HOPEWISER

Hopewiser AutoComplete LWC
(Hopewiser: Global Address Verification)

Administration Manual

Hopewiser Ltd.
Merlin Court
Atlantic Street
ALTRINCHAM
Cheshire
WA14 5NL

Tel: 0161 924 2800
www.hopewiser.com
E-mail: support@hopewiser.com

Produced: Spring 2024
Version: 3.4.9

Contents

1 Introduction

Welcome to **Hopewiser AutoComplete LWC**, a Lightning Web Component for Salesforce. The solution is also referred to as **Hopewiser: Global Address Verification**.

Hopewiser AutoComplete LWC provides an easy way to capture and verify addresses, presenting the most relevant matches as you type. The component can be added to Salesforce Lightning Apps, Lightning Record Pages or from a Lightning Flow Screens. This greatly speeds up the process of entering addresses while at the same time ensuring that accurate address information is stored.

You will need an Address Lookup account from Hopewiser to perform an address search. If you do not have an account yet, you can register free at https://www.hopewiser.com/portal-registration.

Once a final completed address is selected from Hopewiser AutoComplete LWC, other associated address field information can also be stored if the appropriate field mappings information is provided.

1.1 Intended Audience

This manual is aimed at administrators of Hopewiser AutoComplete LWC.

1.2 Operation

Please refer to the User Guide for Hopewiser AutoComplete LWC where more detailed operation instructions are given.

1.3 Solution Configuration Page

To perform address lookup with Hopewiser AutoComplete LWC, your Hopewiser Address Lookup account information and optional field mappings must be provided. The required information can be entered via the configuration page which is accessible as follows:

- Click on the “gear” icon to enter the Setup page from your Salesforce Organisation
- Key in “Visualforce Pages” in the Quick Find box, then select Visualforce Pages. This will display a list of Visualforce Pages
- Click on the hyperlink image on the Visualforce Page identified by the “hpwConfigAutoC” label name.

1.4 Version Information

You can view the version information from the solution configuration page (described above). Version information for other dependency libraries will also be displayed.

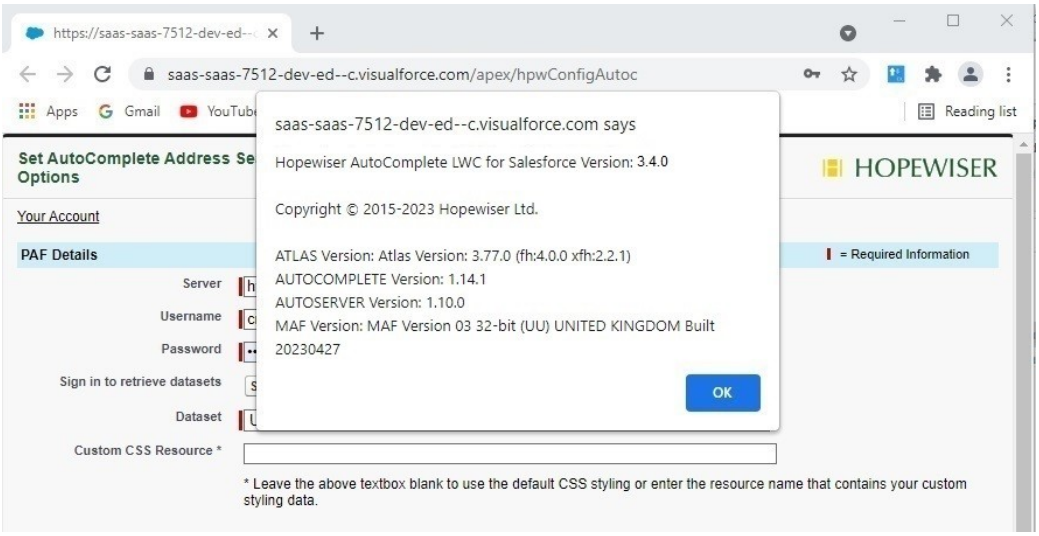


Figure 1. Obtain version information from Configuration Page

Clicking the Hopewiser icon from the top right hand corner will display the version numbers for the application and related libraries, its component parts and the Master Address File (referenced as MAF or dataset in this document) you have selected. Should you need to contact technical support, it is useful to have these version numbers ready to enable the support staff to provide the most efficient response.

2 Installation pre-requisites

Hopewiser AutoComplete LWC can be installed into a target Salesforce Organisation. It has no additional hardware or third-party software requirements beyond those required for Salesforce.

You are required to have an active Hopewiser Address Lookup Bundle, with access to your API Token credentials. To create a new API Token log in to the Hopewiser Web Portal and goto Administration → Tokens → Create Token.

You will need to have system administration permissions on the target Salesforce Organisation where the solution will be installed.

- To **upgrade** an existing installation, please follow the instructions given in Section 2.1.
- For a **new installation**, please find our solution at Salesforce AppExchange (Section 2.2) where more information and installation / user-guide guides are provided. Please ensure to grant access to cloud.hopewiser.com as shown in Figure 4.

  After a fresh installation, you must follow the instructions given in Section 3.0 to **configure the solution.**

2.1 Upgrade Hopewiser AutoComplete LWC

This can be achieved via instructions given in Section 2.2 for "**Installation via Salesforce AppExchange**" as Salesforce will be able to detecte if an upgradeable version of "**Hopewiser: Global Address Verification**" exists or not.

To perform an upgrade, you are recommended to check the namespace prefix for the Hopewiser AutoComplete LWC component that has been installed.

- To obtain the namespace prefix information, click on the "gear" icon to enter the Setup page from your Salesforce Organisation.
- Key in **Installed** in the Quick Find box, then select **Installed Packages**. This will display a list of all the installed packages in your system.
- Locate the packaged called **Hopewiser: Global Address Verification**. The namespace prefix will be displayed amount other information such as version number.

2.1.1 Upgrade the package identified with a hpwSFAC namespace

If you have previously installed Hopewiser AutoComplete LWC that is identified with a namespace prefix of **hpwSFAC**, then you can upgrade to a newer version via Salesforce AppExchange as described in Section 2.2 or by entering the following URL onto a browser.

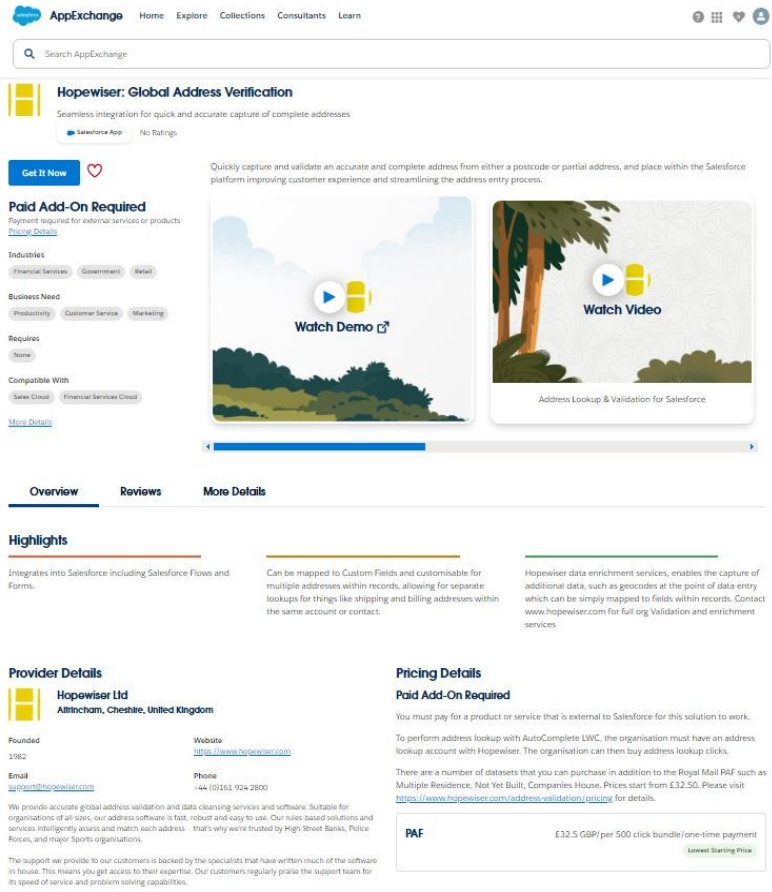
https://login.salesforce.com/packaging/installPackage.apexp?p0=04t4K000002mmas

Figure 2. Upgrade Hopewiser AutoComplete LWC with a hpwSFAC namespace

Select one of the 3 profiles and click the “Upgrade” button to start the upgrade process.

2.2 Installation via Salesforce AppExchange

You can find our solution by entering “Hopewiser” as the search criteria in the Saleforce AppExchange search box at https://appexchange.salesforce.com.



Once you have found our solution, click on the “Get It Now” button to start the installation process.

Figure 3. Salesforce AppExchange

Application information as illustrated from Figure 3 above will be displayed once you have found and clicked on the "**Hopewiser: Global Address Verification**" solution from the Salesforce AppExchange page.

To begin the installation process, simply click on the "Get It Now" button and follow online instructions.

2.3 Grant access to cloud.hopewiser.com

The installation file is released as a managed package which contains static resources, Visualforce pages and the Hopewiser AutoComplete Lightning Web Component.

Installation can be triggered via Salesforce AppExchange. Find our solution by searching the "**Hopewiser"** string within the Salesforce AppExchange page, then click the "**Get it Now**" button to begin the installation.

You will be prompted to sign-in to a Salesforce Organisation where the package will be installed.

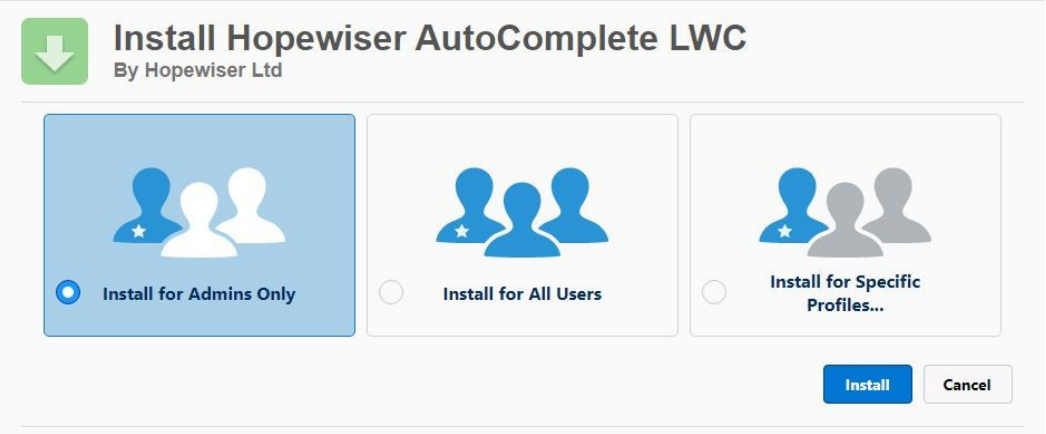


Figure 4. Hopewiser AutoComplete LWC Installation

Select one of the 3 profiles and click the "Install" button to start the installation process, then follow the on screen instructions to install Hopewiser AutoComplete LWC. Tick the "Yes" checkbox and click the "Continue" button to grant access to cloud.hopewiser.com website when the following screen appears.

Figure 5. Grant access to cloud.hopewiser.com

Continue the installation process and click the “Done” button to finish the installation when the installation completed message appears.

Before using the LWC for the first time, please upload a static resource text file as described in Appendix A1. This static resource should be named as “**hpwConfigAutoc_Org**” (the static resource name is case sensitive). It contains the following text.

```
<Hopewiser>
  <Host>Salesforce.com</Host>
  <!-- PAF Details -->
  <Server>https://cloud.hopewiser.com</Server>
  <CustomCSS></CustomCSS>
  <Dataset>uk-rm-paf</Dataset>
  <AvailableDatasets>uk-rm-paf</AvailableDatasets>
  <!-- Common Address Field Mappings -->
  <Common_Fields></Common_Fields>
  <!-- ExtraData Field Mappings one node for each MAF -->
  <uk-rm-paf_XtraData></uk-rm-paf_XtraData>
 </Hopewiser>
```

Once the file is uploaded correctly, a screen that is illustrated below will be shown.

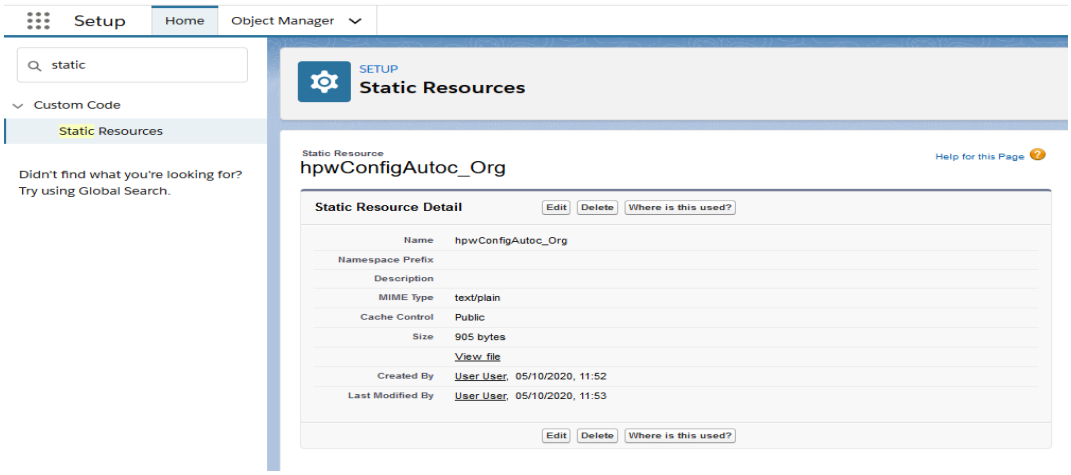


Figure 6. The hpwConfigAutoc_Org static resource.

The above static resource file contains empty configuration information. Please refer to the instructions in Section 3 when you are ready to provide the configuration data. Note that Hopewiser AutoComplete LWC will not function correctly until the credentials and configuration data is entered and saved.

3 Configuration

3.1 Configuration Page and Optional Field Mappings

Your Hopewiser Address Lookup credentials and optional field mappings are managed via configuration page. To access the configuration page, please refer to the bullet points in Section 1.3 above.

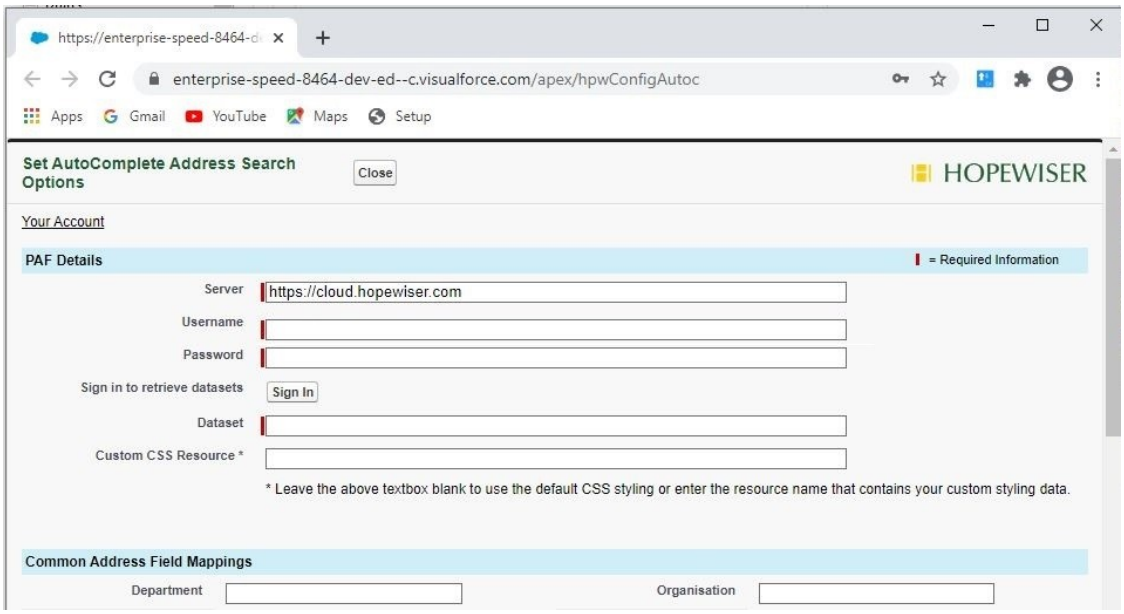


Figure 7. Solution Configuration Page

Fill in your username and password and click the “Sign In” button. Assuming you have provided the correct credentials, the Dataset drop-down box will contain the list of MAFs that you can perform address lookups against. The selected dataset value will become the “Default MAF” when adding a new instance of the Hopewiser AutoComplete LWC to a Lightning Page or Flow Screen.

There are over 20 solution core Common Address Fields and a variable number of Extra Data fields which you can extract and store onto Salesforce fields by mapping the Salesforce API fieldnames in the respective textboxes. Refer to Appendix A4 for detail on how to find the API name of a Salesforce. In the examples given below (assuming the Salesforce SF_Town, SF_OrgKey, SF_OrgKey2 and SF_AddrKey fields exist) the **Town** address field data value from matched address will be returned to the Salesforce **SF_Town** field.

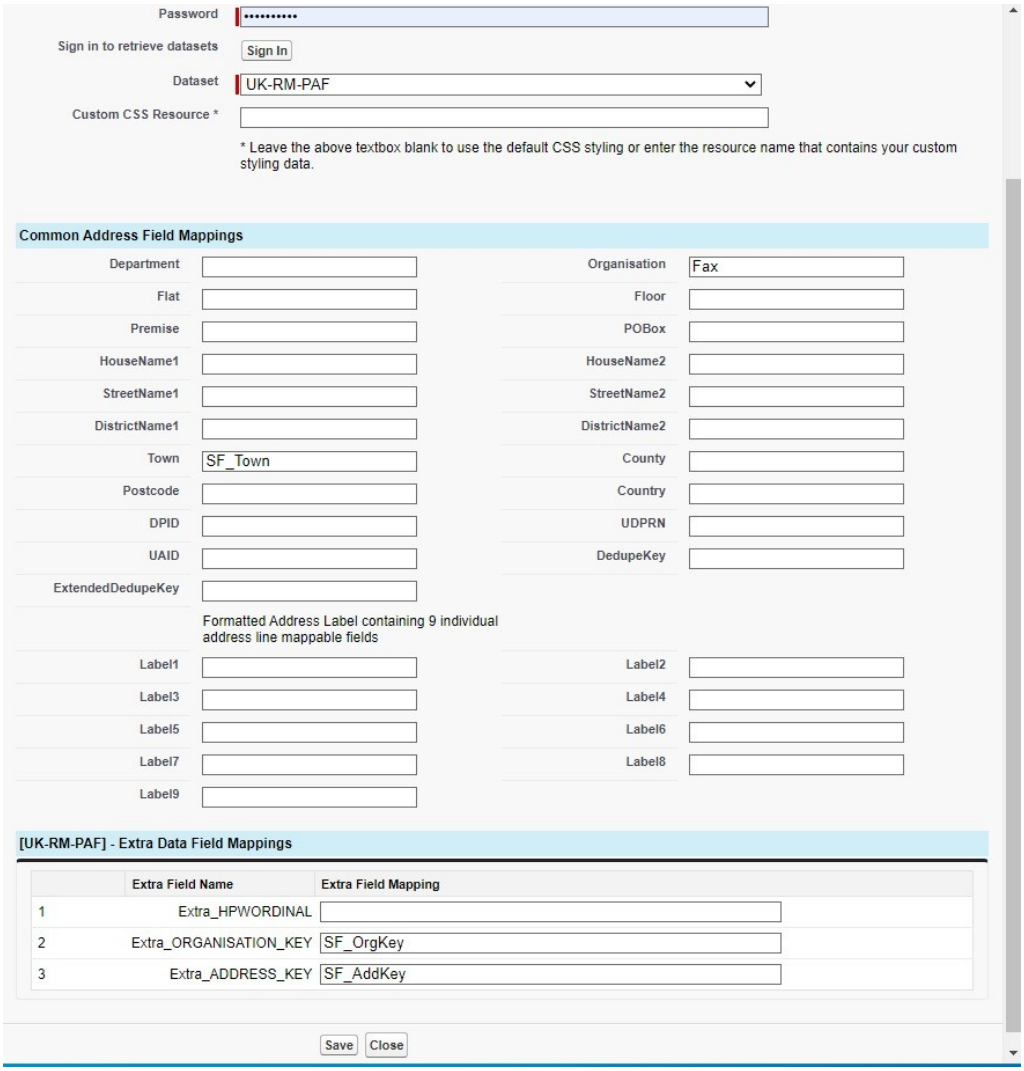


Figure 8. Setting Default Dataset and field mappings information on Configuration Page

The number and the names of Extra Data Fields varies between different datasets. In the above example, the **uk-rm-paf** dataset contains 3 extra data fields (Extra_HPWORDINAL, Extra_ORGANISATION_KEY and Extra_ADDRESS_KEY). The last two extra data fields are mapped to Salesforce's **SF_OrgKey** and **SF_AddrKey** Fields respectively.

Multiple fields mappings are supported as illustrated in Figure 9 below. Simply provide the Salesforce API field names as a pipe-delimited string to enable multiple fields mappings. The example below indicates that the extra data Extra_ORGANISATION_KEY is to be stored in both the **SF_OrgKey** and **SF_OrgKey2** Fields.

Figure 9. Multiple fields mappings are supported.

You can prevent information from being returned from common address fields and/or extra data fields in three ways:

- Qualify the Salesforce API name with a prefix to distinguish the specific target mapped fields when you enter the field mapping information from the Configuration Page.

- Tick the **“Suppress Common Address Fields”** checkbox when you are configuring the Hopewiser AutoComplete LWC properties if you do not want mapped common field data to be returned.

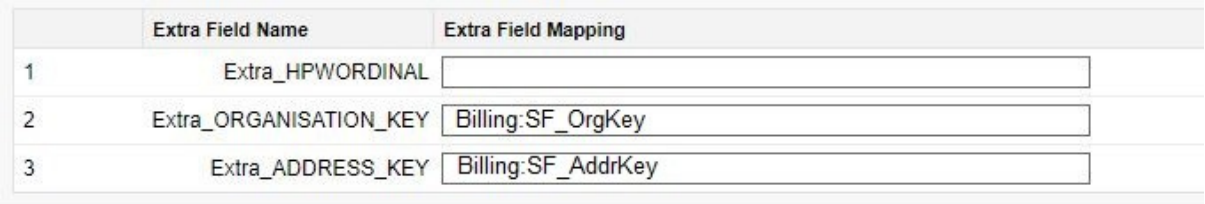
- Tick the **“Suppress Extra Data Fields”** checkbox when you are configuring the Hopewiser AutoComplete LWC properties if you do not want mapped extra data field data to be returned.

3.2 Qualified Field Mappings

You can prevent common data or extra data information from being returned by qualifying the mapped information at the top level via the Configuration Page.

Some Lightning Record Pages contain more than one set of address fields. For example, you will find a set of address fields for Billing information and another set of address fields for Shipping information in the Account entity Record Page. For example, you can add the first instance of the Hopewiser AutoComplete LWC to the Account Record Page and assign the “Destination address” LWC property to “Billing” to update the Billing address information. Then add second instance of the Hopewiser AutoComplete LWC and assign the “Destination address” LWC property to “Shipping” to update the Shipping address information.

If you only want the Billing address data to return the extra data, you can qualify the field mapping information with the “Destination address” followed by the colon symbol. In this case this will be “**Billing:**” as per the illustration below:



	Extra Field Name	Extra Field Mapping
1	Extra_HPWORDINAL	
2	Extra_ORGANISATION_KEY	Billing:SF_OrgKey
3	Extra_ADDRESS_KEY	Billing:SF_AddrKey

Figure 10. Qualifying specific field mappings.

The address and the extra data from the first LWC instance (which is configured to associate with the Billing address) will be returned. Whilst the address data from the second LWC instance (which is configured to associate with the Shipping address) will be returned, the extra data will not be returned as the extra data field mappings have not been qualified for “Shipping” addresses in the above example.

3.3 Suppressing Field Mapping Information

You can also prevent mapped information to be returned from common data fields or extra data fields by enabling the appropriate **“Suppress … Fields”** LWC properties.

In the Account Record Page example below, there are two instances of the Hopewiser AutoComplete LWC for the Account Record Page. The first LWC (near the top) for Billing Address has been configured with default values for all its LWC properties. The checkboxes to suppress the field mappings on the second LWC for Shipping Address have been ticked. This means the second LWC will only return the address data but no common field or extra data fields will be returned.

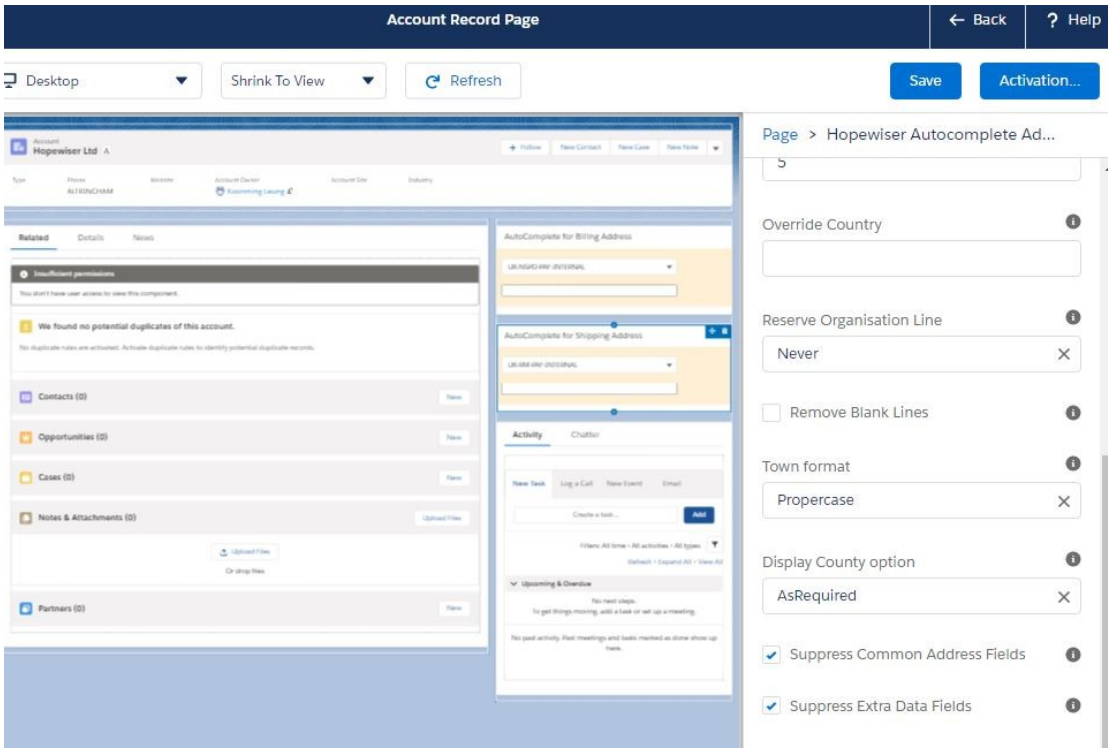


Figure 11. Account Record Page with added Hopewiser AutoComplete Components.

3.4 Custom CSS Resource

A default hpw-autoc-jsclient.css file is used to format the dynamic menu for the Hopewiser AutoComplete LWC. Refer to Appendix 5 to see the default styling definitions.

If you have uploaded an alternative static CSS resource, enter its name here to enable it.

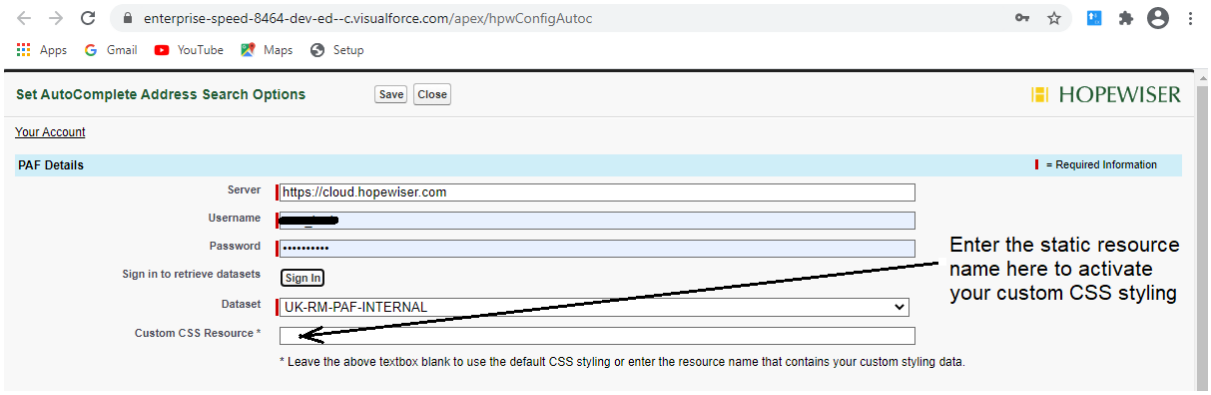


Figure 12. Enable custom CSS from Configuration Page

3.5 Remote Site Settings

Hopewiser AutoComplete LWC references the "https://cloud.hopewiser.com" server by default. If you have changed the server url value from the configuration page, please ensure that you create a new '**Remote Site Settings**' to specify the amended server url or update the "Remote Site URL" value for "**HopewiserLimited**" which can be found with in the "Remote Site Settings" Salesforce setup screen.

4 LWC Usage

The Hopewiser AutoComplete LWC is accessible from Salesforce tools such as Lightning App Builder and Salesforce Flow Builder. Typically you would drag and drop the LWC component to the required page with Lightning App Builder or add the component to a Flow Screen within Salesforce Flow Builder. Set the LWC properties to the required values to control the behaviour of the Hopewiser AutoComplete LWC address lookup capability at runtime.

Five Salesforce entities are supported in Lightning Record Pages, these are Account, Contact, Lead, Order and Contract. On Flow Screens, address lookup feature is supported for existing records in Account, Contact and Lead entities. Salesforce entity records are not linked in Lightning App Pages.

Fore a more detailed description on how to use the Hopewiser AutoComplete LWC can be found in the User Guide document.

4.1 Adding the LWC to a Lightning page

Adding the Hopewiser AutoComplete LWC can be achieved by dragging and dropping the component to the required layout position within the Lightning Page or Flow Screen. Two instances of the LWC have been added to the Account Record Page as illustrated in Figure 11 Account Record Page above.

Once the component is added to a page, its properties will be displayed. The default values for these properties can be amended to change the behaviour of the LWC.

4.2 LWC Properties

The following Hopewiser AutoComplete LWC properties are configurable in Lightning App Pages, Lightning Record Pages and Flow Screens.

- **Destination address**
  Valid values are : Billing, Shipping, Mailing, Other and Standard. On a Flow Screen, the user can enter a custom string to link to a specific Destination address. This value will be used as a prefixed text in order to identify the correct address field to be updated.

- **Minimum number of input characters**
  This will define the minimum number of input characters which will trigger an autocomplete search. A default value of 3 is assigned.

- **Delay in milliseconds**
  The delay in milliseconds is the time between when a keystroke occurs and when a search is performed. A default value of 300 is assigned.

- **Default MAF**
  This is the default MAF for your autocomplete search. The initial value will be derived from the value you have set in the configuration pages.

- **Number of formatted address lines**
  This is the number of address lines in the output label. It defaults to 7. This will only affect the returned address label to custom multiline address box destinations.

  Note : Address data returned from the LWC will be populated to the individual Salesforce address fields (eg **BillingStreet, BillingCity, BillingState, BillingPostalCode** and **BillingCountry** for the Account entity), this LWC property setting has no obvious effect on the composite address fields such as **BillingAddress** and **ShippingAddress** fields in the Account entity, **MailingAddress** and **OtherAddress** fields in the Contact entity and the

Address field in Lead entity as these composite address fields will be formatted by the Salesforce application from the corresponding address field elements.

- **Override Country**
  The country information will be overwritten with the text specified by this property if set.

- **Remove Blank Lines**
  The default value is **false**. Tick this box or set it to **true** to remove blank lines from the completed address output. As Salesforce address elements are normally displayed on separate address lines (street, city, state and postal code), there will be no obvious changes in the formatted address lines as experienced by the end user.

- **Reserve Organisation Line**
  Reserve the first line of the formatted address label for the organisation in the completed address output. The default value is “Never” as address are typically stored without the company name information because a specific company field might already exists. Valid values for this property are **AsRequired, Always** and **Never**.

- **Town format**
  This specifies the town format. The default is **Propercase**. Change this to **Uppercase** or **Lowercase** if required.

- **Display County option**
  This specifies the county format. The default is **AsRequired**. Change this to **Always** or **Never** if required.

- **Suppress Common Address Fields**
  The default value is **false**. The Common Address Field mappings will be ignored if this box is ticked or it is set it to **true**.

- **Suppress Extra Data Fields**
  The default value is **false**. The Extra Data Field mappings will be ignored if this box is ticked or it is set it to **true**.

Salesforce address fields are normally prefixed with a unique identifier if more than one set of address elements are present within the Salesforce entity. For example, you will find a set of Billing and Shipping addresses in Account and a set of Mailing and Other addresses in Contact. There is only one set of address information in Lead, the standard address fields in Lead are not prefixed with an identifier.

An additional **Custom Prefix And Caption** LWC configurable property is available if the component is used in App Pages or Lightning Record Pages.

- **Custom Prefix And Caption**
  Use this property to overwrite the default address prefix and LWC caption values. Use the **colon** character to separate the prefix and caption values. For example, the string “**Shipping:Delivery Address**” specifies the “**Destination address**” is to be identified with an address prefix of “**Shipping**” and the LWC component itself will be displayed with a “**Delivery Address**” caption. Salesforce address fields referenced by Street, City, State, PostalCode and Country will be updated from the result of a complete address that is returned from the Hopewiser AutoComplete LWC address search.

Additional LWC configurable properties described below are only available when the component is used in a Flow Screen.

- **Entity Name**
  Supported Salesforce entities in a Screen Flow are Account, Contact and Lead. If the supported **Entity Name** is provided, then the appropriate Salesforce Address fields will be displayed as illustrated in Figure 13 below.

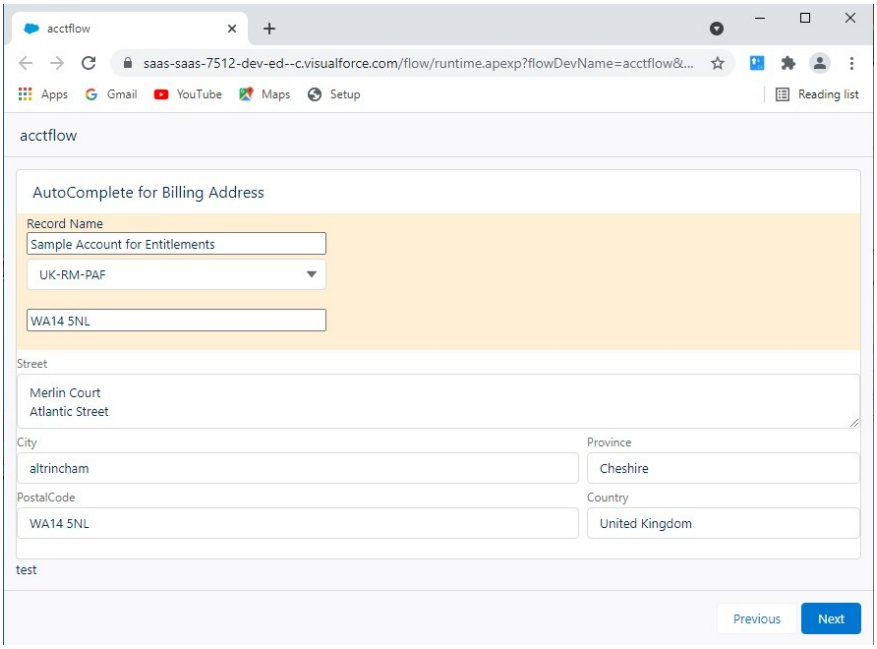


Figure 13. LWC linked to the Billing address of an existing Account record

If the **Entity Name** is not provided, then the “Returned Address Label and requested data” textbox will be displayed instead. An example of this can be seen in Figure 15 where none of the three LWC components have been configured to link to an entity object.

- **Selected Record Id**
  This is the Salesforce entity record Id that uniquely identifies the current record. The LWC will automatically update the linked Salesforce record if a valid **Selected Record Id** is provided for the linked **Entity Name** object.

- **Selected Record Name**
  This is the Salesforce entity record Name (which will be the account name or full contact name for the current record). This is a readonly field to help the user identify the correct record easily.

- **Do not show the record name**
  A default value of **false** is assigned to this property. Flow Screens can be used to capture addresses for existing entity records. The **Selected Record Id** should be linked to the I**d** field from the given **Entity Name**. To help identify the correct record on Screen Flows, user should also link the **Selected Record Name** field correctly. Set this property to **true** if you wish to hide the LWC Record Name field.

- **moaStreet**
  This flow output field will store the multi-lines Salesforce Street value.

- **moaCity**
  This flow output field will store the single-line Salesforce City value.

- **moaState**
  This flow output field will store the single-line Salesforce State value.

- **moaPostalCode**
  This flow output field will store the single-line Salesforce PostalCode value.

- **moaCountry**
  This flow output field will store the single-line Salesforce Country value.

- **hw[Address Label Fields]**
  9 address label data fields (**hwLabel1, hwLabel2, hwLabel3, hwLabel4, hwLabel5, hwLabel6, hwLabel7, hwLabel8, hwLabel9**) are available. The "**Number of formatted address lines**" LWC property will determine how many of these **hw[Address Label Fields]** will be populated.

  For example, when is set to **Number of formatted address lines** 7, then 7 flow label variables will be populated (**hwLabel1-7)**. The first 4 label lines (**hwLabel1-4**) will correspond to each one of the multi-lines in the **moaStreet** data. The remaining 3 label lines (**hwLabel5-7**) will contain the city,state and postcode which should correspond to the **hwTown, hwCounty and hwPostcode** data respectively.

- **hw[Other Common Address Fields]**
  Another 21 additional common data fields are available in the Flow design
  hwDepartment, hwOrganisation, hwFlat, hwFloor, hwPremise, hwPOBox, hwHouseName1, hwHouseName2, hwStreetName1, hwStreetName2, hwDistrictName1, hwDistrictName2, hwTown, hwCounty, hwCountry, hwDPID, hwUDPRN, hwUAID, hwUDPRN, hwDedupeKey, hwExtendedDedupeKey

Note the user can change values on the Salesforce address fields, therefore the combined content of **hw[Address Label Fields]** could be different from the combined Salesforce address fields. Detailed description of the Common Address Fields can be found in https://www.hopewiser.com/developer-document/autocomplete-api/ under the "Common Output Fields" section.

5 Add Hopewiser AutoComplete LWC in Lightning App Builder

The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience. Hopewiser AutoComplete LWC component can be accessed through the Custom Apps in helping to build either a single-page app, a dashboard-style app or a Salesforce entity record page with address capture and verification functionality included.

Click **Setup** in the top right of the dashboard. Enter *Lightning App Builder* in the **Quick Find** box and then select **Lightning App Builder**. Find **Hopewiser Autocomplete Address** component from the Component List on the left hand side. Drag and drop the component onto the centre panel of the Salesforce Lightning App Builder screen. The added AutoComplete LWC component can be seen in Figure 14. The properties of the component can be altered on the right hand side.

The **Destination address** can be chosen as Lightning Record page, such as Billing, Shipping, Mailing and Other. The values available will depend upon the entity in play.

The **Minimum number of input characters** is used to set the minimum number of characters that must be entered before an address search will be started,

The **Default MAF** is the dataset used by default for searching address. Please refer to **Appendix A2** for details of connecting to Hopewiser's website to get the list of available MAF reference dataset for a Hopewiser Portal Organisation or Account.

The other properties included here are all covered in the preceding **LWC Properties** section.

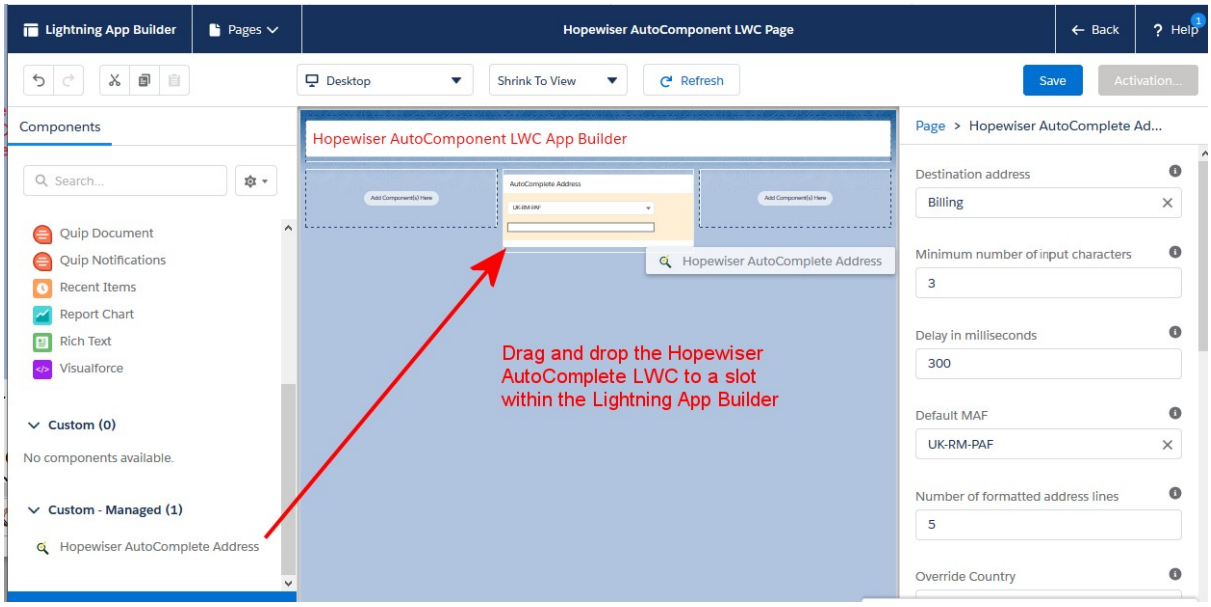


Figure 14: Hopewiser AutoComplete LWC has been added onto a Customer Lightning App Page

An example of using the Salesforce App Builder to add the Hopewiser AutoComplete LWC to a page is shown above. The page contains a Header and Three Regions, with the Hopewiser AutoComplete LWC component being added to the central region.

Figure 15 below shows a simple Lightning App that demonstrates Hopewiser AutoComplete LWC Address Search capabilities. This App can be used to search (or verify) a Billing, a Shipping and a Mailing Address.

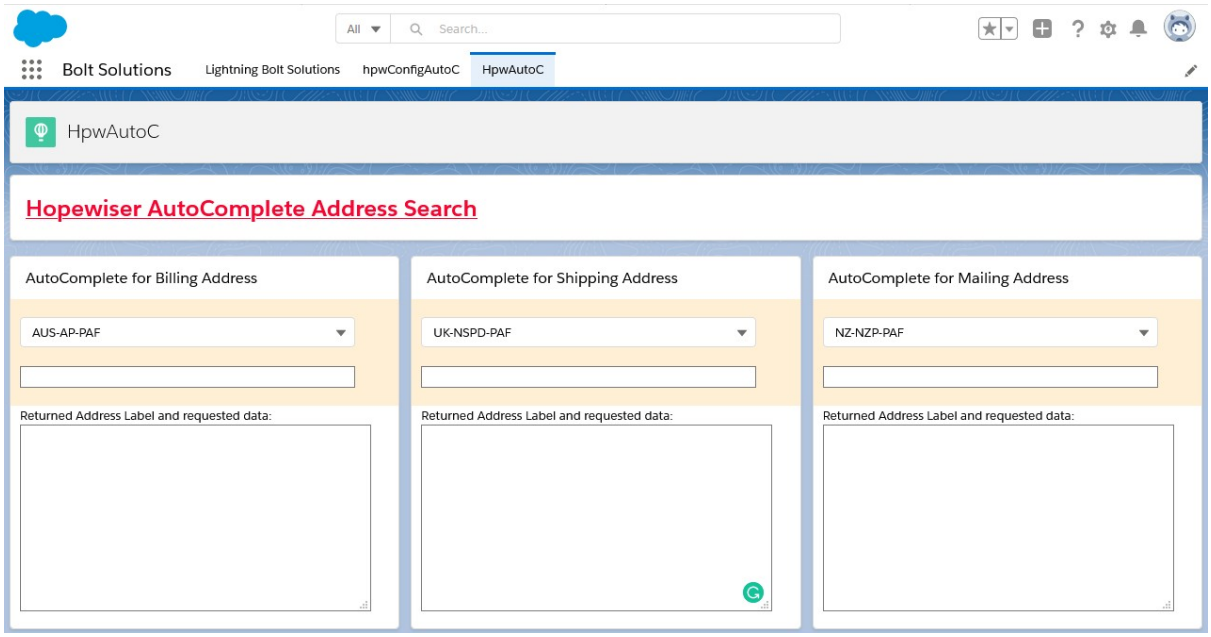


Figure 15: Lightning page with three Hopewiser AutoComplete LWC components for Billing, Shipping and Mailing Address

6 Adding Hopewiser AutoComplete LWC to Salesforce Lightning Record Page

Hopewiser AutoComplete LWC component can also be added into most Salesforce Lightning Record pages with address fields, such as the Account Page, Contact Page, Contract Page, Leads Page or Order Page and any Lightning Record page containing address fields.

This section shows how to add the Hopewiser AutoComplete LWC component into the Lightning Record Account page. In **App Launcher** click **Accounts,** select any account to display account page. Select **Setup** → **Edit Page.**

Figure16 shows the Lightning Record Account page with lookup panels pertaining to Billing address and Shipping Address as two added LWC components. **Hopewiser AutoComplete Address** is a LWC component that can be dragged-and-dropped into the page to verify or alter the Lightning Record Salesforce account fields concerned.

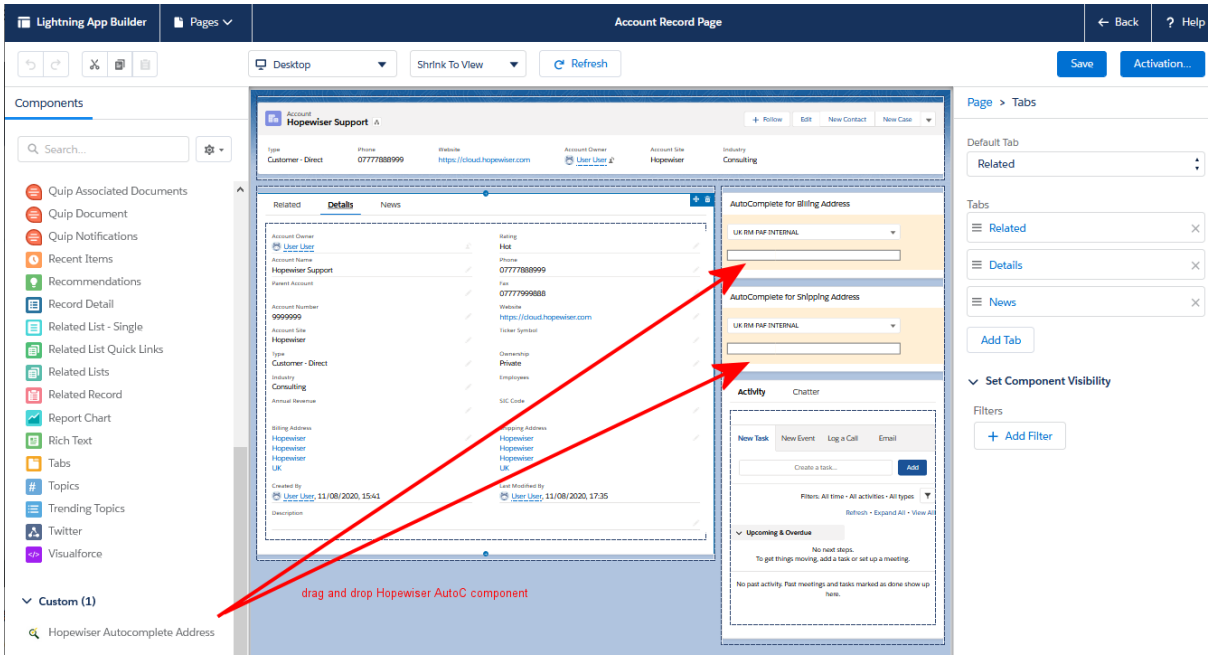


Figure 16: Add Hopewiser AutoComplete LWC component to Salesforce Lightning Record account page.

The altered Lightning Record Salesforce page should be saved and activated via the buttons in the top right of the Edit page. The Activation is only required once in any component editing session. This allows the address lookup panel(s) created to be visible to all users of that entity going forwards.

7 Add Hopewiser AutoComplete LWC into Flow Builder

The Hopewiser AutoComplete LWC component can also be loaded while Salesforce user building a customer flow.

Figure 17 is a simple flow example which will load each account record in the Organisation, validate the Billing Address, then outputs to the account record. The address lookup function in a flow is supported for Account, Contact and Lead entities. Person Account is also supported if the feature is enabled for the Organisation.

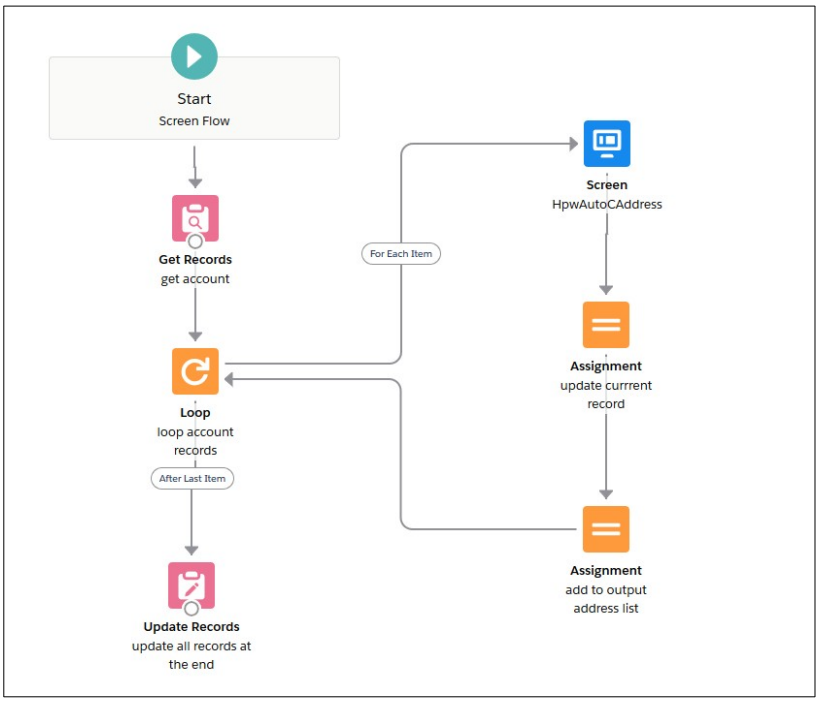


Figure 17. A simple address validation flow for account entity

Figure 18 illustrates the process of adding Hopewiser's AutoComplete LWC component to a Flow Screen by drag-and-drop the component onto the screen.

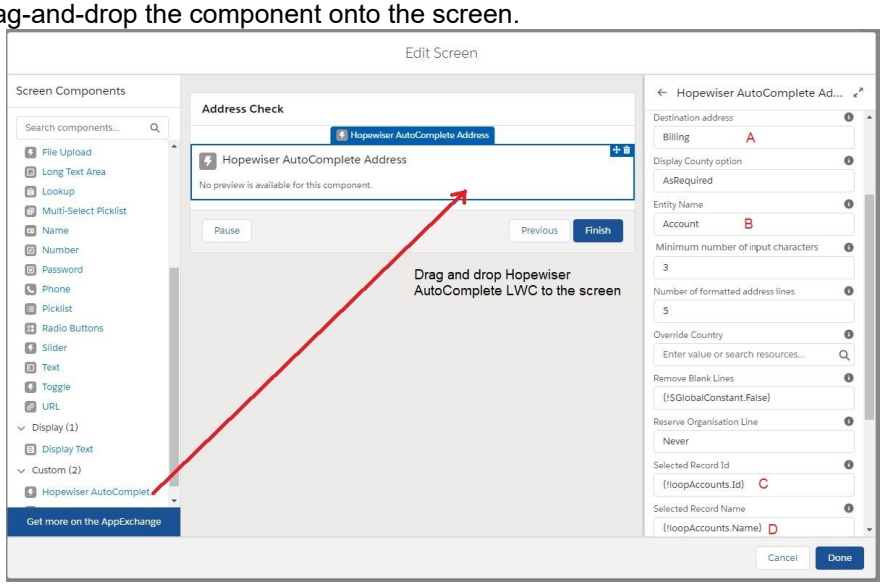


Figure 18: Add Hopewiser AutoComplete LWC to a screen in Flow Builder

The LWC properties are displayed on the right when the component is selected. The following 4 properties (labelled A-D in red) should be configured carefully to ensure the Address Lookup function work correctly.

A. Destination address

You should enter a value here to specify a prefix which can identify the Destination address eg Billing, Shipping, Mailing, Other, Standard or a custom prefix. You can assign a custom caption value with the colon symbol (refer to **Custom Prefix And Caption** LWC properties as described in section 4.2 above for details).

B. Entity Name

Valid values are Account, Contact or Lead.

C. Selected Record ID

This is the Salesforce record ID that will uniquely identify the current record.

D. Selected Record Name

This is account name or contact full name that is associated with the current record.

For the Account entity, **Billing** Address and **Shipping** address are supported. For the Contact entity, **Mailing** Address and **Other** address are supported. For the Lead entity, **Standard** Address is supported. If Person Account is enabled, two extra Destination address can be added to the Hopewiser AutoComplete LWC component. These are **PersonMailing** Address and **PersonOther** Address .

Each Hopewiser AutoComplete LWC component can be configured in the same manner as described in the preceding section for including the LWC in to a Lightning Record page. The are 4 LWC properties that need to be set correctly in order for the customer flow to run smoothly. For example for an account entity address validation flow (Figure 17).

AutoComplete LWC property	set value
Destination address	Billing / Shipping / PersonMailling / PersonOther
Entity Name	Account / Contact / Lead
Selected Record Id	{!Loop_account_record.Id}
Selected Record Name	{!Loop_account_record.Name}

Each Hopewiser AutoComplete LWC component will output the following address fields when a final address is selected:

moaStreet, moaCity, moaState, moaPostalCode, moaCountry, recordId.

Note that the Destination address will be automatically updated when a final address is selected from Hopewiser AutoComplete LWC. Should you wish to pass the address information to the other elements in the Flow, you can assign the above Hopewiser AutoComplete LWC output fields.

In the above example, there are two assignment flow elements. The first assignment is to assign Hopewiser AutoComplete LWC output address fields to a single record variable. The second assignment is to append each record variable into a collection list. In the example below, **curAddress** is an entity record variable, it is used to store the current selected address that has been captured by the Hopewiser AutoComplete LWC which is contained in a Screen Flow (referenced by the API name **hpwAutoCAddress**)

In Flow Build, create a variable for each record, curAddress for example, assign variable's elements equal to the Hopewiser AutoComplete LWC output address fields accordingly. For example Billing Address assignments is illustrated in the list below:

	Flow Variable Element		Hopewiser AutoComplete LWC Address fields
Billing	{!curAddress.BillingStreet}	=	{!hpwAutoCAddress.moaStreet}
	{!curAddress.BillingCity}	=	{!hpwAutoCAddress.moaCity}
	{!curAddress.BillingState}	=	{!hpwAutoCAddress.moaState}
	{!curAddress.BillingPostalCode}	=	{!hpwAutoCAddress.moaCity}
	{!curAddress.BillingCountry}	=	{!hpwAutoCAddress.moaCountry}
	{!curAddress.Id}	=	{!hpwAutoCAddress.recordId}

Figure 19 shows the running of Hopewiser AutoComplete LWC in the Flow. The LWC is configured to capture a Billing Address for the Account entity, the selected address will be displayed in the address field boxes on the Screen Flow where further manual editing can be edited, the address will be committed to the entity record when the "Next" button from the Screen Flow is clicked.

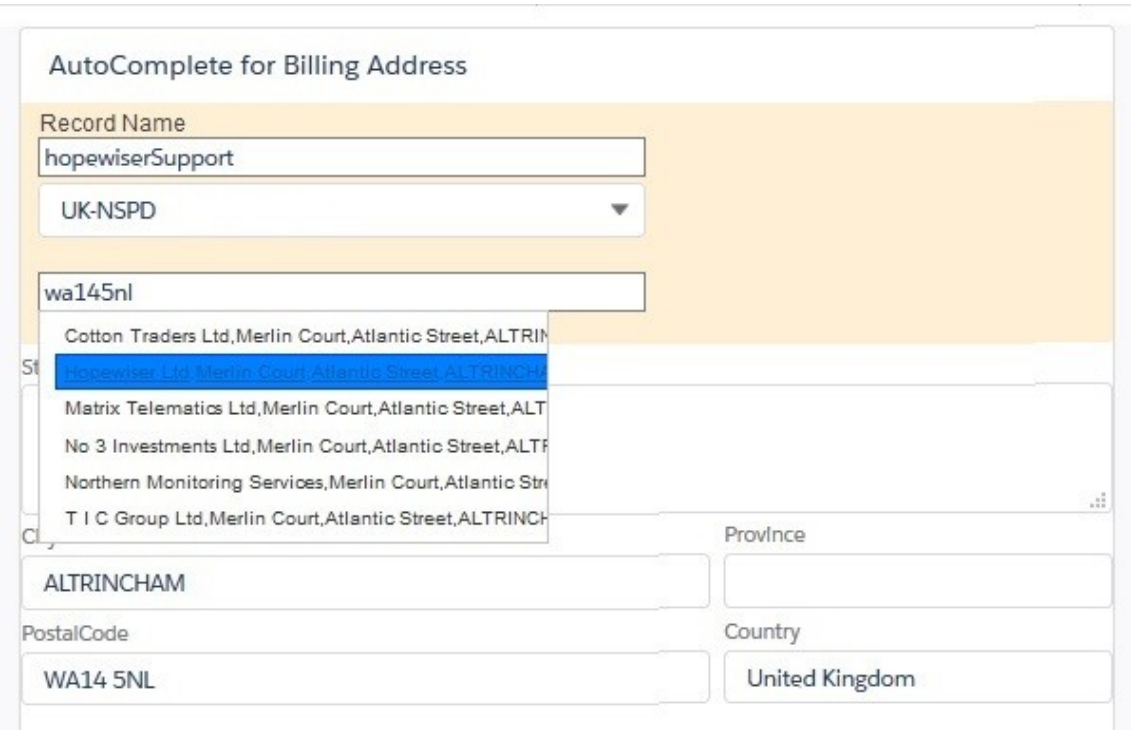


Figure 19. Using Hopewiser AutoComplete Address LWC to capture Billing Address in a Screen Flow.

If the user wishes to capture the Shipping Address for the Account entity, whilst it is possible to an a second instance of the Hopewiser AutoComplete Address LWC to the same Screen Flow, it is recommended to add a new Hopewiser AutoComplete Address LWC to a separate Screen Flow to distinguish the two separate tasks more easily.

8 Uninstalling the Hopewiser AutoComplete LWC Component

If you installed the installation package, navigate to **Setup** > **Develop** > **Installed Packages**.
- Click **Uninstall** next to the package named `Hopewiser AutoComplete LWC` to complete the uninstallation.

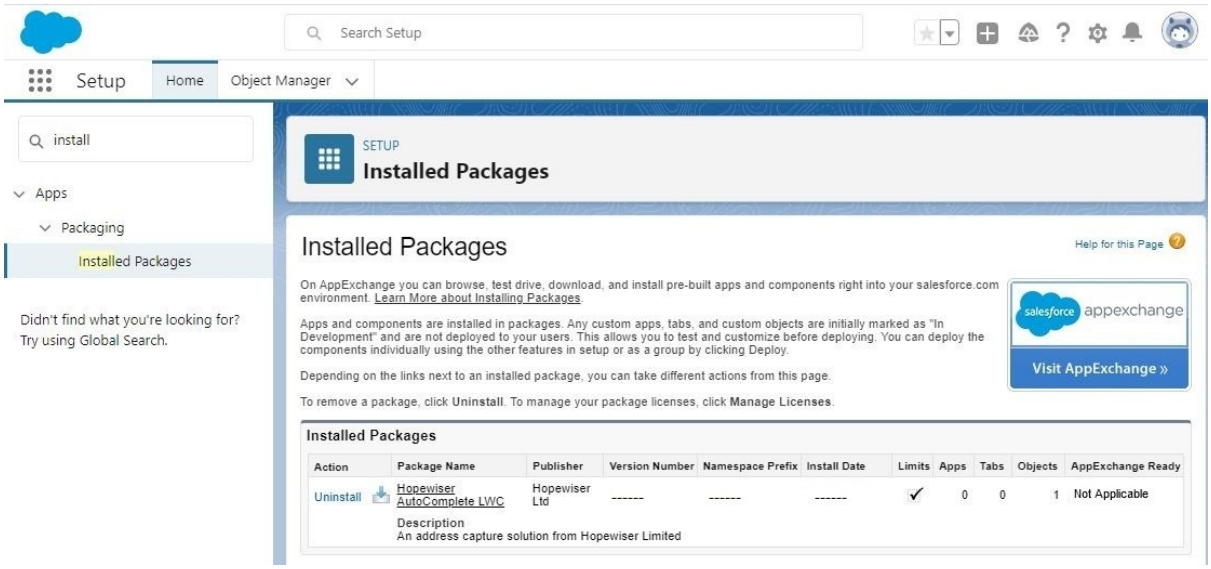


Figure 20. Uninstall Hopewiser AutoComplete LWC

Please follow the on screen instructions to remove the installed package.

Appendix

A1. Upload hpwConfigAutoc_Org Static Resources

In Salesforce.com, navigate to **Setup** > **Home** > **Custom Code** > **Static Resources**.

- A copy of the "**hpwConfigAutoc_Org.txt**" file can be found in the installation folder alongside this administration guide and the user guide. If required, you can create it by saving the following content as a text file and save it with a file name "**hpwConfigAutoc_Org.txt**". This should be uploaded to the target Salesforce Organisation as a static resource file with the name "**hpwConfigAutoc_Org**"

```
<Hopewiser>
  <Host>Salesforce.com</Host>
  <!-- PAF Details -->
  <Server>https://cloud.hopewiser.com</Server>
  <CustomCSS></CustomCSS>
  <Dataset>uk-rm-paf</Dataset>
  <AvailableDatasets>uk-rm-paf</AvailableDatasets>
  <!-- Common Address Field Mappings -->
  <Common_Fields></Common_Fields>
  <!-- ExtraData Field Mappings one node for each MAF -->
  <uk-rm-paf_XtraData></uk-rm-paf_XtraData>
</Hopewiser>
```

- Click the **New** button.
- In **Name** type **hpwConfigAutoc_Org** (case sensitive).
- In **Description** type Hollister **AutoComplete LWC Options**.
- In **File**, click the **Browse** button and navigate to the "**hpwConfigAutoc_Org.txt**" file (from the installation folder or from the folder where you have saved the file earlier), select it and then click **Open**.
- In **Cache Control**, select **Public**.
- Click the **Save** button.

A2. Accessible MAFs

This list of accessible Master Address Files (referred to as **MAFs** or **Datasets**) are selectable from the "Default MAF" drop-down list within the Hopewiser AutoComplete LWC for Lightning App Pages and Lightning Record Pages.

When using the LWC within a Flow Screen, you will have to enter the MAF reference Id manually. The MAF reference Id can be retrieved from the URL below:

https://cloud.hopewiser.com/autoc/json

Once the correct account credential is provided, the MAF reference Id will be returned in the response similar to the one shown below.

{"Status":"OK","Results":{"MAFs":[{"ID":"uk-nspd-paf"},{"ID":"uk-rm-paf"}}]}

Key in one of the ID values into the "Default MAF" textbox on a Flow Screen to specify the default MAF. In the above example, this can either be uk-nspd-paf or uk-rm-paf.

Note that in Lightning App Pages and Lightning Record Pages, the "Default MAF" drop-down list is pre-populated with item labels that are uppercased, each will link to the correct MAF reference Id internally.

A3. Destination addresses

For Account and Order entities Lightning App Pages and Lightning Record Pages, the Destination address is prefixed with “Billing” and “Shipping” values. When the Person Account feature is enabled in your Salesforce Organisation, “PersonMailing” and “PersonOther” address prefixes can are also supported for the “Account” entity.

For the Contact entity Lightning App Pages and Lightning Record Pages, the Destination address is prefixed with “Mailing” and “Other” values.

For the Lead entity Lightning App Pages and Lightning Record Pages, the Destination address is prefixed with the “Standard” value.

If the address is to be written to a set of custom address fields, please provide the prefixed text by setting the “Custom Address Prefix” LWC property accordingly.

Note that the Destination address LWC property is a textbox rather than a drop-down list, it is not set automatically, you will have to enter the value the Billing, Shipping, Mailing, Other, Standard or <custom prefix> in the textbox manually.

A4. Find the API name of a field

You will need to know the Salesforce API name of a field should you wish to provide field mapping information from the Configuration Page. These should be updatable input fields and their Salesforce API names are documented from the URL below.

https://help.salesforce.com/articleView?id=000327123&type=1&mode=1

For example, you can determine the Salesforce API name for “Annual Revenue” in the Account entity object is “AnnualRevenue” by navigating to **Setup** and follow steps below to display the screen as illustrated in Figure 21 for confirmation.

1. Click **Object Manager**.
2. Locate and click the “**Account**” object.
3. In the sidebar, click **Fields & Relationships**.
4. Locate the row with a value of “Annual Revenue” under the **FIELD LABEL** column.
5. The corresponding value for this row will be “AnnualRevenue” which can be confirmed under the **FIELD NAME** column.

Please ensure that valid Salesforce API names are provided. Unknown Salesforce API names will be ignored. Also the Salesforce API target fields will not be updated if one or more of the mapped data failed the target field validation.

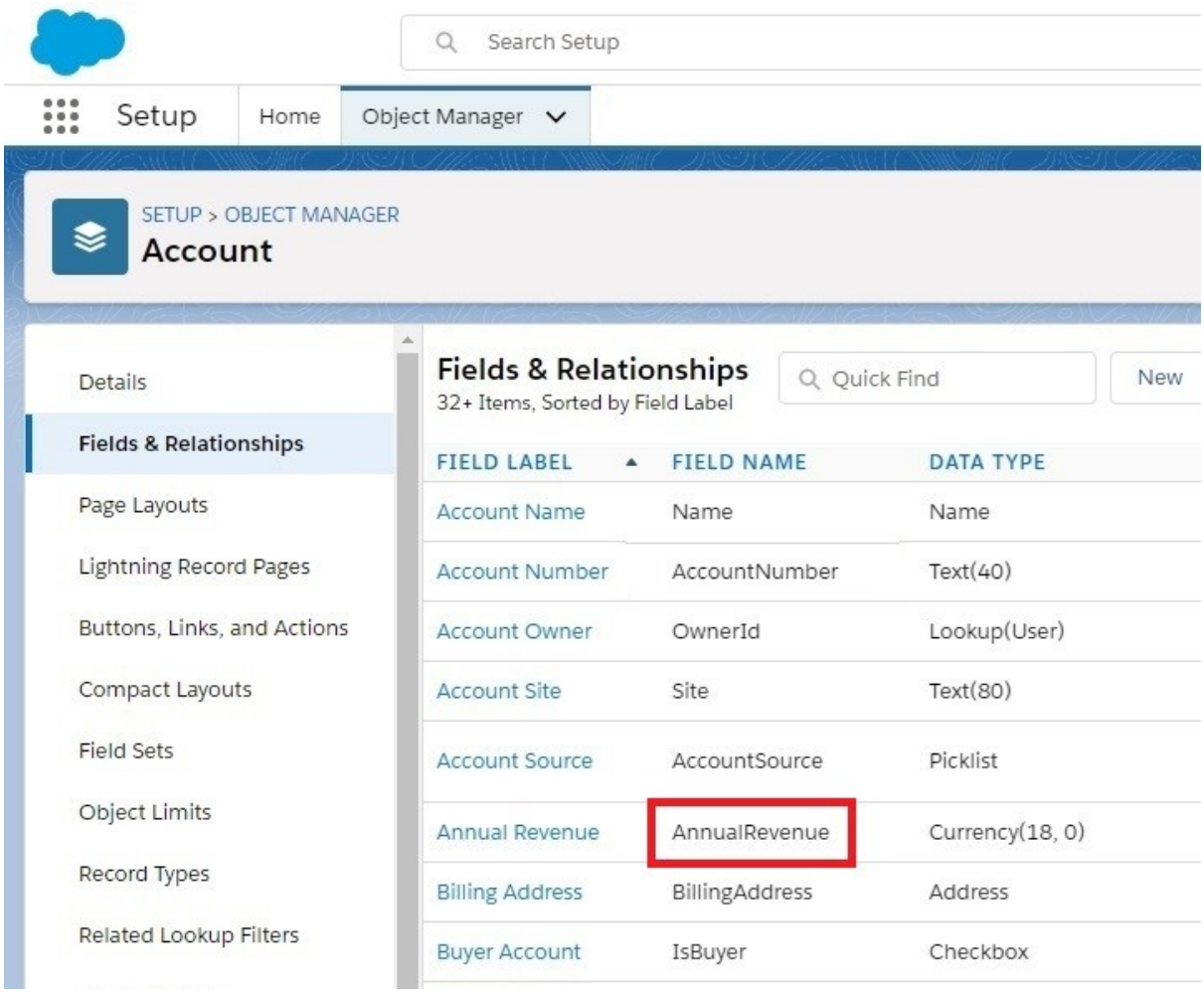


Figure 21. Identifying a Salesforce API field name.

A5. Default “hpw-autoc-jsclient.css” file

A default hpw-autoc-jsclient.css file is used to format the display of the dynamic menu for the Hopewiser AutoComplete LWC. It contains the following element styling definitions:

```
.lwc-hopewiser {
    background: #ffefd5;
}
.ui-widget {
    font-family: Arial,Helvetica,sans-serif;
    font-size: 0.85em;
}
.ui-autocomplete {
  max-height: 200px;
  max-width: 300px;
  overflow-y: hidden;
  overflow-x: hidden;
}
.ui-menu-item
{
  white-space: nowrap;
  text-overflow: ellipsis;
}
```

```
.ui-menu .ui-menu-item
{
  margin-top: 5px;
  margin-bottom: 5px;
  margin-left: 10px;
  margin-right: 10px;
  width: 90%;
}
.ui-widget-content a {
  color: #333;
}
.ui-widget-content i {
  color: red;
}
```

You can control the Hopewiser AutoComplete LWC display with your own css file. To do this, simply save a new static resource file (using the “hpw-autoc-jsclient.css” file as a template is recommended), then edit the CSS definitions for the appropriate elements as required. To activate the new CSS styling, you should enter the appropriate static resource name in the “**Custom CSS Resource**” textbox that is displayed in configuration page that is described in Section 3.4.

For example, change the “**overflow-y: hidden**” value to “**overflow-y: auto;**” will display the vertical scroll bar in the AutoComplete dynamic menu if the number of returned items can not fit into the window height of .ui-autocomplete which has been defined with a “**200px**” value.