Application: **Park and Recreation Finder – HTML5**
Version: **3.2**
Document: **Readme**
Date: **February 28, 2013**
Author: **CyberTech Systems and Software Ltd.**

Description

Park and Recreation Finder is a configuration of ArcGIS Server and a HTML5 application that allows citizens to locate park and recreation opportunities in their community. This application is typically used by citizens to locate a park in their community that offers a recreation activity they are interested in, but can also be used by visitors to locate a park in a community they are visiting.

Package Contents

Park and Recreation Finder: Folder containing web application and configuration file
Readme.pdf: Deployment and configuration guide

Minimum System Requirements

- ***Application Web Server***
  - P-IV with 1 GB RAM and 40 GB Hard Disk
  - IIS 5.0 or higher
  - Win 2K Server or above
- ***GIS Server***
  - P-IV with 1 GB RAM and 40 GB Hard Disk
  - IIS 5.0 or higher
  - ArcGIS Desktop 10.1, ArcGIS Server 10.1 and ArcSDE 10.1
  - .NET framework 3.5 with Service Pack 1
- ***Network Requirements***
  - LAN connectivity
  - Broadband internet connection
- ***Client Requirements***
  - Web browser with JavaScript enabled
- ***Screen Resolution***
  - 1024 x 768 high color, 32-bit or higher

Configuration

You can configure Park and Recreation Finder 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 HTML application.

1. Copy the “Park and Recreation Finder” directory onto your web server so that it can be accessed as a website or virtual directory.

   Example: Copy the “Park and Recreation Finder” 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.
   - Open the source folder as a website in Visual Studio
   - Right-click on website, go to properties and change the .NET target framework to desired version
   - Publish the website
4. The “Park and Recreation Finder” 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>/ParkandRecreationFinder/default.htm
   Substitute "<server>" tag 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: Set application title
ApplicationName: "Park and Recreation Finder",

ApplicationIcon: Set application icon path
ApplicationIcon: "images/appIcon.png",

SplashScreenMessage: Set splash window content - Message that appears when the application starts
SplashScreenMessage: " Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.",

HelpURL: Set URL of help page/portal
HelpURL: "help.htm",

BaseMapLayers: Set baseMap layers

Please note: All base-maps need to use the same spatial reference. By default, on application start the first base-map 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: "topoMap",
    ThumbnailSource: "images/topographic.jpg",
    Name: "Topographic Map",
    MapURL: "http://services.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer"
    }, {
    Key: "parcelMap",
    ThumbnailSource: "images/parcel.png",
    Name: "Parcel Map",
    MapURL: "http://localgovtemplates.esri.com/ArcGIS/rest/services/ParcelPublicAccess/MapServer"
  }],
```

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

```
DefaultExtent: "-9820183.18, 5123332.08, -9807373.74, 5128739.76",
```

DevPlanLayer: URL used for doing query task on the parks layer ,
For parks photos, recommended image format is PNG and each image size should be maximum up to 700 KB.

```
DevPlanLayer: "http://arcgis-two-1334003536.us-west-1.elb.amazonaws.com/arcgis/rest/services/Parks/FeatureServer/0",
```

PrimaryKeyForParks: Set the primary key attribute for parks

```
PrimaryKeyForParks: "${FACILITYID}",
```

ParkCommentsLayer: URL used for doing query task on the comments layer

```
ParkCommentsLayer: "http://arcgis-two-1334003536.us-west-1.elb.amazonaws.com/arcgis/rest/services/Parks/FeatureServer/1",
```

PrimaryKeyForComments: Set the primary key attribute for comments

```
PrimaryKeyForComments: "${FACILITYID}",
```

ParkName: Set the name attribute for parks

```
ParkName: "${NAME}",
```

ReferenceOverlayLayer: Set the URL for the overlay layer

- ServiceUrl: ServiceUrl is the REST end point for the reference overlay layer
- DisplayOnLoad: DisplayOnLoad setting is used to show or hide the reference overlay layer. Reference overlay will be shown when it is set to true

```
ReferenceOverlayLayer:
        {
          ServiceUrl: " http://arcgis-two-1334003536.us-west-
1.elb.amazonaws.com/arcgis/rest/services/TrailsOnlyDynamic/MapServer",
          DisplayOnLoad: true
        },
```

InfoWindowHeader : Set Info-window title. Configure this with text/fields

```
InfoWindowHeader: [{
    FieldName: "${NAME}", Alias:
    "Park Name"
  }],
```

InfoWindowContent: Choose content/fields for the infowindow

```
InfoWindowContent: [{
    FieldName: "${FULLADDR}", Alias:
    "Full Address"
  }],
```

InfoPopupFieldsCollection: Set the content to be displayed on the info-Popup. Define labels, field values, field types and field formats

- DisplayText: Field used for displaying the Text instead of alias names
- FieldName: Field used for getting the details of the particular service feature
- Alias: Alias name of that particular field

```
InfoPopupFieldsCollection: [{
    DisplayText: "Address:",
    FieldName: "${FULLADDR}", Alias:
    "Full Address"
  }, {
    DisplayText: "Days Open:",
    FieldName: "${OPERDAYS}", Alias:
    "Operational Days"
  }, {
    DisplayText: "Hours of Operation:",
    FieldName: "${OPERHOURS}", Alias:
    "Operational Hours"
  }, {
    DisplayText: "Parking Spaces Available:",
    FieldName: "${PARKURL}",
    Alias: "Number of Parking Spaces"
  }
  ],
```

Activities: Activities to be displayed in info window for a park

- FieldName: Field used for getting the details of the particular service feature
- Alias: Alias name of that particular field

- Image: Image used to represent that particular field
- Name: field used to show the name of the Activity

```
Activities: [{
    FieldName: "${RESTROOM}",
    Alias: "Restrooms Available", Image: "images/restrooms.png", isSelected: true
  }, {
    FieldName: "${ADACOMPLY}", Alias: "ADA Compliant",
    Image: "images/ada compliant.png"
  }, {
    FieldName: "${SWIMMING}", Alias: "Swimming",
    Image: "images/swimming.png"
  }, {
    FieldName: "${HIKING}", Alias: "Hiking",
    Image: "images/hiking.png"
  }, {
    FieldName: "${FISHING}", Alias: "Fishing",
    Image: "images/fishing.png"
  }, {
    FieldName: "${PICNIC}",
    Alias: "Picnic Shelters", Image: "images/picnic.png"
  }, {
    FieldName: "${BOATING}", Alias: "Boating",
    Image: "images/boating.png"
  }, {
    FieldName: "${ROADCYCLE}", Alias: "Road Cycling",
    Image: "images/cycling.png"
  }, {
    FieldName: "${MTBCYCLE}", Alias: "Mountain Biking", Image:
    "images/mtb.png"
    },
    {
    FieldName: "${PLAYGROUND}",
    Alias: "Playgrounds",
    Image: "images/playground.png",
    Name: "PLAYGROUND"
  },
  {
    FieldName: "${SKI}",
    Alias: "Skiing",
    Image: "images/skiing.png"
  }, {
    FieldName: "${SOCCER}", Alias:
    "Multi-Purpose Fields", Image:
    "images/soccer.png"
  }, {
    FieldName: "${CAMPING}",
```

```
    Alias: "Camping",
    Image: "images/camping.png"
  }, {
    FieldName: "${HUNTING}",
    Alias: "Hunting",
    Image: "images/hunting.png"
  }, {
    FieldName: "${BASEBALL}",
    Alias: "Baseball Fields", Image:
    "images/baseball.png”
  }, {
    FieldName: "${BASKETBALL}", Alias:
    "Basketball Courts", Image:
    "images/basketball.png”
  }],
```

Set size of the info-Popup - select maximum height and width in pixels (not applicable for tabbed info-Popup)

```
InfoPopupHeight: 270,
InfoPopupWidth: 330,
```

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

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

FormatDateAs : Set date format

```
FormatDateAs: "MMM dd, yyyy",
```

BufferDistance: set distance in miles for drawing the buffer

```
BufferDistance: "1",
```

BufferColor : set buffer color for address search

```
BufferColor: [0, 100, 0],
```

LocatorRippleSize: Set the locator ripple size

```
LocatorRippleSize: 30,
```

GetDirectionsMobile: Set this variable to true/false to enable/disable directions for Mobile/tablet

```
GetDirectionsMobile: true,
```

GetDirectionsDesktop: Set this variable to true/false to enable/disable directions for desktop

```
GetDirectionsDesktop: true,
```

GetDirections: Set this variable to true/false to enable/disable directions

Note: if this master variable is set to false directions cannot be enabled for any of the devices

```
 GetDirections: true,
```

LocatorSettings: Set locator settings such as locator symbol, size, display fields, match score

```
LocatorSettings: {
      DefaultLocatorSymbol: "images/RedPushpin.png",
      MarkupSymbolSize: {
         width: 35,
         height: 35
      },
       ZoomLevel:12,
      Locators: [{
         DisplayText: "Location",
         LocatorDefaultAddress: "971 sylvan cir Naperville IL 60540",
         LocatorParamaters: ["SingleLine"],
         LocatorURL:
"http://tasks.arcgisonline.com/ArcGIS/rest/services/Locators/TA_Address_NA_10/GeocodeServer",
         CandidateFields: "Loc_name, Score, Match_addr",
         DisplayField: "${Match_addr}", AddressMatchScore:
         80,
         LocatorFieldName: "Loc_name",
         LocatorFieldValues: ["US_Streets", "US_StreetName"] }, {
         DisplayText: "Name",
         LocatorDefaultPark: "Knoch Park"
      }, {
         DisplayText: "Activity"
      }]
   },
```

DatabaseFields: Define the database field names, DateFieldName refers to a date database field. All other attributes refer to text database fields.

```
DatabaseFields: {
          ParkIdFieldName: "FACILITYID",
          CommentsFieldName: "COMMENTS",
          DateFieldName: "SUBMITDT",
          RankFieldName: "RANK"
      },
```

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

```
CommentsInfoPopupFieldsCollection: {
      Rank: "${RANK}",
      SubmitDate: "${SUBMITDT}",
      Comments: "${COMMENTS}"
   },
```

GeometryService : Set geometry service URL

```
GeometryService: "http://tasks.arcgisonline.com/ArcGIS/rest/services/Geometry/GeometryServer",
```

RouteServiceURL : Set URL for routing service

```
RouteServiceURL:
"http://tasks.arcgisonline.com/ArcGIS/rest/services/NetworkAnalysis/ESRI_Route_NA/NAServer/Route"
,
```

RouteColor: Set color for the route symbol

```
RouteColor: "#7F7FFE",
```

RouteWidth: Set width of the route

```
RouteWidth: 6,
```

RippleColor: Ripple color for selected feature.

```
RippleColor: [60, 72, 36],
```

InfoBoxWidth: Set width of the boxes in the bottom panel

```
InfoBoxWidth: 422,
```

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=Parks%20and%20Recreation%20Finder",
        TwitterShareURL:
    "http://twitter.com/home/?status=Parks%20and%20Recreation%20Finder ${0}",
        ShareByMailLink: "mailto:%20?subject=Checkout%20this%20map!&body=${0}"
      },
```

Order: Set sequence for info pods in the bottom panel

```
Order: ["search", "park", "directions", "photogallery", "comments"]
});
```