Application:	**Campus Building Viewer – JavaScript**
Version:	**3.2**
Document:	**Readme**
Date:	**February 21, 2013**
Author:	**CyberTech Systems and Software Ltd.**

Package Contents

CampusBuildingViewer : Folder containing web application and configuration file.
Readme.pdf : Deployment and configuration guide.

Configuration

You can configure Campus Building Viewer Template in your environment. To complete the configuration, you will need experience with Microsoft's Internet Information Server (IIS). If you are new to JavaScript Viewers, this template will demonstrate a pattern you can use to deploy your own JavaScript application.

1. Copy the "CampusBuildingViewer" directory onto your web server so that it can be accessed as a website or virtual directory.

   Example: Copy the "CampusBuildingViewer" directory under C:\Inetpub\wwwroot for Microsoft IIS web servers.

2. This application uses a proxy file provided by ESRI. The proxy file is available in three different languages (ASP.NET, PHP and JSP). Current application uses ASP.NET proxy file. If you wish to use a different proxy file please click here. For ASP.NET proxy file, change the REST end point to the ArcGIS REST service URL in the proxy.config file.
3. Go to IIS, right-click on the parent directory of the copied files, select "Convert To Application" and set the application pool to ASP.NET v4.0.

   Note: If you want to use a different version of ASP.NET, please follow the steps below.
   1. Open the source folder as a website in Visual Studio.
   2. Right-click on website, go to properties and change the .NET target framework to desired version.
   3. Publish the website.
4. The "CampusBuildingViewer" directory contains config.js which is the main configuration file. To modify any configuration values, open config.js file using a text editor like Notepad. Make necessary changes to the JSON objects. See the "Description of Configuration Tags" section below for more information.
5. Test the application in a browser by entering the URL to the default.htm page. Example: http://<server>/CampusBuildingViewer/default.htm

   Substitute "<server>" with name of your server. Please note: "default.htm" may not be defined as a default document on your web server.

Description of Configuration Tags (config.js)

ApplicationName: Application name to be displayed in the application header.

```
ApplicationName: "Campus Place Finder",
```

ApplicationImage: Application icon to be displayed in the application header.

```
ApplicationImage: "images/Finder.png",
```

SplashScreenMessage: Application start splash screen message.

```
SplashScreenMessage: "<b>Welcome to the Campus Place Finder</b><hr/>The Campus Place
Finder application helps employees and guests locate people, offices, conference rooms, and
spaces as well as obtain information about those people and places. To locate a person or place,
simply click on the map or select what you are searching for; person or place, then enter who or
what you are looking for in the search box. The person’s office or space  will then be highlighted on
the map and relevant information about the person or space will be displayed in the application.",
```

HelpURL: Path for help file.

```
HelpURL: "help.htm",
```

DefaultExtent: Initial map extent. Use comma (,) to separate values and don't delete the last comma

```
DefaultExtent: "-13046368.774366917,4036413.338302078,-
13046063.02625387,4036524.7094409126",
```

GeometryService: Geometry service URL

```
GeometryService:
“http://localgovtemplates2.esri.com/ArcGIS/rest/services/Geometry/GeometryServer",
```

QueryTaskURL: URL for querying total buildings and floors

```
QueryTaskURL: {
 QueryURL:
"http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapS
erver/0",
BuildingKey: "${BUILDINGKEY}",
Floor: "${FLOOR}"
 },
```

BaseMapLayers: Set baseMap layers
Please note: All base maps need to use the same spatial reference. By default, on application start the first basemap will be loaded

- Key: Key for every map. This has to be unique in this collection.
- ThumbnailSource: Source path of the image.
- Name: Name of the layer.

- MapURL: URL of the Map layer.

```
BaseMapLayers: [{
    Key: "worldTopoMap",
    ThumbnailSource: "images/TopoCampus.png",
    Name: "Topo + Campus",
    MapURL: [{
      LayerId: "worldTopoMap",
      MapURL:
"http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer"
    }, {
      LayerId: "campusMap",
      MapURL: "http://arcgis-tenone2012-1974758903.us-west-
1.elb.amazonaws.com/arcgis/rest/services/Campus/MapServer"
    }]
  }, {
    Key: "worldImageryMap",
    ThumbnailSource: "images/imagery.jpg",
    Name: "Imagery",
    MapURL: [{
      LayerId: "worldImageryMap",
      MapURL: "http://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer"
    }]
  }],
```

ServiceRequest: Set configuration fields for Service Request layer

```
 ServiceRequest: {
    Instructions: "Please enter an address above or click directly on the map to locate the area
you'd like to report. Fill out the form below and click Submit to initiate your service request. A
service request number will be issued immediately, please take note of this number in order to
track the status of your request. <br /> <br/> <b>Contacting Us by Phone</b> <br/> <hr/> <br/>
Customer Service <br/> Hours: 8 am - 4 pm<br/> 555-555-1212 <br/>",
    isEnabled: true,
    LayerInfo: {
      Key: "srFloor0",
      ServiceURL:
"http://203.199.47.146/arcgis/rest/services/ServiceRequest/ServiceRequestCPF/FeatureServer/0",
      WhereQuery: "floor = '${0}' AND status = 'Unassigned' AND (building = '${1}' or building =
'outside')",
      OutFields: "*",
      RequestId: "${requestid}",
      ShareFields: "${requestid}",
      ShareQuery: "objectid = '${0}'",
      Building: "${building}",
      Floor: "${floor}",
      RequestTypeFieldName: "requesttype",
      CommentsLayerURL:
"http://203.199.47.146/arcgis/rest/services/ServiceRequest/ServiceRequestCPF/FeatureServer/4",
      CommentsOutFields: "*",
      CommentId: "${requestid}",
      BuildingFloorPlan:
"http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapS
erver/2",
      BuildingAttribute: "BUILDINGKEY",
      BuildingKey: "${BUILDINGKEY}"
    },

    InfoPopupFieldsCollection: [{
      DisplayText: "Name:",
      FieldName: "${name}",
      HideCondition: true
    }, {
      DisplayText: "Phone:",
      FieldName: "${phone}",
      HideCondition: true
    }, {
      DisplayText: "Email:",
      Email: true,
      FieldName: "${email}",
      HideCondition: true
    }, {
      DisplayText: "Place:",
      FieldName: "${building}-${floor}",
      DataType: "string"
    }, {
      DisplayText: "Type:",
      FieldName: "${requesttype}",
      DataType: "string"
    }, {
      DisplayText: "Description:",
      FieldName: "${comments}",
      DataType: "description",
      id: "comments"
    }, {
      DisplayText: "Date Submitted:",
      FieldName: "${requestdate}",
      DataType: "date",
      FormatDateAs: "MMM dd, yyyy"
    }]
  },
```

OperationalLayers: Feature layer URL for getting feature details (Configure operational layers)

- Name:Name of the layer.
- Key: Key for every operational layer. This has to be unique in this collection.
- MapURL: URL of the layer.
- DateFields: Field for the Date values.
- MapURL: The URL on which the query task is performed.
- ShareQuery: This field is used when sharing is done to collect the Space id.
- OutFields: this is used when performing the query task to specify the output fields.

InfoPopupFieldsCollection: Info-popup is a popup dialog that gets displayed on selecting a feature Configured as

- key: value pairs. Please note that keys should be unique.
- Title: Fields which will be shown in header of the info window.
- isLayerVisible: This value can be true or false which defines weather the operational will be visible at the moment project is initialized.
- WhereQuery: Condition based on which the operational layer will be queried and layer selection will be rendered.
- BuildingAttribute: Layers among the collection which has “buildingAttribute” value set as “BUILDING” will be displaying the layer info with infoWindow.

```
  OperationalLayers: [{
    Name: "Building Interior Spaces Type",
    Key: "BuildingInteriorSpacesType",
    MapURL:
"http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapServer/1",
    Building: "${BUILDING}",
    Floor: "${FLOOR}",
    ObjectID: "${OBJECTID}",
    isLayerVisible: true,
    WhereQuery: "BUILDING = '${0}' AND FLOOR = '${1}'",
    RelationshipQuery: "SPACEID = '${LOCATION}'",
    ShareQuery: "SPACEID='${0}'",
    ShareFields: "${SPACEID}",
    SpaceType: "SPACETYPE",
    SpaceID: "SPACEID",
    BuildingAttribute: "BUILDING",
    hasDynamicMapService: true
  }, {
    Name: "Building Floorplan Lines",
    Key: "BuildingFloorplanLines",
    MapURL:
"http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapServer/0",
    isLayerVisible: true,
    WhereQuery: "BUILDINGKEY = '${0}' AND FLOOR = '${1}'"
  }],
```

PlaceLayer: Set layer details for searching different space types

```
  PlaceLayer: {
      Key: "PlaceLayer",
      QueryURL:
"http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapServer/1",
      OutFields: "*",
      QueryFields: "BUILDING,FLOOR,SPACEID",
      SpaceType: "${SPACETYPE}",
      SpaceID: "SPACEID",
      DateFields: [{
         DisplayField: "Last Update Date",
         ValueField: "LASTUPDATE",
         AliasField: "Last Update Date"
      }],
      InfoPopupFieldsCollection: [{
         DisplayText: "Name:",
         FieldName: "${FIRSTNAME} ${LASTNAME}",
         HideCondition: true
      }, {
         DisplayText: "Email:",
         FieldName: "${EMAIL}",
         isLink: true,
         HideCondition: true
      }, {
         DisplayText: "Phone:",
         FieldName: "${EXTENSION}",
         HideCondition: true
      }, {
         DisplayText: "Building:",
         FieldName: "${BUILDING}"
      }, {
         DisplayText: "Floor:",
         FieldName: "${FLOOR}"
      }, {
         DisplayText: "Wing:",
         FieldName: "${WING}"
      }],

      Title: "${SPACETYPE}: ${SPACEID}"
  },
```

PersonLayer: Set layer details for searching a person

```
PersonLayer: {
      Key: "PersonLayer",
  QueryURL: "http://203.199.47.146/arcgis/rest/services/CampusPlaceFinder/BuildingInterior_norelation/MapServer/4",
      OutFields: "OBJECTID,FIRSTNAME,LASTNAME,EMAIL,EXTENSION,BUILDING,FLOOR,WING",
      QueryFields: "FIRSTNAME,LASTNAME",
      WhereQuery: "LOCATION = '${SPACEID}'",
      FirstName: "FIRSTNAME",
      LastName: "LASTNAME",
      DateFields: [{
        DisplayField: "Last Update Date",
        ValueField: "LASTUPDATE",
        AliasField: "Last Update Date"
      }],
      InfoPopupFieldsCollection: [{
        DisplayText: "Name:",
        FieldName: "${FIRSTNAME} ${LASTNAME}",
        HideCondition: true
      }, {
        DisplayText: "Email:",
        FieldName: "${EMAIL}",
        isLink: true,
        HideCondition: true
      }, {
        DisplayText: "Phone:",
        FieldName: "${EXTENSION}",
        HideCondition: true
      }, {
        DisplayText: "Building:",
        FieldName: "${BUILDING}"
      }, {
        DisplayText: "Floor:",
        FieldName: "${FLOOR}"
      }, {
        DisplayText: "Wing:",
        FieldName: "${WING}"
      }],

      Title: "${SPACETYPE}: ${SPACEID}"

    },
```

DefaultBuilding: Set default building selected when map is loaded

```
DefaultBuilding: 'M',
```

DefaultFloor: Set default floor selected within the selected building

```
DefaultFloor: '1',
```

PersonText: Set default watermark text displayed in search box for person search

```
PersonText: "Enter first name or last name or both",
```

PlaceText: Set default watermark text displayed in search box for place search

```
PlaceText: "Enter name of place e.g. MA",
```

ShowNullValueAs: Set string value to be displayed for null or blank values

```
ShowNullValueAs: "N/A",
```

InfoPopupHeight,InfoPopupWidth: Set size of the info-Popup - select maximum height and width in pixels(minimum height should be 200 for the info-popup in pixels), (minimum width should be 330 for the info-popup in pixels)

```
InfoPopupHeight: 275,
InfoPopupWidth: 330,
```

DefaultSearch: Set default search

Please note: Both values should not be same

- Person: If the default search should be person set it as true else false.
- Place: If default search should be Place set it as true else false.

```
DefaultSearch:
  {
     Person: true,
     Place: false
  },
```

FloorSwitcher: Set floor switcher

```
  FloorSwitcher:
  {
     IsExpressVisible: true,
     IsAccordionVisible: true
  },
```

DatabaseFields: Define the database field names

```
DatabaseFields: {
     RequestIdFieldName: "requestid",
     CommentsFieldName: "comments",
     DateFieldName: "submitdt",
     RankFieldName: "rank"
  },
```

ServiceRequestFields: Define service request input fields for submitting a new request

```
ServiceRequestFields: {
     RequestTypeFieldName: "requesttype",
     CommentsFieldName: "comments",
     NameFieldName: "name",
     PhoneFieldName: "phone",
     EmailFieldName: "email",
     StatusFieldName: "status",
     RequestDateFieldName: "requestdate",
     BuildingFieldName: "building",
     FloorFieldName: "floor",
     RequestIdFieldName: "requestid"
  },
```

BuildingFloorFields: Define building and floor input fields for getting the exact location on map click

```
 BuildingFloorFields: {
      BuildingFieldName: "building",
      FloorFieldName: "floor",
      SpaceTypeFieldName: "SPACETYPE",
      SpaceIdFieldName: "SPACEID",
      SectionFieldName: "SECTION"
   },
```

CommentsInfoPopupFieldsCollection: Set info-pop fields for adding and displaying comment for existing service request

- Rank: Rank field for the comments field is set here.
- SubmitDate: Submit date for comments field is set here.
- Comments: Comments field is set here

```
 CommentsInfoPopupFieldsCollection: {
      Rank: "${rank}",
      SubmitDate: "${submitdt}",
      Comments: "${comments}"
   },
```

MapSharingOptions:Set URL for TinyURL service, and URLs for social media

- TinyURLServiceURL: used for converting the long URL to short URL
- TinyURLResponseAttribute: It is a response for the URL for converting long URL to short URL and it behaves as a key for that URL.
- FacebookShareURL: Social networking site URL

```
 MapSharingOptions:
        {
           TinyURLServiceURL:
"http://api.bit.ly/v3/shorten?login=esri&apiKey=R_65fd9891cd882e2a96b99d4bda1be00e&uri=${0}&format=json",
           TinyURLResponseAttribute: "data.url",
           FacebookShareURL:
"http://www.facebook.com/sharer.php?u=${0}&t=Campus%20Building%20Viewer",
           TwitterShareURL: "http://twitter.com/home/?status=Campus%20Building%20Viewer ${0}",
           ShareByMailLink: "mailto:%20?subject=Checkout%20this%20map!&body=${0}"
    }
    });
```