AC - Quick Integration Guide - Address Predict

Integration Introduction

Integrating our DOTS Global Address Complete solution can now be completed with anyone familiar with HTML and JavaScript. We make the integration process so easy that almost anyone will be able to get it up and running in little time. Our solution is designed for web based technologies that can consume our JavaScript helper file. We have simplified the integration process down to three easy steps: Include our JavaScript and CSS in your application, field mapping and customization by setting options. This integration guide will take you through a typical HTML web form integration.

Step 1 - Include Files

The first thing to be done on the page you want to integrate Global Address Complete is to make a reference to the JavaScript and CSS files. The JavaScript will handle all the things that should happen under the hood of the field that the Global Address Complete is connected to for the type ahead solution. And, the CSS file will help with the formatting of the field. The CSS file is just as critical as the JavaScript file for integrating this solution.

Starting with the CSS file ACStyle.css, add it to the page using the link tag in the head section.

HTML - Include CSS

```
<head>
    <link rel="stylesheet" type="text/css" href="https://trial.
serviceobjects.com/Resources/AC/CSS/ACStyle.css" />
    .
        .
        .
</head>
```

For the JavaScript file ACSvriptV1.02.js, add it using the script tag at the top inside the body.

HTML - Include JavaScript

```
<body>
    <script type="text/javascript" src="https://trial.serviceobjects.com
/Resources/AC/JS/ACScriptV1.02.js"></script>
        .
        .
        .
```

Step 2 - Map Fields

Step 2 involves mapping the fields on the form to the fields in the JavaScript file that we added in the previous step. In this example, I will be adding the mapping inside the script tag on the same HTML page that we have been working on but you are free to create a separate JavaScript file and add the mapping code there. If you do add it to a new JavaScript page, be sure to reference it in the HTML page the same way we did in step 1 when we referenced the ACScriptV1.02.js page.

Here is an example of an HTML section with the input fields on a typical web form. Make note of the ID values on the input tags because those will be used in the mapping part of the script tag.

Input Fields to Map

```
<div id="sAddress" class="AddressInputBlock">
        <div id="lAddress2" class="AddressLabels">Address One</div>
        <input id="iAddress1" type="text" class="AddressInputs" />
</div>
<div id="sAddress2" class="AddressInputBlock">
        <div id="lAddress2" class="AddressLabels">Address Two</div>
        <input id="iAddress2" type="text" class="AddressInputs" />
</div>
<div id="sAddress3" class="AddressInputBlock">
        <div id="lAddress3" class="AddressLabels">Address Three</div>
        <input id="iAddress3" type="text" class="AddressInputs" />
</div>
<div id="sUnits" class="AddressInputBlock">
        <div id="lUnits" class="AddressLabels">Unit</div>
        <input id="iUnits" type="text" class="AddressInputs" />
</div>
<div id="sLocality" class="AddressInputBlock">
        <div id="lLocality" class="AddressLabels">Locality</div>
        <input id="iLocality" type="text" class="AddressInputs" />
</div>
<div id="sAdminArea" class="AddressInputBlock">
        <div id="lAdminArea" class="AddressLabels">Administrative Area<
/div>
        <input id="iAdminArea" type="text" class="AddressInputs" />
</div>
<div id="sPostal" class="AddressInputBlock">
        <div id="lPostal" class="AddressLabels">Postal Code</div>
        <input id="iPostal" type="text" class="AddressInputs" />
</div>
<div id="sCountry" class="AddressInputBlock">
        <div id="lCountry" class="AddressLabels">Country</div>
        <input id="iCountry" type="text" class="AddressInputs" />
</div>
```

Now that we have made note of the iAddress1, iAddress2, iAddress3, iUnits, iLocality, iAdminArea, iPostal, and iCountry ID's of the inputs we are ready to map. In the script tag, add the following code. Here we are mapping the iAddress1 input field to the searching mechanism in the referenced JavaScript. This is where the address suggestions will be displayed. All of the other fields are set to POPULATE so upon selection of an address those fields will be filled with the result automatically.

Field Mapping

```
<script>
     var fields = [
                                  { element: "iAddress1", field: "Address1",
mode: so.fieldMode.SEARCH | so.fieldMode.POPULATE },
                                  { element: "iAddress2", field: "Address2",
mode: so.fieldMode.POPULATE },
                                  { element: "iAddress3", field: "Address3",
mode: so.fieldMode.POPULATE },
                                  { element: "iUnits", field: "SubPremise",
mode: so.fieldMode.POPULATE },
                                  { element: "iLocality", field: "Locality",
mode: so.fieldMode.POPULATE },
                                  { element: "iAdminArea", field:
"AdminArea", mode: so.fieldMode.POPULATE },
                                  { element: "iPostal", field: "PostalCode",
mode: so.fieldMode.POPULATE },
                                  { element: "iCountry", field: "Country",
mode: so.fieldMode.POPULATE }
                             ];
</script>
```

Step 3 - Set Options: (A) Address Search Type

In the final step is where you can customize how Global Address Complete works for your solution. Here will will continue where we left off in the last step and add the next lines of code after the ending bracket of the fields assignment but still inside the script tag. If using the Address Search Type, continue with set up (A). If using the Locality Search Type, continue with set up (B).

To complete the setup, you will need to have a Custom Key and you will need to know if you are using trial key or live key. We will pass the Custom Key to the options variable and if we are using a trial key no additional options need to be set. If you were setting up for a live implementation adding isTrial: false would be needed in the options variable so the entire version of what I have below would look like this in that situation: var options = { key: CustomKey, isTrial: false, setBiasToIP: true };. No other settings need to be made, however, in this example I added setBiasToIP and set it true. This just means that the service is going to be looking for addresses to suggest starting from your IP address location.

Setting Options

```
var fields = [
                                { element: "iAddress1", field: "Address1",
mode: so.fieldMode.SEARCH | so.fieldMode.POPULATE },
                                .
                                .
                                .
                                { element: "iCountry", field: "Country",
mode: so.fieldMode.POPULATE }
                           ];


var CustomKey = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
var options = { key: CustomKey, setBiasToIP: true };
var DOTSGlobalAddressComplete = new so.Address(fields, options);
```

*Note, the true/false values are boolean values and not strings.

Step 3 - Set Options: (B) Locality Search Type

Set Options: (B) is just a bit of a helper so you can see the difference side by side between the address prediction and place prediction. The guide Quick Integration Guide - Place Predict will be exclusively about this functionality.

Setting Options

```
var fields = [
                                { element: "iAddress1", field: "Address1",
mode: so.fieldMode.SEARCH | so.fieldMode.POPULATE },
                                .
                                .
                                .
                                { element: "iCountry", field: "Country",
mode: so.fieldMode.POPULATE }
                           ];


var CustomKey = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
var options = { key: CustomKey, setBiasToIP: true, DoPlace: true,
SearchType: "Locality" };
var DOTSGlobalAddressComplete = new so.Address(fields, options);
```

*Note, the true/false values are boolean values and not strings.

Once you have the options set your form will now be activated to suggest addresses (or localities) as you type.