Chapter 7: Fusion API introduction

Flexible web layout

Autodesk

Fusion viewer / Flexible web layout

Autodesk
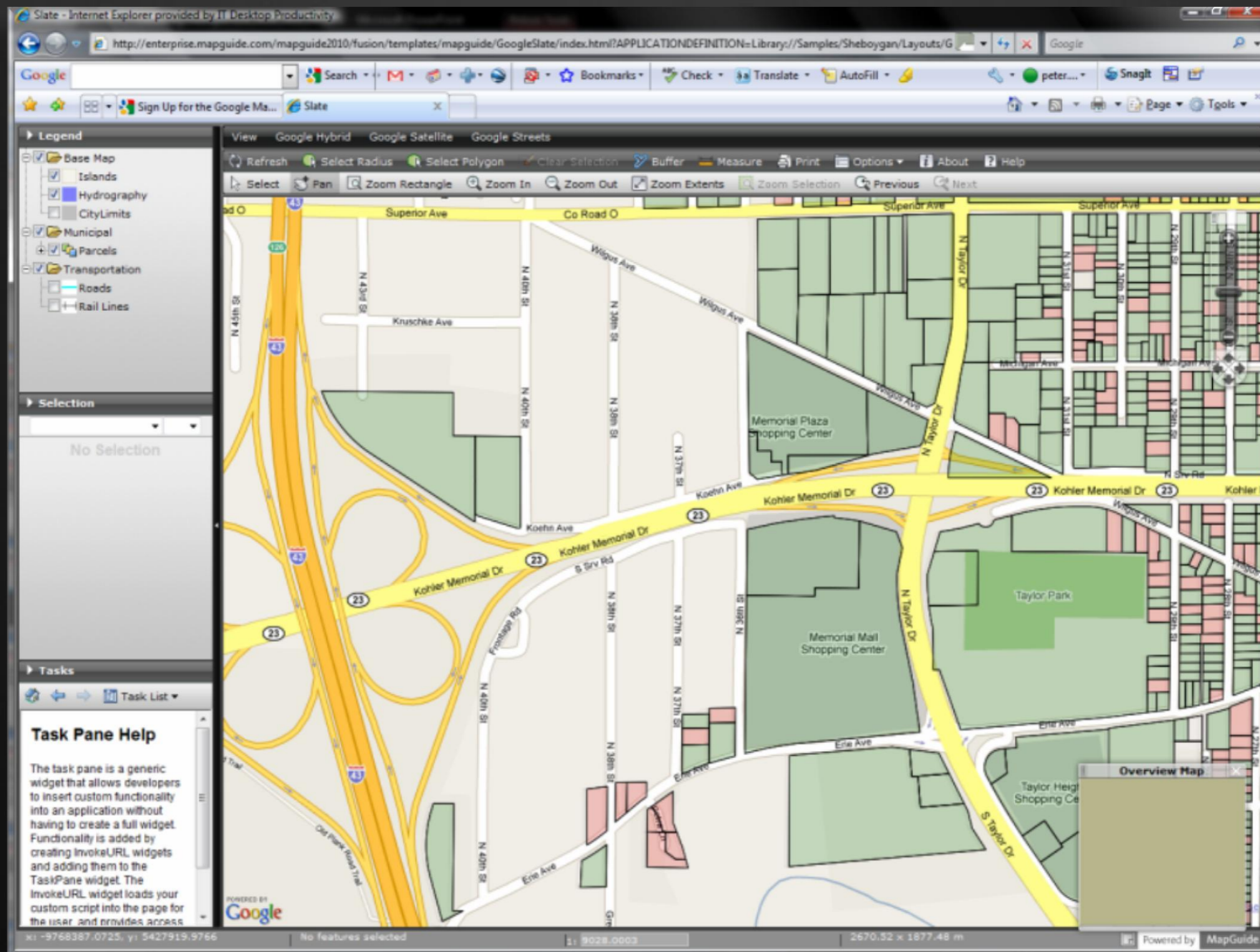
What is Fusion?

- A web-mapping application development framework
- Built primarily in JavaScript.
- Based on the Openlayers, easy to extend
- Using widgets, standard-compliant HTML and CSS.
- No plug-ins and works in all major browsers
- **Flexible Layouts** inside Autodesk Infrastructure Studio

http://trac.osgeo.org/fusion/

Autodesk

Fusion Capabilities

- Dynamic application generation from ApplicationDefinition resource and an HTML template
- Supports MapGuide Tiles and Untiled overlayers
- Internationalization support
- Authentication integration
- Functional parity with the AJAX viewer

Components of Fusion

- Core
  - Platform-independent (entirely javascript)
  - Provides main API for building applications
  - OpenLayers (www.openlayers.org)
- Widgets
  - Discrete, independent functionality (zoom, pan, …)
  - Plug-'n-play (zero programming)
- UI Library
  - Jx provides common application building blocks
  - Tabs, Menus, Buttons etc …
  - Prototype, Scriptaculous
- Platform-specific plugins
  - MapGuide, MapServer, OGC, and potentially others

# Fusion Developers are:

1. Web designers
   - Work in DreamWeaver, other HTML editor
   - Use HTML, CSS, minimal javascript
   - Concerned primarily with aesthetics, layout, look and feel, usability
2. Web developers
   - Work in favorite IDE (Zend, text editor)
   - Use JavaScript, PHP, ASP .NET, JSP
   - Concerned with functionality

Why Fusion?

- The AJAX viewer is:
  - not flexible
  - frames based
  - difficult to customize functionality
  - difficult to modify layout

Autodesk
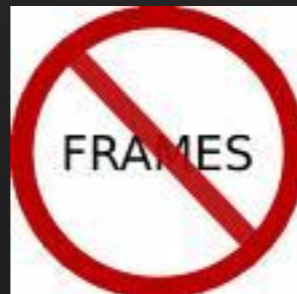
Creating a Fusion Application

- Fusion needs two things to make an application:

1. An XML Application Definition
   - Typically ApplicationDefinition.xml

2. An HTML web page
   - called a Template in the docs

Autodesk

Application Definition

- New resource type for MGOS/AIMS
- Structured XML similar to WebLayout
- Major blocks:
  - MapSet - lists maps available to the application
  - WidgetSet - relates widgets to a map
- Widgets
  - Are the control block for a widget
  - Name tag is where the widget goes
  - Type tag identifies the widget to use

Autodesk

Templates

- A template is just an HTML file
- *Should* have a valid doctype
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- Uses 'id' of tags to identify where widgets go
- *Any* tag will do
  - div, span, a, li are commonly used
- Includes fusion.js
- Calls Fusion.initialize()

Autodesk

Example of a Template

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- `<html>`
- `<head>`
- `<title>Simple Fusion Application</title>`
- `<script type="text/javascript" src="../../../lib/fusion.js"></script>`
- `<script type="text/javascript">`
- `window.onload = function() { Fusion.initialize(); }`
- `</script>`
- `</head>`
- `<body>`
- `<div id="ZoomIn"></div>`
- `<div id="Map"></div>`
- `</body>`
- `</html>`

Autodesk

How does it Work?

- Load ApplicationDefinition.xml
- Loop through WidgetSet/Widget and
    - Find elements by ID using Widget Name
    - Request widget JavaScript file
- Load widget JavaScript files
- Load widget dependencies
- Create widgets
- Create a Session

Fusion Widgets

- ## Navigation & Map State
  - Zoom, Pan, Legend, Layer Manager, Extent History, Overview Map
- ## Selection
  - Rectangle, circle, polygon
- ## Info
  - Cursor Position, MapTips, Scale, etc
- ## Tasks
  - Buffer, Search, InvokeURL, About, Help, …

Jx

- JavaScript-based UI library
- Open Source (MIT) www.jxlib.org
- Cross-browser compatible
- Provides basic building blocks for creating applications

Button	Menu
Toolbar	Color
Tab	Layout
Panel	Splitter
Grid	Tree

Using Jx in Fusion

- Fusion uses:
  - Toolbar
  - Button
  - Menu
  - Tree
  - Layout
- Application can use any Jx component to build the look and feel

Customizing Jx

- Jx components use
  - Simple, semantic HTML structures
    - UL, LI, A, DIV, SPAN, IMG
    - Tables are only for tabular data!
  - CSS 2 for presentation
    - CSS is 'hack' free
    - May use browser-specific stylesheets (IE)
- Change entire look and feel through a single CSS style sheet (500-600 lines)

Jx Builds On

- Prototype
  - www.prototypejs.org
  - Javascript framework for class-driven development in an AJAX environment
- Scriptaculous
  - script.aculo.us
  - Javascript visual effects library built on Prototype
  - Animation, drag-n-drop, ajax controls …

OpenLayers

- www.openlayers.org
- OpenLayers is a javascript framework for creating tiled and untiled maps
- Unique layering system allows combining many different map sources in a single map view
- Advanced 'Mashup' capability
- Active development community

Major Fusion API Components

- ApplicationDefinition
- CSS (Cascading Style Sheet)
- <DIV> Skeleton
- Jx JavaScript Functions

Flexible Layout Editor

- Layout editor for Fusion viewer

- Determine viewer's features and contents

- Does NOT determine the look and feel

- Generate ApplicationDefinition XML
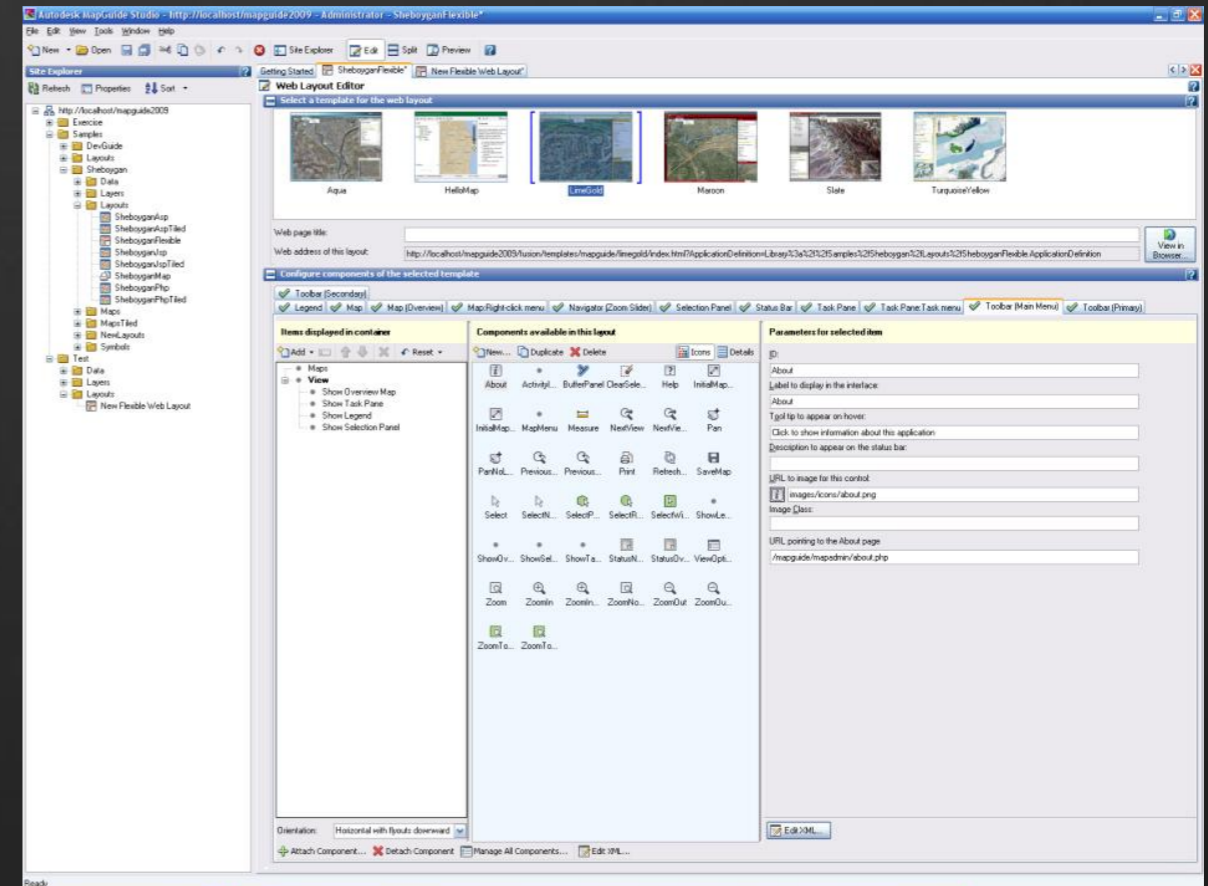
Map Tab

MapSet XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<MapSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MapGroupType id="MgMainMapId">
    <Map>
      <Type>MapGuide</Type>
      <SingleTile>true</SingleTile>
      <Extension>
        <ResourceId />
      </Extension>
    </Map>
    <Extension />
  </MapGroupType>
```

```xml
  <MapGroupType id="Sheboygan">
    <Map>
      <Type>MapGuide</Type>
      <SingleTile>true</SingleTile>
      <Extension>
        <ResourceId>Library://Samples/Sheboygan/Maps/Sheboygan.MapDefinition
        </ResourceId>
      </Extension>
    </Map>
    <Extension />
  </MapGroupType>
</MapSet>
```

Task Menu Tab

Task Menu XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<UiItemContainerType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Name>TaskMenu</Name>
  <Type>ContextMenu</Type>
  <Position>top</Position>
  <Extension />
  <Item xsi:type="WidgetItemType">
    <Function>Widget</Function>
    <Widget>BufferPanel</Widget>
  </Item>
  <Item xsi:type="WidgetItemType">
    <Function>Widget</Function>
    <Widget>Measure</Widget>
  </Item>
  <Item xsi:type="WidgetItemType">
    <Function>Widget</Function>
    <Widget>ListSelection</Widget>
  </Item>
</UiItemContainerType>
```

Widget Editor - ShowOverviewMap

ShowOverviewMap XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<UiWidgetType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Name>ShowOverviewMap</Name>
  <Type>InvokeScript</Type>
  <Location />
  <Extension>
    <Script>showOverviewMap()</Script>
    <Target />
  </Extension>
  <ImageUrl />
  <ImageClass />
  <Label>Show Overview Map</Label>
  <Tooltip>Display the overview map window</Tooltip>
  <StatusText />
  <Disabled>false</Disabled>
</UiWidgetType>
```

Autodesk

Widget Editor - ListSelection

List Selection XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<UiWidgetType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Name>ListSelection</Name>
  <Type>InvokeURL</Type>
  <Location />
  <Extension>
    <Url>ListSelectionTask.aspx</Url>
    <DisableIfSelectionEmpty>true</DisableIfSelectionEmpty>
    <Target>TaskPane</Target>
  </Extension>
  <ImageUrl>images/icons/invoke-url.png</ImageUrl>
  <ImageClass />
  <Label>List Selected Features</Label>
  <Tooltip>List Selected Features</Tooltip>
  <StatusText>List Selected Features</StatusText>
  <Disabled>false</Disabled>
</UiWidgetType>
```

ApplicationDefinition XML

- Aggregation of the earlier XMLs
- Determine viewer's features
- Determine viewer's contents
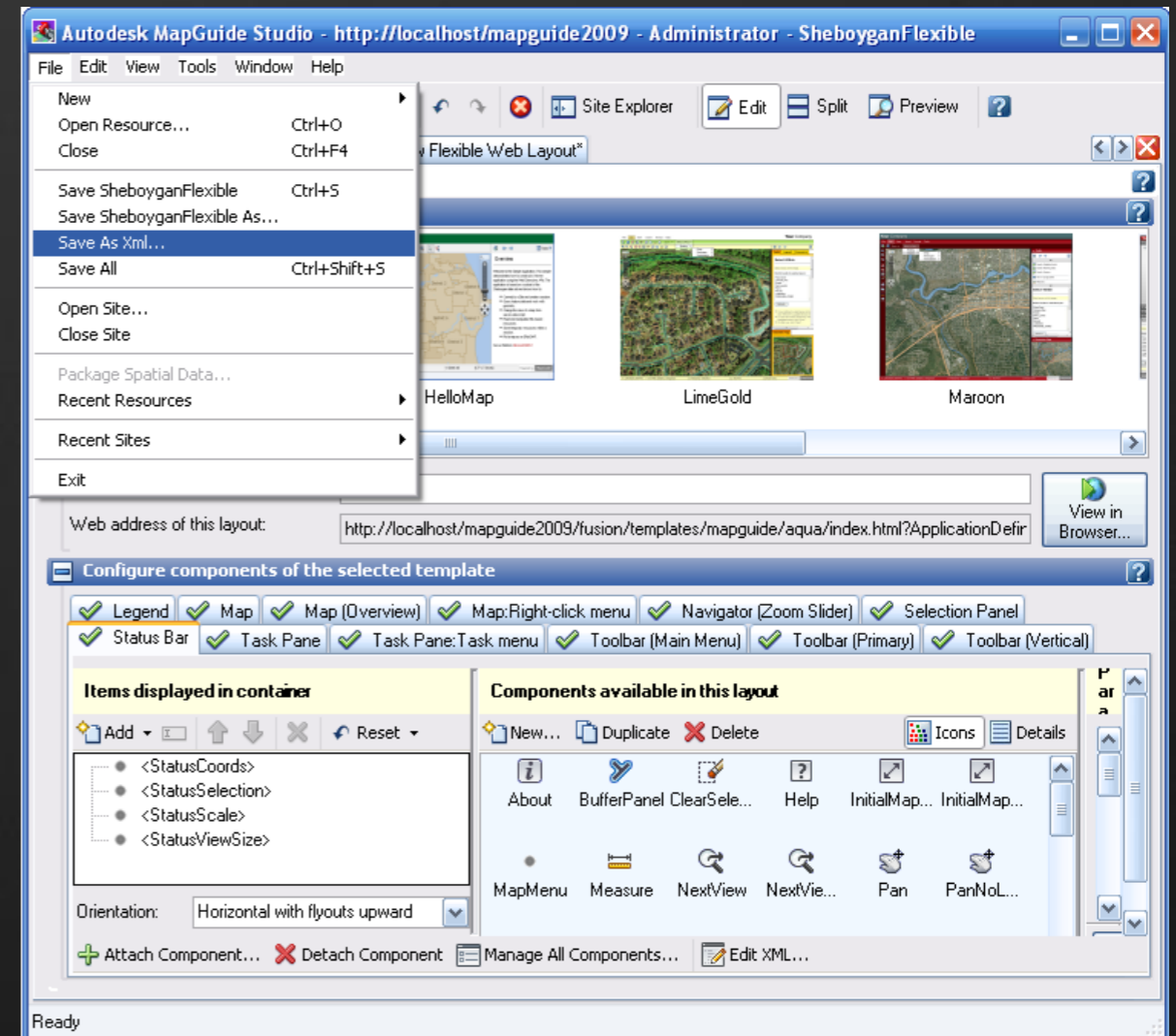
Major Fusion API Components

- ApplicationDefinition
- CSS (Cascading Style Sheet)
- <DIV> Skeleton
- Jx JavaScript Functions

CSS – Visual Style

- Color, font, style, etc.
- Cascading Style Sheet
- Change of CSS file changes the visual style

CSS Example

```
<style type="text/css">
  @import url(css/limegold_fusion.css);
</style>



  /********************************/

  #StatusbarContainer {
    border-top: 1px solid #000000;
    background: #fffebe;
    color: #ffffff;
    line-height: 22px;
    font-size: 10px;
    vertical-align: middle;
  }
```

Major Fusion API Components

- ApplicationDefinition
- CSS (Cascading Style Sheet)
- <DIV> Skeleton
- Jx JavaScript Functions

<DIV> Skeleton – Viewer's Anatomy

- Define containers
- Use HTML <DIV> to define the skeleton
- <DIV> order doesn't matter
- <DIV> replaced with actual HTML code during run time

\<DIV\> Skeleton

```
<body>
<div id="AppContainer">

  <div id="SplitterArea">
        <div id="Sidebar">
            <div id="SidebarCollapse"></div>
            <div id="Top"></div>
            <div id="Bottom"></div>
            <div id="Middle"></div>
         </div>
        <div id="Map">
            <div id="Navigator"></div>
        </div>
   </div>

   <div id="Toolbar"></div>
   <div id="TaskPane"></div>

   <div id="StatusbarContainer">
      <div id="Statusbar"></div>
   </div>
</div>
<div id="Maptip"></div>
</body>
```
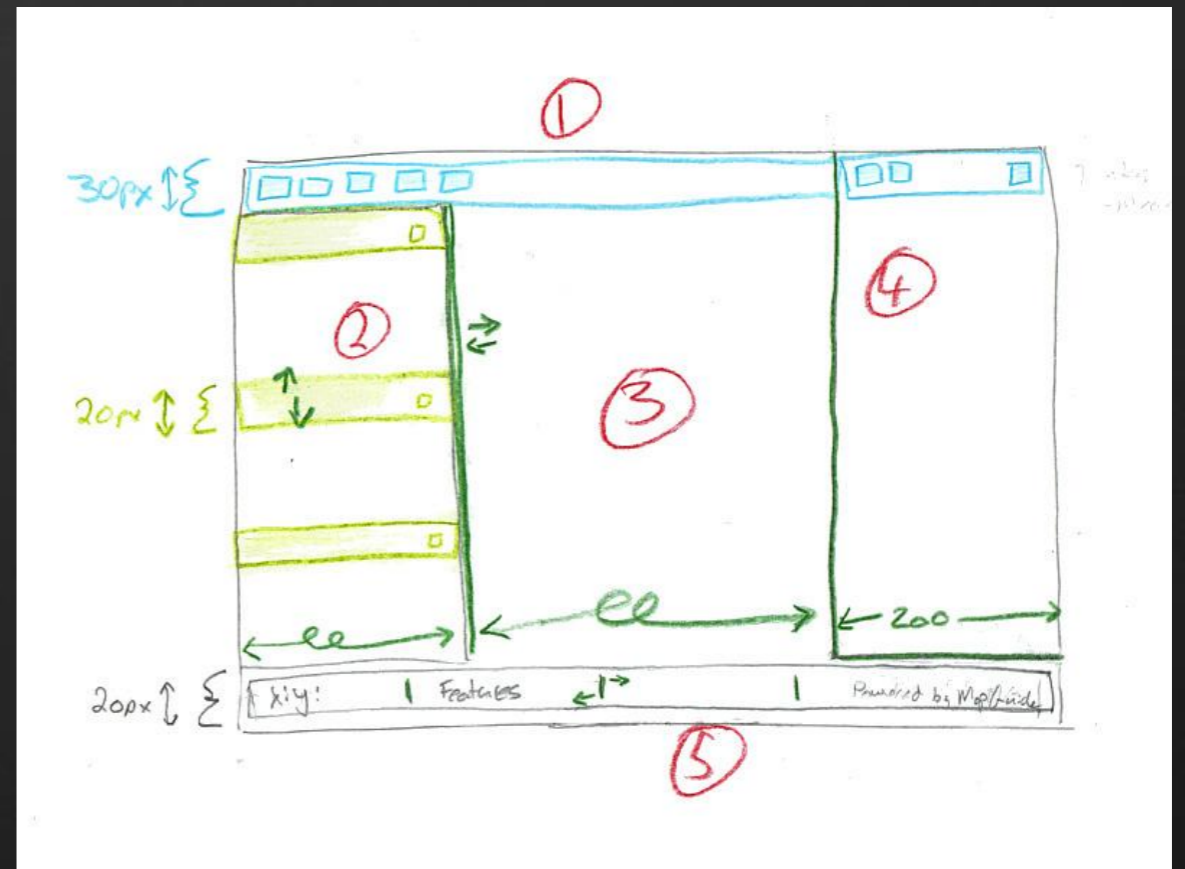
Autodesk

Major Fusion API Components

- ApplicationDefinition
- CSS (Cascading Style Sheet)
- <DIV> Skeleton
- Jx JavaScript Functions

Jx JavaScript –Layout Creation

- Jx JavaScript functions used to lay out the viewer
- Create containers, ie. panels, splitters, dialogs
- Initialize containers, ie. title, position, size
- Hook up containers with widgets in ApplicationDefinition
- Set up event handlers

Jx JavaScript Programming

1. Reference Fusion.js JavaScript library
2. Invoke layout creation in window.onload event
3. Create and initialize containers
4. Set up Fusion event handlers

Autodesk

Step 1: Reference Fusion.js

```html
<script type="text/javascript" src="../lib/fusion.js"></script>
```

Autodesk

Step 2: window.onload event

```
<script type="text/javascript">
   window.onload = function() {
        Fusion.initializeLocale();

        //…

        Fusion.initialize();
   }
</scrip>
```

Autodesk

Step 3: Create Layout

```
<script type="text/javascript">
window.onload = function() {
       Fusion.initializeLocale();


       var main = new Jx.Layout('AppContainer', {left: 20, right: 20, top: 20, bottom: 20});
       new Jx.Layout('Toolbar', {height: 25, left: 0, right: 0, top: 0, bottom: null});
       new Jx.Layout('Map', {width: null, height: null, left: 0, right: 0, top: 40, bottom: 0});

       dOverviewMap = new Jx.Dialog({id: 'dialogOverviewMap', imageBaseUrl: 'images/',
              title: OpenLayers.String.translate('ovmapTitle'),
              modal: false, resizeable: false,
              right: 50, bottom: 50, width: 200, height: 200,
              contentID: 'OverviewMap', helpID: '' } );


       dOverviewMap.show();



       Fusion.initialize();
}
</scrip>
```

Autodesk

Step 4: Set Up Events

```javascript
Fusion.registerForEvent(Fusion.Event.FUSION_INITIALIZED, fusionInitialized);
Fusion.registerForEvent(Fusion.Event.FUSION_ERROR, fusionError);

var fusionInitialized = function() {
    dTasks.hide();
    dTasks.domObj.style.right = '100px';
    dTasks.domObj.style.left = '';
    dLegend.hide();
    dLegend.domObj.style.left = '20px';
    dSelection.hide();
    dSelection.domObj.style.left = '100px';
    $('AppContainer').resize({forceResize: true});
}

var fusionError = function(eventId, error) {
    //console.log('Fusion Error: \n' + error.toString());
}
```

Autodesk

Demo

Questions

Questions ?