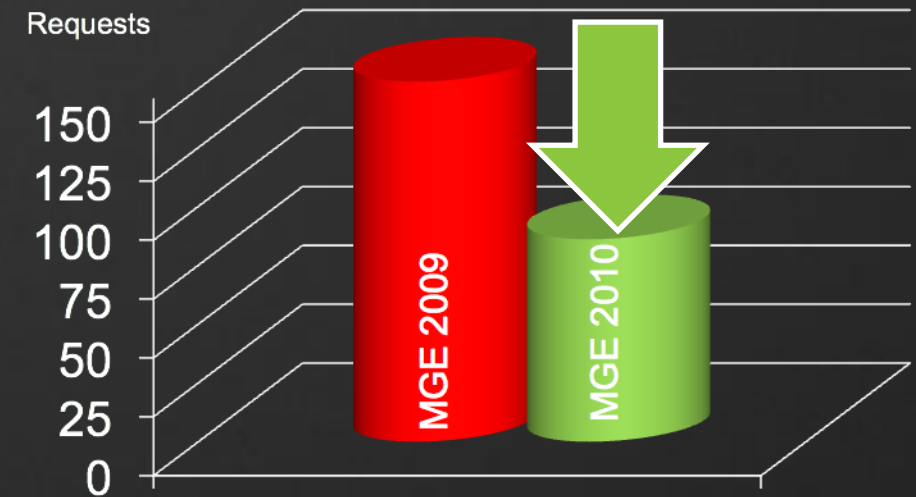
Improved Performance

Faster Loading of Flexible Web Layouts

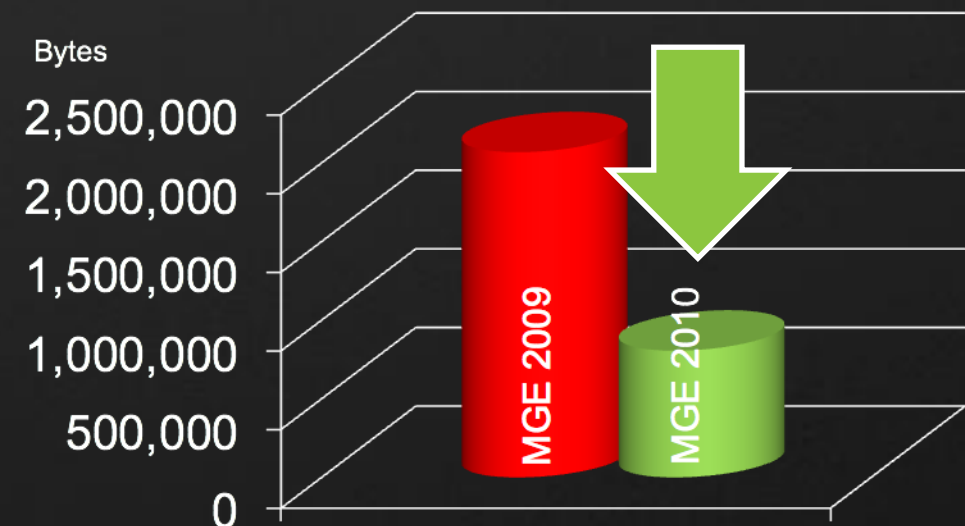
- Reduce the number of HTTP requests to the server
- Load maps faster
- Utilize JSON output from the server
- Use new libraries (MooTools replaces script.aculo.us and Prototype) that reduce script size
note: modified templates must be updated
- Faster Selects/Deselects

Please note: All performance numbers are indicative based on preliminary testing during development.

HTTP Request count
(the number of requests from the browser to the client before the map draws)



Bytes transferred
(this number does not include the map image)



AMGE 2010:

- 6 seconds at 128 KBs or 1 Mbps
- 1 min 54 sec over 56.6 Kbps modem

AMGE 2009:

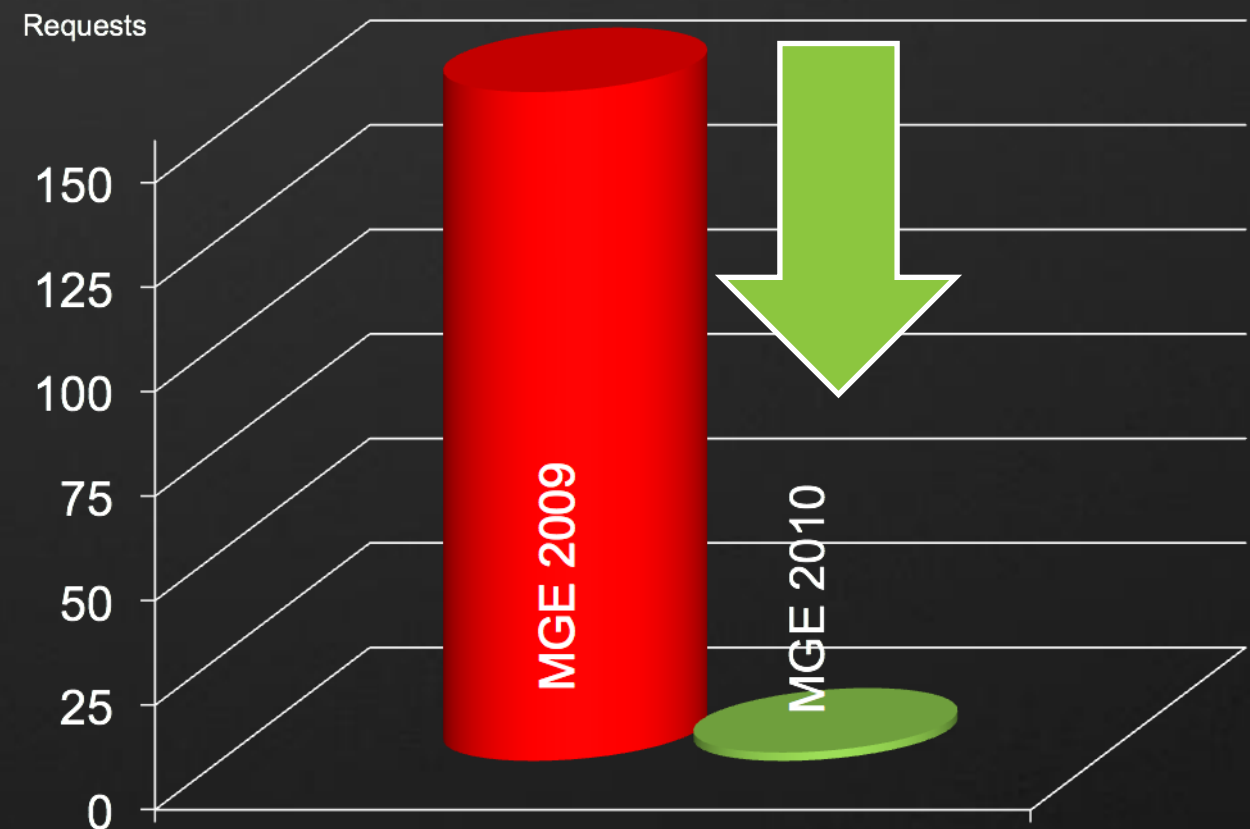
- 16 seconds at 128 KBs or 1 Mbps
- 4 min 55 sec over 56.6 Kbps modem

Improved Performance

Save to Large Resource Repository

- Improved dependency checking in resource repositories with thousands of resources

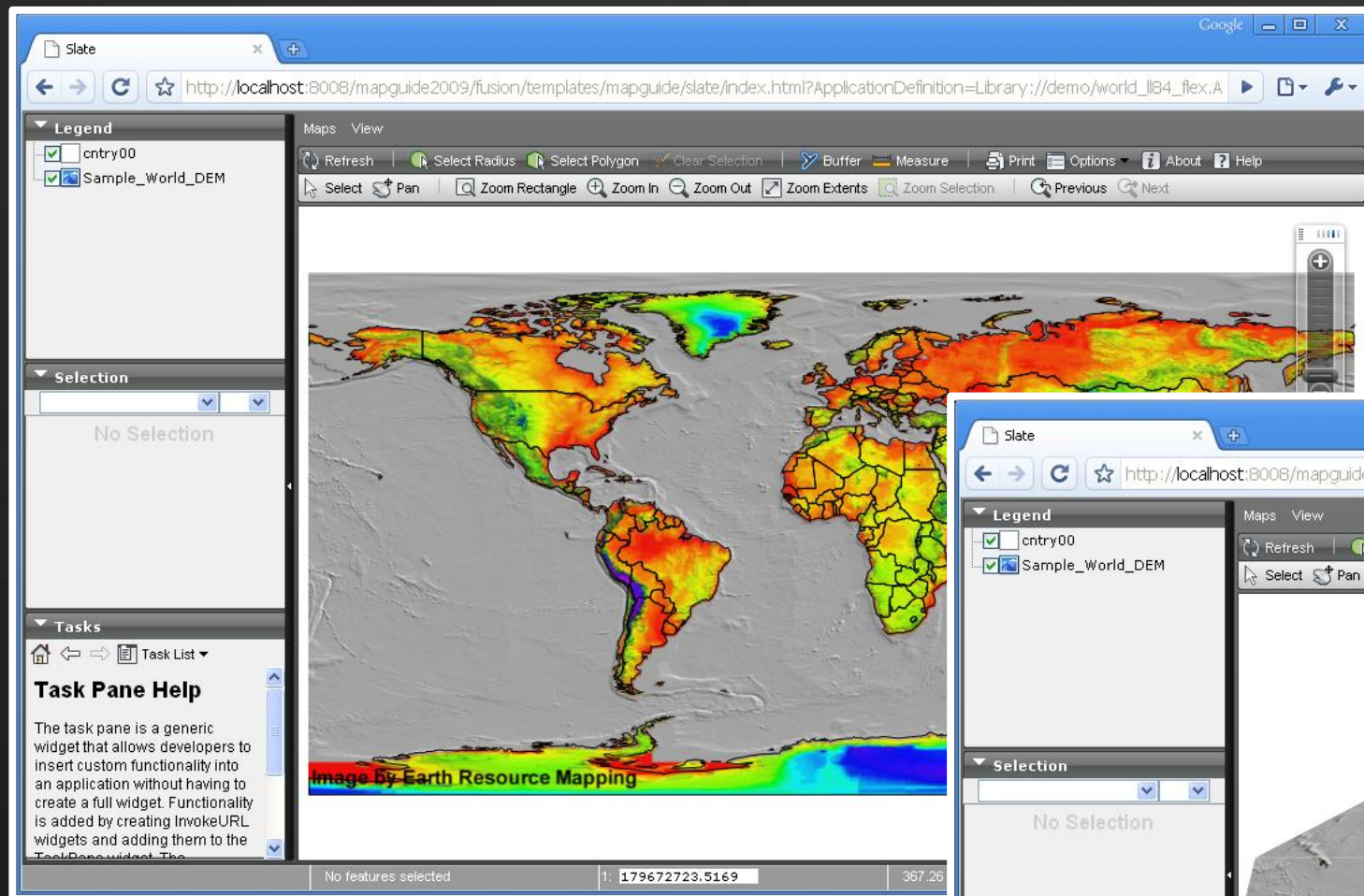
Save to resource repository with approximately 7000 resources



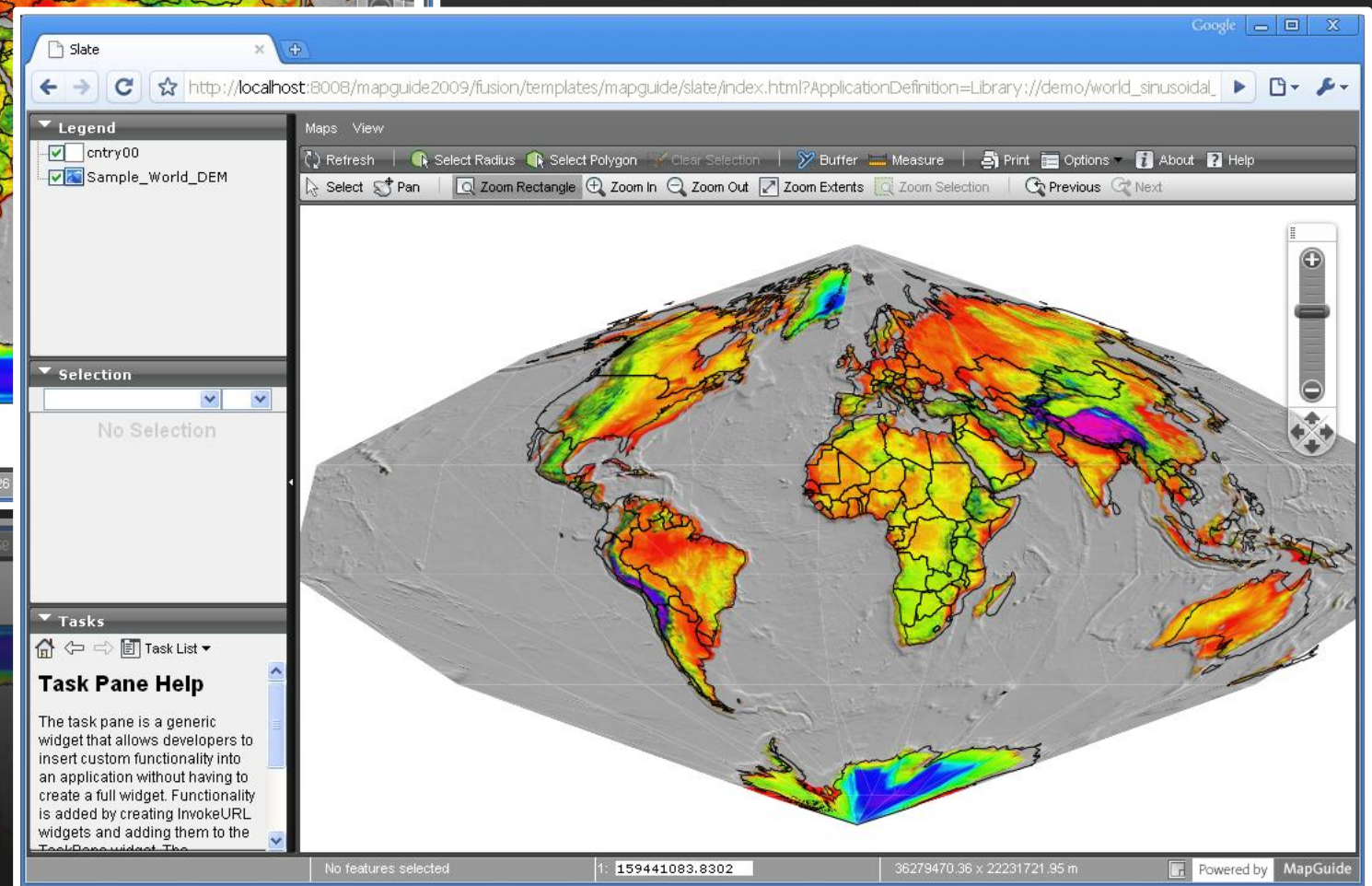
Please note: All performance numbers are indicative based on preliminary testing during development.

Vector and Raster Reprojection

LL84



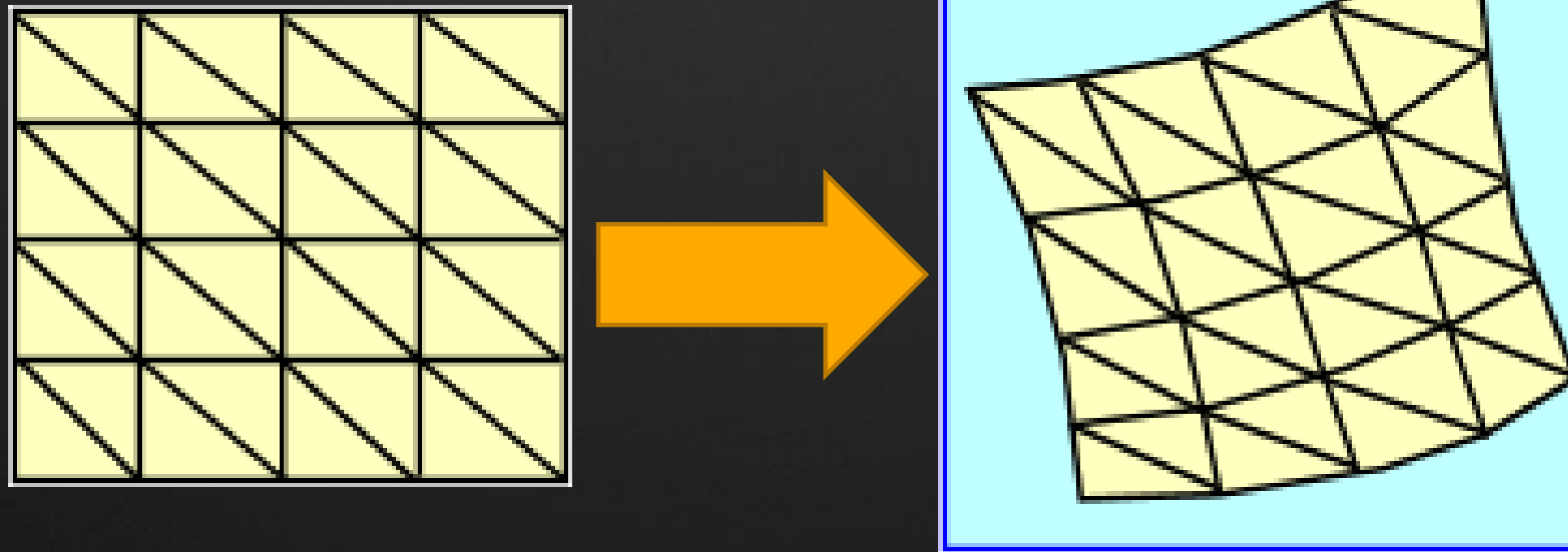
WORLD-SINUSOIDAL



New Raster Reprojection Algorithm

Common to AutoCAD® Map 3D 2010 and Autodesk MapGuide® Enterprise 2010

- Transform calculation code is in Geospatial Platform

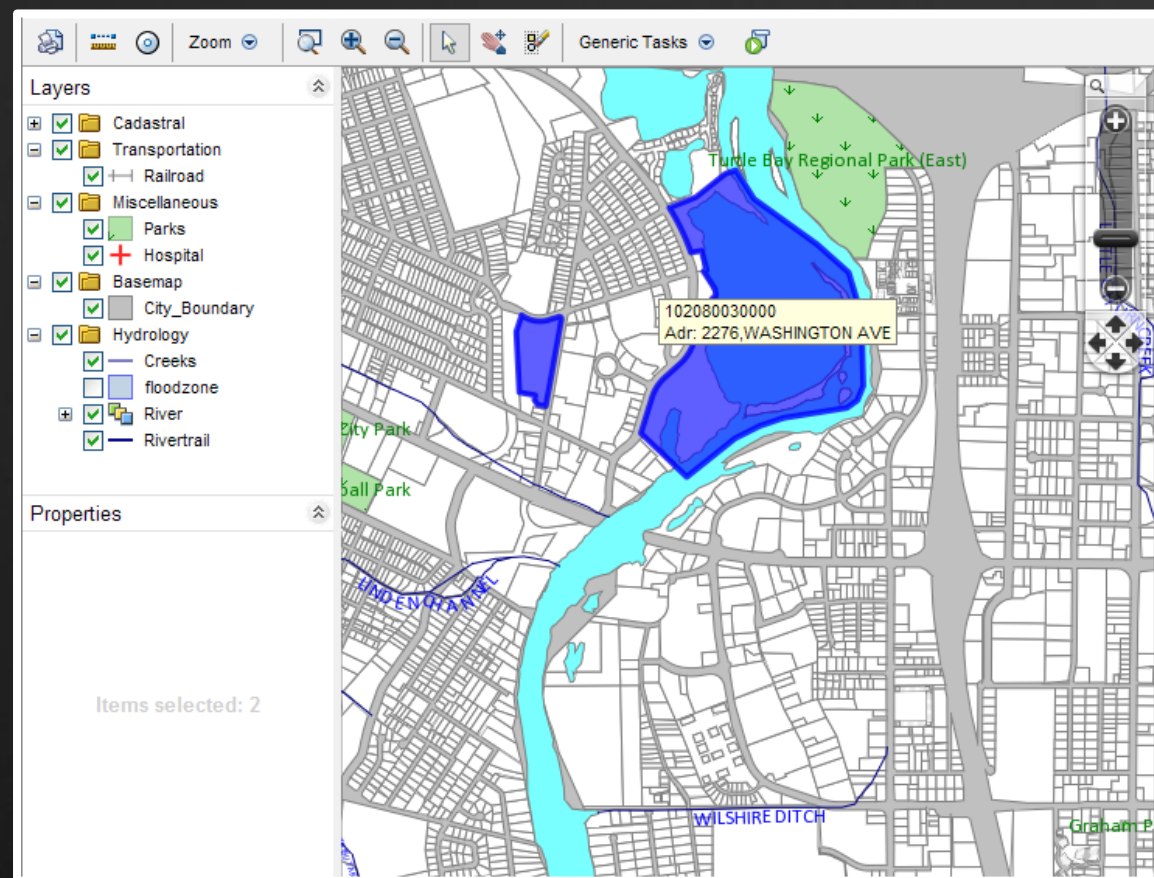


Note: Triangle size can be changed through a server setting. Initial size is optimized for speed.

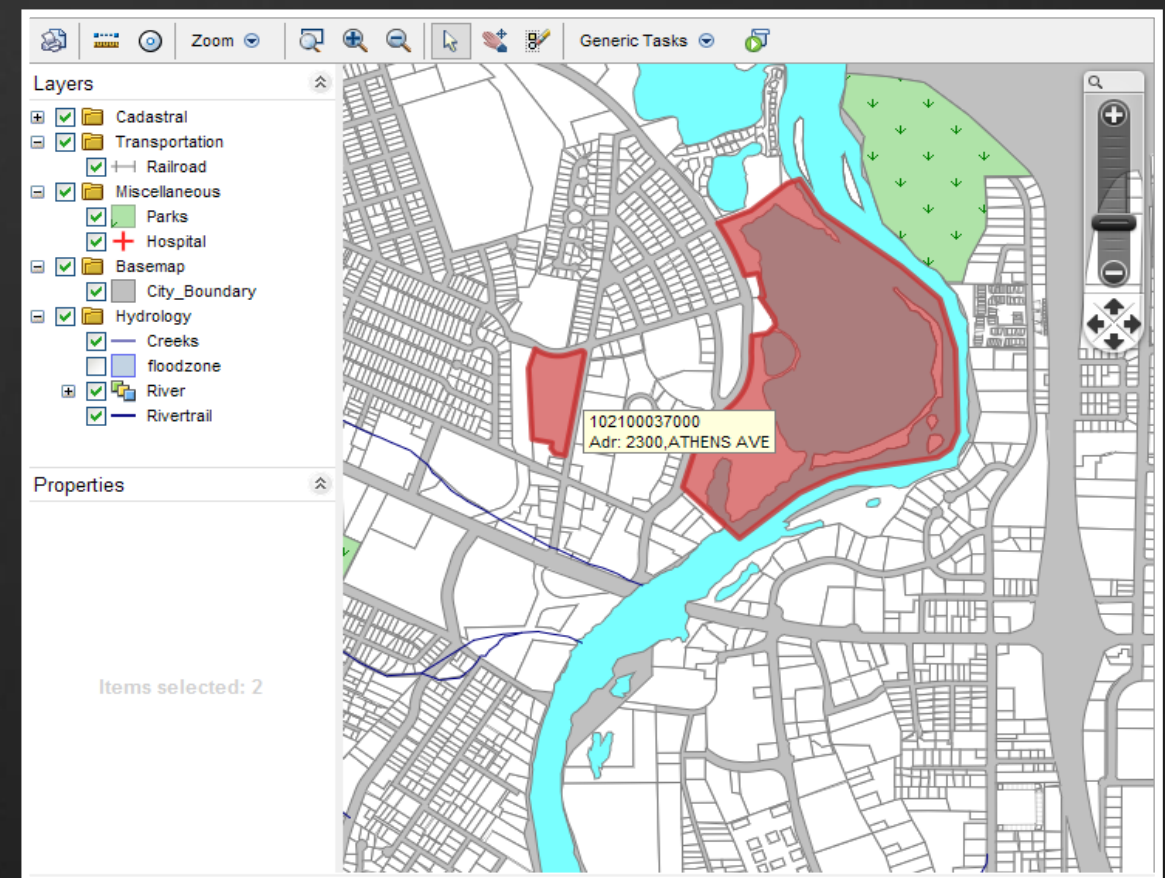
Change Default Selection Color

Default selection color can be changed in ajaxmappane.templ

MGE 2009

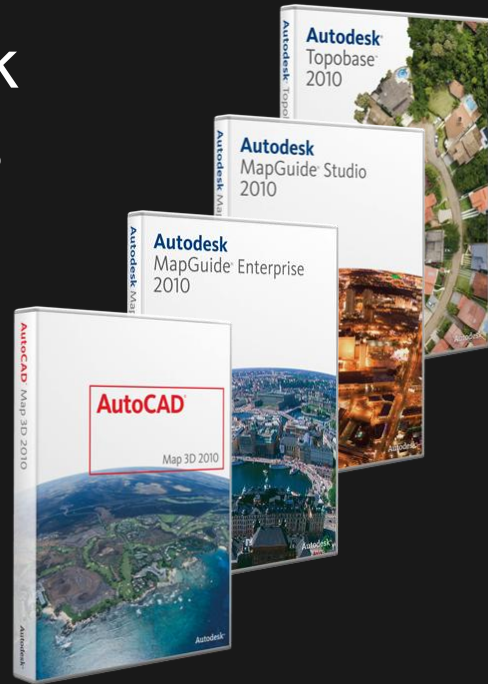


MGE 2010

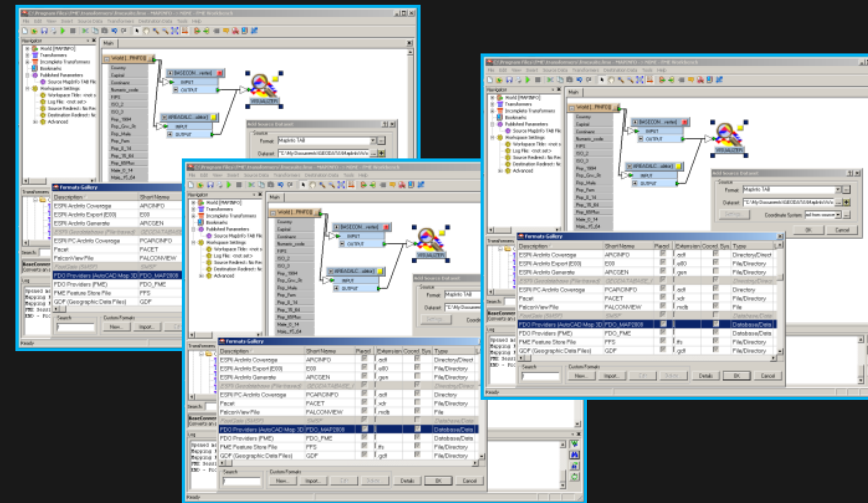


FDO Data Access Technology—New Providers

Autodesk Products



Third-Party Solutions



Autodesk Certified Providers

Providers from Third-Party & Open Source

Providers supporting **multiple** formats

ODBC

Oracle® and Microsoft® Office Access®, Excel®

Raster

15+ raster formats

ArcSDE®

Oracle and Microsoft SQL Server®



150+ vector and raster formats

OGR

25+ vector formats



25+ raster formats

NEW

Providers supporting **single** format

SDF

SHP

WMS

WFS

Microsoft SQL Server 2005

ORACLE®

MySQL® Sun

Microsoft SQL Server 2008

GE Energy Smallworld

PostGIS

SuperMap

ORACLE®

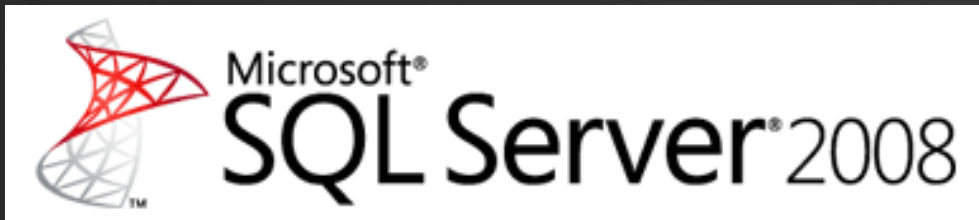
IBM formix Dynamic

Microsoft SQL Server 2005

KML

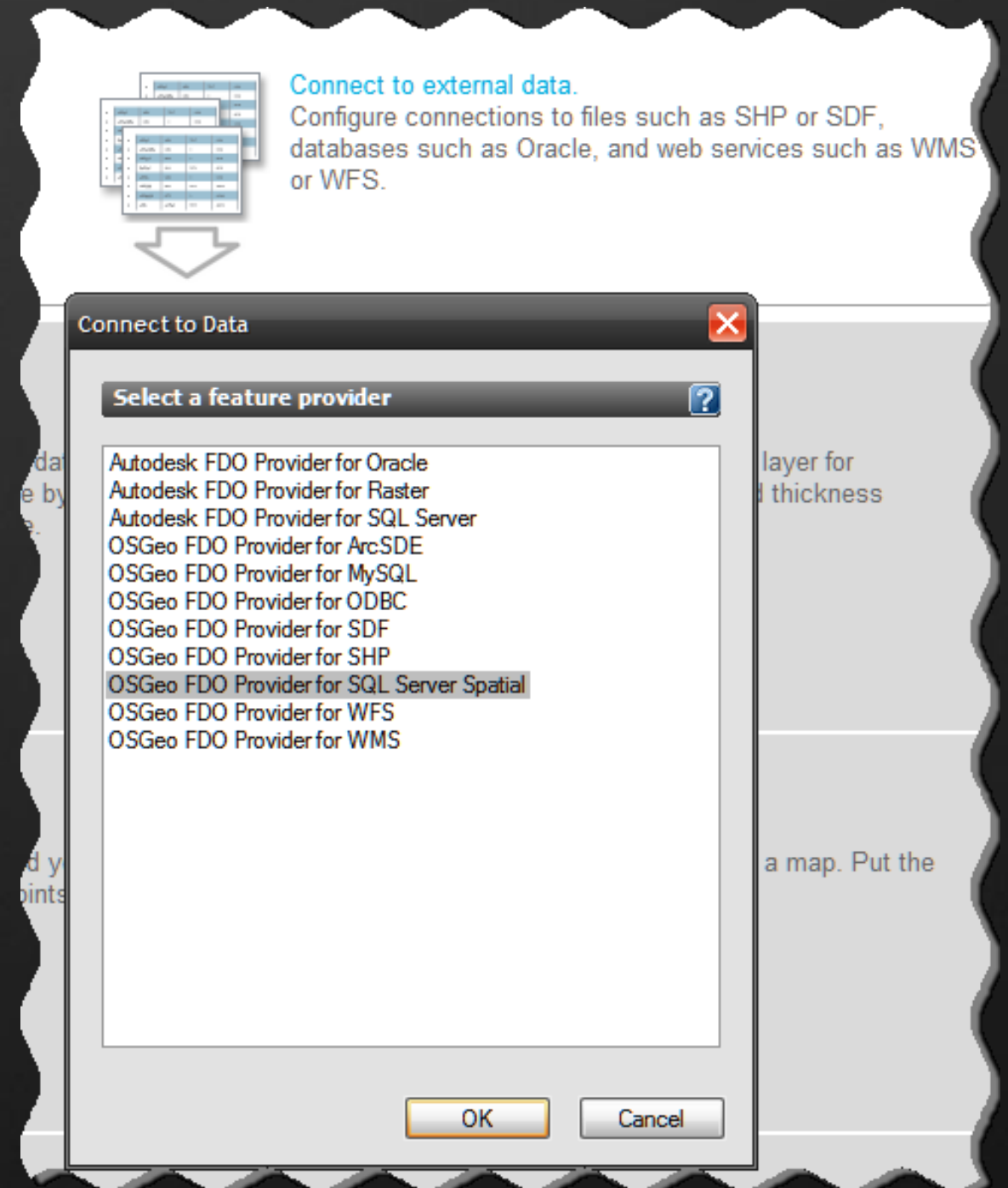
Autodesk

New Provider for Microsoft SQL Server 2008



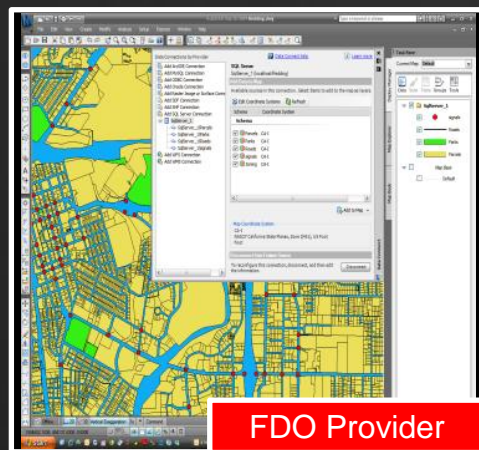
- Support for native SQL Server 2008 geometry and spatial indexing
- Coexist with Autodesk® FDO provider for SQL Server 2005
- For more info...

www.microsoft.com/isv/autodesk/



New FDO Provider for Microsoft SQL Server 2008

AutoCAD® Map 3D



- Retrieve data
- Create/Edit
- Analyze
- Publish to Autodesk MapGuide Enterprise

FDO Provider

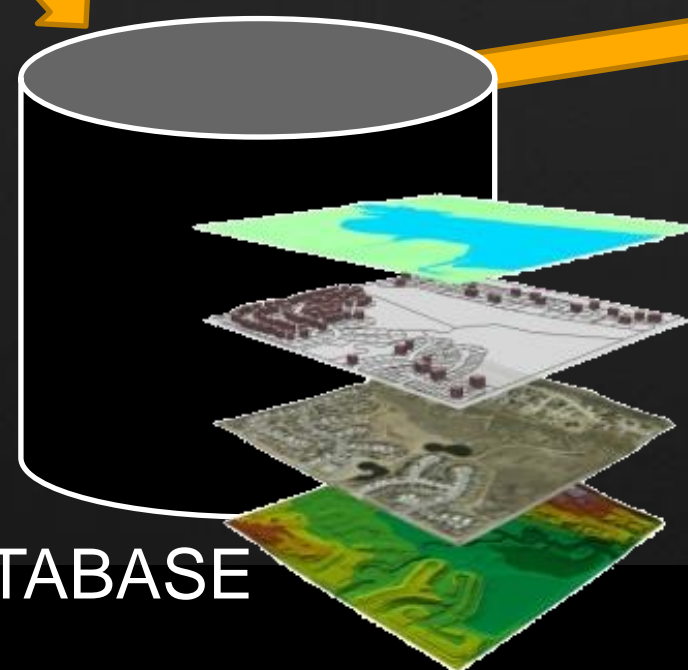
DESKTOP

Publishing

Transactions

Microsoft SQL Server® 2008 Spatial

DATABASE



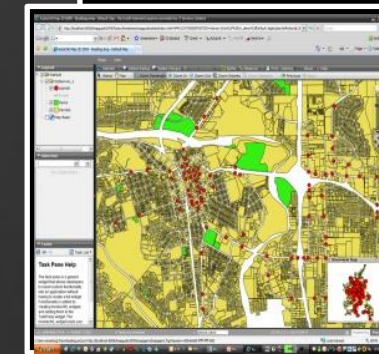
Autodesk MapGuide® Enterprise

FDO Provider

APPLICATION SERVER

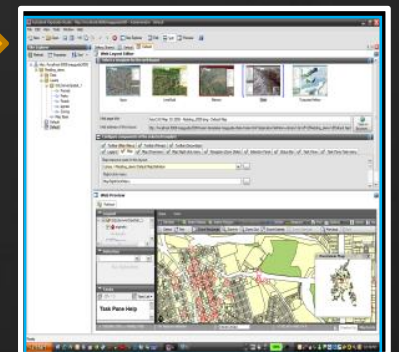
Browser

- Browse
- Search
- Analyze



WEB

Autodesk MapGuide® Studio



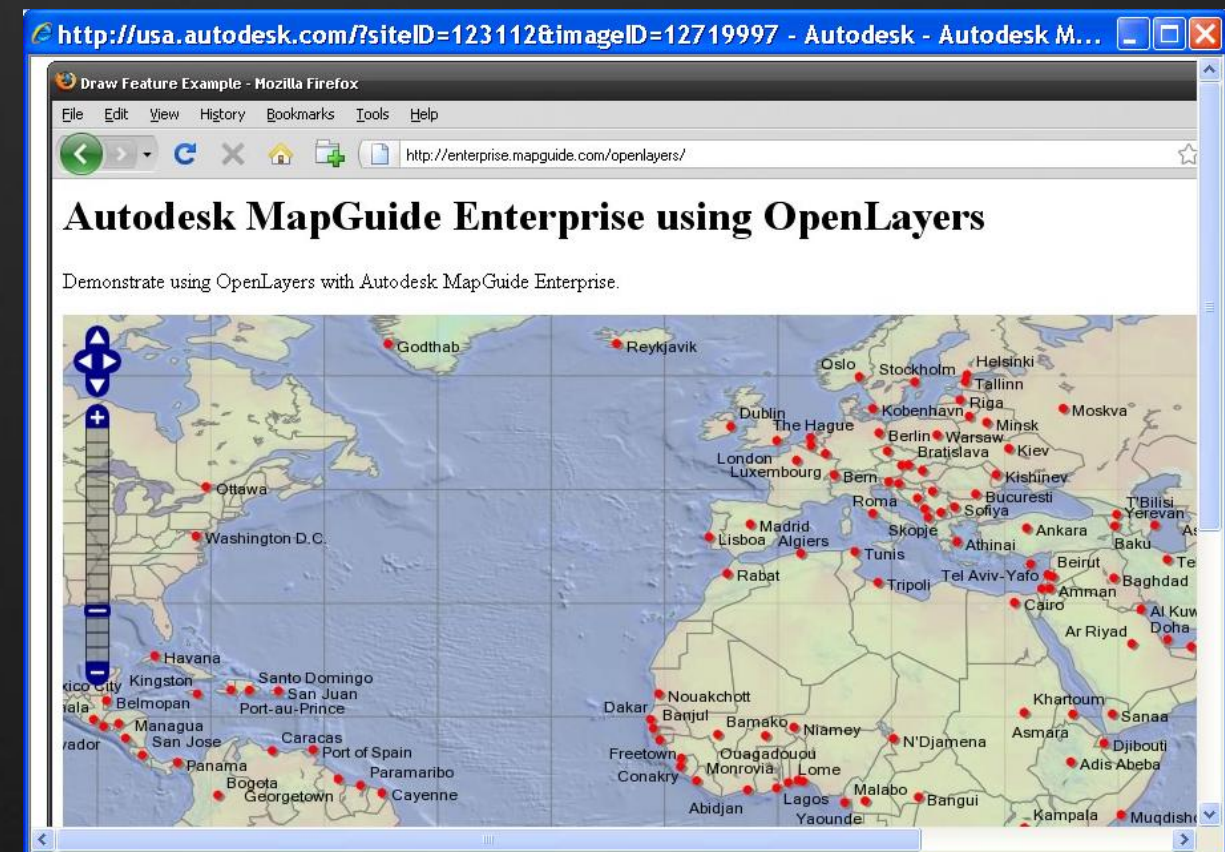
- Add layout
- Tools
- Additional data
- Queries/Filters

Other Improvements

- Faster raster performance
- Improved legend display
 - Remove unwanted geometry types in the legend
- Coordinate system
 - Resolve unknown coordinate systems faster
- Better server error reporting
- **API enhancement / changes**
DescribeSchema (MgResourceIdentifier id, String SchemaName,
MgStringCollection ClassNames)

OpenLayers Support

- **OpenLayers Support** - The OpenLayers JavaScript library can use MapGuide Open Source and Autodesk MapGuide Enterprise as a map data source, offering users a client-side scripting toolkit to create richer client-side interactions, including:
 - Client-side redlining tools
 - The ability to use multiple map data sources
 - Markers and pop-up windows



Autodesk MapGuide Enterprise 2010

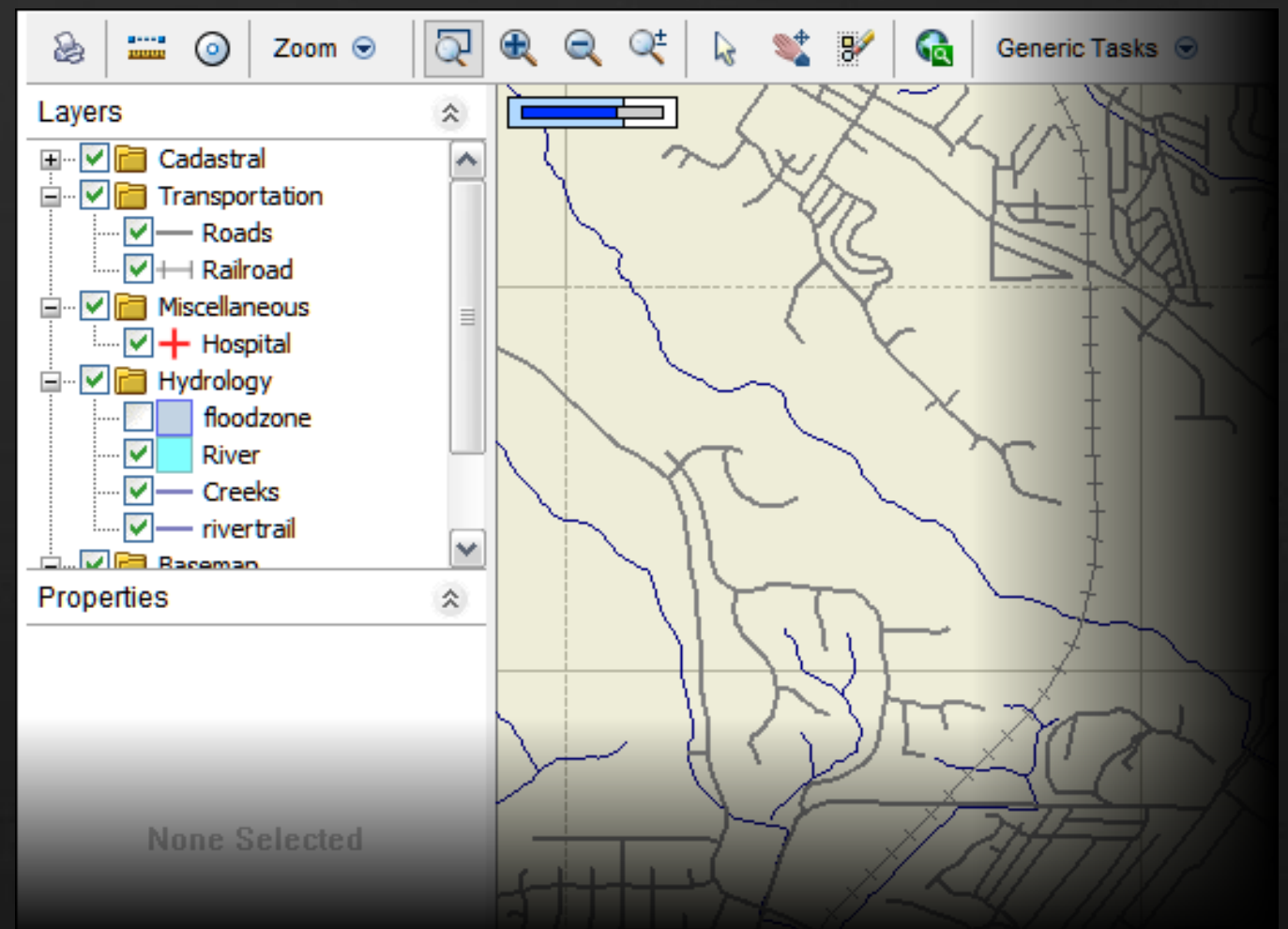
Continuous innovation

Feature	Release			
	2007	2008	2009	2010
Built using MapGuide Open Source	●	●	●	●
Multiple viewer options (DWF™ and AJAX)	●	●	●	●
Autodesk MapGuide Studio	●	●	●	●
Multiple operating system support (Windows and Linux)	●	●	●	●
Multiple development environments (PHP, .NET, and Java)	●	●	●	●
Raster support (including ECW and MrSID®)	●	●	●	●
OGC Web Services (WMS/WFS)	●	●	●	●
Buffer zones	●	●	●	●
Multipurpose authoring environment with Autodesk MapGuide Studio	●	●	●	●
Direct publish from AutoCAD Map 3D	●	●	●	●
FDO Data Access Technology	●	●	●	●
Direct Google Earth support		●	●	●
User-defined line styles and symbols		●	●	●
Built-in pragmatic load balancing		●	●	●
Unmanaged data source support		●	●	●
Flexible web layouts and templates			●	●
Integrated open layers technology on web client			●	●
Uniform expression language and expression editor (shared with AutoCAD Map3D 2010)			●	●
New and improved AGG renderer support			●	●
Native integration with Apache and IIS			●	●
Improved Performance: Oracle, Raster and Flexible Web Layout				●
Raster reprojection capability				●
Improved Legend				●
New FDO providers for Microsoft SQL Server 2008 , GE Smallworld				●
OpenLayers integration with Autodesk MapGuide Enterprise				●

Deprecation of DWF Viewer

- The DWF™ Viewer functionality is available in Autodesk MapGuide® Enterprise 2010, but it is no longer supported and defects will not be fixed.
- In order for DWF Viewer to work in Autodesk MapGuide Enterprise 2010, Autodesk® Design Review 2009 must be installed on client system.
- **Please note: It is still possible to create and plot DWF files to be used with Autodesk Design Review !!**

- **DO NOT** upgrade to Design Review 2010 if your MGE application makes use of the DWF Viewer. Design Review 2010 does not include the component used by Autodesk MapGuide Enterprise DWF Viewer
- Design Review 2009 and Design Review 2010 cannot coexist on the same system
- Future versions of Autodesk MapGuide Enterprise will NOT include the DWF Viewer



Agenda

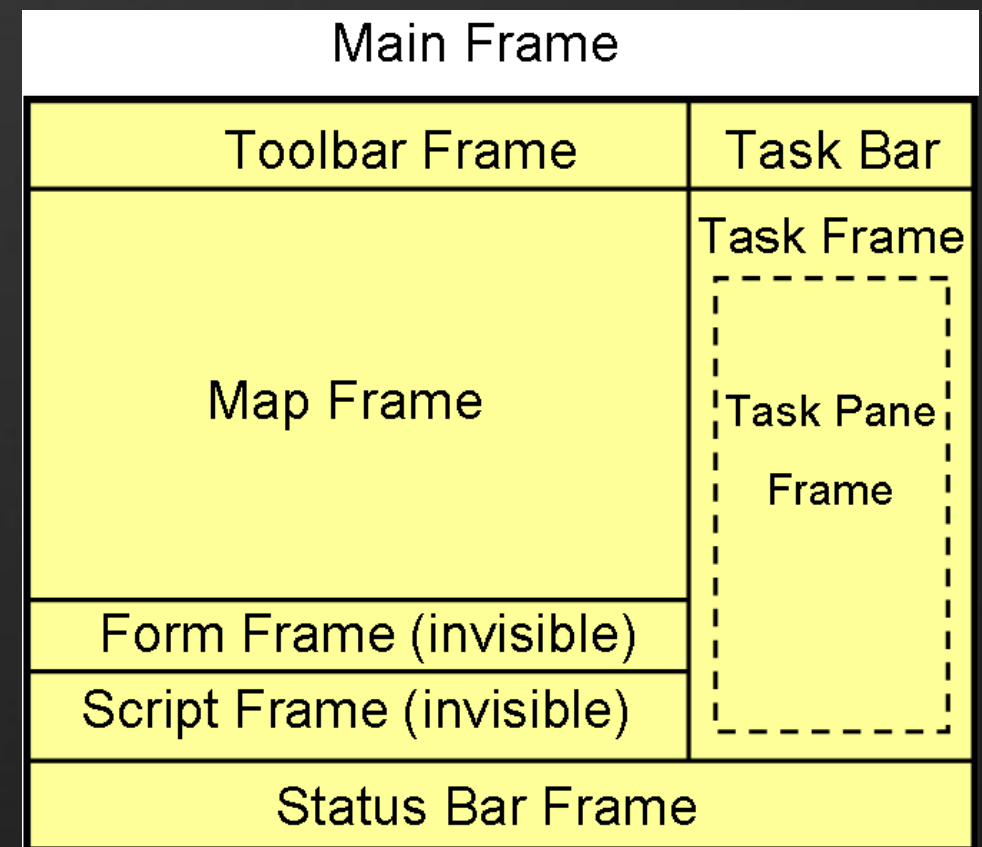
- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- [MapGuide API Overview](#)
- Integration of Fusion & Google Maps
- Data Migration Tool

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.
- For example

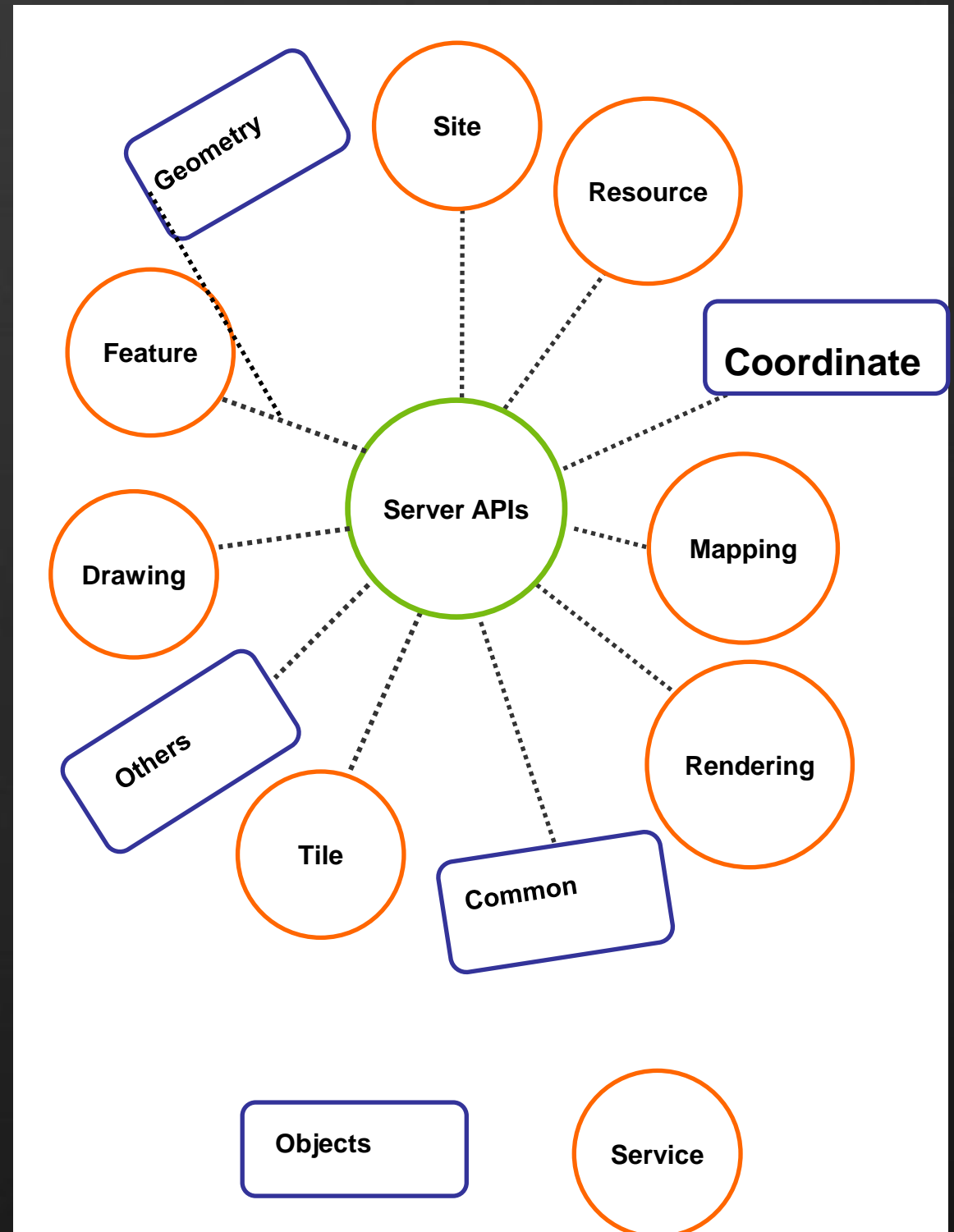
```
//From taskPaneFrame to refresh map  
parent.parent.mapFrame.Refresh();  
//From sbFrame to refresh map  
parent.mapFrame.Refresh();
```

Viewer Frame Structure



Server Side API Overview

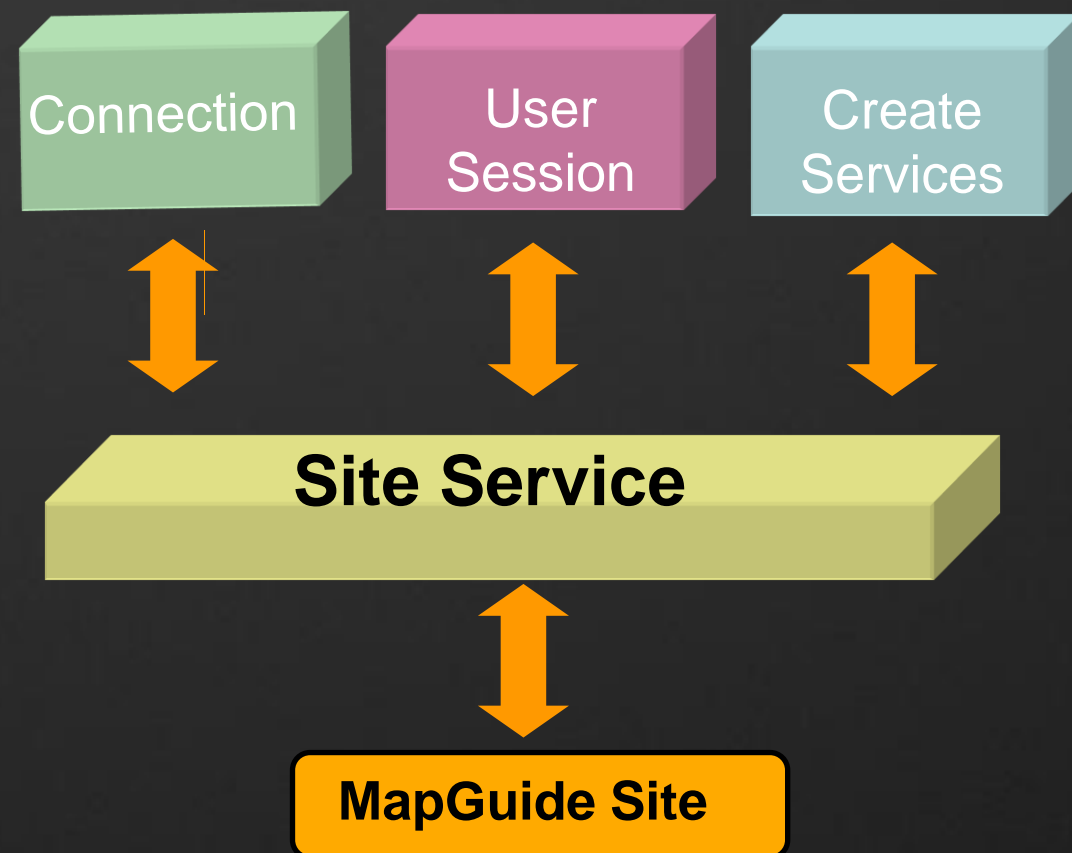
- Server side API is organized by service types
 - Site service
 - Resource service
 - Mapping service
 - Rendering service
 - Tile service
 - Drawing service
 - Feature service
- Accompanied by auxiliary objects
 - Geometry
 - Common
 - Coordinate
 - Objects



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



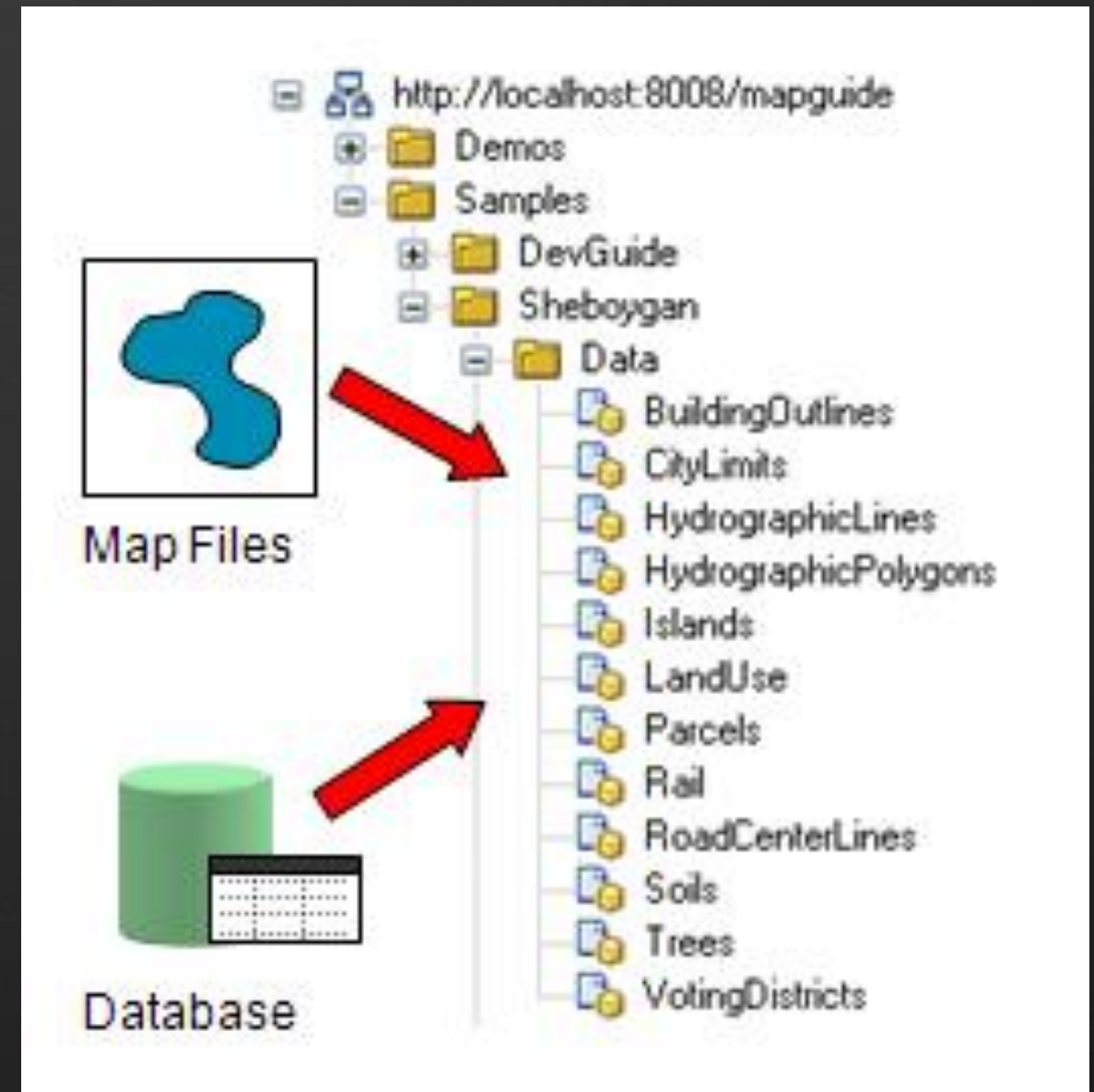
Resource Service

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

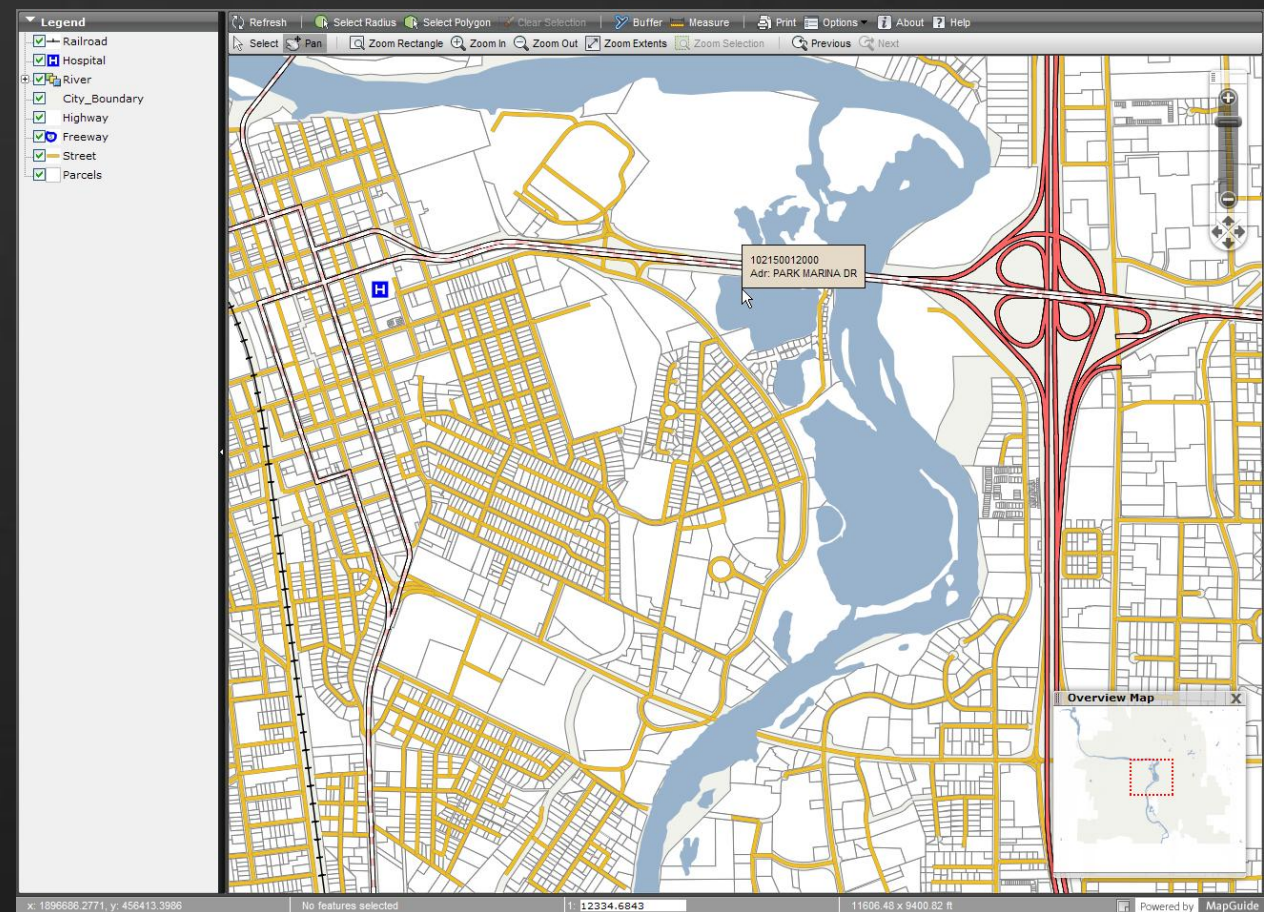
- Display and plot maps
- Manage map layers and layer groups
- Create map legends

Mapping service

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

Rendering service

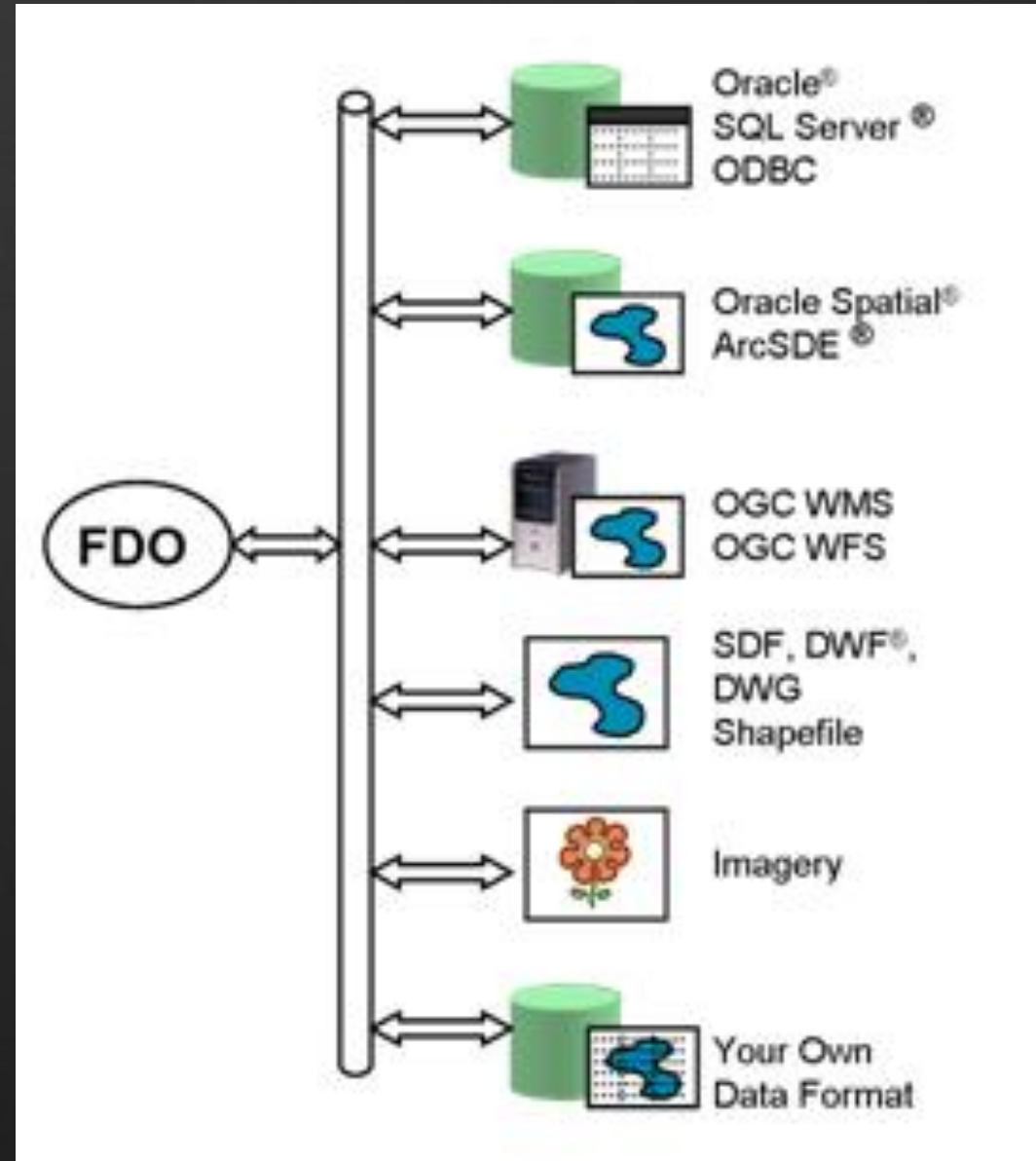
- Create maps in plain images



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

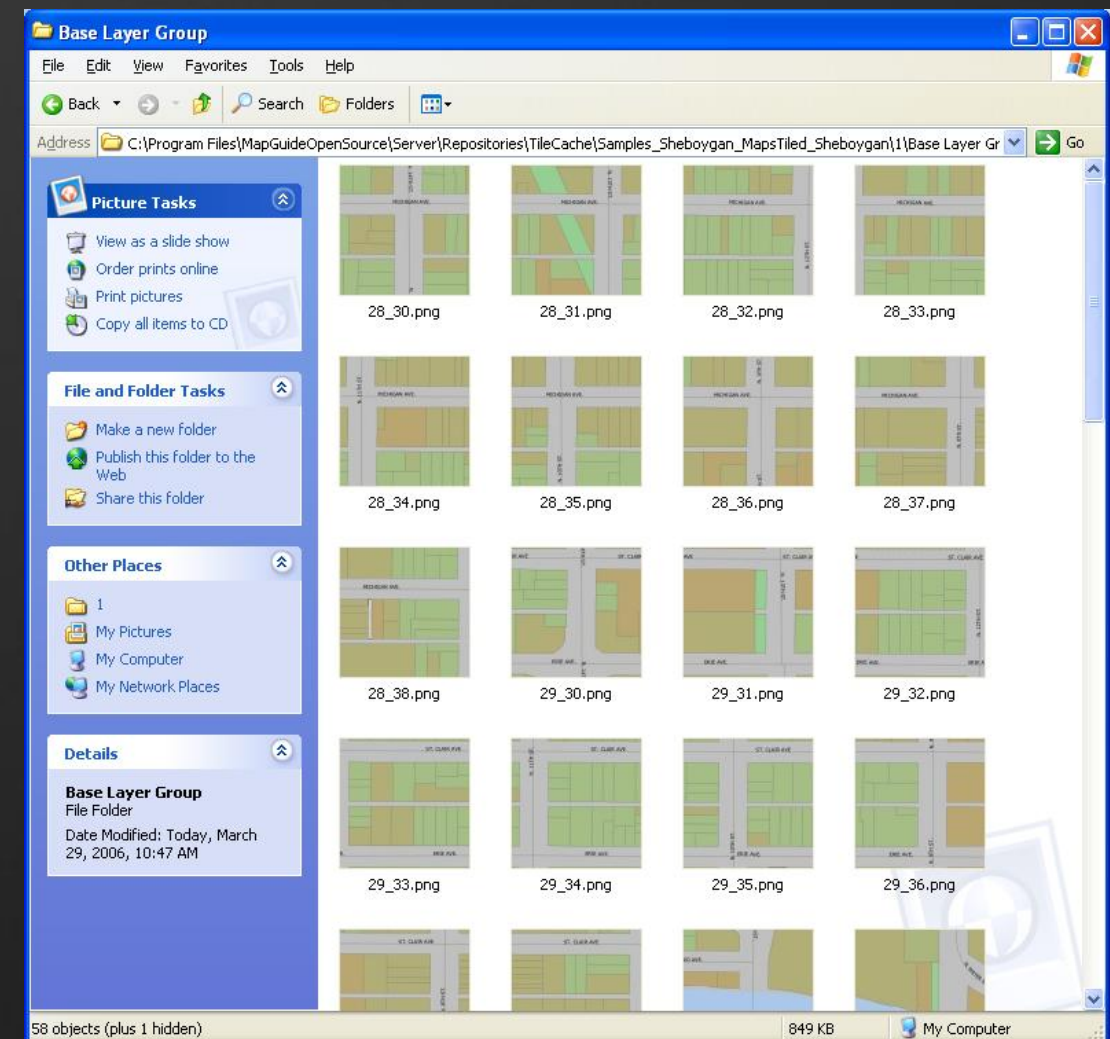


Tile Service

- 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.



Creating Resource Services

- A resource service manages data in a repository
- To use a service, web page must open a site connection and create an instance of a resource service
- MgSiteConnection object is used in creating the resource service

```
MgUserInformation userInfo = new MgUserInformation(sessionID);  
siteConnection = new MgSiteConnection();  
siteConnection.Open(userInfo);  
MgResourceService resourceService =  
    (MgResourceService) siteConnection.CreateService(  
        MgServiceType.ResourceService);
```

Enumerating Repository Resources

- Use `MgResourceService::EnumerateResources()` to enumerate resources
- Create an `MgByteSink` object from the returned `MgByteReader` and save it to a file.

```
MgResourceService resService =  
    (MgResourceService)siteConn.CreateService(  
        MgServiceType.ResourceService  
        //resources are located in a Library repository  
MgResourceIdentifier resourceID = new  
    MgResourceIdentifier "Library://"  
    //return info on MapDefs in folder and all its descendants  
MgByteReader byteRdr = resService.EnumerateResources(  
    resourceID, -1, "MapDefinition")  
    //convert to string  
String byteRdrStr = byteRdr.ToString();  
    //Create a byte sink object with the byte reader  
MgByteSink byteSink = new MgByteSink(byteRdr);  
    // write to file  
byteSinkToFile("MapDef.xml");
```

Resource Contents

- Resource contents can be obtained using `GetResourceContent()`
- `MapDefinition`, `LayerDefinition`, `PrintLayout`, `SymbolLibrary`, `LoadProcedure` have associated XML Schemas
 - See `C:\Program Files\Autodesk\MapGuideEnterprise2010\Server\Schema`

```
//A map definition resource
MgResourceIdentifier resourceID = new MgResourceIdentifier(
    "Library://Samples/Sheboygan/Maps/Sheboygan.MapDefinition");

//return resource content
MgByteReader byteRdr = resService.GetResourceContent(resourceID);

//Create a byte sink object with the byte reader
MgByteSink byteSink = new MgByteSink(byteRdr);

// write to file
byteSinkToFile("MapDef.xml");
```


Resource Contents

```
<?xml version="1.0" encoding="UTF-8" ?>
- <MapDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  1.0.0.xsd">
  <Name>Sheboygan Map</Name>
  <CoordinateSystem>GEOGCS["WGS84 Lat/Long's, Degrees, -180 ==> +180",DATUM["D_WGS84",SPHEROID["World_Geodetic_System_of_1984",6378137,298.257222932867]],PRIMEM["Greenwich",0],UNIT["Degree",0.7853981633974483]]</CoordinateSystem>
- <Extents>
  <MinX>-87.764986990962839</MinX>
  <MaxX>-87.695521510899724</MaxX>
  <MinY>43.691398128787782</MinY>
  <MaxY>43.797520000480347</MaxY>
</Extents>
  <BackgroundColor>FFCDBD9C</BackgroundColor>
- <MapLayer>
  <Name>Roads</Name>
  <ResourceId>Library://Samples/Sheboygan/Layers/Roads.LayerDefinition</ResourceId>
  <Selectable>>false</Selectable>
  <ShowInLegend>>true</ShowInLegend>
  <LegendLabel>Roads</LegendLabel>
  <ExpandInLegend>>true</ExpandInLegend>
  <Visible>>true</Visible>
  <Group>Transportation</Group>
</MapLayer>
- <MapLayer>
  <Name>Rail Lines</Name>
  <ResourceId>Library://Samples/Sheboygan/Layers/Tracks.LayerDefinition</ResourceId>
  <Selectable>>false</Selectable>
  <ShowInLegend>>true</ShowInLegend>
  <LegendLabel>Rail Lines</LegendLabel>
  <ExpandInLegend>>true</ExpandInLegend>
  <Visible>>false</Visible>
  <Group>Transportation</Group>
</MapLayer>
```

Working with Resources

- Copy an existing resource to another location using CopyResource()
- Move an existing resource to another location with MoveResource()
- Delete a resource with DeleteResource()

```
//Rename UKMapI.MapDefinition to UKMapII.MapDefinition
MgResourceIdentifier oldID = new MgResourceIdentifier(
    "Library://Samples/UK/Maps/UKMapI.MapDefinition");

MgResourceIdentifier newID = new MgResourceIdentifier(
    "Library://Samples/UK/Maps/UKMapII.MapDefinition");

resService.MoveResource(oldID, newID, true);
```

Resource Data

- Data used by a resource, but not stored in the resource itself, e.g. a binary SDF file, stored separately from the XML used to store the SDF resource itself
- Resource data can be stored as *files*, *streams*, or *strings*
 - Files are used when the data is large, e.g. SDF files
 - Streams are used for faster access for smaller pieces of binary data, e.g. symbols
 - Strings are used for small pieces of text data, e.g. database access credentials

```
//Create the resource ID
MgResourceIdentifier resourceID = new
MgResourceIdentifier("Library://Samples/UK/Data/UKBase.FeatureSource")
;
//Data name as specified in XXX.FeatureSource
String name = "UKBase.sdf";
// Get the layer SDF data and write it out to file
MgByteReader byteRdr = resService.GetResourceData(resourceID,name) ;
MgByteSink byteSink = new MgByteSink(byteRdr) ;
byteSinkToFile(SDFPath) ;
```


Get Feature Service

- Feature service can be created from MgSiteConnection object.
- It can restore the previous MapGuide session state because the MgSiteConnection object was created with session ID.

```
MgUserInformation userInfo = new MgUserInformation(sessionID);  
siteConnection = new MgSiteConnection();  
siteConnection.Open(userInfo);  
MgFeatureService featureService =  
    (MgFeatureService) siteConnection.CreateService(  
        MgServiceType.FeatureService);
```

Query Features from Source

- Feature service provides methods to retrieve features from a source.
- Retrieved features are stored in MgFeatureReader.
- Enumerate through the MgFeatureReader object to get individual feature.
- Get the property values based on the types defined in the feature schema.
- You can select a set of features from a source according to the criteria set in MgFeatureQueryOptions.
- Selection can be performed on both feature attributes and feature geometries.

Query Features from Source

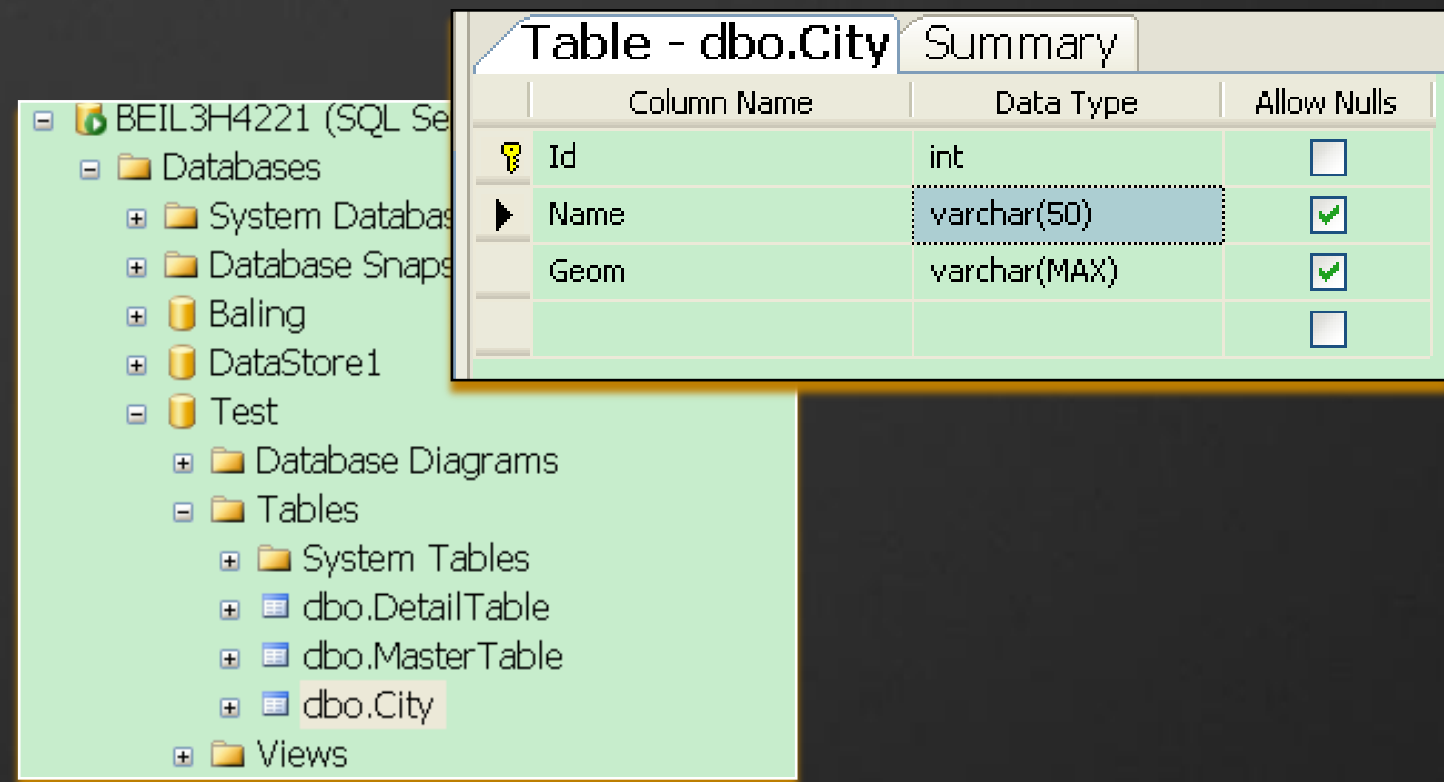
```
MgFeatureQueryOptions query = new MgFeatureQueryOptions();
query.SetFilter("Year >= 1990");
query.SetSpatialFilter("SHPGEOM", geometry,
    MgFeatureSpatialOperations.Inside);
MgFeatureReader featureReader =
    featureService.SelectFeatures(resId, "feature_class_name",
    query);

while (featureReader.ReadNext()) {
    int key = featureReader.GetInt32("Autogenerated_SDF_ID");
    MgByteReader byteReader = featureReader.GetGeometry("Data");

    MgGeometry geometry = geometryReaderWriter.Read(byteReader);
    MgPoint point = geometry.GetCentroid();
    double x = point.GetCoordinate().GetX();
    double y = point.GetCoordinate().GetY();
}
```


Feature Class Structure

- Database-table-like structure
- Feature class contains properties corresponding to table columns.



The screenshot shows a SQL Server Enterprise Manager interface. On the left, a tree view displays the database structure for 'BEIL3H4221 (SQL Server)'. It includes folders for 'Databases', 'System Databases', 'Database Snapshots', 'Baling', 'DataStore1', and 'Test'. Under 'Test', there are 'Database Diagrams', 'Tables', 'System Tables', and 'Views'. The 'Tables' folder is expanded, showing 'dbo.DetailTable', 'dbo.MasterTable', and 'dbo.City'. The 'dbo.City' table is selected. On the right, a 'Table - dbo.City' summary window is open, showing the following columns:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
Geom	varchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

MG classname	MG concept	Database concept	sample
MgFeatureSchema		Database	Test
MgClassDefinitionCollection			
MgClassDefinition		Table	City
MgPropertyDefinitionCollection			
MgPropertyDefinition		Columns definition	ID,Name...

Create a New Feature Class

Step 1

- Create a new MgClassDefinition object
- Get the property definition collection from the class

```
string ll84Wkt =  
@"GEOGCS[\"LL84\", DATUM[\"WGS_1984\", SPHEROID[\"WGS  
84\", 6378137, 298.25722293287], TOWGS84[0, 0, 0, 0, 0, 0, 0]], PRIMEM[\"Gre  
enwich\", 0], UNIT[\"Degrees\", 1]]\";  
  
MgClassDefinition parcelClass = new MgClassDefinition();  
  
parcelClass.SetName("tempParcel");  
  
MgPropertyDefinitionCollection props =  
    parcelClass.GetProperties();
```

Create a New Feature Class

Step 2

- Populate the property collection with proper values

```
MgDataPropertyDefinition id = new MgDataPropertyDefinition("ID");
id.SetDataType(MgPropertyType.Int32);
id.SetReadOnly(true);
id.SetNullable(false);
id.SetAutoGeneration(true);
props.Add(id);

MgPropertyDefinitionCollection idProps = parcelClass.GetIdentityProperties();
idProps.Add(id);
MgGeometricPropertyDefinition geom = new MgGeometricPropertyDefinition("GEOM");
geom.SetGeometryTypes(MgFeatureGeometricType.Surface);
geom.SetHasElevation(false);
geom.SetHasMeasure(false);
geom.SetSpatialContextAssociation("LL84");
props.Add(geom);
parcelClass.SetDefaultGeometryPropertyName("GEOM");

MgDataPropertyDefinition acre = new MgDataPropertyDefinition("ACRE");
acre.SetDataType(MgPropertyType.String);
acre.SetLength(256);
props.Add(acre);
```


Create a New Feature Class

Step 3

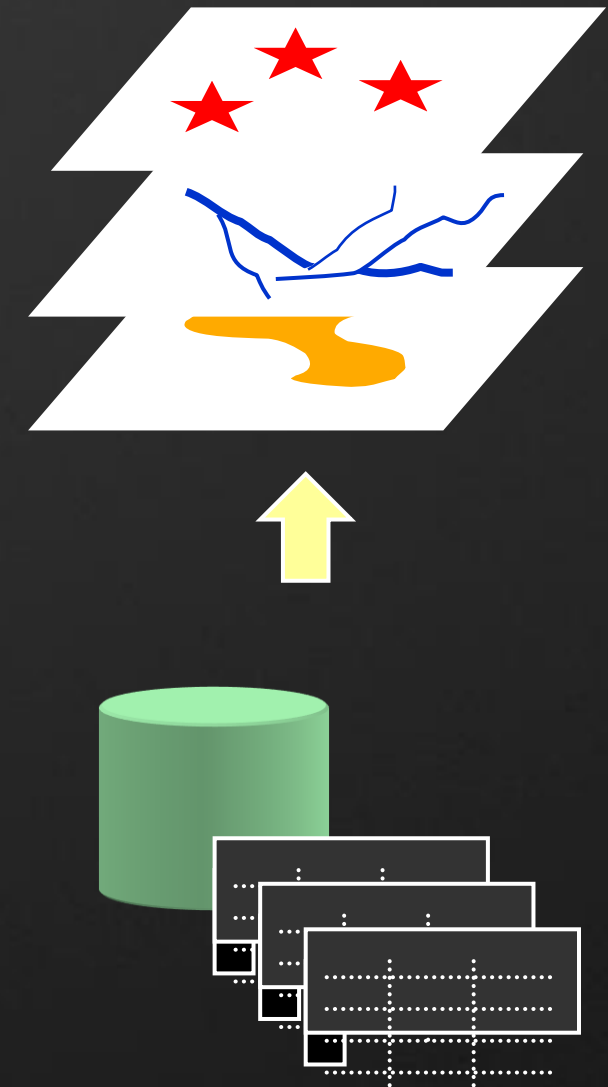
- Create and set up schema
- Create the physical storage format

```
MgFeatureSchema schema = new MgFeatureSchema();  
schema.SetName("SchemaParcels");  
schema.GetClasses().Add(parcelClass);  
  
MgCreateSdfParams sdfParams = new  
    MgCreateSdfParams("LL84", 1184Wkt, schema);  
featureService.CreateFeatureSource(resId, sdfParams);
```

Map Layer

Map Layer

- Visual representation of feature class
- Based on feature class
- Composition and style are defined in LayerDefinition file
- Like feature sources, map layers can be created on permanent library or temporary session repository
- Can be dynamically created on-the-fly



Create a Map Layer Programmatically

- Create or parse the LayerDefinition XML file.
- Create a new map layer resource.
- Set the map layer's properties.

```
Byte[] layerDef = new byte[length - 1];
MgByteSource byteSource = new MgByteSource(layerDef,
    layerDef.Length);
byteSource.SetMimeType(MgMimeType.Xml);

MgResourceIdentifier parcelLayerId = new
MgResourceIdentifier("Session:" + sessionId +
    "//parcelLayer.LayerDefinition");
resService.SetResource(parcelLayerId,
byteSource.GetReader(), null);

MgLayer parcelLayer = new MgLayer(tempParcelLayerId, resService);
parcelLayer.SetName("NewParcels");
parcelLayer.SetLegendLabel("New Parcels");
parcelLayer.SetDisplayInLegend(true);
parcelLayer.SetSelectable(false);
```


Add Layer to Map Display

- Get the current map display, MgMap.
- Add the layer to map display.
- Refresh map display and save it.

```
MgMap map = new MgMap();  
map.Open(resService, "Sheboygan");  
  
MgLayer parcelLayer = getLayerByName(map, "NewParcels");  
map.GetLayers().Insert(0, parcelLayer);  
  
parcelLayer.SetVisible(true);  
parcelLayer.ForceRefresh();  
map.Save(resService);
```

Feature store concept model

More like database table

Table - dbo.City		Summary	
	Id	Name	Geom
▶	1	Beijing	slkdfj20923jrns...
	2	Shanghai	20jksdfj98nj43k...
	3	Newyork	lksfl309kg0934m...
*	NULL	NULL	NULL

MG classname	Database concept	sample
MgProperty	Field and value	Id=1
MgPropertyCollection	Partial Record/ Record Set	{id=1,name=Beijing}

MgPropertyCollection Class

Example on using MgPropertyCollection class

- Assume all the properties are nullable and we need to update the feature with these values

```
MgPropertyCollection props = new MgPropertyCollection();  
MgAgfReaderWriter agfWriter = new MgAgfReaderWriter();  
  
props.Add(new MgGeometryProperty("GEOM", agfWriter.Write(geom)));  
props.Add(new MgStringProperty("LOTDIM", "540X400"));  
props.Add(new MgInt32Property("SQFT", "6600"));  
props.Add(new MgStringProperty("ZONE", "RES" ));
```

Editing Feature Class

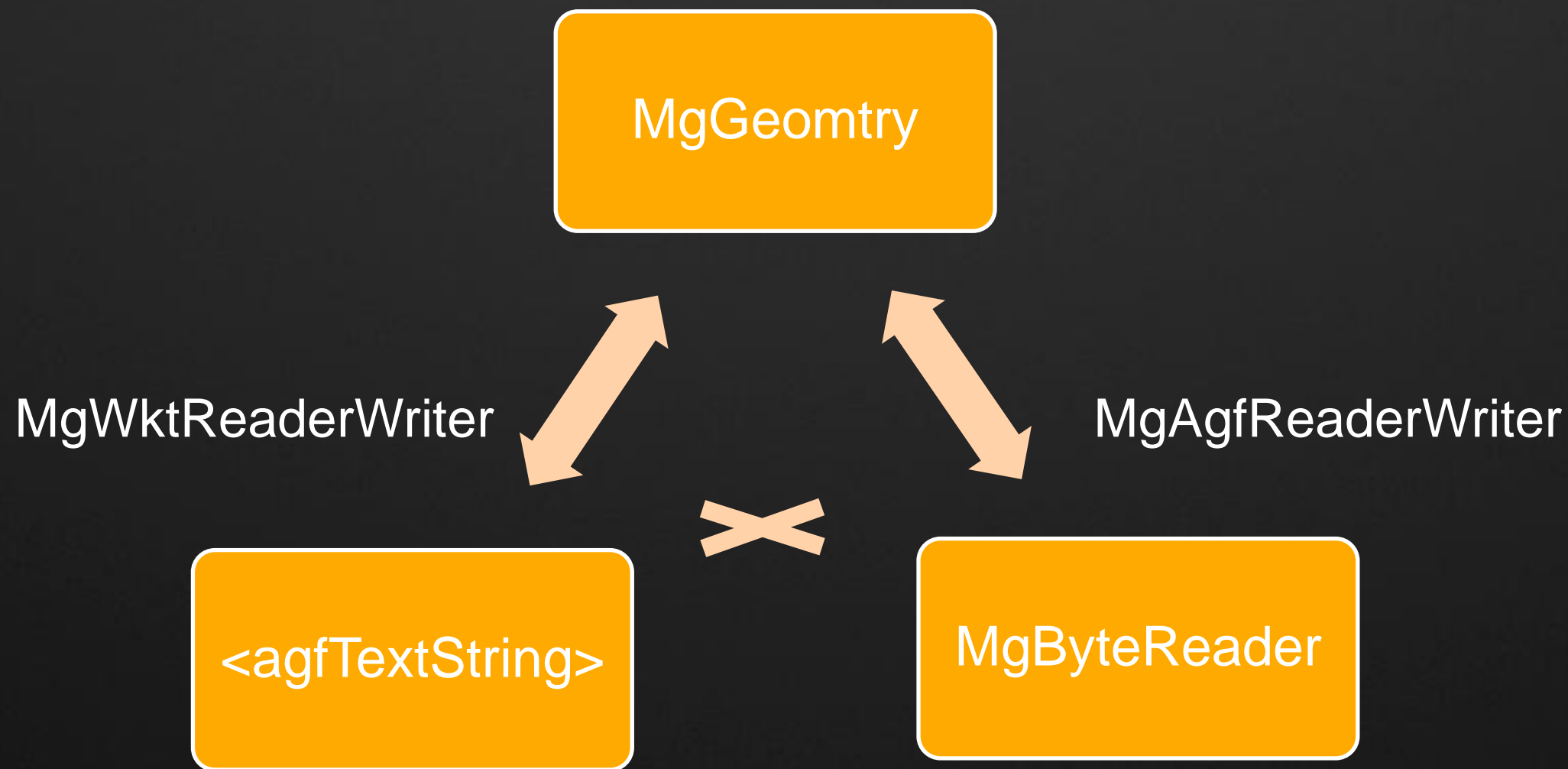
- Feature class can be edited.
- **Three operations**
 - Delete(**MgDeleteFeatures**)
 - Update(**MgUpdateFeatures**)
 - Insert(**MgInsertFeatures**)
- Feature editing is carried out on FDO data source

```
MgDeleteFeatures deleteFeatures = new MgDeleteFeatures
    ("Parcels", "ID=2354");
//Variable "properties" is the MgPropertyCollection object
MgInsertFeatures insertFeatures = new MgInsertFeatures
    ("Parcels", properties);
//Variable "properties" is the MgPropertyCollection object.
MgUpdateFeatures updateFeatures = new MgUpdateFeatures
    ("Parcels", properties, "ID=2354");
MgFeatureCommandCollection commands = new
    MgFeatureCommandCollection();
commands.Add(deleteFeatures);
commands.Add(updateFeatures);
commands.Add(insertFeatures);
//true means the execution is a transaction
featureService.UpdateFeatures(sourceId, commands, true);
```


Representation of Geometry

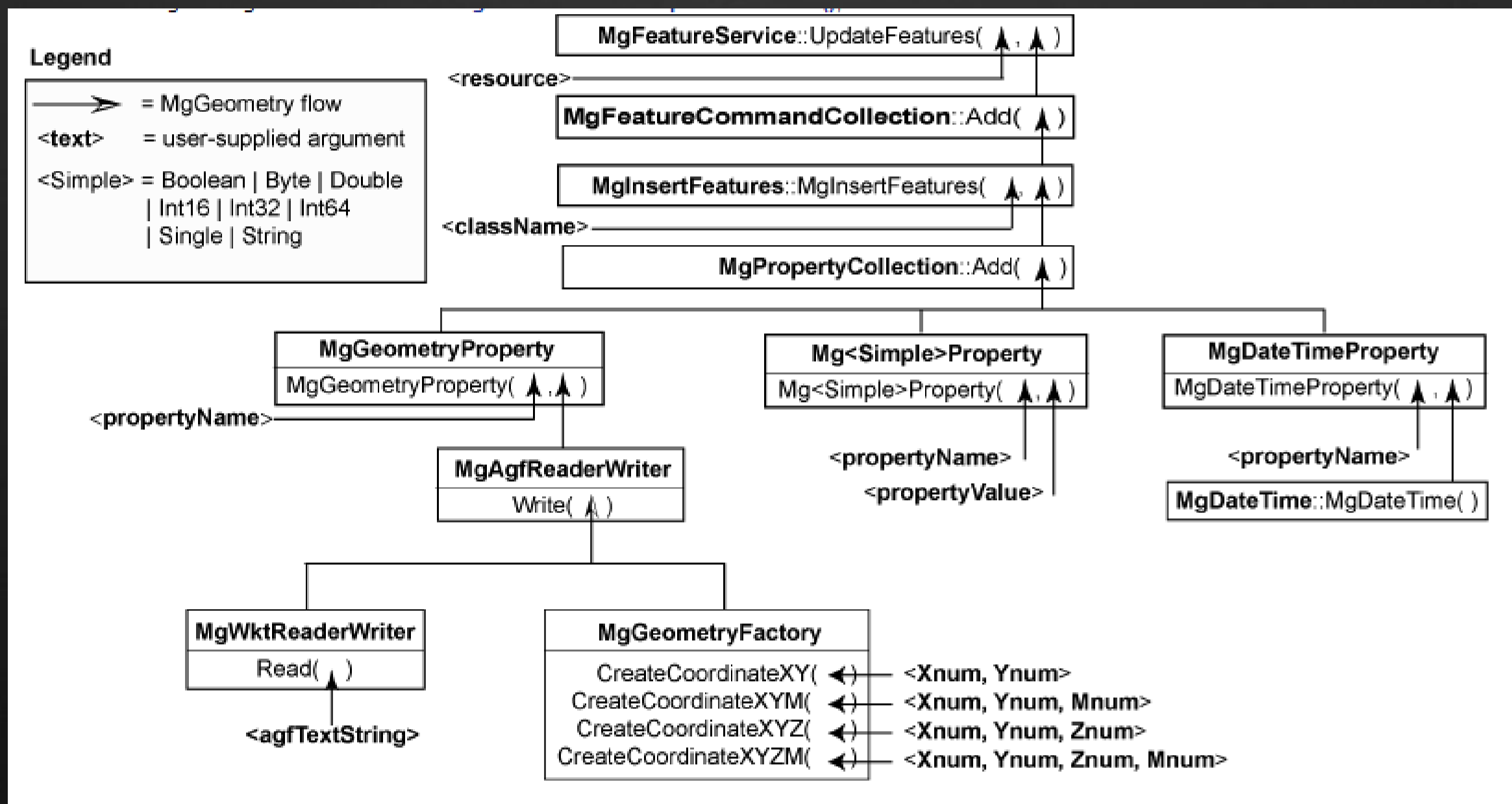
- **Three different forms:**

- AGF(Autodesk Geometry Format) text format, superset of WKT
- Binary AGF format, used by FDO technology supporting the Feature Service
- MgGeometry and derived, Autodesk MapGuide internal representation



Roadmap of Inserting Features

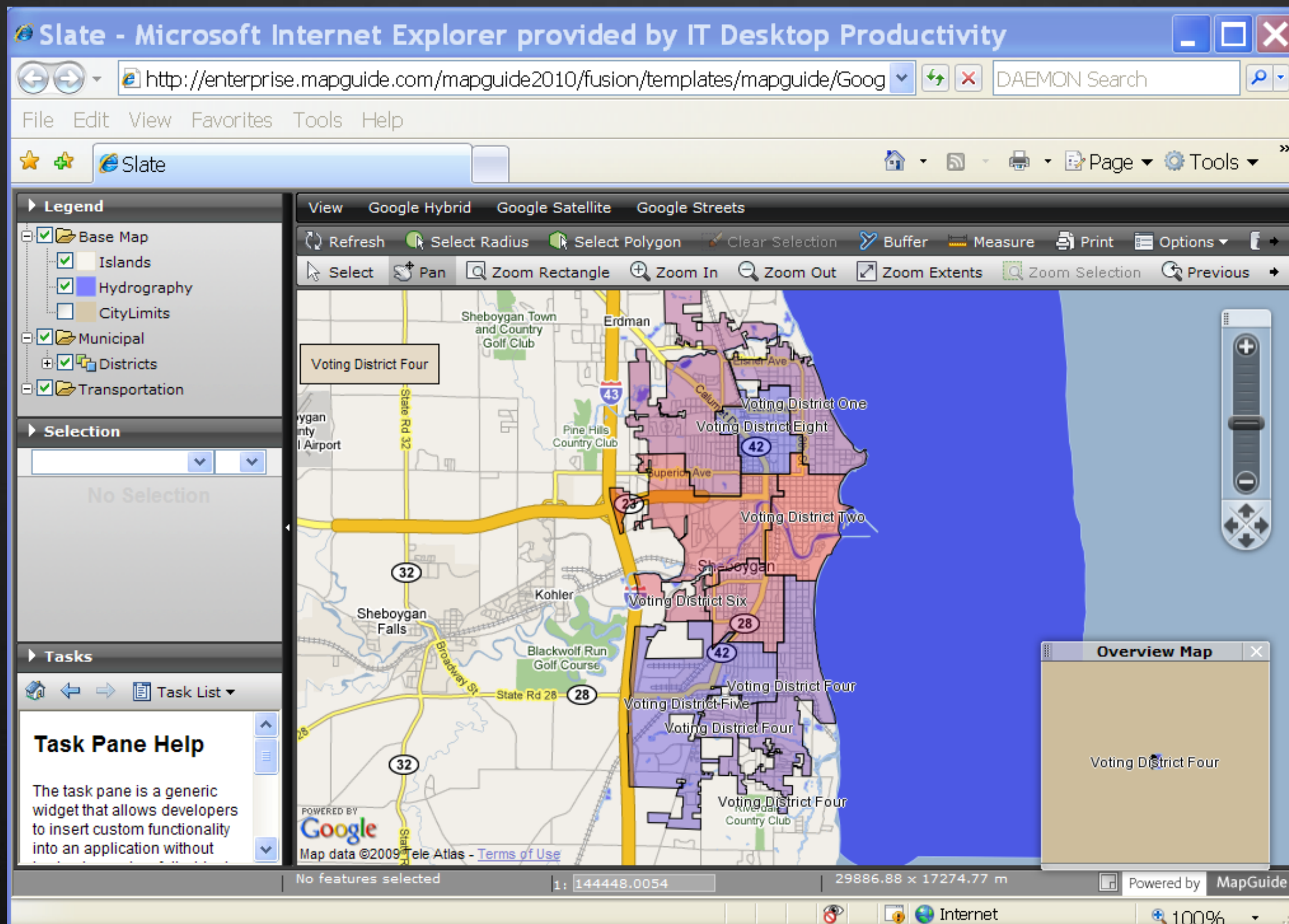
The following illustration shows the relationships among the classes used in creating the arguments for a call to [MgFeatureService::UpdateFeatures\(\)](#), which contains one insertion command:



Agenda

- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- MapGuide API Introduction.
- Integration of Fusion & Google Maps
- Data Migration Tool

Integration of Fusion & Google Maps



OpenLayers & Fusion

- Open Layers

- An open source JavaScript library for displaying mapping data in a web browser
- All client side – there are no server side dependencies
- All source code can be downloaded
- Website: <http://www.openlayers.org>

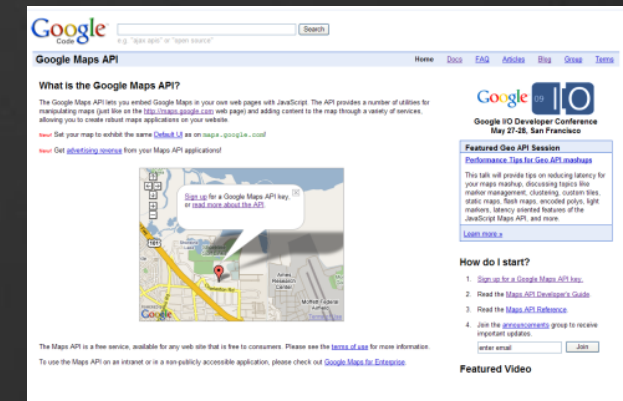


- MapGuide Fusion Viewer

- Fusion uses a subset of OpenLayers
- Allows access to the underlying OpenLayers API

How to do?

- Sign up for a Google Maps API key.
- <http://code.google.com/apis/maps/>
- `<!-- Add the Google Map API Javascript library--><script src='http://maps.google.com/maps?file=api&v=2&key=ABQI.....IUiOiZObZE SPg'></script>`
- Copy an existing web layout folder and corresponding xml file and rename
- Update the <Name> and <LocationUrl> in xml file
- Update index.html to include Google API Key
- Update index.html to include Google base map of choice
- [Optional] use Studio to add scripts that call the API to update base layer



How to do?

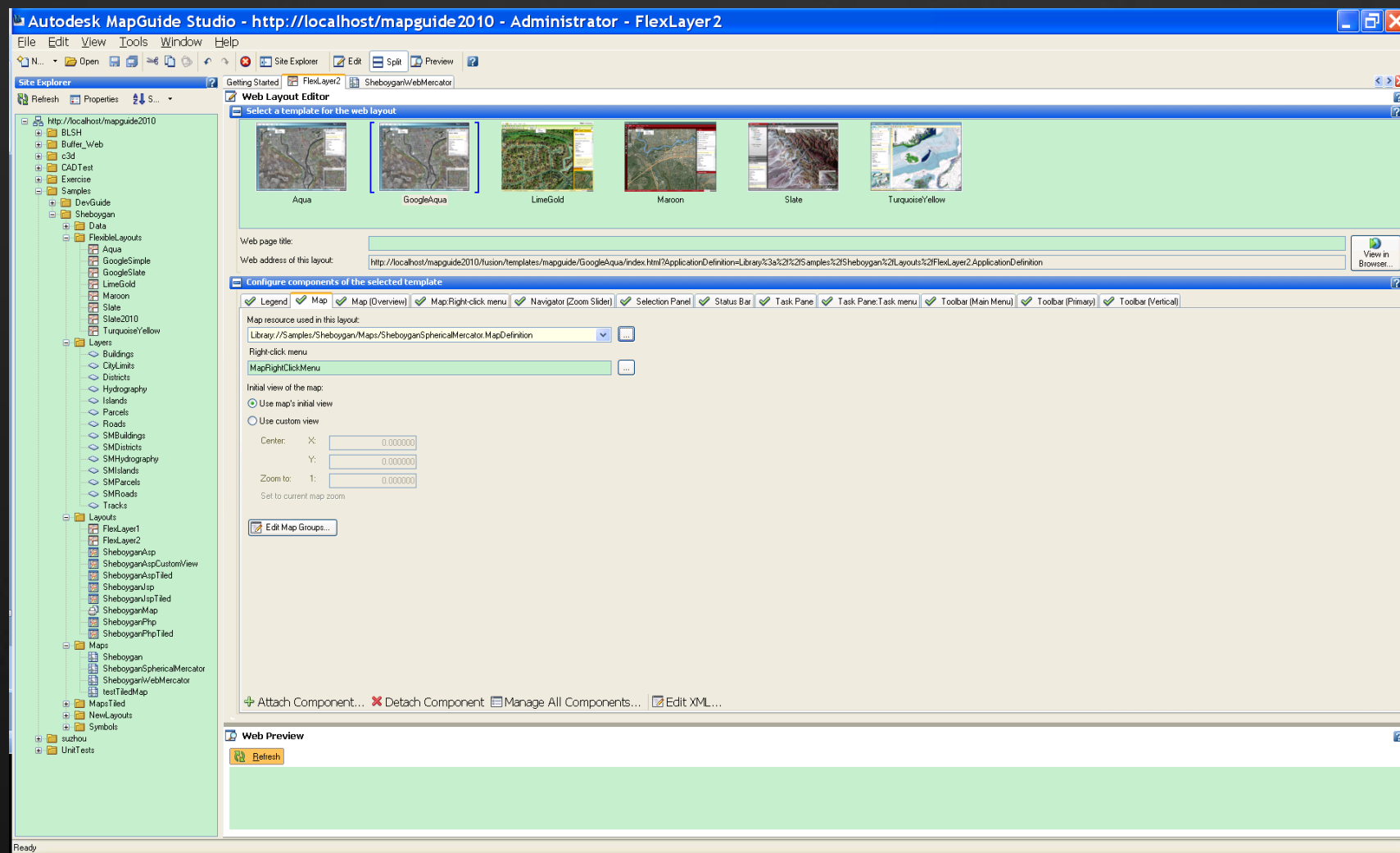
- Edit the index.html of Fusion template

```
//Register the MAP_LOADED event to load Google base map when map is loaded
Fusion.getWidgetById('Map').registerForEvent(Fusion.Event.MAP_LOADED, addGoogleBaseMap);
```

```
// Add the google map as base map
var addGoogleBaseMap = function()
{
    var map = Fusion.getWidgetById('Map').oMapOL;
    //create google layer by OpenLayer API
    var googleStreetsLayer = new OpenLayers.Layer.Google('Google Streets',
    {type : G_HYBRID_MAP, isBaseLayer:true, sphericalMercator:true});
    map.addLayer(googleStreetsLayer);
    //...
    map.setBaseLayer(googleStreetsLayer);
}
```

How to do?

- Create a Map
 - CS: EPSG:3785 (Popular Visualisation CRS / Mercator)
- Create a flexible weblayout using the custom template and selecting the map



Agenda

- MapGuide Enterprise introduction
- MapGuide Enterprise 2010 new features
- How to create my first MapGuide application
- MapGuide API Introduction.
- Fusion viewer API introduction
- Integration of Fusion & Google Maps
- Data Migration Tool

Data Migration Tool – What is it ?

Autodesk MapGuide Enterprise

[Print](#)

MapGuide Family

Features

System Requirements

White Papers

Customer Stories

Product Trial

News & Events

Events

Articles & Reviews


Press Releases

Migrate

Related Products

How to Buy

Subscription Benefits

 [Product Brochure \(pdf - 2745Kb\)](#) [FAQ \(pdf - 112Kb\)](#)

Services & Support

Resource Center

Support

Documentation

Data Migration Tool

Data Migration Made Easy

You can use the Autodesk MapGuide Data Migration Tool to migrate your MapGuide 6.5 data in these file formats to Autodesk MapGuide Enterprise 2010.

- MWF
- MWX
- UDL
- MLF
- SMB (Symbol Library)


Hardware Requirements

The Data Migration Tool can be installed on either Microsoft® Windows® 2000 Server or Microsoft Windows 2003 Server or Microsoft Windows XP Pro SP2 and requires the following: Intel Pentium, III/IV 1 GHz, 512MB of RAM.

Software Requirements

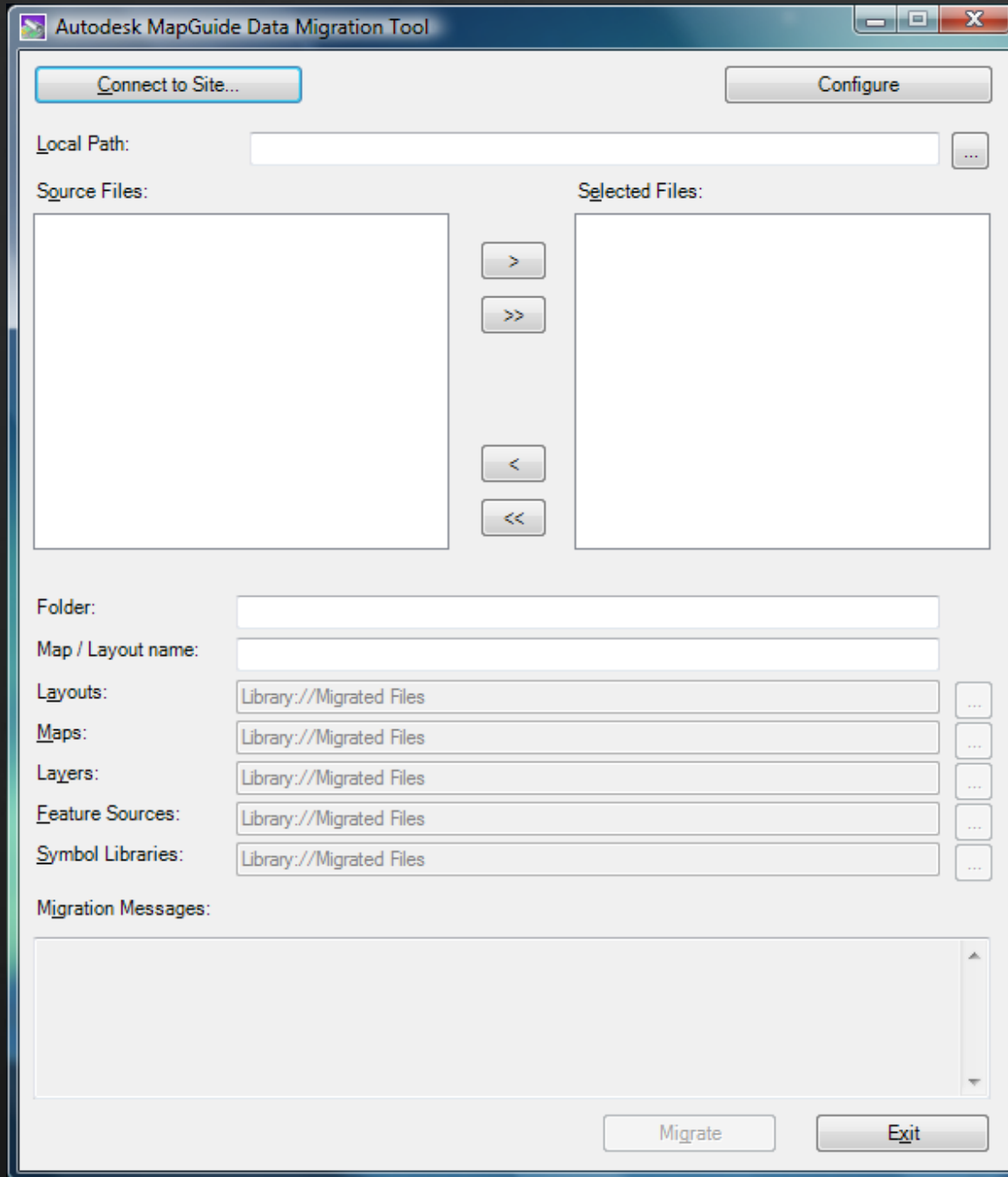
The Data Migration Tool requires Autodesk MapGuide Enterprise 2010 and Autodesk MapGuide Server 6.5 to be installed on the same system to migrate data files and to access UDL files. Note: Files can also be migrated from Autodesk MapGuide Server 6.5 to Autodesk MapGuide Enterprise 2008 or 2009 software.

Documentation

 [Data Migration Tool: User Guide \(pdf - 245Kb\)](#) [MapGuide2010_MigrationTool.zip \(zip - 14592Kb\)](#)

<http://www.autodesk.com/datamigrationtool>

Data Migration Tool – What is it ?



<http://www.autodesk.com/datamigrationtool>

Why to migrate ?

Why should you migrate from Autodesk MapGuide® software to Autodesk MapGuide® Enterprise web mapping technology?

Here are our top 10 reasons:

- Platform Flexibility
- Single Authoring Tool
- Programming Flexibility
- Simpler Viewing
- More Innovation

Why to migrate ?

- Server-Side Processing
- Integration with AutoCAD Map 3D Software
- High Quality, consistent Maps
- Easier Data Access
- Remote Administration

Data Migration - Limitations

- No Support for current DWG file format
- Themed Layer and Object Data of DWG Layers are not migrated
- MWF based on Map queries will not be migrated.
- No display of symbols – SMB files.
- Zoom GoTo functionality from Autodesk MapGuide 6.5 will not be migrated.
- Spatial filters will not be migrated

MapGuide Resources -

Autodesk MapGuide® Enterprise Product Site Page

<http://www.autodesk.com/mapguideenterprise>

DevTVs, Recorded Webcasts, Code Samples @ A D N

<http://adn.autodesk.com>

API Training Classes

www.autodesk.com/apitraining

Autodesk Developer Network

www.autodesk.com/joinadn

Developer Center

www.autodesk.com/developautocad

MapGuide Best Practices Wiki

http://sandbox.mapguide.com/index.php/Main_Page

MapGuide Resources -

Autodesk MapGuide Enterprise

MapGuide Family

Features

System Requirements

White Papers

Customer Stories

Product Trial

News & Events


Migrate

Related Products

How to Buy

Subscription Benefits

 [Product Brochure](#) (pdf - 2745Kb)

 [FAQ](#) (pdf - 112Kb)

Services & Support

[Resource Center](#)

[Support](#)

[Documentation](#)

White Papers

Creating a Secure Autodesk MapGuide Enterprise Site

This document provides step-by-step instructions for creating a secure Autodesk MapGuide® Enterprise software site for different configuration types. The setup includes an open component for viewing maps, and a secured component for administering and authoring the site.

 [Creating a Secure Autodesk MapGuide Enterprise Site](#) (pdf - 277Kb)

Mashup Autodesk MapGuide Enterprise with Google Earth and Google Maps

A mashup is the integration of publicly accessible data with privately owned data sources. This white paper provides background and examples for creating mashups of Autodesk MapGuide® Enterprise software with Google Earth™ and Google Maps™ mapping service technology. Using MapGuide Enterprise as the business logic tier to the base map information from Google Maps or Google Earth provides a compelling way to visualize data in relation to the planet as a whole.

 [Mashup Autodesk MapGuide Enterprise with Google Earth and Google Maps](#) (pdf - 853Kb)


Moving to MapGuide Enterprise: A Guide for ESRI ArcIMS Users

Are you looking for a new web-based geographic information system (GIS) solution? Learn how to migrate an ArcIMS application to MapGuide Open Source or Autodesk MapGuide® Enterprise software.

 [Moving to MapGuide Enterprise: A Guide for ESRI ArcIMS Users](#) (pdf - 3331Kb)

FDO Data Access Technology - New FDO Providers

Extend data access for Autodesk software using FDO Data Access Technology and third-party or open source FDO providers.

 [FDO Data Access Technology - New FDO Providers](#) (pdf - 475Kb)

AutoCAD Map 3D and Autodesk MapGuide Enterprise: Powerful, Affordable, and Open GIS

With open data standards, CAD integration, and a shared API, AutoCAD® Map 3D® and Autodesk MapGuide Enterprise streamline workflows and maximize the value of geospatial data seamlessly—from desktop to web. Learn how your organization can work seamlessly to create, manage, and share engineering design and geospatial data.

 [AutoCAD Map 3D and Autodesk MapGuide Enterprise: Powerful, Affordable, and Open GIS](#) (pdf - 547Kb)

Print



Email



MapGuide Resources -

Welcome to the Autodesk MapGuide Enterprise Demo Server - Microsoft Internet Explorer provided by IT Service Centers

http://enterprise.mapguide.com/landing/index.php

File Edit View Favorites Tools Help

Welcome to the Autodesk MapGuide Enterprise...

Welcome to enterprise.mapguide.com.
This server hosts Autodesk MapGuide Enterprise demonstrations and sample applications.

Home

search for demo

Resources

- [Best Practices Wiki](#)
- [Product Page](#)
- [Resource Center](#)
- [MapGuide OSGEO Live Gallery](#)
- [Autodesk Developer Network](#)

Downloads

- [Autodesk MapGuide Enterprise 2010 Sample Application](#)

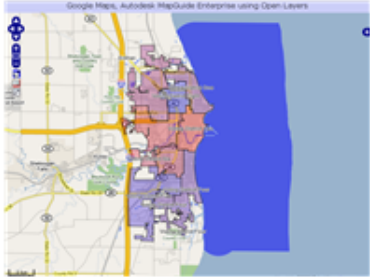
Latest Demos

- [Open Layers with Google Maps & Autodesk MapGuide Enterprise](#)
- [Flexible Web Layouts - Custom Symbology](#)
- [ERDAS Image Web Server Demo](#)
- [Flexible Web Layouts - Google Maps API](#)
- [Flexible Web Layouts - Widgets](#)

Popular Demos

- [Open Layers with Google Maps & Autodesk MapGuide Enterprise](#)
- [Generic Autodesk MapGuide Enterprise Demo](#)
- [Topobase Web](#)
- [Flexible Web Layouts - Custom Symbology](#)
- [Utility Map](#)

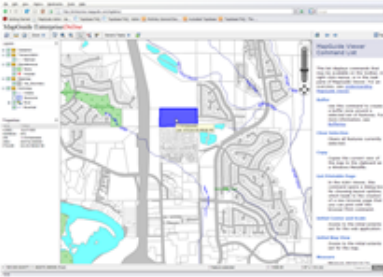
Open Layers with Google Maps & Autodesk MapGuide Enterprise



This demo showcases the use of a Open Layers with the Google Maps API and Autodesk MapGuide Enterprise.

For best results use IE7 or Firefox 2.0

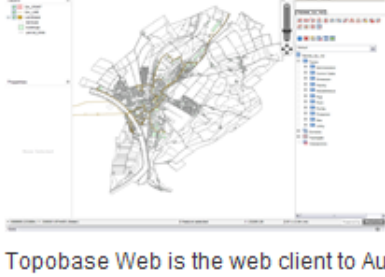
Generic Autodesk MapGuide Enterprise Demo



Try out the cool features of Autodesk MapGuide Enterprise

- [AJAX Viewer \(no plugin required\)](#)
- [Enhanced map rendering](#)
- [True Color Support](#)

Topobase Web



Topobase Web is the web client to Autodesk Topobase and is built on Autodesk MapGuide Enterprise

Login with the following credentials

Username: WEB
Leave the password field empty.

When accessing the site for the first time, you



Thank You !
Questions ?

partha.sarkar@autodesk.com

Autodesk®

Autodesk®