provenworks™

How Can I Add AddressTools Functionality To My VisualForce Page?

This guide will cover:

- ☑ Adding the essential code to the VisualForce page
- ☑ Setting up the address fields
- ☑ Adding the fields to the AddressTools setup
- ☑ Creating an AddressTools trigger for the VisualForce page
- ☑ An example VisualForce page

Note: This guide assumes you are familiar with VisualForce and are using the Premium version of AddressTools, if you are not using Premium and are interested in finding out more, you can trial it for free at http://sforce.co/1DfEoWP.

Adding the essential code to your VisualForce page.

1. The following statements need to be added at the top of your VisualForce page just inside the apex:page tag.

```
<apex:includeScript value="/soap/ajax/27.0/connection.js" />
<apex:includeScript value="/soap/ajax/27.0/apex.js" />
<apex:includeScript value="/apex/pw_ccpro__CountriesJavaScript?core.apexpages.devmode.url=1" />
<apex:includeScript value="{!URLFOR($Resource.pw_ccpro__CountryCompleteResources, '/javascript/CountryAutoComplete.js')}" />
<apex:includeScript value="{!URLFOR($Resource.pw_ccpro__AddressCompleteResources, '/javascript/AddressComplete.js')}" />
```

You must now choose **one** of following two approaches to execute the AddressTools Javascript on your VisualForce page.

Option A) Add a new <script> block to your code as follows:

```
<script type="text/javascript">
sforce.connection.sessionId = '{!$Api.Session_ID}';
</script>
```

Option B) Add the same statement to the page's existing document.onload() function:

```
<script type="text/javascript">
 // Other JS here
 window.document.onload = new function(e)
 {
 // Other initialisation here
 sforce.connection.sessionId = '{!$Api.Session_ID}';
 // Other initialisation here
 }
 </script>
```

If your VisualForce page consists of any Apex or Salesforce Operations such as showing/hiding fields, you will need to paste the following code into your VisualForce page after the operation is called which will re-initialise the tool.

```
function OnComplete() {
preInit();
preInitCC();
ProvenWorks.AutoComplete.Init();
ProvenWorks.AddressComplete.Init();
ProvenWorks.ZipCodeLookup.enforceValidation('zipcodeID');
}
```

Note: the zipcodeID text highlighted above must be changed to the zipcode ID you have used on your Address field. The IDs are explained in the next step.

Setting up the address fields

The address fields must follow two rules:

1) The fields must be linked to a pre-existent field in a Salesforce object, see the snippet of code below which shows the county field linked to the Custom Object, ***NewObject__c*** which has a custom country field by the name of ***Country__c***. This section is in bold in the snippet below.

```
<apex:inputField id="CountryID" value="{!NewObject__c.Country__c}" />
```

2) Each field must have a unique ID, we use IDs to reference the fields so that AddressTools knows where to add functionality. Notice that ***id=""*** is added to the ***apex:inputField*** tag to achieve this.

```
<apex:inputField id="CountryID" value="{!NewObject__c.Country__c}" />
```

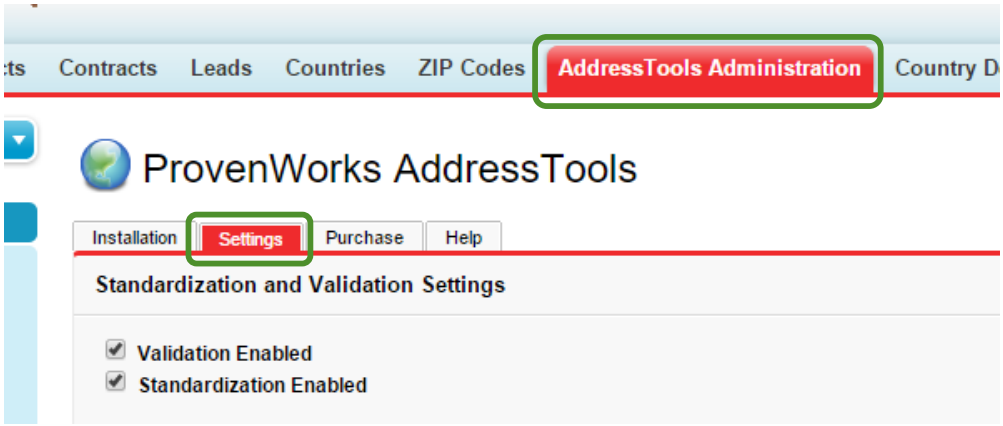
A completed address field in VisualForce may look similar to the example below:

```
<apex:form>

<apex:inputField id="StreetID" value="{!NewObject__c.Billing_Street__c}"/>
<apex:inputField id="CityID" value="{!NewObject__c.City__c}" />
<apex:inputField id="StateID" value="{!NewObject__c.State__c}" />
<apex:inputField id="ZipID" value="{!NewObject__c.Zip__c}" />
<apex:inputField id="CountryID" value="{!NewObject__c.Country__c}" />
<apex:inputField id="LonID" value="{!NewObject__c.lon__c}" />
<apex:inputField id="LatID" value="{!NewObject__c.Lat__c}" />

</apex:form>
```

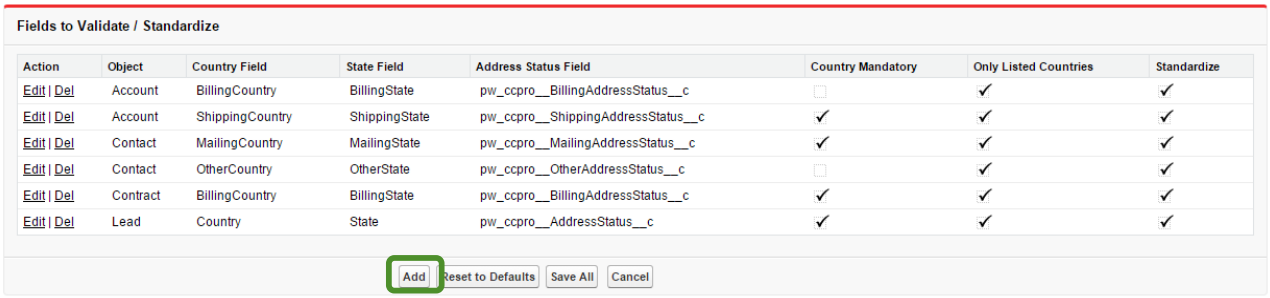
Adding the fields to the AddressTools Settings

Now that we have configured the address field, we can continue onto adding the fields and IDs to the AddressTools Settings.

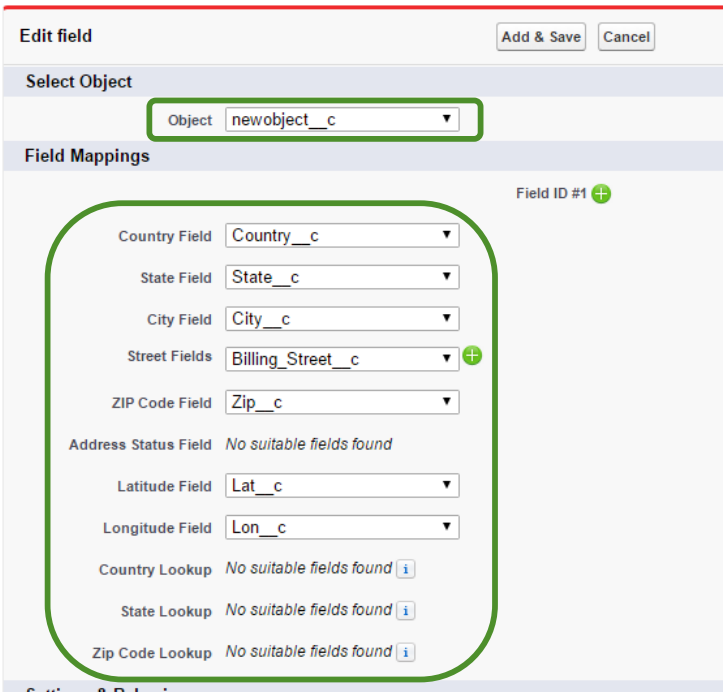
1) Navigate to ***AddressTools Administration*** Tab and select the ***Settings*** sub-tab.



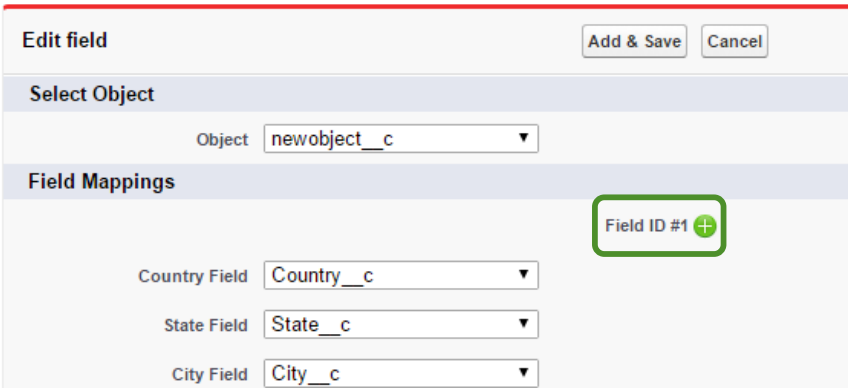
2) Locate the ***Fields to Validate/Standardize*** section and select ***Add***.



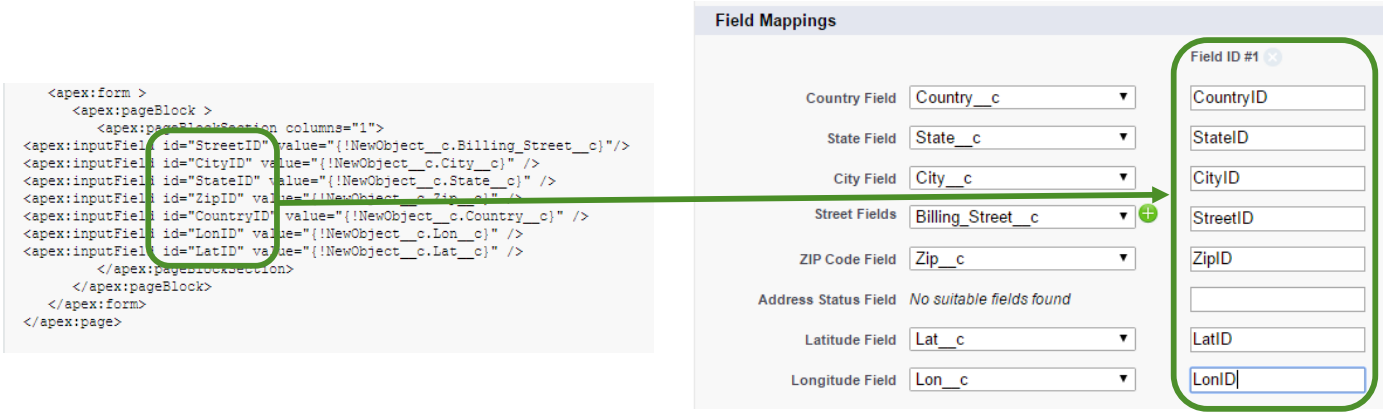
3) Find the Object that the fields are located on, this will be the same object that we referenced in the last step. {!**NewObject__c**.Billing_Street__c} and choose the appropriate fields from the dropdowns provided on the page.



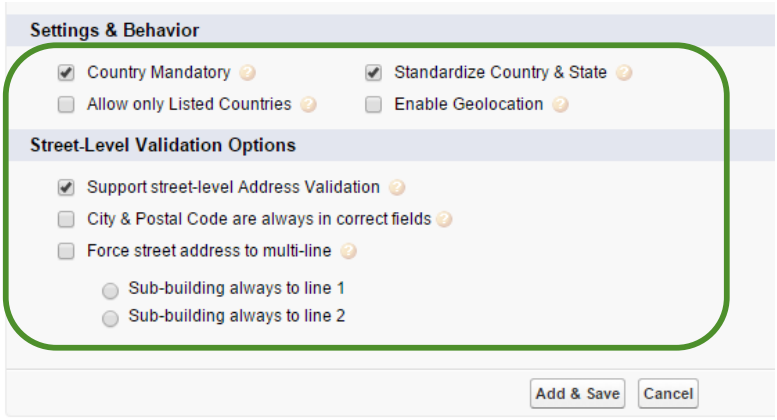
4) Select the '**+**' next to ***Field ID #1*** to display the fields where you can input your field IDs.



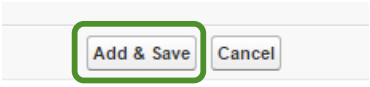
5) Take the IDs that we assigned each field in the VisualForce page and place them into the correct ID field in the ***Field Mappings*** section.



6) Before saving the fields, check the options at the bottom of the page to enable settings such as ***Country Mandatory*** or ***Support Street-Level Validation***.



7) Press ***Add & Save***.

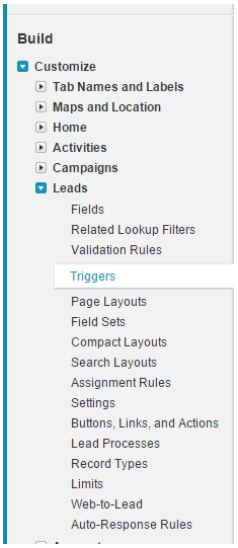


Creating an AddressTools trigger for the VisualForce page

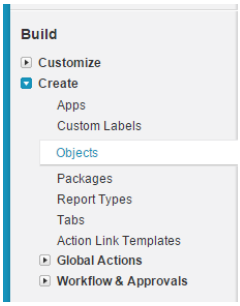
After successfully linking the fields to AddressTools, we want to create a trigger so that the information gets standardized on save and checks whether all requirements are met.

1) Goto ***Setup*** in the top-right corner of your Salesforce environment.

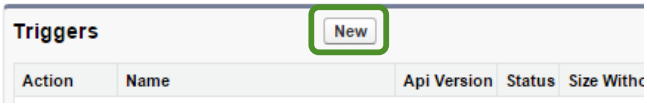
2a) If the object is a standard object, navigate to ***Build → Customize → 'Object Name' → Triggers***



2b) If the object is a custom object, navigate to ***Build → Create → Objects → 'Object Name'*** then locate the section ***Triggers***.



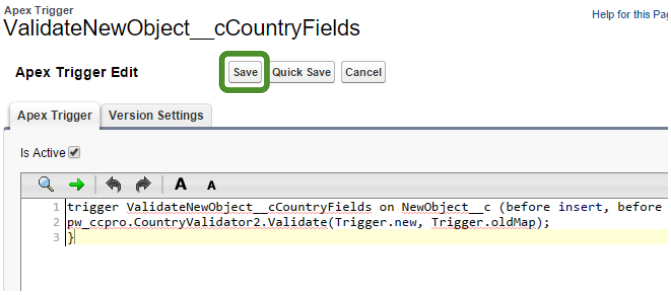
3) Select ***New.***



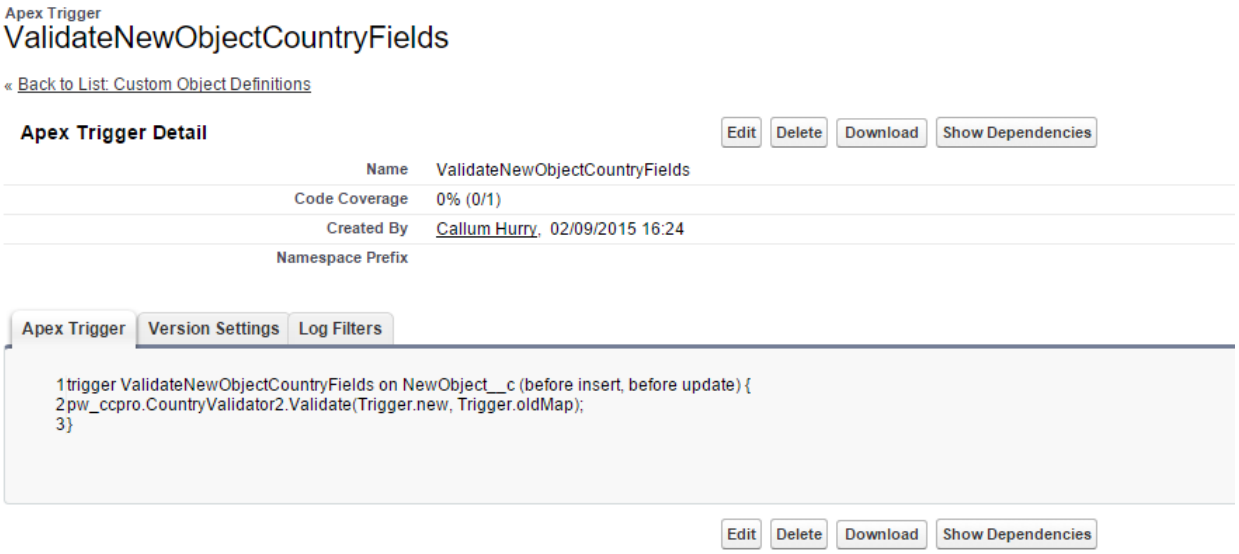
4) You will now be met with a text area to write your trigger, delete what it automatically generated in the field and paste the code below into the trigger's code field, note that the example below is for the 'NewObject' object. Replace the highlighted text to the object's API name. *Be sure to remove the* ***__c*** *from the first instance if it is a custom object to avoid Invalid identifier errors.*

```
trigger ValidateNewObjectCountryFields on NewObject__c (before insert, before update) {
pw_ccpro.CountryValidator2.Validate(Trigger.new, Trigger.oldMap);
}
```

5) Press ***Save.***



Once saved, your view will now look some like this and the trigger will have now been successfully created.

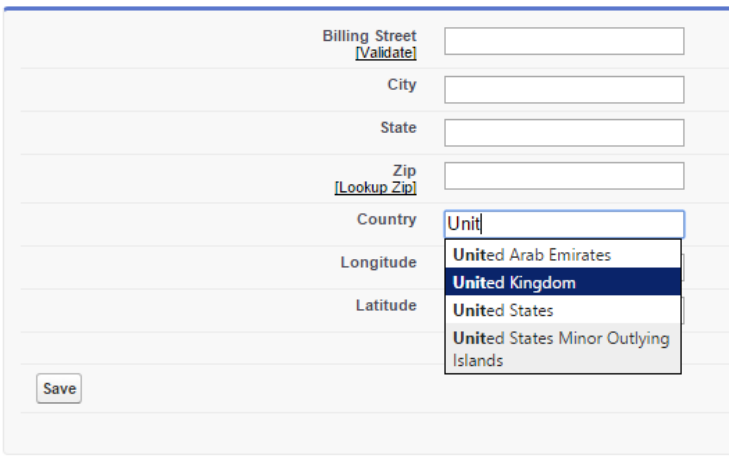


An example VisualForce Page

You will now have all the required information on how to implement AddressTools functionality on your Salesforce VisualForce page. Below is an example of what your VisualForce page's code may look like and a visual of the front end result.

```
<apex:page standardController="NewObject__c">

<apex:includeScript value="/soap/ajax/27.0/connection.js" />
 <apex:includeScript value="/soap/ajax/27.0/apex.js" />
 <apex:includeScript value="/apex/pw_ccpro__CountriesJavaScript?core.apexpages.devmode.url=1" />
 <apex:includeScript value="{!URLFOR($Resource.pw_ccpro__CountryCompleteResources, '/javascript/CountryAutoComplete.js')}" />
 <apex:includeScript value="{!URLFOR($Resource.pw_ccpro__AddressCompleteResources, '/javascript/AddressComplete.js')}" />

<script type="text/javascript">
 sforce.connection.sessionId = '{!$Api.Session_ID}';
 </script>

<p>
<apex:pageMessages id="msg" />
</p>
  <apex:form >
    <apex:pageBlock >
      <apex:pageBlockSection columns="1">

          <apex:inputField id="StreetID" value="{!NewObject__c.Billing_Street__c}"/>
          <apex:inputField id="CityID" value="{!NewObject__c.City__c}" />
          <apex:inputField id="StateID" value="{!NewObject__c.State__c}" />
          <apex:inputField id="ZipID" value="{!NewObject__c.Zip__c}" />
          <apex:inputField id="CountryID" value="{!NewObject__c.Country__c}" />
          <apex:inputField id="LonID" value="{!NewObject__c.Lon__c}" />
          <apex:inputField id="LatID" value="{!NewObject__c.Lat__c}" />

      </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

When going to test the VisualForce page above, you will be met with a display similar to the one here. You can test its functionality by beginning to type a country and seeing that a dropdown appears.



FAQs

Note: All of these questions are in relation to the example above with the Custom Object being named 'NewObject' and all of the address fields going by the names from above, please read through the questions and acknowledge that the custom objects and fields may be of a different name to yours.

Q) When I try to save my new VisualForce page, I am receiving the error the following error:
Error: Unknown property 'NewObject__c' referenced in NewObject

A) You must add the object you are using as a standardController tag to the first <apex:page> so it would appear like this: <apex:page standardController="NewObject__c">

Q) When I try to save my new VisualForce page, I am receiving the error the following error:
Error: NewObject__c does not exist

A) Ensure that you have an object in your Org by the name of NewObject

Q) When I try to save my new VisualForce page, I am receiving the error the following error:
Error: Could not resolve field 'Billing_Street__c' from <apex:inputField> value binding '{!NewObject__c.Billing_Street__c}' in page NewObject.

A) You must ensure that there is a field created in your Object by the name of Billing Street.

Q) When I preview my VisualForce page, it looks something like this:

A) The tutorial we provide is for a very quick and simple VisualForce page, if you add

```
<apex:pageBlock >
        <apex:pageBlockSection columns="1">
```

Just after the form tag and matching closing tags for these just before the closing form tag, the layout should then match the one shown in the final example.

Q) When I try to save my trigger, I am met with the following error:
Error: Compile Error: Invalid identifier: ValidateNewObject__cCountryFields at line 1 column 9

A) Making sure you remove the ***__c*** from the trigger name as this will cause the trigger to be classed as an Invalid identifier.

Q) I went to test my VisualForce page and the dropdown list isn't appearing when I begin to type a state or country.

A) Make sure you clear your browser's cache and reload the page. Most browser's cache can be cleared by pressing [Ctrl+Shift+Del]

Contact Us

If you are still unable to successfully place AddressTools functionality on your VisualForce page, please feel free to get in contact with us at: support@provenworks.com