



ADDRESS VERIFICATION

IMPLEMENTATION GUIDE



TO A **FINISH**

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

Implementation Guide

© 2011-2026 To A Finish LLC
12407 MoPac Expwy N • Suite 250
Austin, TX 78758

www.AddressVerificationApp.com

Table of Contents

<i>Introduction to this Implementation Guide</i>	5
Introducing To A Finish	6
Introducing the Address Verification App	7
Before You Begin, Requirements	8
<i>Initial Configuration</i>	9
Connecting Salesforce to Smarty.com	11
Entering Smarty API Keys	11
International Smarty Verification Options	12
Optional Settings	13
Authentication Tab Options	13
Options Tab Options	14
Configuring Physical Addresses	16
Configuring Phone Numbers	21
Configuring Email Addresses	22
Status Results for Verified Records	23
Physical Address Return Codes	23
Phone Number Return Codes	23
Email Address Return Codes	23
Email Address Return Sub Codes	24
Remote Site Entries	25
<i>Configuring Standard Objects</i>	26
Accounts, Contacts & Leads	27
Turning on Field History (Optional)	27
Modifying Layouts	28
Adding “Verify” Components to the Layout	30
<i>Configuring Custom Objects</i>	32
Physical Address Configuration	32
Creating Mandatory Fields	32
Strongly Recommended – But Still Optional – Fields	33
Additional Optional Fields	33
Creating Optional Formula Fields	34
Creating International Fields	36
Creating Custom Address Records	37

Phone Number Configuration	38
Creating Mandatory Fields	38
Creating Optional Fields	38
Creating Custom Phone Number Records	38
Creating Optional Formula Fields	38
Email Address Configuration	39
Creating Mandatory Fields	39
Creating Optional Fields	39
Creating Custom Email Address Records	39
Creating Optional Formula Fields	40
Creating a Custom “Verify” Action	41
Adding “Verify” Components to the Layout	42
Adding the “Verify and Confirm” Component	42
Adding the “Verify and Save” Button Component	43
Adding the “Background Verification Tracker” Component	44
Un-Verifying Records on Change	45
Finalizing Custom Object Setup	47
Automation and Bulk Verification	48
Manual Bulk Verification	48
List Verification for Standard Objects	48
List Verification for Custom Objects	50
Automated Verification Options	51
Turning on Trigger Verifications	51
Creating Triggers for Custom Objects	52
Check for Existing Records	53
Verifications Using Flows	54
Bulk Verifications With Batch Jobs	57
Email-Powered Record Verification	60
Turning Off All Triggers	62
Logging Verification Results	62
Advanced Options	63
Setting Up Autocomplete	64
Initial Configuration	65
Named Credential for Security	66
Child Object Usage	71
Important Notes About Overriding New/Edit Functions	74
Advanced Topics	75
Using Google Search to Improve Verifications	75
Address Verification in a Flow	76
Feedback and Remote Access	77



Introduction to this Implementation Guide

Introducing all the key components to the Address Verification App for Salesforce

This manual will help you through the implementation of the Address Verification App for your Salesforce.com Org. You can read this chapter if you desire to do so, but most users can probably start at the next chapter, where the installation and configuration instructions begin.

Thank you for considering our app in your organization.

I C O N K E Y	
 Details You Can Skip	To succeed in the installation process , you will need to pay close attention to all of the information contained in this manual. We recommend reading through this first chapter before you begin, and then reading through each section as you work on it. The Icon Key to the left shows the icons you will be seeing as you go through this manual. The more advanced readers can skip some of the details as indicated by the appropriate icon.
 Don't Miss This!	
 Sample Code	
 Danger!	

This manual is a work in progress, so please do take notes and let us know if you run into any issues with the text or if you believe there is a better way to accomplish the tasks we describe.

Introducing To A Finish



To A Finish™ is a concept company that initially grew out of a desire to improve the Salesforce CRM consulting industry, and eventually moved exclusively to creating business apps for Salesforce. Too often, we have found the priorities of consulting companies as well as app-makers skewed towards speed and profitability, rather than excellence and long-term success. While not a blanket statement about all consulting and app companies, it is a general statement about the industry at large.

We believe there is a different way of consulting that will bring ultimate success to any CRM venture. The biggest difference is time. We give special attention to steps that are often rushed or even overlooked entirely in a typical project. Things like in-depth analysis, getting to really know a business, documentation and end-user involvement in the process help us set up a project for success. Ultimately, we must spend the time to do the work until it is absolutely complete—To A Finish™.

We are not guaranteeing our clients will never see a bug or have to rethink initial strategies. Reworking is actually part of the process of achieving success. What we are saying is that our strategies and focus will minimize unforeseen problems and achieve for us an ultimately successful project completion. We put our name on our work and stake our reputation on quality service.

Test our apps and our service. Allow us to show you work done To A Finish™.

You can find more about To A Finish at www.toafinish.com



TO A FINISH

When you finish a thing, you ought to be able to say to yourself: "There, I am willing to stand for that piece of work. It is not pretty well done; it is done as well as I can do it; done to a complete finish. I will stand for that. I am willing to be judged by it."

- O.S. Marden

Introducing the Address Verification App



Address Verification for Salesforce features a very simple way to cleanse physical addresses, email addresses, and phone numbers within your Salesforce system. Automatically, or with the touch of a button, contact information is verified and cleansed. Works in standard or custom objects.

Try it for free for 30 days!



The primary use of this app is to work with the top-rated Address Cleansing service, Smarty.com, with Email and Phone Number verifications also supported fully. Other functions are planned, including data augmentation with company information.

The Address Verification for Salesforce app provides triggers and buttons on the Account, Contact and Lead layouts which instantly verify and cleanse the standard physical addresses, emails and phone numbers. Easy instructions for adding buttons for custom objects and fields are provided in later sections of this manual.

Regardless of the type of verification you do, not only statuses are returned, but also additional data like Latitude and Longitude, County Name, and much more, are included, depending on your configuration and needs.

Address Verification is simple in its execution, and powerful in its delivery. Built for service, not for selling.

Before You Begin, Requirements



Before you begin working with our app, you must have several components already in place and configured. We will not cover these components in detail, but will at least point you to where you can get the information you need.

Salesforce.com A production system Salesforce.com Org will be needed in order to set up the integration. A sandbox will be very helpful in this process as well. Any edition of Salesforce.com, starting with Enterprise, should work.

This Implementation Guide only covers configuration in the Lightning Experience. Most of it will work with Classic, except for a few specific differences, which are covered in the [Implementation Guide for Classic Features](#).



Smarty Account Option Smarty.com (formerly SmartyStreets) offers an address cleansing service, which our app uses. Please keep in mind that Smarty.com is a separate company from To A Finish LLC, which we partner with. They curate the address information, and we provide the functionality to use their service inside of Salesforce.

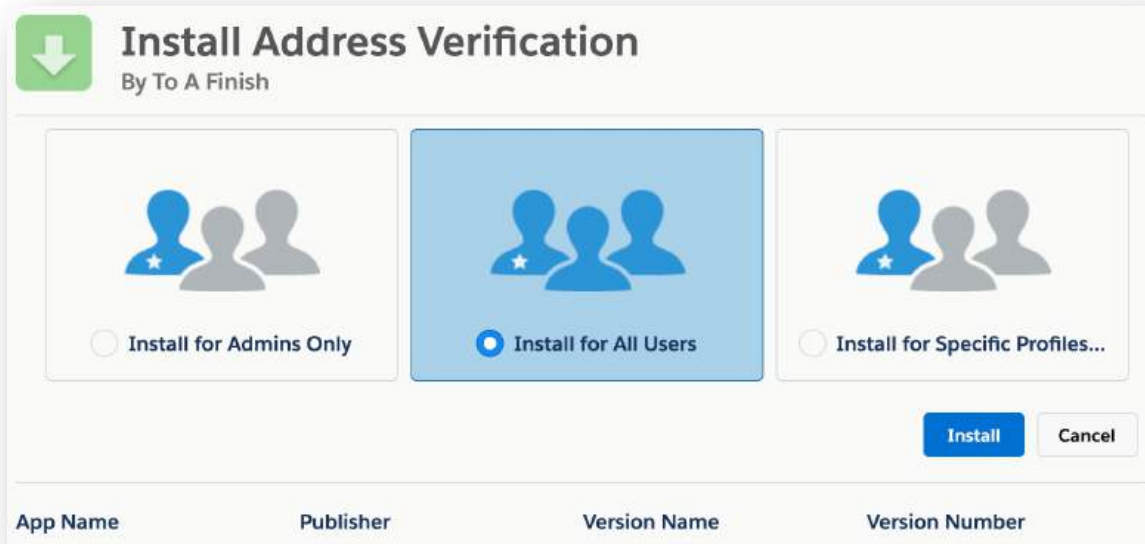
We resell their Address Verifications (and provide a discount on the app if you purchase through us), but you can use your own account with them directly if you choose to do so. This is recommended when initially testing the app. Note:

- Smarty offers a free trial plan with 1000 US address lookups. Please use this as a way to test our app at no cost.

Initial Configuration

The steps necessary to set up the app for first-time use.

The Address Verification app can be installed through the AppExchange.com listing or via a direct link in the case of installations that require an invoice, or when purchasing both the app and a Smarty package at the same time. Either way, the first question the installation asks is very important. Who should the app be installed for? The answer almost always is: **Install for All Users**.



Install Address Verification
By To A Finish

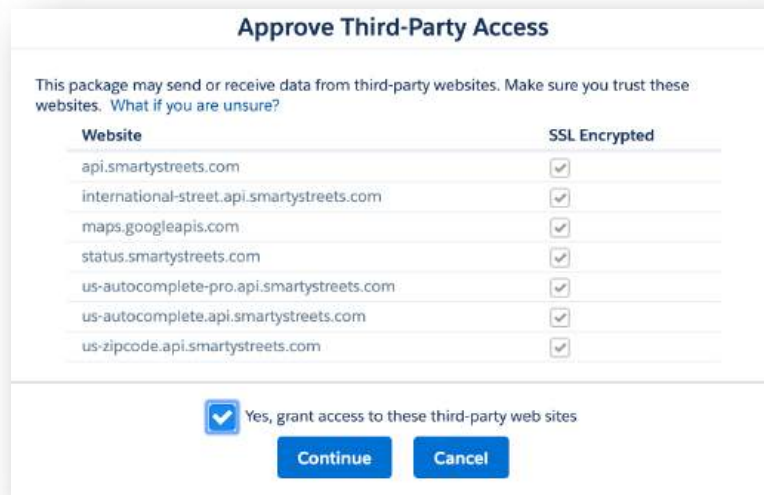
☐ Install for Admins Only

☒ Install for All Users

☐ Install for Specific Profiles...

Install **Cancel**

App Name	Publisher	Version Name	Version Number
----------	-----------	--------------	----------------



The next question is also equally important: Grant Access to Third-Party Websites? You will need to allow access to these APIs, so for now, check the box and press the “Confirm” button.

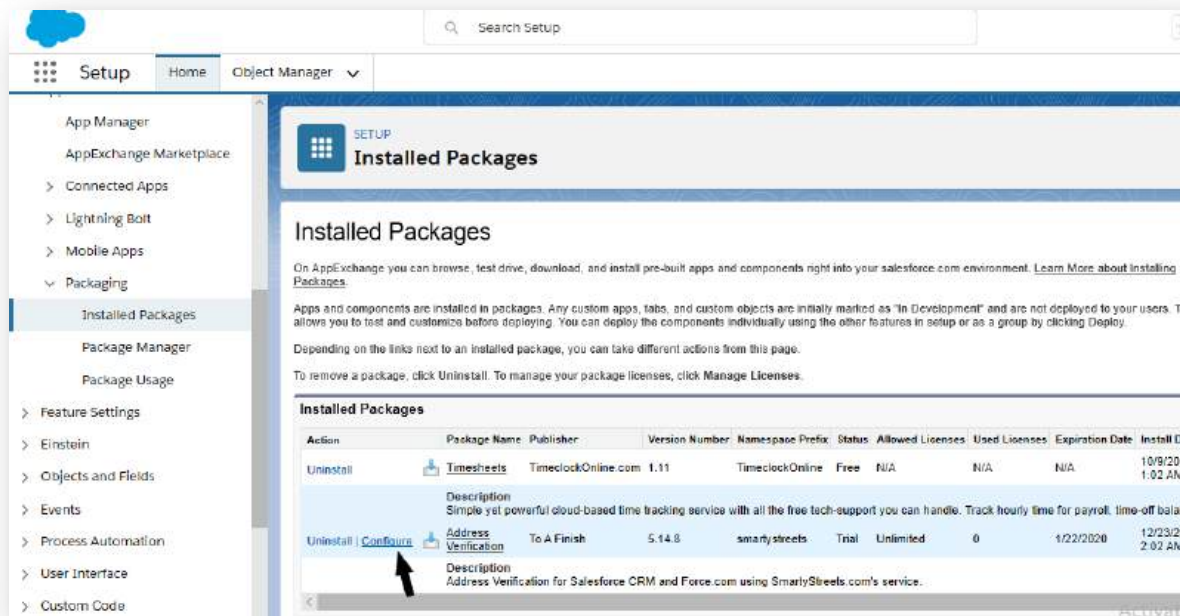
See the last page in this chapter for which ones you can optionally turn off if you want to.

After the installation, the initial steps in the configuration are completed by clicking the “Configure” button on the Address Verification package

and then pasting in the Credentials you will pull from the installation process detailed below. If you have already left the package installation screen, you can get back to the configuration page by clicking:

Setup → Installed Packages → Address Verification → Configure

Click on the “Configure” link next to the Address Verification package.



Connecting Salesforce to Smarty.com

After installing the Address Verification app, please verify that you have received at least one set of authentication credentials from our team.

Below, you will begin with Physical Address configuration and then move on to Email and Phone number configuration. Skip over whatever is not applicable to you.

Entering Smarty API Keys

We are now ready to set up the connection to Smarty.com in order to verify physical addresses. At this point you should have the Smarty.com API Authentication Keys ready to be used. Switch to the configuration page for the Address Verification app in Salesforce.com and simply paste them in the location provided. Then press the green **Save All Settings** button *towards the bottom of the screen*, and you are done.

Authentication Data Mapping Options Automation More...

SMARTY.com ACCOUNT CONFIGURATION

You will need to log into Smarty.com to get the auth-id, auth-token and website key. See the [Implementation Guide](#) if you run into issues finding them. **If** you do not have an account with Smarty.com, click the button below to sign up for a trial account with 250+ API calls.

Please note that International Address Verifications are now supported.

 Authentication Configuration

▼ Address Verification Default

Auth-Id (Default)
.....

Auth-Token (Default)
.....

License
Select an Option ▼

Verify

> Address Verification International

> Autocomplete Domestic

> Autocomplete International

> Enrichment

Save All Settings Reveal Encrypted Settings

You may press the **Verify** button *after* saving, to ensure the authentication codes work, but keep in mind that this will use up one of your address lookups.

- *Note: The secondary sets of authentication credentials are not necessary unless you have multiple accounts. They are there so that IF you want to use a different Smarty.com account for one of the Autocomplete APIs or for International address lookups than the account you indicated in the default set of authentication credentials, then you can enter more credentials in these boxes. If you do not enter other sets, then it will attempt to use the default set.*

International Smarty Verification Options

You can turn ON or OFF international verifications from the Options tab. Please realize that if you turn it on, then any time you have an international address, it will try to verify it using the International API from Smarty. If this is not checked, then any address with a country that is not blank, or some iteration of “USA” will be ignored.

In order for International verifications to work, you need to purchase an International Plan and possibly enter the auth-id and auth-token for that plan in the Configuration page. However, even if you haven’t turned it on with Smarty the app will attempt to verify the address, so in order to not see errors, you want to make sure Smarty is set up for International verifications before turning it on.

- ❖ *Note that you have the ability to add an additional set of Authorization Credentials that will be used exclusively for International Verifications. This is helpful in case you have multiple accounts.*

Optional Settings

The following options can be configured in order to change the overall way that Address Verification works. For options that are specific to each individual type of address, see the section a few pages forward called *Setting Up Addresses*.

Authentication Tab Options

Option	Description
fixed name for USA verifications	By default, when you verify an address in the United States, the Country field is ignored. If you want to automatically populate something in this field, set this field to USA , US or United States in order for it to function correctly.
default autocomplete country name	This setting should be left at “USA” or similar unless an International Autocomplete package is set up. Keep in mind that the 2-character code (e.g. “US”) must be put here if State/Country Picklists are turned on.
default autocomplete country name recordtypes	This option works with the one above. Put the actual ID of the recordtype in this field if you want the default country to show up only sometimes. E.g.: if you have a specific record type that is not for your most common country, then you would put all recordtype ids in this field, except for the id of the record type for foreign addresses.
(pro only) Autocomplete Filters	The place to put filters for the AutoComplete Pro API. For example, if you want to have only results from a certain State or range of Zip Codes, you could use a filter as described in this documentation from Smarty.
only use states in these countries (use for country/state picklists only)	When using Country/State picklists, Salesforce enforces the list of States that you set up. So, if Smarty returns a value for a Region/State that is slightly different from what is configured, then Salesforce will reject the update. It is recommended that you populate this field so that only the countries you list will use States. <i>Note: use ISO 2-character country codes (can be found online.) Include the US. For example, if you are maintaining lists of States/Regions in the United Kingdom, France and Spain, you would enter: US, GB, FR, ES in this field.</i>
Countries with Basic Package	Seldom used option, only used when a specific agreement is in place for address verification for certain countries.
Admin Emails	Put an email address, or multiple email addresses separated by commas, in order to specify where any errors should be sent. If nothing is entered here, then the last three administrators who logged into Salesforce will be sent the errors.
Verification Premise Minimum	This is for the International API only. If you do not select this option, then addresses might be “confirmed” even if the house or building number cannot be found, and they cannot be verified beyond the Thoroughfare (Street). The International API can verify at various levels, which are different for each country , so turn on this option if you want to be as sure as possible that you can mail something to the address. The levels of verification are, from best to worst: <i>Delivery Point — rooftop level, Premise —</i>

Option	Description
	<i>property or building level, Thoroughfare — street level, Locality — city or town level, Administrative Area — state or province level, None — unknown address.</i>
Ignore User ID	This field should contain the Salesforce User ID of the one User who our application should ignore. This could be a Site Guest User, or an Administrative user tasked with doing updates to the system. If the 15 or 18-character User ID is saved here, that user will be ignored by any trigger verifications or un-verifications. It also stops any flow verifications, though it does not stop a flow from un-verifying a record if set up to do so. (Manual verifications are not at all affected.)
Name of Named Credential	The name used when creating a Named Credential for use in Autocomplete screen building. This is set up in Chapter 5: Autocomplete.

Options Tab Options

Option	Description
Verify International	Note: This requires an addon Purchase. Contact sales@toafinish.com for pricing
Use Enhanced Matching for US Addresses	Uses new "core" license, which includes enhanced matching and 15,000,000+ non-USPS addresses. <i>Note: You will also need to turn on the “Less Strict Verifications/Enhanced Data” option for EACH address you use this on. → This option is now turned on by default and can only be turned off via a ticket with our support team.</i>
Use Google Search	Will use Google Search to “fix” any addresses that are not found when verifying. Please see the full description at the beginning of chapter seven. Addresses failing verification, or only partially verified, will be searched on Google and if a valid "Did You Mean?" entry is returned, will be verified a second time (causing 2 API calls).
Use Long Country Names	Turns the 2-character country abbreviations to the full, English, country name. Do not use this if you have State/Country picklists turned on in Salesforce.
Use 3 Character Country Codes	Turns the 2-character country abbreviations to ISO 3-character code. Do not use this if you have State/Country picklists turned on in Salesforce.
Use Long State Name	Turns the 2-character state abbreviation to the full, English, State name. Do not use this if you have State/Country picklists turned on in Salesforce.
Use AutoComplete Pro	Turns on the AutoComplete Pro API in the standard AutoComplete Edit and New Lightning components in Salesforce. This will only work if you have purchased an AutoComplete Pro package. → This option is now turned on by default and can only be turned off via a ticket with our support team.

Option	Description
Do Not Verify After AutoComplete	Will restrict the AutoComplete New and Edit Lightning components from using the standard US or International Street Verifications after having already verified the address via the AutoComplete Pro API. Use this only if you need to save verification calls. Please note that none of the additional fields will be populated if you choose this option (Zip Plus 4, County, etc.)
Turn OFF All Triggers	Checking this box will turn off all triggers configured on the Automation tab. This includes auto-verification and also the function that changes the Address Status to Unverified when address fields are changed. If this is selected, trigger options below are ignored. (Flows and Processes are not affected.)
Log Responses	Checking this box will log calls and the response from Smarty into the Verification Logs object. Usually, only errors get logged. Please keep in mind that this object could grow very large, depending on how many verifications are performed. To see all of these logs, create a tab for the "Verification Logs" object.
Allow 1-Line Address Verifications	Checking this box will allow us to try to send Smarty addresses for verification if they do not have a zip code, and also do not have a city and state. (For cases where the whole address might be packed into the Street field.) Turning this on could cause a potentially significant number of addresses to be sent to Smarty for verification when they are not entirely populated. So do this only if you are sure you want to verify no matter what, when a Street is populated.
Restrict Reverifications	Checking this box modifies the way triggers sense if a change has been made to the address. If it is unchecked, then any change to the address will trigger a change to "Not Yet Verified" status. If checked, it will attempt to determine if the change is small enough to be insignificant. This setting is recommended if addresses are being updated by an external system, these changes may not require a re-verification.
Copy Account Contact Verifications	Checking this box will attempt to push Account Billing address verifications to any contacts with the same address, and a Contact's verification to the Account Billing address if it has the same address.
Verification Premise Minimum	This is for the International API only. If you do not select this option, then addresses might be "confirmed" even if the house or building number cannot be found, and they cannot be verified beyond the Thoroughfare (Street). The International API can verify at various levels, which are different for each country , so turn on this option if you want to be as sure as possible that you can mail something to the address. The levels of verification are, from best to worst: <i>Delivery Point — rooftop level, Premise — property or building level, Thoroughfare — street level, Locality — city or town level, Administrative Area — state or province level, None — unknown address.</i>
Treat US Territories as USA (not their own country)	When this option is turned on, US Territories will be treated as a part of the United States, and not as their own country. For example: an address in Puerto Rico would set "USA" as the Country, rather than "PRI" (assuming 3-character countries were being used).

Configuring Physical Addresses

Now we must set up the addresses that the Address Verification app will support. Click on the “Data Mapping” subtab in the Setup page, and then press the **Create Default Addresses** button in order to create the list of 5 default addresses (6 if you have the “Person Accounts” feature turned on.). You might need to refresh the browser page to see them.

Authentication **Data Mapping** Options Automation More...

The button below will create addresses with all necessary settings for the standard objects. You can access them later through this [Data Mapping] tab.

[Create Default Records](#) [Add Supported Object](#)

▼ Address Configuration List

Action	Name	Active ?	Object	Street	City	State	Zip
Edit Del	Account Billing	☒	Account	BillingStreet	BillingCity	BillingState	BillingPostalCode
Edit Del	Account Shipping	☒	Account	ShippingStreet	ShippingCity	ShippingState	ShippingPostalCode
Edit Del	AddressPicklist	☒	AddressPicklist__c	PStreet__c	City__c	PickState__c	PZip__c
Edit Del	Campaign	☒	Campaign	Street__c	City__c	State__c	Zip_Code__c
Edit Del	Contact Mailing	☒	Contact	MailingStreet	MailingCity	MailingState	MailingPostalCode

Editing Existing Addresses

If you click the **Edit** link on any of the addresses, a popup will show you many different fields and options. Following is a description of all fields:

Data Element	Notes about the field
Required Fields	
Name	Can be anything you want
Object	Select the object that you want to create an address for. If it is not on the list, close the popup and use the Add Supported Object button to add the object.
Active?	Unchecking this field will turn off any verification of the current address. This has the same effect as deleting the address record from the list, but allows you to return and make it active at a later date.
Street	Required
City	Required
State	Required
Zip Code	Required
Country	Required
Is Person Account?	This only is available for the Account object and indicates a Person Account record.
Verified	Required. Checkbox field that will be checked when an address is verified. This is used to ensure that it is not verified again if a verification process is run against the record.
Last Verified	Required. Date/Time field that keeps track of the last time the address was verified. (Even if it is no longer verified.)
Return Code	Required. This field will be populated with the result of the verification once it is attempted. Possible values are listed after this table.
Footnotes	Required. Notes about the verification that was performed

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

Data Element	Notes about the field
Optional Fields	
Street2	Second line of the address. If no field selected, then the 2 nd line of the address will be put on the end of the first Street field.
Street3	Third line of the address. Not used for US Addresses, but may include an “Attention” indicator or non-address information.
Street4	Fourth line of the address. Not used for US Addresses.
Unit	Secondary Number, like Apartment, Unit or Suite
County	
County FIPs Code	
Island	This information is not from Smarty, and only supports the Hawaiian and two sets of Spanish islands.
Do Not Verify	Checkbox field that can be used to ensure the particular record is not verified.
Clean Street	Populated with a stripped-down version of the street address, calculated manually, to be used in matching addresses against other records.
Time Zone	
DST	Daylight Savings?
UTC Offset	Number of hours from the UTC standard. Used along with DST to calculate the current address.
Barcode	Barcode value, used in a formula field to visually create a barcode.
Urbanization	Designation only for Puerto Rico
RDI	Residential Delivery Indicator (Residential vs Commercial)
Latitude	
Longitude	
Record Type	Accuracy of verification in descending order: (F)irm, (G)eneral Delivery, (H)igh-Rise, (P).O. Box, (R)ural Route, and (S)treet.
Carrier Route	
Congressional District	
Is Active?	From the USPS, currently delivering mail at this address.
Is Vacant?	From the USPS, have delivered mail, but marked vacant after nobody picked up for a certain amount of days.
Verification Used	Which verification service was used: Smarty, Google Maps, or UPS
Verified by User	UserID of the Salesforce user who verified the address
Zip Plus 4	Field to use in order to split the Zip Code saved into two values,
Match Details	Will be populated with one of the following and is useful for analysis: • none — No address match was found. • non-postal-match — A match was found within additional, non-postal address data. • postal-match — A match was found within postal address data. • missing-secondary — The address should have a secondary (e.g., apartment), but none was found in the input.

Data Element	Notes about the field
	unknown-secondary — The provided secondary information did not match a known secondary within the address data. ignored-input — The provided input contained information that was not used for a match.
Smarty Key	Smarty's unique identifier for an address. This data element is needed for calling the Address Enrichment APIs, so we recommend always populating a field with this value.
Smart Key Version (Principal)	Used in Address Enrichment API calls. Specifically for US Property Data (principal) calls. This versioning reduces the number of calls made to the Address Enrichment APIs since a lookup call is not counted if the version did not change. Also includes an “Ext” field.
Smart Key Version (Geo)	Used in Address Enrichment API calls. Specifically for GeoReference calls. Also includes an “Ext” field.
Smart Key Version (Financial)	Used in Address Enrichment API calls. Specifically for US Property Data (financial) calls. Also includes an “Ext” field.
Smart Key Version (Business)	Used in Address Enrichment API calls. Specifically for Business Data. (Currently in development.). Also includes an “Ext” field.
Smart Key Version (Secondary)	Used in Address Enrichment API calls. Specifically for US Secondary Data. (Not currently used in this version of the app.). Also includes an “Ext” field.
Internal Zip Code Database Fields	
Area Code(s)	Not from Smarty. Only available with the optional internal Zip Code Database. Valid Phone Area Code(s) for the zip code.
Urban Area (Code)	Not from Smarty. Only available with the optional internal Zip Code Database.
Urban Area Name	Not from Smarty. Only available with the optional internal Zip Code Database. Description of the code. If used, must include also a field for the Urban Area Code.
International Only Fields	
Organization	Seldom populated. The name of the recipient, firm, or company at this address
Address Format	Technical information about how the address should be formatted for shipping.
Address Precision	How precise is the address? From best to worst: Delivery Point — rooftop level, Premise — property or building level, Thoroughfare — street level, Locality — city or town level, Administrative Area — state or province level, None — unknown address. The International API can verify at various levels, which are different for each country .
Administrative Area	The most common administrative division within a country (e.g. Region)
Complete Address	Completely formatted address using the format in the “Address Format” element in order to maintain a perfectly formatted address for the intended destination.
Dependent Locality	
Dependent Locality Name	
Geocode Precision	Indicates how precise the geocoding of the address may be. Different Smarty.com APIs provide different precisions. Please contact sales@toafinish.com if a high level of precision is needed.) Same values as the Address Precision.
Premise	Alphanumeric code pertaining to an individual location. Usually a house or building number.

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

Data Element	Notes about the field
Premise Extra	Extra information about the premise that is not necessarily authoritative but might still be useful
Post Box	
Sub Building Number	The alphanumeric component of the sub building
Sub Building Type	The leading sub-building type of the sub building
Super Administrative Area	The largest administrative division within a country
Thoroughfare (Complete)	The name component of the thoroughfare along with the designation (e.g. Main St)
Thoroughfare Name	The name component of the thoroughfare (e.g. Main)
Force Latin?	Will force the address to use Latin characters when saved to Salesforce
Force Native?	Will force the address to use characters for the local language, if different from Latin characters. Only available in certain countries.
Address Options	
Don't Overwrite Street1	Will not overwrite the entered Street line with what Smarty returns.
Don't Overwrite City	Will not overwrite the entered City with what Smarty returns.
Don't Overwrite State	Will not overwrite the entered State with what Smarty returns.
Don't Overwrite Zip Code	Will not overwrite the entered Postal/Zip Code with what Smarty returns.
Truncate Values Longer Than Mapped Fields	When this option is selected, the Lightning component will check to see if the fields are long enough to contain the data, and if not, will automatically truncate it in order to fit in the fields. This might leave bad data that should not be used for shipping. Use the "Complete Address" field as a backup if this option is necessary.
Clear all verified fields after address change	By default, when the Street, City, State or Zip fields are changed, only the Verified and Return Code are modified. If this is selected, then all configured fields are cleared.
Clean Address Prior to Verification	With this option selected, the field indicated in the "Clean Street" (see above) will be populated with a modified version of the Street 1 address which seeks to standardize the address using a proprietary formula. This "Clean Street" should NOT be used for shipping purposes, it is exclusively to be used in deduplication efforts.
Only 5 Digits on Zip Code	Will only populate the first 5 digits of the Zip Code (not the additional 4 digits)
Separate Unit Field	When this option is selected AND when the Unit field is populated (above), the app will save the Unit only in the Unit field and not on the first line of the address, as would typically be the case in order to CASS-certify the address according to USPS standards.
Less Strict Verifications /Enhanced Data	This will allow addresses that are believed to exist but are not registered officially with the USPS to verify with a "Confirmed – May be Undeliverable" return code. <i>Note: This option is required in order to use the Enhanced Matching of the new US Core API for US Street Address verifications.</i>

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

Data Element	Notes about the field
Populate Info from Unknown Addresses	Brings in as much information as is known about an address even if the address is not registered officially with the USPS or is even known to be a bad address. If no record an address is found at all, this will do nothing.
Remove Additional Info from Street 1	With this option selected, it will try to remove anything that doesn't look like it is part of the main part of the street address from the Street 1 field prior to sending it to Smarty. <i>Note: this includes Suites, Apartments, Buildings, etc., and in many cases will render an address undeliverable. Use with caution.</i>
Run Enrichment APIs	Must be checked to call the Address Enrichment API when running a verification on an address. Address Enrichment calls are secondary calls that happen after the first lookup provides the Smarty Key data element.
Allow Verifications with no Street	Will use the Smarty Zip Code Lookup API to verify only City, State and Zip Code fields if the Street is not populated. Please realize that this will count as a verification the same as any other, even though it has very little information.
Use FedEx Address Standards	This will enforce standards from FedEx on addresses, specifically length of address fields. Any additional information on the Address Line 1 field would be moved to the Address Line 2 field, and other fields, like the City would be truncated in the event that the value returned is longer than the FedEx standards (not very common).
Always Use Locality for City on International	International address data sometimes varies depending on the country. This setting seeks to standardize the "City" field to always be populated with what is in the "Locality" data element. Please contact our team if you need more information on this option.
Autocomplete Address Options	
Stop If Not Fully Confirmed	Will stop the user from creating or editing a record if the address will not result in a completely confirmed status. If the address would be bad, or even partially confirmed, the user will not be allowed to continue.
Stop record creation on bad addresses	When using the AutoComplete feature, checking this box will remove the "Use as Is" button when an address is not able to be verified.
Use Street 2	Adds a field for Street 2 in the AutoComplete function. For this to work, you also need to select a Street 2 field to populate.
Use Street 3	Adds a field for Street 3 in the AutoComplete function. For this to work, you also need to select a Street 3 field to populate.
Use Street 4	Adds a field for Street 4 in the AutoComplete function. For this to work, you also need to select a Street 4 field to populate.
Do Not Check for Addresses	When this option is selected, the Autocomplete feature will not enforce entry of an address for a record of this type. If an address is entered, it will be processed, but if empty, no error will be shown.
Copy Parent to Child Address	When this option is selected and a Contact is being entered using a custom Lightning Component (see the <i>Child Object Usage</i> section in Chapter 6), For example, for Contacts, the Billing Address from the Account would be copied to the Contact Mailing address fields.

Configuring Phone Numbers

Now, if we have a Phone Number Verification addon package, we can configure as many phone numbers as we want to verify. Contrary to Physical Addresses, there are no phone numbers configured by default. To configure a phone number, you must click the **New Phone** button.

Following is a description of all fields:

Data Element	Notes about the field
Required Fields	
Name	Can be anything you want
Object	Select the object that you want to create a phone number for. If it is not on the list, close the popup and use the Add Supported Object button to add the object.
Active?	Unchecking this field will turn off any verification of the current phone number. This has the same effect as deleting the phone number record from the list, but allows you to return and make it active at a later date.
Phone	Required
Verified	Required. Checkbox field that will be checked when a phone number is verified. This is used to ensure that it is not verified again if a verification process is run against the record.
Country	Required. A field containing the 2-character Country Code of the Phone number. If this field is blank, then "US" will be assumed. If the field has the full country name, or the 3-character abbreviation, it will still work, but has to go through the extra step of figuring out which country it is.
Return Code	Required. This field will be populated with the result of the verification once it is attempted. Possible values are listed after this table.
Last Verification	Required. Date/Time field that keeps track of the last time the phone number was verified. (Even if it is no longer verified.)
Phone Response	Required. This should be a long text field and will contain the actual data that is returned by the API call. It is not intended for display.
Optional Fields	
Carrier	Phone provider
Location	Location of owner, usually the city, often where it was purchased.
Line Type	Type of phone line (mobile, landline, etc.)
Country Full Name	Full Name of the Country that was passed in via the Country field
Clear all verified fields after Phone Change	By default, when the Phone field is changed, only the Verified and Return Code are modified. If this is selected, then all configured fields are cleared.

Configuring Email Addresses

Now, if we have an Email Address Verification add-on package, we can configure as many email addresses as we want to verify. Contrary to Physical Addresses, there are no email addresses configured by default. To configure an email address, you must click the **New Email** button.

Following is a description of all fields:

Data Element	Notes about the field
Required Fields	
Name	Can be anything you want
Object	Select the object that you want to create an email address for. If it is not on the list, close the popup and use the Add Supported Object button to add the object.
Active?	Unchecking this field will turn off any verification of the current email address. This has the same effect as deleting the email address record from the list, but allows you to return and make it active at a later date.
Email	Required
Verified	Required. Checkbox field that will be checked when an email address is verified. This is used to ensure that it is not verified again if a verification process is run against the record.
Last Verification	Required. Date/Time field that keeps track of the last time the email address was verified. (Even if it is no longer verified.)
Return Code	Required. This field will be populated with the result of the verification once it is attempted. Possible values are listed after this table.
Optional Fields	
Domain Age Days	Number of days since the email domain was registered on the internet. (Divide by 365 for the number of years)
Free Email Account	True/False checkbox field indicating a “throwaway” email address, typically set up only for the purpose of providing a temporary email address, but not a real address to be used going forward. We recommend never emailing these addresses.
First Name	First Name of registered owner. (If known.)
Last Name	First Name of registered owner. (If known.)
Email Response	Actual JSON returned by the verification request.
Gender	Sex/Gender of owner, if known. (male, female)
SMTP Provider	Email host
Sub Status Code	Additional information about the email status, if available
Clear all verified fields after Phone Change	By default, when the Phone field is changed, only the Verified and Return Code are modified. If this is selected, then all configured fields are cleared.

Status Results for Verified Records

Physical Address Return Codes

Following are the possible return codes on addresses configured.

Status	Description
Confirmed	Fully verified address
Confirmed - Address Truncated	Verified, or partially verified address, too long for one or more mapped fields
Confirmed - By Dropping Secondary Info	Fully verified only after dropping secondary number on the street
Confirmed - May be Undeliverable or Confirmed - Partially	Partially verified as a possible address; not likely to receive mail from the USPS
Confirmed - Missing Secondary Info	Partially verified with missing secondary number on the street
Not Submitted	Ignored address when processing verifications (rarely used)
Changed	Previously verified address changed and not yet submitted for reverification
Bad or Blank Address	Submitted to verification and failed or blank and not submitted at all

Phone Number Return Codes

Following are the possible return codes on phone numbers configured.

Status	Description
Valid	Confirmed valid phone number
Error	Unknown phone number, or known wrong number

Email Address Return Codes

Following are the possible return codes on email addresses configured.

TEXT CODE	DESCRIPTION	OK TO SEND?
valid	Verified as real address	Yes
invalid	Verified as not valid	No (see sub-codes on next page)
disposable	A temporary, disposable address	No
catch-all	A domain-wide setting, where all are accepted (Unverifiable)	Not for Sales/Marketing, unless on private server
do_not_mail	Verified email, usually role-based	Not for Sales/Marketing (see sub-codes on next page)
abuse	Verified emails of abuse-clickers	Not for Sales/Marketing
spamtrap	Verified bad email	No
unknown	The server cannot be reached	No (see sub-codes on next page)

Email Address Return Sub Codes

Following are the possible return codes with some of the possible sub-codes on email addresses configured.

CODE	SUB CODE	DESCRIPTION
invalid	does_not_accept_mail	These domains only send mail and don't accept it.
	failed_syntax_check	Emails that fail RFC syntax protocols
	possible_typo	These are emails of commonly misspelled popular domains. These emails are marked invalid.
	mailbox_not_found	These emails addresses are valid in syntax, but do not exist. These emails are marked invalid.
	no_dns_entries	These emails are valid in syntax, but the domain doesn't have any records in DNS or have incomplete DNS Records. Therefore, mail programs will be unable to or have difficulty sending to them. These emails are marked invalid.
	mailbox_quota_exceeded	These emails exceeded their space quota and are not accepting emails. These emails are marked invalid.
	unroutable_ip_address	These emails domains point to an un-routable IP address, these are marked invalid.
do_not_email	global_suppression	These emails are found in many popular global suppression lists (GSL), they consist of known ISP complainers, direct complainers, purchased addresses, domains that don't send mail, and known litigators.
	possible_trap	These emails contain keywords that might correlate to possible spam traps like spam@ or @spamtrap.com. Examine these before deciding to send emails to them or not.
	role_based	These emails belong to a position or a group of people, like sales@ info@ and contact@. Role-based emails have a strong correlation to people reporting mails sent to them as spam and abuse.
	disposable	These are temporary emails created for the sole purpose to sign up to websites without giving their real email address. These emails are short lived from 15 minutes to around 6 months. There is only 2 values (True and False). If you have valid emails with this flag set to TRUE, you shouldn't email them.
	toxic	These email addresses are known for abuse, spam, or bot created emails. If you have valid emails with this flag set to TRUE, you shouldn't email them. We also provide other additional fields that you should take into consideration before emailing. If any of your valid emails have the disposable or toxic flag set to true, we recommend that you don't email them.
	role_based_catch_all	These emails are role-based and also belong to a catch_all domain.
unknown	antispam_system	These emails have anti-spam systems deployed that are preventing us from validating these emails. You can submit these to us through the contact us screen to look into.
	exception_occurred	These emails caused an exception when validating. If this happens repeatedly, please let us know.
	failed_smtp_connection	These emails belong to a mail server that won't allow an SMTP connection. Most of the time, these emails will end up being invalid.
	forcible_disconnect	These emails belong to a mail server that disconnects immediately upon connecting. Most of the time, these emails will end up being invalid.
	greylisted	Emails where we are temporarily unable to validate them. A lot of times if you resubmit these emails they will validate on a second pass.
	mail_server_did_not_respond	These emails belong to a mail server that is not responding to mail commands. Most of the time, these emails will end up being invalid.
	mail_server_temporary_error	These emails belong to a mail server that is returning a temporary error. Most of the time, these emails will end up being invalid.
	timeout_exceeded	These emails belong to a mail server that is responding extremely slow. Most of the time, these emails will end up being invalid.

Remote Site Entries

If you have a brand-new install and you are configuring for the first time, you do not need to look at this section. If you have an older install, make sure you have these remote site settings set up.

In Salesforce, please navigate to Setup → Security → Remote Site Settings (or search for "Remote Site Settings") and add the following remote sites. You can call them anything you like, but the URLs must be as shown:

- 1) For International Verifications add:
<https://international-street.api.smartystreets.com>
- 2) For Autocomplete add:
<https://us-autocomplete.api.smartystreets.com>
- 3) For Email Address and/or Phone Number Verification add:
<https://api.addressverificationapp.com>
- 4) (Optional) For the Pro Autocomplete add:
<https://us-autocomplete-pro.api.smartystreets.com>
- 5) (Optional) For the International Autocomplete add:
<https://international-autocomplete.api.smartystreets.com>
- 6) (Optional) For the Zip-Code Only Smarty API add:
<https://us-zipcode.api.smartystreets.com>
- 7) For Smarty status in the Setup page add:
<https://status.smarty.com/>
- 8) For (Optional) Google Maps Verifications add:
<https://maps.googleapis.com>

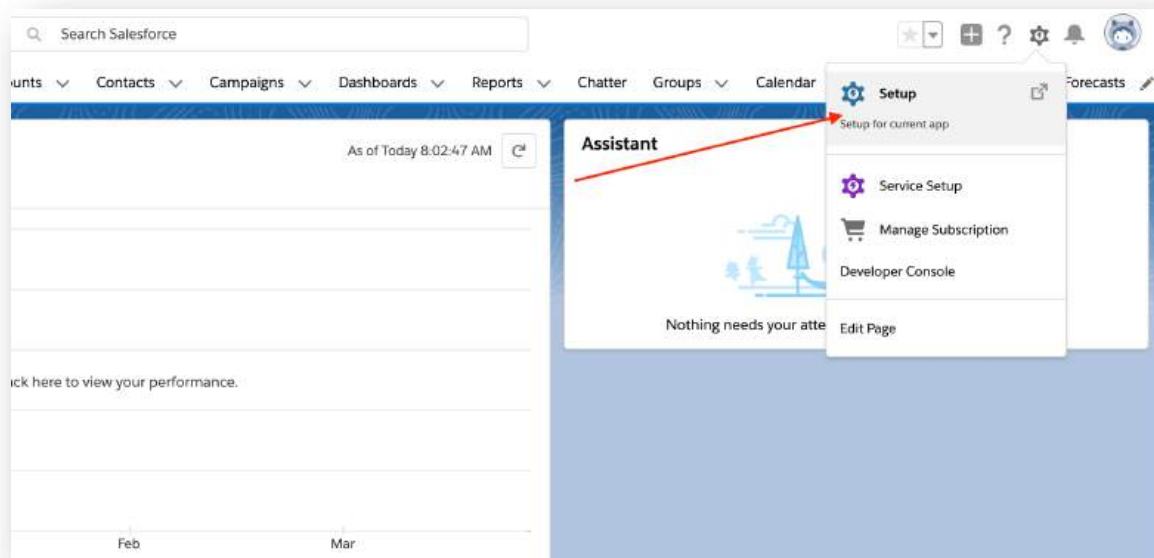
Congratulations! You have now completed all mandatory steps and everything from here on out is optional. In the next section you will set up the “Verify Address” functionality for Accounts, Contacts, and Leads.

Configuring Standard Objects

This chapter will teach you how to configure the app to be used the standard Account, Contact and Lead objects.

In this section you will learn to configure the standard objects included with Salesforce CRM. The Accounts and Contacts come with most editions of Salesforce CRM and the Lead object is included with the Sales Cloud and other editions. For custom objects, please see the next section of this document.

To do all of these steps, we need to switch to the “Setup” area inside Salesforce.

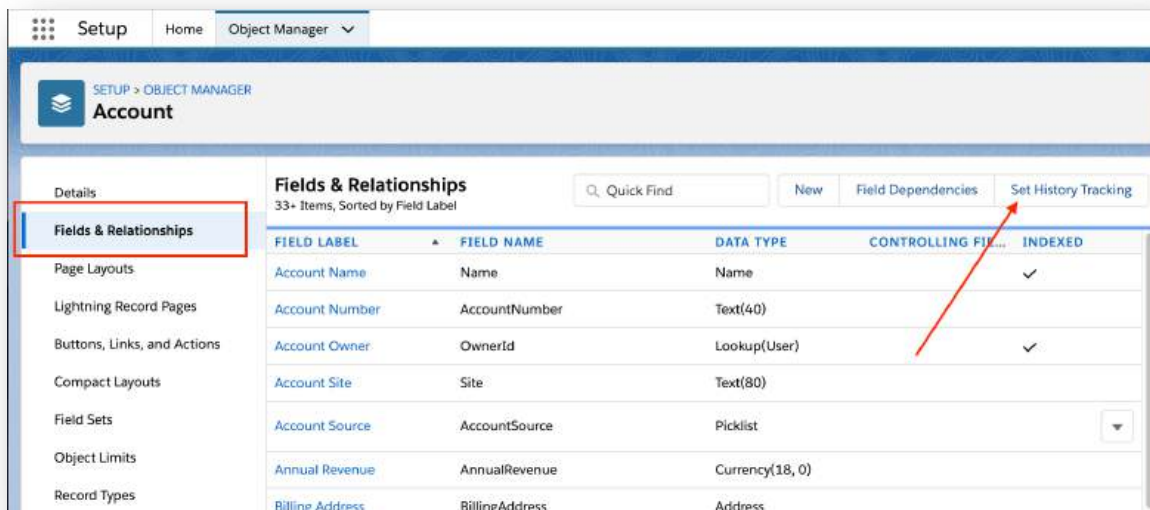
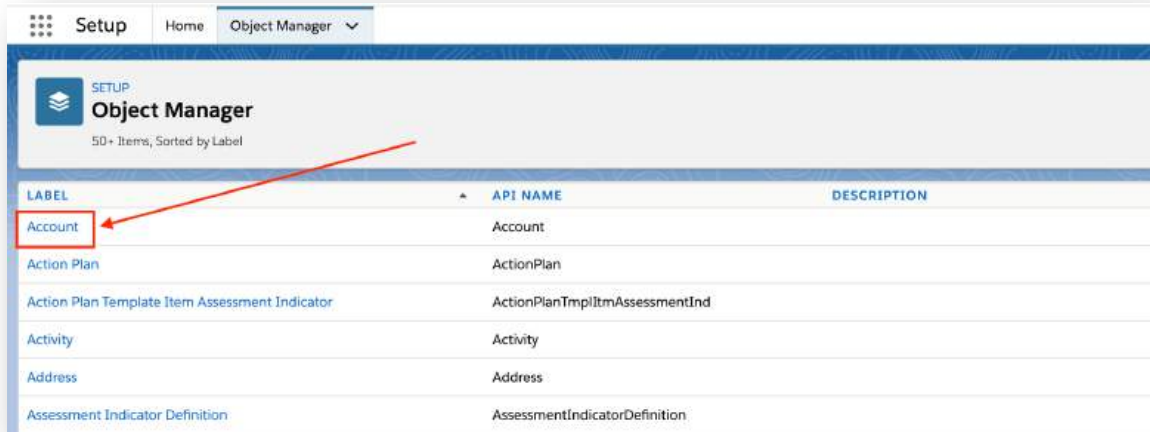


We will display instructions for Physical Address verifications on Accounts below. The Contact and Lead steps will be similar. Email and Phone Number steps would be similar as well.

Accounts, Contacts & Leads

For standard objects we only have three steps: optionally set up field change tracking for the address fields, optionally add address fields to the layout, and add the button and history to the layouts. Note, these steps must be done from the Setup section.

Turning on Field History (Optional)



ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

You might need to Enable Account History before selecting the fields to track.

The screenshot shows the 'Account Field History' configuration page. On the left sidebar, 'Fields & Relationships' is selected. The main content area has a red box around the 'Enable Account History' checkbox, which is checked. Below this, there is a 'Save' button and a 'Cancel' button. A red arrow points to the 'Save' button. Under the 'Track old and new values' section, there are two columns of checkboxes. In the left column, 'Billing Address' and 'Shipping Address' are checked, with red arrows pointing to them. The right column contains various other fields like 'Account Number', 'Account Site', 'Annual Revenue', etc., all of which are unchecked.

Modifying Layouts

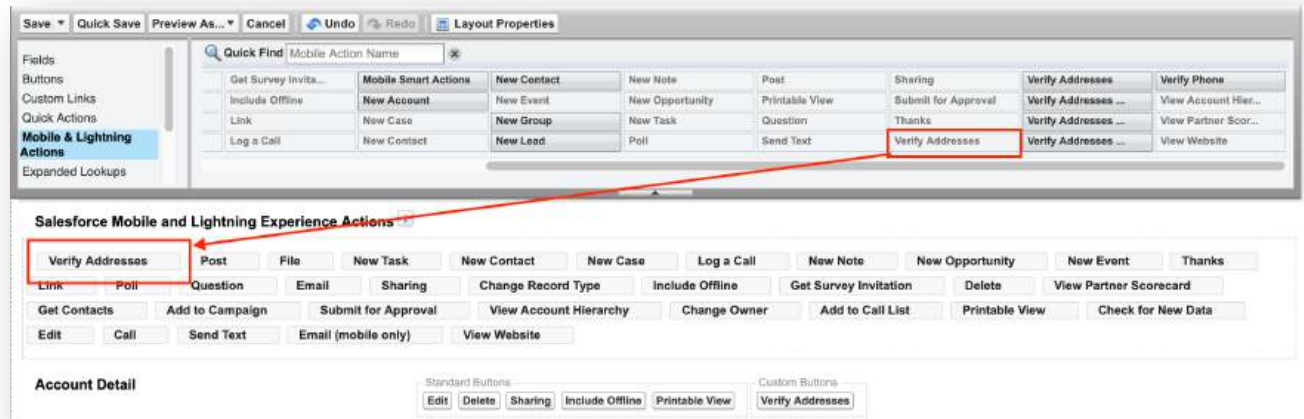
Once again, for Contacts and Leads, the steps will be similar.

The screenshot shows the 'Page Layouts' configuration page for the 'Account' object. On the left sidebar, 'Page Layouts' is selected. The main content area shows a table with one item, 'Account Layout'. The table has columns for 'PAGE LAYOUT NAME', 'CREATED BY', and 'MODIFIED BY'. The 'Account Layout' row shows it was created by 'Joseph Dindinger' on '12/24/2019, 6:59 AM' and modified by the same user at the same time. A red arrow points to the 'Edit' button in the actions column next to the 'Account Layout' row.

* Note that you might have multiple layouts, in this case, you might want to click on the “Page Layout Assignment” button to see which ones are being used, and which ones need to be updated.

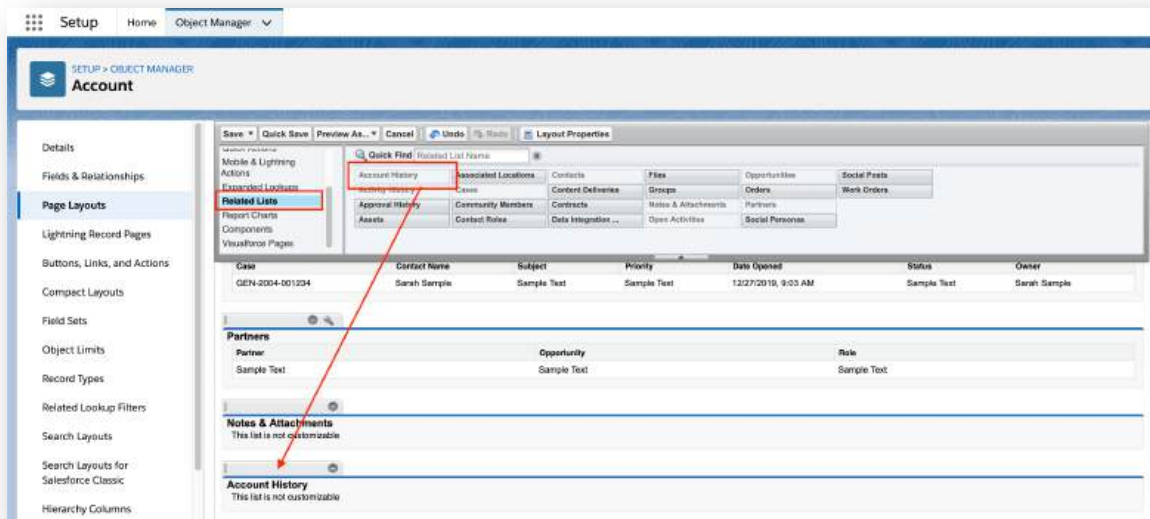
ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

After bringing in as many fields as desired, the most important and only required step is to bring in the **Verify Addresses** action (for Lightning) as shown below.



* For the Lightning component, use the one called **smartystreets__Verify_AddressesComp** – you can see the name if you hover over each action.

You may also want to drag down the **Account History** related list into the layout, so that you can see the field changes on each record.



After making these changes, **Save** the layout and you are finished.

The same basic steps apply to the contact and lead objects.

Adding “Verify” Components to the Layout

We have two Lightning Web Components that can be added directly to the layout of any object’s page, in order to verify Physical Address records (not yet Emails or Phone Numbers):

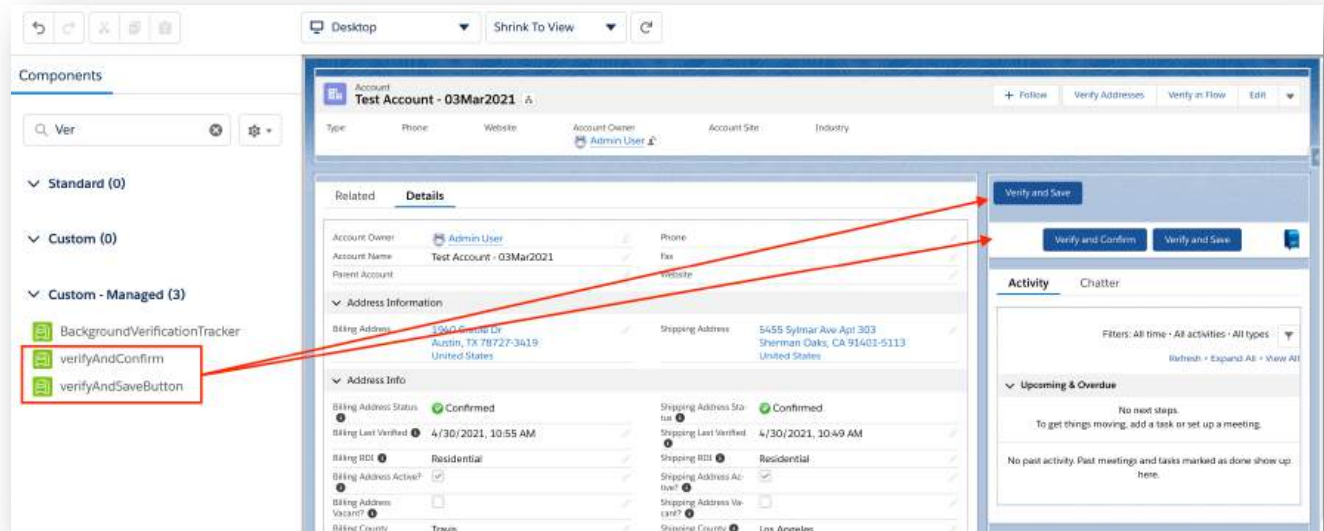
- **Verify and Confirm** button – this button will provide a confirmation of the updates about to happen on the addresses to be verified, showing any changes in red, as shown below. The user is allowed to Confirm or Cancel. Either way, the verification is made, but if the user cancels, the verification is not saved to the record. (This component also has a Verify and Save button.)
- **Verify and Save** button – this button is an alternate to the button in the prior section and does exactly what the other button does, but is a bit faster, so if you want to give up a bit of real estate on the page, it should improve performance.
- **Verify and Confirm Enrichment** – for use only with help from tech support.



The screenshot displays the Salesforce 'Details' page for an Account. The left sidebar shows 'Related' and 'Details' tabs. The main content area is divided into sections: 'Address Information' and 'Address Info'. The 'Address Information' section shows 'Billing Address' (55 Shuman Blvd #1025, Naperville, IL, USA) and 'Shipping Address' (2300 N Field St, Dallas, TX, USA). The 'Address Info' section shows 'Billing Address Status' (Not Yet Verified) and 'Shipping Address Status' (Not Yet Verified). The right sidebar shows 'Account Billing' and 'Account Shipping' sections. The 'Account Billing' section shows 'Street' (55 Shuman Blvd Ste 1025), 'City' (Naperville), 'State' (IL), 'Postal Code' (60563-8455), 'Country' (USA), 'Status' (Confirmed), and 'Footnote' (City/state/zip + Street are all valid ZIP+4 matched; Confirmed entire address; address is valid Verified by SmartyStreets). The 'Account Shipping' section shows 'Street' (2300 N Field St), 'City' (Dallas), 'State' (TX), 'Postal Code' (75201-2476), 'Country' (USA), 'Status' (Confirmed - Missing Secondary Info), and 'Footnote' (City/state/zip + Street are all valid. Confirmed with missing secondary information; address is valid but it also needs a secondary number (apartment, suite, etc.) Verified by SmartyStreets). At the bottom of the right sidebar are 'Confirm' and 'Cancel' buttons.

In order to set up either of these components, follow these steps:

- 1) Go to a record where you want to add the component.
- 2) Click the  icon at the top-right of the page. Select **Edit Page** from the dropdown.
- 3) This will put you in the page editor. From the bottom of the panel on the left, under the “Custom – Managed” header, you will find the **verifyAndSaveButton** and **verifyAndConfirm** components.
- 4) Drag either of these components (you don’t need both since the **verifyAndConfirm** has both included) to the area of your choosing as shown on the screenshot below.



- 5) Click the **Save** button at the top of the page. If this is the first time you have edited this page, a popup will come up asking about activation of the page. If that is the case, we recommend making it the Org Default for all profiles, which will maintain the same layout for everyone.
- 6) Click the **← Back** button at the top, to take you back to the record where you started.
- 7) Now you can test the component you added.

Adding Address Verification for Lists

You can set up a button to verify multiple addresses at one time from a standard Salesforce.com list. Please see the first part of *Chapter 5: Automation and Bulk Verification* for instructions on setting this up.

Lead Conversion Setup

If you are verifying addresses on the Leads object, we recommend updating the Lead to Account/Contact Mapping so that when you convert Leads which have already been verified, you will not need to verify the Accounts and Contacts that are created.

To do this mapping, go to Setup → Object Manager → Lead, and then click on “Fields & Relationships” and press the “Map Lead Fields” button to enter the mapping area. Map all the fields you can, but at least the Address Verified checkbox, the Address Return Code text field, and Address Last Verified date/time field.

Configuring Custom Objects

Expanding Address Verification to Custom Areas

In this section you will learn to configure custom objects built within Salesforce CRM or any edition on the Salesforce platform. As with the steps for Standard Objects, we need to switch to the “Setup” area.

Physical Address Configuration

Creating Mandatory Fields

The following fields must be created for each object where you want to verify an address. If there is more than one address to verify, then one of these fields must be created for each address.

Sample Field Name	Field Type	Field Description
Street	Text(255)	Will typically already be created
City	Text(128)	Will typically already be created
State	Text(20)	Will typically already be created
Zip Code	Text(20)	Will typically already be created
Country	Text(64)	Will typically already be created
Address Verified?	Checkbox	Used to determine if the address needs to be verified
Address Last Verified	Date/Time	Used to determine the last time the address was verified
Address Return Code	Text(40)	Used for a return value from the address verification. Possible return values are listed towards the end of Chapter 2.
Footnotes	Long Text Area	Notes on the details of the verification

Strongly Recommended – But Still Optional – Fields

Sample Field Name	Field Type	Description
Smarty Key	Text (100)	Smarty's unique identifier for an address for the Address Enrichment APIs.
Smarty Key Version	Text (100)	Version for Smarty Key (saves calls)
Address County	Text(64)	Shows the county name returned on verification
Address Latitude	Number (12, 6)	Used for geocoding the Latitude of the address
Address Longitude	Number (12, 6)	Used for geocoding the longitude of the address.
Active?	Checkbox	Shows if the Post Office thinks this is an active address
Do Not Verify	Checkbox	Use this field to indicate that you do not want the address to be verified in the future. Used mostly for automated processes.

Additional Optional Fields

Sample Field Name	Field Type	Description
Street 2	Text(128)	Optionally, the second line of the address. If this is not present, then the second line of the address can be included in the “Street” field.
Street 3	Text(128)	Sometimes used to include the fourth line of the address. This is not used for US addresses, and only rarely for International addresses.
Street 4	Text(128)	Sometimes used to include the fourth line of the address. This is not used for US addresses, and only rarely for International addresses.
Address Barcode	Text(32)	Shows the barcode returned on verification
Address County FIPs Code	Text(32)	Shows the FIPs code for the county returned on verification
Address Unit	Text(64)	Shows the Apartment, Suite or Unit included in the address on verification
Urbanization	Text(64)	Shows the urbanization information (for Puerto Rico) returned on address verification
Carrier Route	Text(4)	Post Office carrier route code
Congressional District	Text(2)	Code for the Congressional District in which the address is located.
Daylight Savings?	Checkbox	Shows if the address is in an area that uses daylight savings time.
RDI	Text(32)	Shows the Residential Delivery Indicator from the Post Office.
Record Type	Text(1)	Shows the returned accuracy of the verification.
Time Zone	Text(48)	Shows the name of the time zone in which the address is located.
Vacant?	Checkbox	Shows if the Post Office believes this address to be vacant.
Clean Street	Text(255)	Used to copy a “clean” version of the street field. This is used in the feature that tries to auto-update previously verified addresses.
Verification Used	Picklist	Choices are: Smarty, Google Maps, UPS
Verified By	Lookup(User)	Lookup to the User object (Can also be set to Text(18) field.)
UTC Offset	Number (16,2)	Used to indicate how many hours different from GMT/UTC time
Zip Plus 4	Text(4)	This fields is to be used optionally in case you want to have the first 5 digits of the zip code separated from the last 4 digits.
Match Details	Text (100)	Populated none, non-postal-match, postal-match, missing-secondary, unknown-secondary, or ignored-input.

Creating Optional Formula Fields

The following fields are formula Text fields that pull from the data in the fields in the list above. Use the formula code below to create your own formulas to display these fields instead of some of the ones above.

Address Status To use this formula, please replace “**RETURNCODE**” with the API name of the field containing return code field returned from the address verification. This formula field will provide the status along with a helpful icon.

```
If (RETURNCODE = "Confirmed", IMAGE("/img/msg_icons/confirm16.png", "Confirmed", 16, 16) & " Confirmed",
If (RETURNCODE = "Partial", IMAGE("/img/msg_icons/warning16.png", "Confirmed - Partially", 16, 16) & " Confirmed - Partially",
If (left (RETURNCODE,9) = "Confirmed", IMAGE("/img/msg_icons/warning16.png", RETURNCODE, 16, 16) & " " & RETURNCODE,
If (RETURNCODE = "Bad or Blank Address", IMAGE("/img/msg_icons/error16.png", "Blank Address", 16, 16) & " Bad or Blank Address",
If (RETURNCODE = "Not Submitted", IMAGE("/img/msg_icons/error16.png", "Not Submitted", 16, 16) & " Not Submitted",
If (RETURNCODE = "Not Confirmed", IMAGE("/img/msg_icons/error16.png", "Not Confirmed", 16, 16) & " Not Confirmed",
IMAGE("/img/msg_icons/infol6.png", "Not Yet Verified", 16, 16) & " Not Yet Verified")))))))
```

Barcode Image To use this formula, please replace “**BARCODE**” with the API name of the field containing barcode field returned from the verification. This formula field will provide a visual representation of that barcode, which can be printed as a barcode.

```
if ( not ( isBlank ( BARCODE ) ), IMAGE("https://www.barcodesinc.com/generator/image.php?code=" + XXX +
"&style=196&type=C128B&width=206&height=50&xres=1&font=3", "Address Barcode" ), "" )
```

Geolocation Link To use this formula, please replace “**LAT**” and “**LONG**” with the API name of the field containing the latitude and longitude field returned from the verification. This formula field will show the latitude and longitude with a link to see the location on a map.

```
if ( not ( isBlank ( LAT ) ), text ( LAT ) + " " + text ( LONG ) + " " +
HYPERLINK ("http://www.google.com/maps/place/" + text ( LAT ) + "," + text ( LONG ), "map", "_blank"), "" )
```

Record Type Description To use this formula, please replace “**RECORDTYPE**” with the API name of the field containing the record type in the list of fields above. This will give you a readable description of what the codes mean.

```
If (RECORDTYPE = "F", "Firm",
If (RECORDTYPE = "G", "General Delivery",
If (RECORDTYPE = "H", "High-Rise",
If (RECORDTYPE = "P", "P.O. Box",
If (RECORDTYPE = "R", "Rural Route",
If (RECORDTYPE = "S", "Street", "" ) ) ) ) ) )
```

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

Current Time To use this formula, please replace “**HOUROFFSET**” with the API name of the field containing the numeric offset from UTC official time, and “**DAYLIGHTSAVINGS**” with the checkbox field containing a TRUE or FALSE value for daylight savings. This formula will only work for time in the US.

```
IF(ISBLANK(HOUROFFSET), "",
  IF (AND( DAYLIGHTSAVINGS,
    TODAY() >=
      CASE(MOD( DATE(YEAR(TODAY()), 3, 8) - DATE(1900, 1, 7), 7),
        0, DATE(YEAR(TODAY()), 3, 8),
        1, DATE(YEAR(TODAY()), 3, 8)+6,
        2, DATE(YEAR(TODAY()), 3, 8)+5,
        3, DATE(YEAR(TODAY()), 3, 8)+4,
        4, DATE(YEAR(TODAY()), 3, 8)+3,
        5, DATE(YEAR(TODAY()), 3, 8)+2,
        DATE(YEAR(TODAY()), 3, 8)+1),
    TODAY() <
      CASE(MOD( DATE(YEAR(TODAY()), 11, 1) - DATE(1900, 1, 7), 7),
        0, DATE(YEAR(TODAY()), 11, 1),
        1, DATE(YEAR(TODAY()), 11, 1)+6,
        2, DATE(YEAR(TODAY()), 11, 1)+5,
        3, DATE(YEAR(TODAY()), 11, 1)+4,
        4, DATE(YEAR(TODAY()), 11, 1)+3,
        5, DATE(YEAR(TODAY()), 11, 1)+2,
        DATE(YEAR(TODAY()), 11, 1)+1)),
    /* CALCULATE FOR IT BEING CURRENTLY DST */
    MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 6, 2 ) & "/" &
    MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 9, 2 ) & "/" &
    LEFT( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 4 ) & " " &
    TEXT( IF( OR(
      VALUE( MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 12, 2 ) ) = 0,
      VALUE( MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 12, 2 ) ) = 12), 12,
      VALUE( MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 12, 2 ) ) -
        IF( VALUE( MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 12, 2 ) ) < 12, 0, 12 ))
    & ":" & MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 15, 2 ) & " " &
    IF( VALUE( MID( TEXT( NOW() + (HOUROFFSET + 1) / 24 ), 12, 2 ) ) < 12, "AM", "PM"
    ),
    /* CALCULATE FOR IT NOT BEING CURRENTLY DST */
    MID( TEXT( NOW() + HOUROFFSET / 24 ), 6, 2 ) & "/" &
    MID( TEXT( NOW() + HOUROFFSET / 24 ), 9, 2 ) & "/" &
    LEFT( TEXT( NOW() + HOUROFFSET / 24 ), 4 ) & " " &
    TEXT( IF( OR(
      VALUE( MID( TEXT( NOW() + HOUROFFSET / 24 ), 12, 2 ) ) = 0,
      VALUE( MID( TEXT( NOW() + HOUROFFSET / 24 ), 12, 2 ) ) = 12), 12,
      VALUE( MID( TEXT( NOW() + HOUROFFSET / 24 ), 12, 2 ) ) -
        IF( VALUE( MID( TEXT( NOW() + HOUROFFSET / 24 ), 12, 2 ) ) < 12, 0, 12 )
    )
    )
    & ":" & MID( TEXT( NOW() + HOUROFFSET / 24 ), 15, 2 ) & " " &
    IF( VALUE( MID( TEXT( NOW() + HOUROFFSET / 24 ), 12, 2 ) ) < 12, "AM", "PM"
    )))
)))
```

Verified By To use this formula, please replace “**VERIFIEDUSERID**” with the ID of the Verified User returned by the verification, “**VERIFIEDUSERNAME**” with the first and last name of the same user, and “**LASTVERIFIED**” with the API name of the field containing the last verification date/time returned from the verification. This formula field will show the user along with a timestamp of the verification date/time.

```
if ( not ( isblank ( VERIFIEDUSERID ) ),
  HYPERLINK("/" & VERIFIEDUSERID, VERIFIEDUSERNAME, " Top") & ", " &
  TEXT(MONTH( DATEVALUE( LASTVERIFIED ) ) ) + "/" + TEXT(DAY( DATEVALUE( LASTVERIFIED ) ) ) + "/" +
  TEXT(YEAR( DATEVALUE( LASTVERIFIED ) ) ) & " " &
  IF(OR(VALUE( MID( TEXT( LASTVERIFIED - (5/24) ), 12, 2 ) ) = 0,
    VALUE( MID( TEXT( LASTVERIFIED - (5/24) ), 12, 2 ) ) = 12), "12",
    TEXT( VALUE( MID( TEXT( LASTVERIFIED - (5/24) ), 12, 2 ) ) -
      IF( VALUE( MID( TEXT( LASTVERIFIED - (5/24) ), 12, 2 ) ) < 12, 0, 12))) & ":" &
    MID( TEXT( LASTVERIFIED - (5/24) ), 15, 2 ) & " " &
    IF( VALUE( MID( TEXT( LASTVERIFIED - (5/24) ), 12, 2 ) ) < 12, "AM", "PM"), "" )
  )
)
```

Creating International Fields

If you are going to be verifying international addresses, there are a few fields you should create if you want to have helpful data.

Sample Field Name	Field Type	Description	
Address Format	Text (255)	Formatting order of address elements – to be used for advanced formatting of addresses	
Address Precision	Text (255)	How precise is the address? From best to worst:	
		Value	Description
		DeliveryPoint	rooftop level
		Premise	property or building level
		Thoroughfare	street level
		Locality	city or town level
		AdministrativeArea	state or province level
		None	unknown address
Administrative Area	Text (255)	The most common administrative division within a country (e.g. Region)	
Complete Address	Text Area	The complete address, formatted for shipping as recommended by Smarty, formatted as described in the Address Format field	
Dependent Locality	Text (64)	If there is additional information about the city, it will be here. (e.g., neighborhood in Turkey)	
Dependent Locality Name	Text (64)	If the dependent locality has a name, you'll find it here. (E.g., the dependent locality "Dong Cheng Qu" is named "Dong Cheng.")	
Geocode Precision	Text (255)	Indicates level of geocoding precision	
Organization	Text (255)	The name of the recipient, firm, or company at this address	
Post Box	Text (64)	Post Office Box Type and Number	
Premise	Text (64)	Alphanumeric code pertaining to an individual location	
Premise Extra	Text (64)	Extra information about the premise that is not necessarily authoritative but might still be useful	
Sub Building Number	Text (64)	The alphanumeric component of the sub building	
Sub Building Type	Text (64)	The leading sub-building type of the sub building	
Super Administrative Area	Text (255)	The largest administrative division within a country	
Thoroughfare Name	Text (255)	The name component of the thoroughfare (E.g., if thoroughfare contains "Main St" thoroughfare name would contain "Main.")	
Thoroughfare (Complete)	Text (255)	Complete Thoroughfare Name, including (E.g "N Main St SW")	

Creating Custom Address Records

Every custom address to be verified must first be set up in the configuration page. Once in the setup page, click the “Data Mapping” tab and then click the “New Address” button.

The screenshot shows the 'Data Mapping' tab in Salesforce. It includes a 'Create Default Records' button and an 'Add Supported Object' button. Below these is an 'Address Configuration List' table with columns for Action, Name, Active?, Object, Street, City, State, and Zip. The table lists several address types: Account Billing, Account Shipping, Address Picklist, Campaign, and Contact Mailing, each with its corresponding field mappings.

Action	Name	Active ?	Object	Street	City	State	Zip
Edit Del	Account Billing	☒	Account	BillingStreet	BillingCity	BillingState	BillingPostalCode
Edit Del	Account Shipping	☒	Account	ShippingStreet	ShippingCity	ShippingState	ShippingPostalCode
Edit Del	Address Picklist	☒	AddressPicklist__c	PStreet__c	City__c	PickState__c	PZip__c
Edit Del	Campaign	☒	Campaign	Street__c	City__c	State__c	Zip_Code__c
Edit Del	Contact Mailing	☒	Contact	MailingStreet	MailingCity	MailingState	MailingPostalCode

Enter a name for the new address, which begins with the object name. Next, select the object, make the address Active, and then fill in at least the Required Fields and press the “Save” button. This will add the current address as a valid address to verify in your org.

The screenshot shows the 'Edit Account Billing' configuration page. It includes a 'System Addresses Information' section with fields for Name (Account Billing), Select an Object (Account), and checkboxes for Active and Is Person Account. Below this is a 'Required Fields' section with fields for Street (Billing Street), City (Billing City), Verified (Billing Verified), and Last Verified (Billing Last Verified).

*Note: If you want to add an address from an object that does not show up in the Object dropdown, you will need to first add this object using the **Add Supported Object** button before you click to create a New Address.*

Phone Number Configuration

Creating Mandatory Fields

The following fields must be created for each object where you want to verify a phone number. If there is more than one number to verify, then one of these fields must be created for each one.

Sample Field Name	Field Type	Field Description
Mobile	Phone	Will typically already be created
Mobile Verified?	Checkbox	Used to determine if the phone number needs to be verified
Mobile Country	Text(40)	Used to select the country (ISO-2) where the phone number needs to be verified. If blank it will default to "US"
Mobile Last Verified	Date/Time	Used to determine the last time the phone number was verified
Mobile Return Code	Text(40)	Used for a return value from the phone number verification. Possible return values are listed towards the end of Chapter 2.
Mobile API Response	Long Text Area	Notes on the details of the phone number verification

Creating Optional Fields

Several other fields are not required, but available to be created for each phone number to be verified. Please read the description of the field and create if the data would be helpful.

Sample Field Name	Field Type	Description
Mobile Carrier	Text(255)	Phone provider company
Mobile Line Type	Text(40)	Type of phone line (mobile, landline)
Mobile Location	Text(100)	City or other location of registered owner. (If known.)
Mobile Full Country Name	Text(100)	Full Country Name of the 2-character code provided.

Creating Custom Phone Number Records

Every custom phone number to be verified must first be set up in the configuration page. Once in the setup page, click the "Manage Addresses" tab and then click the **New Phone** button.

Creating Optional Formula Fields

The following fields are formula Text fields that pull from the data in the fields in the list above. Use the formula code below to create your own formulas to display these fields.

Phone Status To use this formula, please replace "**RETURNCODE**" with the API name of the field containing the return code from the phone verification.

```
If (RETURNCODE = "Valid", IMAGE("/img/msg_icons/confirm16.png", "Valid", 16, 16) & " Valid",
If (RETURNCODE = "Error", IMAGE("/img/msg_icons/error16.png", "Error", 16, 16) & " Error",
IMAGE("/img/msg_icons/infol16.png", "Not Verified", 16, 16) & " Not Verified"))
```

Email Address Configuration

Creating Mandatory Fields

The following fields must be created for each object where you want to verify an email address. If there is more than one email address to verify, then one of these fields must be created for each one.

Sample Field Name	Field Type	Field Description
Email	Email	Will typically already be created
Email Verified?	Checkbox	Used to determine if the email needs to be verified
Email Last Verified	Date/Time	Used to determine the last time the email was verified
Email Return Code	Text(40)	Used for a return value from the email address verification. Possible return values are listed towards the end of Chapter 2.

Creating Optional Fields

Several other fields are not required, but available to be created for each email address to be verified. Please read the description of the field and create if the data would be helpful.

Sample Field Name	Field Type	Description
Email Domain Age Days	Number	Number of days since the email domain was registered on the internet. (Divide by 365 for the number of years)
Email Free Account?	Checkbox	True/False checkbox field indicating a “throwaway” email address, typically set up only for the purpose of providing a temporary email address, but not a real address to be used going forward. We recommend never emailing these addresses.
Email Owner First Name	Text(100)	First Name of registered owner. (If known.)
Email Owner Last Name	Text(100)	First Last of registered owner. (If known.)
Email Verification Response	Long Text Area	Actual JSON returned by the verification request.
Email Owner Gender	Text(40)	Sex/Gender of owner, if known. (male, female)
Email SMTP Provider	Text(100)	Email host
Email Sub Status Code	Text(100)	Additional information about the email status, if available

Creating Custom Email Address Records

Every custom email address to be verified must first be set up in the configuration page. Once in the setup page, click the “Manage Addresses” tab and then click the **New Email** button.

Creating Optional Formula Fields

The following fields are formula Text fields that pull from the data in the fields in the list above. Use the formula code below to create your own formulas to display these fields.

Email Status To use this formula, please replace “**RETURNCODE**” with the API name of the field containing barcode field returned from the email verification. This formula field will provide the status along with a helpful icon.

```
If(RETURNCODE = "valid", IMAGE("/img/msg_icons/confirm16.png", "Valid", 16, 16) & " Valid",  
If(RETURNCODE = "catch-all", IMAGE("/img/msg_icons/warning16.png", "Accept all (Unverifiable)", 16, 16) & " Accept  
all (Unverifiable)",  
If(RETURNCODE = "do_not_mail", IMAGE("/img/msg_icons/warning16.png", "Do Not Mail", 16, 16) & " Do Not Mail",  
If(RETURNCODE = "invalid", IMAGE("/img/msg_icons/error16.png", "Invalid", 16, 16) & " Invalid",  
If(Or(RETURNCODE = "disposable", RETURNCODE = "spamtrap"), IMAGE("/img/msg_icons/error16.png", "Disposable", 16,  
16) & " Disposable",  
If(RETURNCODE = "unknown", IMAGE("/img/msg_icons/error16.png", "Unknown", 16, 16) & " Unknown",  
IMAGE("/img/msg_icons/infol6.png", "Not Verified", 16, 16) & " Not Verified")))))
```


Creating a Custom “Verify” Action

1) Go back to Setup → Object Manager and then find and click on the custom object you want to add the Verify button to.

2) Click on "Buttons, Links, and Actions" on the left and click on the "New Action" button.

3) Select "Lightning Component" as the Action Type, and then select the **smartystreets:VerifyAddressesLWC** component to verify Physical Addresses.

Set it to 0px height and give it a name like "Verify Address" and Save.

- To verify Emails, use the **VerifyEmail** Lightning component instead
- To verify Phone Numbers, use the **VerifyPhone** Lightning component instead

4) Now edit the Page Layout for your custom object. If you have multiple layouts, you will need to do this several times. Go to the "Mobile and Lightning Actions" menu at the top and find the action you just created. Drag it down into the "Salesforce Mobile and Lightning Experience Actions" section.

** Note: this section of the layout might need to be opened in order to add the Action.*

5) Now test the button in the Lightning experience. You can also test it on Salesforce Mobile.

Adding “Verify” Components to the Layout

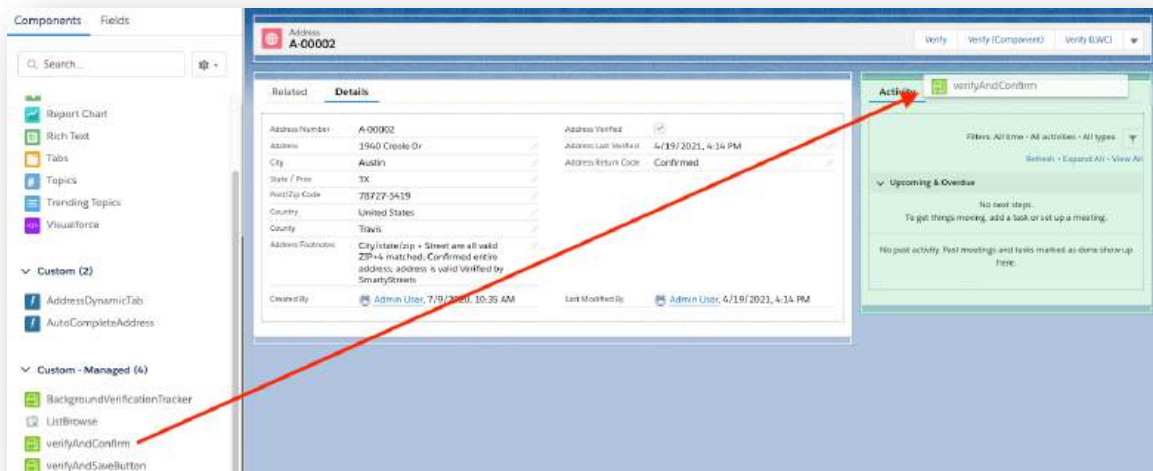
You can use the instructions at the beginning of Chapter 3 to add verification components to custom objects as well.

Adding the “Verify and Confirm” Component

If you want to have the user verify changes to the addresses prior to varication, you can add a component that sits in the page and allows you to do just that.

Note: This component is available for Physical Addresses only.

- 1) Go to a record of the object you want to set up the Verify and Confirm function.
- 2) Press the  icon at the top of the page and choose **Edit Page**.
- 3) This will drop you in the page editor. On the left, scroll down to the “Custom – Managed” section and find the **VerifyAndConfirm** component.
- 4) Drag it into the middle panel, usually at the top of one of the columns on the page.



- 5) If you click on the component you just dragged in, you will be able to change the “Hide Verify And Save” option to:
 - a. NO – the “Verify and Save” button will be shown, allowing user to verify the address without previewing the result, or
 - b. YES – this button will be hidden, forcing users to first preview the changes.
- 6) You can also change the “Always Display Addresses” option to “No” or “YES”. If this is turned on, the address will always be displayed on the screen, even after verification.
- 7) Now click the **Save** button, and if this is the first time creating a page for this object:

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- a. Click the **Activate** button on the “Page Saved” dialog.
 - b. Click the **Assign as Org Default** button, and then select “Desktop” and press the Next, and then Save buttons.
- 8) Click the ← arrow at the top, left of the page to go back to your record.
- 9) You can now press the **Verify and Confirm** button to open the dialog with verified results for any addresses currently configured for the current record. You should see a dialog similar to this one, on the Account detail page:

Account
Test Account V&C

+ Follow Verify Addresses Verify in Flow Edit

Related Details

Account Owner Admin User Phone

Account Name Test Account V&C Fax

Parent Account Website

Address Information

Billing Address 1940 Creole Drive Austin, TX Shipping Address 5455 Sylmar Avenue 301 Van Nuys, CA 91401

Address Info

Billing Address Status Not Yet Verified Shipping Address Status Not Yet Verified

Billing Last Verified 3/30/2021, 4:05 PM Shipping Last Verified 3/30/2021, 4:07 PM

Billing RDI Residential Shipping RDI Residential

Billing Address Active? Shipping Address Active?

Billing Address Vacant? Shipping Address Vacant?

Billing County Travis Shipping County Los Angeles

Billing Address Latitude 30.426998 Shipping Address Latitude 34.169727

Billing Address Longitude -97.686371 Shipping Address Longitude -118.445846

Account Billing

Street 1940 Creole Dr

City Austin

State TX

Postal Code 78727-3419

Country United States

Status Confirmed

Footnote City/state/zip + Street are all valid ZIP+4 matched; Confirmed entire address; address is valid Verified by SmartyStreets

Account Shipping

Street 5455 Sylmar Ave Apt 301

City Sherman Oaks

State CA

Postal Code 91401-5113

Country United States

Status Confirmed

Footnote City/state/zip + Street are all valid ZIP+4 matched; Confirmed entire address; address is valid Verified by SmartyStreets

Confirm Cancel

* Note that changes to the current address will be highlighted in red.

- 10) If you press the **Confirm** button, the displayed results, along with any other data elements configured will be populated on the current record.

Adding the “Verify and Save” Button Component

Three pages back, a section called *Creating the Custom “Verify” Button* can be found, which explains how to add a “Verify” button on the layout. That button will be in the header, at the top of the page.

Note: This component is available for Physical Addresses only.

One other option is to add a “Verify and Save” button somewhere below the header, allowing the button to be displayed in other places, or if there are already too many buttons at the top.

To add this button, simply follow the instructions for the “Verify and Confirm” component on the prior page, but use the **verifyAndSaveButton** component instead of the one listed above.

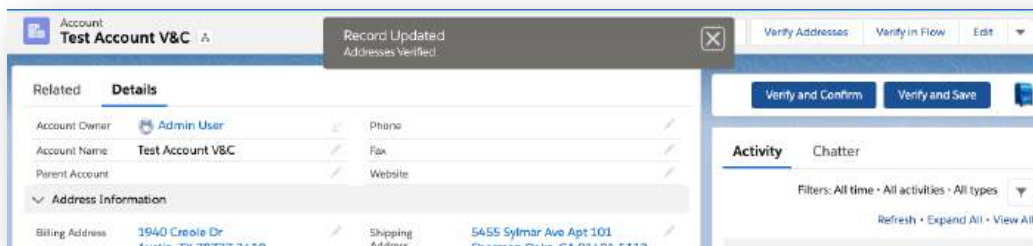
Adding the “Background Verification Tracker” Component

Much of the Address Verification application runs in real time, when a button is pressed, or a screen is used. However, much of the power of Address Verification, is that it can be set to verify records in the background – upon creation, or when the address fields are updated.

Note: This component is available for Physical Addresses only.

The downside to background verification is that it happens in the background, with no way of notifying the user that it was done. If a user is currently seeing a record, they will not know that the address is verified until they refresh the page. For example, if they create a record through the standard screens (not our Autocomplete component), they would see the address they entered when they view the record. A few seconds later, that address would be verified, but unless they refresh the page, a user would never know.

In order to get around this limitation, we now provide a component that sits on your record layout and tries to determine if the record is being verified. It doesn’t *always* succeed as it runs for only 5 or so seconds after an update or insert, but in many cases, it will determine that the address was updated, and will refresh the fields and alert the user of the update.



- 1) Go to a record of the object you want to set up the Background Verification Tracker.
- 2) Press the  icon at the top of the page and choose **Edit Page**.
- 3) This will drop you in the page editor. On the left, scroll down to the “Custom – Managed” section and find the **BackgroundVerificationTracker** component.
- 4) Drag it somewhere on the page layout in the middle panel. This component will be largely invisible, so we recommend putting it at the bottom of a column.
- 5) Now click the **Save** button, and if this is the first time creating a page for this object:
 - a. Click the **Activate** button on the “Page Saved” dialog.

- b. Click the **Assign as Org Default** button, and then select “Desktop” and press the Next, and then Save buttons.
- 6) Click the ← arrow at the top, left of the page to go back to your record.

Un-Verifying Records on Change

Now that we have a button to verify the record, we need to automate one very important feature: we need to make sure that addresses are only verified when we need them to.

Note: This Flow must be separate from a Flow that verifies a record, because it must run in “Fast Field Updates” mode.

Basically, the Verify buttons will only work if the “Verified” field is unchecked. That is done in order to minimize the number of API calls you make. Every time a Verify button is pressed and there is successful verification, we check the box, stating that the address or email or phone number has been verified and does not need to be verified again.

We have two supported options: Record-triggered Flow and Trigger. Flows can be done directly in Production, while Triggers need to be created in a sandbox and deployed, along with test code, into production.

- ❖ *In the examples below, we will assume you are setting up Physical Addresses, which are the hardest to do because you have to track if a change was made to any of the address fields (Street, City, State, Postal Code, Country), where with Email Addresses or Phone Numbers, you only have to check for a change on the one field.*

Option 1: Making Records Un-Verified via Flow

An easy way to mark a record as needing verification is via a process. This option might be the preferable way of handling this task, because it is easy to set up, easy to turn off if necessary, and is quickly becoming the recommended way of making updates to Salesforce records going forward.

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- 1) To build it with a Flow, go to Setup → Process Automation → Flows and click on the “New Flow” button to get started.
- 2) Use the “Start From Scratch” Option.
- 3) Select **Record-Triggered Flow** and press the “Create” button.
- 4) Select your object, choose “A record is created or updated” in the **Trigger the Flow When** question, and “Custom Condition Logic Is Met” for the **Set Entry Conditions** section.
- 5) Leave the **Condition Logic**, for now, and create entry conditions as shown below. In this example, we will assume you are on the Lead object:

The screenshot shows the 'Configure Start' section of a Salesforce Flow. It has three main parts: 'Select Object', 'Configure Trigger', and 'Set Entry Conditions'. In 'Select Object', 'Lead' is chosen. In 'Configure Trigger', 'A record is created or updated' is selected. In 'Set Entry Conditions', 'Custom Condition Logic Is Met' is selected from a dropdown menu.

- a. Select “Street” (assuming Leads) as the **Field**, “Is Changed” as the **Operator**, and “True” as the **Value**.
 - b. Press the + Add Condition button, and select “City” as the **Field**, “Is Changed” as the **Operator**, and “True” as the **Value**.
 - c. Do the same for “State”, “Postal Code”, and “Country”.
 - d. Now add another condition and this time select the checkbox field that indicates if the address was already verified, for example, on Leads it would be “smartystreets__Address_Verified__c” for the **Field**, “Equals” as the Operator, and “True” as the **Value**.
- 6) Now, once the conditions are created, go back to the Condition Logic and enter something like this:

(1 OR 2 OR 3 OR 4 OR 5) AND 6

– What this statement means is that if ANY of the first 5 conditions are met, meaning if any of the 5 address fields are changed AND if the sixth condition is also met (the record is already verified), then this flow will proceed.

- 7) Leave the “Every time a record is updated and meets the condition requirements” selected for When to Run the Flow for Updated Records.
- 8) Change the Optimize the Flow for to use the “Fast Field Updates” option. This is very important.

9) Now click the + action under the first action in the flow:

Record-Triggered Flow Start

Object: Location [Edit](#)

Trigger: A record is created or updated

Conditions: 6

Optimize for: Fast Field Updates

[Open Flow Trigger Explorer for Location](#)

Configure Start

Condition Requirements

Custom Condition Logic Is Met

* Condition Logic

{1 OR 2 OR 3 OR 4 OR 5} AND 6

Field	Operator	Value
1. Street Address	Is Changed	True
2. City	Is Changed	True
3. State	Is Changed	True
4. Zip	Is Changed	True
5. Country	Is Changed	True
6. Address Verified	Equals	True

[+ Add Condition](#)

When to Run the Flow for Updated Records

☒ Every time a record is updated and meets the condition requirements

☐ Only when a record is updated to meet the condition requirements

10) Select “Update Triggering Record”

11) Now, click on and edit the “Update Records” action. Give it whatever Label and API Name you want. Leave “Use the X record that triggered the flow” and do not set any Filter Conditions. Add the following fields to update the following values:

- Verified – set to False
- Last Verified – set to null (this is optional, some users prefer to leave this populated in order to track the last time it was updated)
- Return Code – set to “Changed”

12) Click the **Save** button, give it a Flow Label and Flow API Name, and Save. Then click the **Activate** button so that you can test.

Option 2: Making Records Un-Verified via Trigger

Another way to un-verify and verify records is by creating an Apex trigger in code. This is the most difficult to do from a technical perspective, but perhaps the cleanest and most efficient from a performance standpoint. To set this up, see the *Turning On Trigger Verifications* section in the next chapter.

Finalizing Custom Object Setup

Modifying the layout, and turning on the field history is very similar to what is done for standard objects. The exception being for Physical Addresses, because custom addresses usually have individual fields for Street, City, State, Zip and Country. For effective tracking, all of these should be turned on.

Automation and Bulk Verification

Cleaning existing records, or records imported in bulk

In this section you will learn what you can do in order to clean up existing data in Salesforce, and how you can make sure that your data stays clean. In the previous section we explain how to let users verify individual records and in the next section, we get into the new Autocomplete feature for Physical Addresses, but here we discuss the back-up plan for all of this – what to do when a record is already saved in Salesforce and is not perfectly clean.

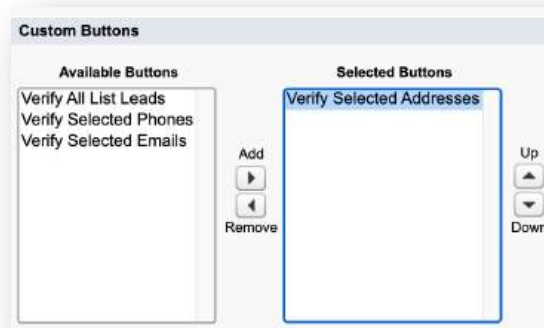
Manual Bulk Verification

If the number of records you need to clean up is not huge, you might be able to clean them 200 at a time using our list view button. Even if the number of records is too large to do cleanup manually, it is always a good idea to configure the option in case you want to use it in the future.

List Verification for Standard Objects

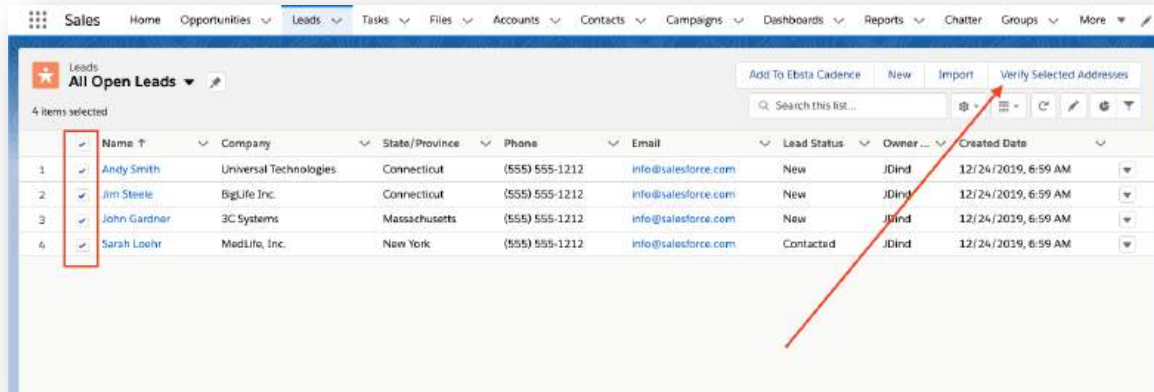
You can set up a button to verify multiple addresses at one time from a standard Salesforce list. We will step through how to do it for Physical Addresses on Leads, but the process is very similar for Accounts and Contacts, and even custom objects, whether you are doing physical addresses or emails or phone numbers.

- 1) Go to **Setup → Object Manager → Lead** and click on List view button layout to Edit.
- 2) From the list of available buttons, select the one called “Verify Selected Addresses” (for Physical Addresses) and press the → arrow button. Next, press the **Save** button.



ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

The button will now be available from any Lead list view. You can select as many records as you want, up to 200 and then press the Verify Selected Addresses.



Please note several things when verifying multiple records at once:

- 1) You will use up a Lookup for every individual physical address, email address, or phone number on each record. So, even though you verify 100 at a time, you are still charged for 100 lookups.
 - a. The exception is if an address is the same in multiple fields on the same record (e.g. Matching Billing and Shipping addresses on an Account). In that case, only one lookup would be done.
- 2) If addresses are very long, sometimes the Salesforce request (not the Smarty API) will choke on multiple verifications at once, and will show you an error. If that happens, please try selecting fewer records and trying again.

List Verification for Custom Objects

A button can also be added for multiple-record validation on custom objects. Please follow these steps and you should be able to add the button fairly easily. This page uses Physical Address verification as an example, but matching code can be found in our GitHub repository, in the “Pages” folder for Email Address or Phone Number verification in a list.



GitHub.com Address Verification Sample Code Repository

<https://github.com/ToAFinish/addressverification>

First, create a simple Visualforce page by going to Setup → Custom Code → Visualforce Pages, and press the “New” button. Then, paste the code below into the page area:

```
<apex:page
    standardController="Object_Name__c"
    extensions="smartystreets.VerifySelectedAddressesExt"
    recordSetVar="leads"
    sidebar="false"
    showHeader="false">
    <smartystreets:VerifyAddressesList
        Controller="{!setCon}" />
</apex:page>
```

Replace the highlighted “Object_Name__c” with the API name of the custom or standard object.

- *Note: Make sure to select the "Available for Lightning Experience, Lightning Communities, and the mobile app" option when creating the page, as this will allow it to appear in Lightning as well as Classic.*

Next, go to the object you want to verify, and find the “Buttons, Links and Actions” section and then create a new button. Set it as a “List Button” with the “Display Checkboxes (for Multi-Record Selection)” option ON, and have the behavior “Display in Existing Window without sidebar or header.” Under the Content Source, select “Visualforce Page” and then select the page you created in the steps above.

Now, you will need to add this button to the search layout as shown at the beginning of this chapter, and you are good to go!

Automated Verification Options

Depending on the number of records in Salesforce, and the ongoing process of ingesting new records into the org, many companies need an automated solution that will do the work for them in the background. Address Verification provides several options.

- ❖ Note: Please keep in mind that Salesforce does not allow verifications to happen in real time when a record is being updated or created. The reason is that in order to verify the address, we are making a call out to an external API, and we are forced to make it in the background (asynchronous) instead of in real-time (synchronous). What this means is that if you create a record with the automated trigger turned on, you will first see the record is not verified, but if you refresh it after a couple seconds, you then see the verification completed.

Turning on Trigger Verifications

You can turn on automatic verifications via Trigger for the Lead, Account and Contact objects in one step, and for other objects by doing this and also manually creating a trigger. After going through this process, records will automatically verify on these objects whenever they are imported or the mapped (required) fields are updated.

This feature is easy to turn on:

- 1) Go to Setup → Apps → Installed Packages → Address Verification → Configure.
- 2) In the Configuration page, click the Automation tab, click the “Edit” link next to the object where you want to configure the trigger.
- 3) Once in the page, configure the options as desired:

a. The “Label” should indicate the object name or nickname.

b. The “API Name” should be the actual API name of the object.

c. The Active checkbox will turn on trigger functionality (but NOT yet verify the record). With this turned on, the trigger on the current object will run (assuming the trigger exists) and will **un-verify** any record which has address or email or phone fields modified, when it was previously

marked as verified. *This Active checkbox by itself will not verify records.*

- d. The “Layout Section Name” is to indicate what the Section on the record layout is called where the address fields are included. If this is empty, then “Address Information” (the Salesforce default) is assumed.

- e. The “Active Job ID Field” is the API name of a Text field (at least 18 characters long) which currently exists in the object. This field should not be on the layout on the object, but the current user should have read/write access to it. *If left blank, no batch jobs can be run on the current object.*
- f. To run verifications automatically turn on the “Verify Address” option for physical addresses, the “Verify Emails” option for email addresses and the “Verify Phones” option for phone numbers.
- g. The “Skip Completion Check” option is for the Autocomplete components on this object, and with this option on, it will allow blank or incomplete addresses without challenging the user.
- h. The “Large Database” option should be turned on for any object with over 200,000 records. It performs a few functions that are overkill, and potentially slowing for smaller object, but critical in order to avoid governor limits on larger objects.
- i. (Physical Address Verifications Only) The “Check for Existing Addresses” option will attempt to save API calls (counting toward your verification package) by checking for a previously verified address in the same object before trying to verify each record. Obviously, this will slow down the process somewhat, so it is only recommended if you have a small verification package and cannot purchase any more. See the “Check for Existing Records” section below for more information.

4) Press the **Save** button when done.

This will now turn on the automated verification for these objects. For objects other than the Lead, Account and Contact, see the next section for the step to create the trigger.

Creating Triggers for Custom Objects

The custom trigger is very simple in what it does, though a bit harder to set up because it requires creating the trigger in a sandbox and then deploying to production. We will not be providing instructions on creating a trigger, as this must be done by an administrator with sufficient rights and understanding to create one in the first place.

As the administrator, you can find Apex code to use in the “trigger” folder on our GitHub.com repository, linked to below. The code from a file called **verifyTrigger.java** should be pasted entirely inside the available text area and then modified to match the custom object that contains the address you want to verify.

In the example code you will pull from GitHub.com, you will want to change all instances of **Office__c** with the API Name of the custom object.



GitHub.com Address Verification Sample Code Repository

<https://github.com/ToAFinish/addressverification>

When finished, press the **Save** button, and the trigger is now ready to run every time a record is updated. Remember, however, that you need to turn the trigger ON, as described in the previous section in this chapter.

Before you can use this trigger in production, you will need to deploy it via a change set or some other deployment method. This might involve creating test code, if triggers for this object are not already covered by a test Apex class.

Check for Existing Records

This feature allows you to immediately verify addresses entered if they already exist in the same object without having to make an API call and re-verify the same address again. This feature is helpful if duplicate values are often entered into your org. For example, if multiple Leads with the same address are entered, you can avoid re-verifying the same address multiple times.

Note: This function is available only for Physical Addresses.

❖ Note that currently we have a couple limitations for this feature:

- *Only addresses in the same object will be checked for existing verifications. So if you are creating a Lead and an Account with a verified address already exists, it will not see it.*
 - *For organizations with a large number of records, usually above 200,000 in an object, this may fail to work depending on several factors. If this is your case, please contact support@toafinish.com so that we can work with you to see if you can use this feature.*
- 1) First, you will need to configure a “Clean Street” field for each address where you want to start using this feature. This is a field where a stripped down, “clean” version of your street address will be hosted. This is the field, along with City and State, which is used to find a match. On the **Data Mapping** tab in the Setup page, edit the appropriate addresses and set this field. Note that you might need to create these fields first. If so, make it Text (255).

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

The screenshot shows the Salesforce Setup page for Address, Email, and Phone Verification. The page is divided into several sections, each with a dropdown menu for configuration. The 'Clean Street' field is highlighted with a red box. The fields are as follows:

UTC Offset Billing UTC Offset	Clean Street Billing Clean Street
Is Active Billing Address Active?	Do Not Verify Billing Do Not Verify
Is Vacant Billing Address Vacant?	Verification Used Billing Verification Used
Island Billing Island	Verified By User Billing Verified By User
Match Details Billing Match Details	Smarty Key Billing Smarty Key

- 2) Next, switch to the **AUTOMATION** tab in the Setup page follow the instructions in the *Turning on Trigger Verifications* section earlier in the chapter.
- 3) Finally, you might want to update your existing data to use this functionality. You don't have to do this, but if you don't, then it will only work with data moving forward. In order to clean all data, you can either re-validate all records, which might be costly, or you can run a script in the Developer Console to "clean" all of your data. Experienced administrators or developers should only attempt this last option. In the future we may have a button that will clean up your data for you, but for now, it is a manual process through code. Here is a sample that would clean 100 Lead records:

```
List<Lead> leadlist = [select Id, Street, CleanStreet__c
                        from Lead
                        where Street != null
                        and CleanStreet__c = null
                        and isConverted = false
                        limit 100];
for(Lead l:leadlist) {
    l.CleanStreet__c = SmartyStreets.Clean.AddressCleanse(l.Street);
}
update leadlist;
```

The key function is **SmartyStreets.Clean.AddressCleanse()**. This function will clean any given street address and standardize it as needed by our function.

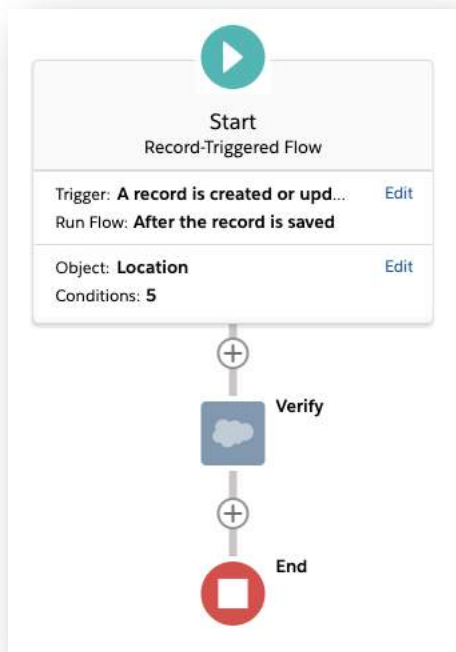
Please do not hesitate to contact us using the support@toafinish.com email address. We will be happy to talk this through with you and help you to get your data cleaned. It is an administrative function that isn't easy, so we can help.

Verifications Using Flows

Another easy way to verify a record when it is changed, is via a Record-Triggered Flow. This option might be preferable for the learned point-and-click administrator, and Salesforce has begun to recommend Flows over Processes.

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- 1) To build it with a Flow, go to Setup → Process Automation → Flows and click on the “New Flow” button to get started.
- 2) Select the “Start from Scratch” option and press Next.
- 3) Select **Record-Triggered Flow** and press the “Create” button.
- 4) Currently you are now presented with a “How do you want to start building?” option. If you see this screen, select “Auto-Layout” and you will be taken to the flow.
- 5) At the top of the flow, click the “Edit” link in the Start component.
- 6) In the Configure Trigger, make a few changes. First, set the Trigger the Flow When to “A record is created or updated”



- 7) Next, set the Run the Flow to “After the record is saved”
- 8) Now click the “Choose Option” link in the Start component and then search for the object you are building the flow for. When found, click on it.
- 9) After selecting the object, you will want to enter conditions to determine when to run. The recommended conditions are:
 - a. The Verified flag (Checkbox field) should be set to = False
 - b. All the address fields: Street, City, State and Zip/Postal Code fields should be set to “Is Null” = False (we will change the logic below so that not ALL of them have to be populated).
 - c. The Address Return Code should be set to “Does Not Equal” = Bad or Blank Address.
- 10) Change the “Condition Requirements” to “Custom Condition Logic is Met” and change it so that it will force the Verified flag to be False, and the Street to be populated, but it should force only either the Zip Code to be populated OR the City and State to be populated. Something like what is shown on the next page:

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

* Condition Logic ⓘ

1 AND 2 AND ((3 AND 4) OR 5) AND 6

	Field	Operator	Value	
1	Address_Verified_c	Equals	False	✕
2	Street_c	Is Null	False	✕
3	City_c	Is Null	False	✕
4	State_c	Is Null	False	✕
5	Zip_Code_c	Is Null	False	✕
6	Address_Return_Code_c	Does Not Equal	Bad or Blank Address	✕

+ Add Condition

- 11) When you have set up the conditions, make sure that **When to Run the Flow for Update Records** is set to “Every time a record is updated and meets the condition requirements”.
- 12) Make sure that **Optimize the Flow for:** is set to “Actions and Related Records” and you are done with this screen.
- 13) Press the “+” sign and select Action.
- 14) In the Action search, type in “Verify” and select either **Verify Address** to verify physical addresses, **Verify Email** to verify email addresses, or **Verify Phone** to verify phone numbers.
- 15) Give the action a label such as “Verify”
- 16) On the “recordIds” Input value, click the On switch, and then search for the “Id” field in the record you started with.
- 17) Press the **Done** button, and then **Save** and finally **Activate** to turn on the automatic verification.

Set Input Values

A recordIds

{!\$Record.Id}

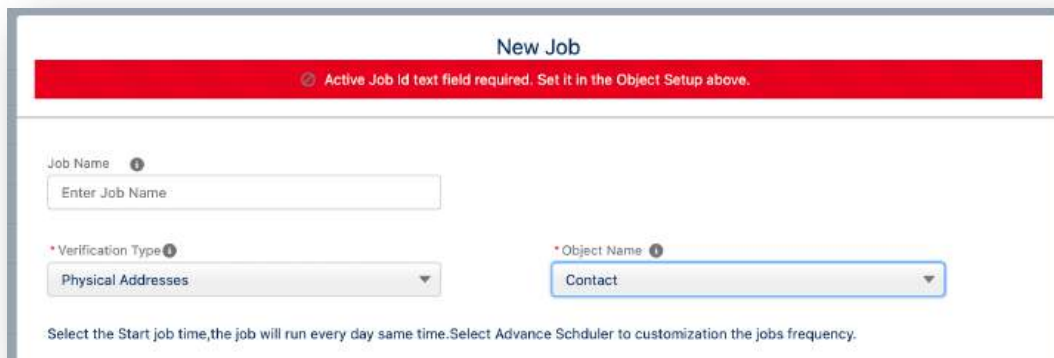
Include

Cancel Done

Bulk Verifications With Batch Jobs

If you do not want to have addresses verified via triggers or flows as shown in the sections above, you can also schedule Apex jobs that will verify addresses as often as needed.

- ❖ Please note that these jobs can take a long time to run, depending on how many records you are trying to verify. Each batch will contain up to 50 records, so with hundreds of thousands or millions of records, the batch jobs will take a while to run. Apex jobs are not particularly fast in Salesforce, due to the need to maintain a fast experience for all customers on each Salesforce instance.
 - ❖ Note also that our clients typically will run these jobs to clean up existing data when they first install the application. These jobs are used more as cleanup or maintenance jobs than an ongoing solution, since triggers, processes or flows usually work better for that type of near-real-time verification.
- 1) To set up a job, go to **Setup → Installed Packages → Address Verification → Configure** to take you to the setup page.
 - 2) Click on the **AUTOMATION** tab and scroll down to the “Batch Jobs” section and expand the section. Here is where you can turn on this job.
 - a. Click on New Job
 - b. Optionally give this job a name.
 - c. Select the **Verification Type**: Physical Addresses, Emails, or Phone Numbers you want to verify. These must be done in separate jobs.
 - d. Select the **Object Name**. If you do not see the object you want to work with, you will need to close this screen and press the “Add Supported Object” button to add a new object.



New Job

Active Job Id text field required. Set it in the Object Setup above.

Job Name ⓘ
Enter Job Name

* Verification Type ⓘ
Physical Addresses

* Object Name ⓘ
Contact

Select the Start job time, the job will run every day same time. Select Advance Scheduler to customization the jobs frequency.

If you see the error above after selecting the Object Name, you must close this popup window, and enter the **Active Job ID Field** for this object as described in the *Turning on Trigger Verifications* section earlier in the chapter. Once you are done entering this field, wait a few minutes, refresh the page, and then come back here and start over adding the job.

- e. Select the **Job Start Time** Or click the **Enable Advance Job Scheduler** options if you are going to schedule the job to run for a time in the future. If you are going to run it now, you can leave these blank.
- f. Click the **Email when Job Complete** option to have an email sent to the current user when the job completes, IF it validated any addresses.
- g. Enter one or more **Email(s)** separated by commas.
- h. If you enter a value other than 0 in the **Expire validations > X days** field, then any validations that were done prior to that number of days in the past will be re-validated when the job runs. This is helpful for organizations that must validate their addresses periodically.
- i. Expand the “Job Filters” section and choose one of three ways to filter the records.
 - i. Select a report from the “Address Verification” report folder.
 - ii. Select a list view from the selected object. Note: *The first time you use this filter, you will have to add a remote site setting.*
 - iii. For the more technical, enter a value into the **SOQL Filter** field or press the Open Query Builder button to launch the SOQL Query Builder.

Note: Do NOT turn on this Email option on large jobs. It slows down the job by adding an extra step that, depending on how many records you have, could be very significant. Rather, monitor longer jobs in the Apex Jobs area in Setup.

The screenshot shows the SOQL Builder interface. At the top, it says "SOQL Builder". Below that, there's a "Filter" section with two radio buttons: "AND" (selected) and "OR". To the right of these is an "Add" button. Below the filter section is a table with five columns: "No", "Field", "Operator", "Value", and "Action".

No	Field	Operator	Value	Action
1	Type	=	Customer	×
2	smartystreets__Billing_Verified__c	=	True	×

Below the table, there's a "Filter Logic" section showing "1 AND 2".

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- iv. Add as many fields as necessary to restrict the query to only verify the records you really want to verify. Once done, press the **Validate** button to check your results.
- v. If everything looks good, press the Use Query button to take you back to the main scheduling area.

Note: You might want to manually add a “Limit X” at the end of the query to further restrict your query, in particular when you are testing out the job.

The SOQL language is fairly straightforward, but if you have not used this type of language in the past, it might be a bit hard to work your way through writing filters. The SOQL Query Builder should help, and here is a resource that can turn you into an expert on SOQL. If you still cannot figure out how to set up your filter, email our support team and we will try to help.

https://resources.docs.salesforce.com/latest/latest/en-us/sfdc/pdf/salesforce_soql_sosl.pdf

- ❖ Please note that the **ORDER BY** clause is NOT supported in the filters.
 - ❖ Please note that if you want to filter on an ID field, you **MUST** use the 18-character (case-**ins**sensitive) version, and never the 15-character (case-sensitive) version.
- j. Now press the **Test the Query** button next to the type of filter you have chosen to use. This will tell you how many records *might* be impacted (it doesn’t take into account records that might be blank or have already been verified).
 - k. Press the **Schedule Job** button to schedule the job for the time you selected in the future. Or, if you prefer to run the job one time, you can press the **Run Job** button.
- 3) You are done! You can now press the **Go to Apex Jobs** to watch the “Run Now” jobs, or the **Go to Scheduled Jobs** to see the job you scheduled for the future.
 - 4) You can also View or Delete jobs from this screen. Do not abort them directly in the Apex Jobs, if at all possible.

Batch Jobs						
Action	Job Name	Verification Type	Object	Created Time	Status	Job Type
View Delete	Test Job 1	ADDRESS	Account	10/3/2022, 03:52 PM	Completed	BATCH
View Delete	10/3/2022, 1:55 PM-EMA-00Q-268	EMAIL	Lead	10/3/2022, 03:55 PM	Aborted	BATCH
View Edit Delete	10/3/2022, 1:56 PM-ADD-001-256	ADDRESS	Account	10/3/2022, 03:56 PM	WAITING	SCHEDULE

- 5) When running batch jobs, a hidden object called Deputy is used. These records could grow to be significant and are not needed after they have been processed. Please follow these steps to schedule a nightly job that will delete them from your system:
 - a. Go to Setup → Custom Code → Apex Classes and click the “Schedule Apex” button.
 - b. For the Job Name, call it something like “Address Verification Batch Cleanup”
 - c. For the Apex Class, select or type in “DeputyRecordDeletionBatch”
 - d. Schedule it to run once a week, or nightly if you prefer, at a specific time. It is best to choose a time off hours, and after any Address Verification batch jobs are scheduled to run, so that it does not slow any other process down.
 - e. Press Save and these records should now be cleaned out.

Email-Powered Record Verification

If you want to verify records using a Salesforce Guest user of some sort, or an external process, one option is to set up an email service that will process one or more records at a time.

- ❖ This can also be used if for some reason you need to change the context user in Salesforce. If you have one user running a flow, for example, but cannot give that user rights to verify a particular record, you can use the Email Record Verification option in order to switch the verification to a user with the correct rights (e.g. with the Address Verification permission set assigned.)

Following are the simple steps to using this verification:

1. Go to Setup → Custom Code → Email Services and click the **New Email Service** button.
2. Enter the following and then press the **Save and New Email Address** button:
 - a. Enter a name in the **Email Service Name**
 - b. Select the “AddressVerificationForSiteRecords” **Apex Class**
 - c. Leave **Accept Attachments** as “None” and **Advanced Email Security Settings** OFF
 - d. Usually you want to use **Accept Email From** blank. But if there is only one email address that will be sending emails to verify them, then you might want to enter an email address here
 - e. Turn ON the **Active** checkbox

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- f. You can decide if you want to use **Error Routing** of any kind in the section below. It might be a good idea so that you can ensure that emails are being sent correctly.
3. On the next screen enter the items below, and then press the **Save** button.
 - a. Any **Email Address Name** and **Email Address** are fine
 - b. Leave **Active** checked
 - c. Select the **Context User** which you want to run the verification as. This user should have the “Address Verification” or “Address Verification (NonAdmin)” permission set assigned.
 - d. As above, remove any email address from the **Accept Email From** and leave it blank unless you want to restrict this service.
4. You will see an email address record will be created. Copy the whole email address from the **Email Address** field (it will be very long) and save this as the email address that you will need to use.
5. Now you can test your new verification setup by sending an email like this:
 - a. From: whoever (unless you filtered it above)
 - b. To: The Email Address you copied in the prior step
 - c. Subject: Anything you want
 - d. Body: A simple text of JSON as shown below. The highlighted “Object Name” should be replaced with the API name of the object which you are verifying, and the “Record Id” should be the actual Id of the record to verify (note that if you want to, you could include multiple Record Ids, as long as they are separated by commas).

```
{
  "vType": "Address",
  "sObjectName": "Object Name",
  "recordIds": [
    "Record Id"
  ],
  "isRecheck": false
}
```
6. Give it a minute or so to take effect, and then check the record in Salesforce to confirm that it was verified.
7. If all looks good, you will now need to turn this on. Go to the Configuration page for the app (go to **Setup → Installed Packages → Address Verification → Configure** to take you to the configuration page.)

8. Scroll down to the bottom and paste the email address from Step #4 into the **Email Service Address** field.
9. Press the green **Save All Settings** button to save. From now on, guest users who have the “Address Verification (Guest User)” permission set attached and create a record with an address, should be verified via this new email service.
 - a. Please note that the ability to **Send Email must be turned on** for this to work, since the User Account (often a guest license user) needs to send the email as configured above. For this a few settings might be necessary:
 - i. The Email address for the Site Guest User (Setup → Feature Settings → Digital Experiences → All Sites → Workspaces → Administration → Pages → Go to Force.com → Public Access Settings → Assigned Users → Guest User) and set, and verify, the email address.
 - ii. The Email address for the Digital Experience (Setup → Feature Settings → Digital Experiences → All Sites → Workspaces → Administration → Emails → Email Address) and set, and verify, the email address.
 - iii. An **Organization-Wide Email Address** (Setup → Email → Organization-Wide Address) may need to be set up and verified, for the guest user who is sending the email.

Turning Off All Triggers

You can turn off the Account, Contact and Lead triggers installed with our app, along with any you create using the information above. This is useful mostly for troubleshooting or before doing large data processing jobs where you want as little overhead as possible.

If you turn off these triggers permanently, you will probably want to re-create the functionality using two separate Flows, as has already been described earlier in this document.

Logging Verification Results

You are able to store logs of the verifications you have performed. Please note that turning on this option might result in a lot of records being created in the Verification Logs object included with the app.

To see these logs, you first have to make sure that you have access to the object and all of the fields in the object. Once you do, you also need to make sure that the tab is not hidden for your profile. You will need to talk to your Salesforce administrator for both of those things. If you don't have an administrator, please reach out to us at support@toafinish.com and we can do it for you if you can provide us remote access.

Advanced Options

Please explore our GitHub.com repository for ways that you can automate some of the address verification features within Salesforce.com. We may also have code that is not included in the public repository yet, so let us know if have any specific requirements and can't find what you need. We cannot support this code for free, but please use it and contact us if you need technical assistance.



GitHub.com Address Verification Sample Code Repository

<https://github.com/ToAFinish/addressverification>

Setting Up Autocomplete

Setting Up Auto completion and Immediate Verification on Insert and Edit of records

In this section you will learn to configure the autocomplete feature in your Salesforce org. This feature needs to first be purchased, and then must be configured. When you press the “Save” button, it then sends a regular verification call to Smarty, and this one is charged. As with the previous steps, we need to switch to the “Setup” area.

Note: This component is available for Physical Addresses only.

New Lead: Regular

Address Information

Street

415 Mission St, San

415 Mission St, San Angelo TX

415 Mission St, San Marino CA

415 W Mission St, San Jose CA

415 E Mission St, San Jose CA

415 Mission St, San Miguel CA

415 Mission St, San Antonio TX

415 Mission St, San Francisco CA

415 Mission St, San Luis Obispo CA

415 Mission St, San Juan Bautista CA

415 Mission St, San Juan Capistrano CA

City/Province

Country

State

Description

Initial Configuration

- 1) Ensure that you have completed the “Enabling Autocomplete” section in *Chapter 2: Initial Configuration* to set up the web key in the configuration page. This will open up the Smarty Autocomplete API to allow processing your addresses. Note that you may need to do this in any Sandbox as well as in your Production org.
- 2) Next, for the object that you want to set up, you will need to look at the layouts that are being used, and for each one, please ensure that it contains a section called “Address Information” (or whatever you have set the section name to be in the Object Setup entry in the Configure page) with the address field(s) inside of it; ONLY these fields in it.

The screenshot shows a Salesforce record layout for a Lead. The layout is divided into several sections:

- Lead Information (Header visible on edit only):** Contains fields for Lead Owner (Sample Text), Name (Sarah Sample), Company (Sample Text), Phone (1-415-555-1212), Mobile (1-415-555-1212), and Email (sarah.sample@company.com).
- Address Information:** This section is highlighted with a red box. It contains the Address field with the value: Suite 300, The Landmark @ One Market, San Francisco, CA 94105, US.
- Description Information (Header visible on edit only):** Contains the Description field with the value: Sample Text.
- International Information:** Contains fields for Registered Organization (Sample Text), Thoroughfare Name (Sample Text), Sub-Building Number (Sample Text), and Super-Administrative (Sample Text).

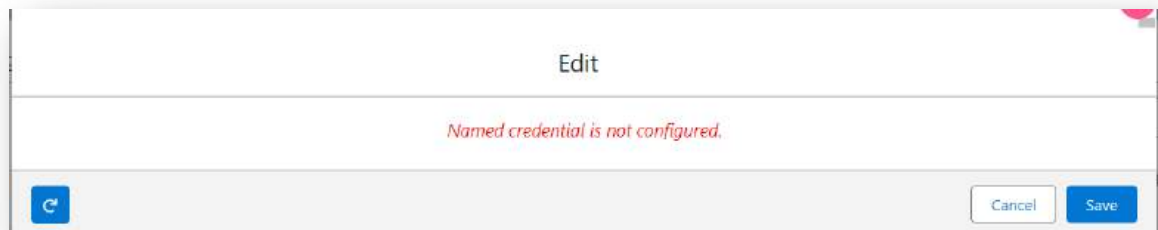
- 3) Finally, make sure that the address you want to use with Autocomplete is set up in the configuration page. If you are not sure, go to **Setup → App → Installed Packages → Address Verification → Configure**, and click on the “Data Mapping” tab. The address must be listed here. If not, then it must be added.
 - a. Please note that the country field, even if you are only verifying US addresses, must be mapped for each address, and included in this section.
 - b. You can turn on the Street 2 field by populating this field and also turning on the “Use Street 2” feature at the bottom of the address configuration. (Also, Street 3 and Street 4, if necessary.)
 - c. You can enforce a correct address by removing the “Use as Is” button when saving, by turning on the “Stop record creation on bad addresses (Autocomplete only)” feature also towards the bottom of the address configuration.
- 4) Now, to **Setup → Object Manager** and then go to the object where you want to set Autocomplete up.

- 5) Select "Buttons Links and Actions" and edit either the standard "New" or "Edit" actions, depending on if you want to set it up for Inserts or Edits, or both.
- 6) From the override section, update the "Lightning Experience Override" item and select **smartystreets:AutoCompleteNew** if you are doing the "New" action, or **smartystreets:AutoCompleteEdit** if you are doing the "Edit" action.
- 7) Now, wait 10 seconds, and then go back to Salesforce, to the tab of the object where you set it up, and click the "New" button (or go to an existing record and press the "Edit" button). You might have to refresh the page, and do it a couple times, because sometimes Lightning takes a bit to refresh the page layout*. But once you press the button, and scroll down to the address section, you will notice it looks a bit different. And when you start typing into the Street, you will see a dropdown.

** You can also turn off this feature by going to Setup → Session Settings → and unchecking the “Enable secure and persistent browser caching to improve performance.” (You might want to turn it back on after you have refreshed the pages.*

Named Credential for Security

IF when you test the AutoComplete screen, you see an error indicating that the “Named credential is not configured” then you must follow these additional instructions.



- 1) In Lightning, go to Setup → Search for “Auth. Providers” and click on the **New** button and follow the below steps.
 - a. Choose “Salesforce” for the **Provider Type**.
 - b. Enter a name for both the **Name** and the **URL Suffix** fields.
 - c. For the **Consumer Key** and **Consumer Secret**. Put “NA” for both. They will be changed later.
 - d. For the **Default Scopes**, enter “refresh_token full”
 - e. Check the **Use Proof Key for Code Exchange (PKCE) Extension** option

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- f. Check the **Include Consumer Secret in SOAP API Responses** option and then **Save**.
- g. Copy the **Callback URL** and note it somewhere. It will be needed later.

Auth. Provider

Auth. Provider Edit Save Save & New Cancel

Auth. Provider ID	0SOKd0000008lq3
Provider Type	Salesforce
Name	AName
URL Suffix	AName
Consumer Key	NA i
Consumer Secret	NA i
Authorize Endpoint URL	https://login.salesforce.com/services/oauth2/authorize i
Token Endpoint URL	https://login.salesforce.com/services/oauth2/token i
Use Proof Key for Code Exchange (PKCE) Extension	☒ i
Default Scopes	refresh_token full i
Include Consumer Secret in SOAP API Responses	☒ i
Custom Error URL	
Custom Logout URL	i
Registration Handler	i Automatically create a registration handler template
Execute Registration As	i
Portal	--None--
Icon URL	 Choose one of our sample icons
Use Salesforce MFA for this SSO Provider	☐ i
Created Date	10/22/2024, 8:38 AM

Save Save & New Cancel

- 2) Now, you must Set up an **External Client App** in order to provide sufficient rights to our app for the Autocomplete functionality.
- 3) If you are not there, navigate to **Setup** in Lightning Experience. In the Quick Find box, search for “External Client App Manager” and open it.
- 4) Click on the **New External Client App** button.
- 5) Enter the required basic details:
 - a. **External Client App Name**
 - b. **Contact Email**

Setting Up an External Client App

The following instructions are somewhat technical, and could be in flux for some time. Salesforce recently introduced External Client Apps as a replacement for Connected Apps, and they are a bit trickier to use in this environment.

An “**INSUFFICIENT PRIVILEGES**” error on one of the upcoming steps (usually step #15) means you will need to open a case with Salesforce tech support so that they enable “External Client Apps” in your org.

- c. **Contact Phone**
 - d. Set **Distribution State** to *Local*
- 6) Enable OAuth settings:
- a. Check **Enable OAuth Settings** under the API section.
 - b. Once enabled, the **App Settings** section will appear. Provide the following:
 - i. **Callback URL**: Paste the URL copied in Step 1.
 - ii. **OAuth Scopes**: Select the following:
 - Manage user data via APIs (*api*)
 - Full access (*full*)
 - Perform requests at any time (*refresh_token, offline_access*)
- 7) Configure security settings:
- a. Enable **Require Secret for Web Server Flow**
 - b. Enable **Require Secret for Refresh Token Flow**
- 8) Click **Create**. A success message will be displayed upon successful creation.
- 9) After creation:
- a. Go to the **Settings** tab.
 - b. Navigate to **OAuth Settings**.
 - c. Click on **Manage Consumer Details** under the Consumer Key and Secret section.
 - d. You may be prompted to enter a one-time verification code sent to your registered email.
 - e. Copy and securely store both the **Consumer Key** and **Consumer Secret** for future use. (A notepad usually works well.)
- 10) Now, Go to the **Auth Provider** you created in step 1 and **Edit** it
- 11) Replace the consumer key and consumer secret with those you got from step 9 and **Save**.

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

SETUP
Auth. Providers

Auth. Provider Edit

Save Save & New Cancel

Auth. Provider ID: 0SO5j000000qFhm
Provider Type: Salesforce
Name: CallMeBack_AuthProvik
URL Suffix: CallMeBack_AuthProvik
Consumer Key: 3MVG9pRzvMkjMb6lu7
Consumer Secret: 75544BBA8A79AAA5Fi
Authorize Endpoint URL: https://login.salesforce.com/services/oauth2/authorize
Token Endpoint URL: https://login.salesforce.com/services/oauth2/token
Use Proof Key for Code Exchange (PKCE) Extension: ☐
Default Scopes: refresh_token full
Include Consumer Secret in SOAP API Responses: ☒
Custom Error URL:
Custom Logout URL:
Registration Handler: Automatically create a registration handler template
Execute Registration As:
Portal: --None--

12) Switch to salesforce Classic. This next step doesn't currently work in Lightning.

13) Go to Setup → Search “Named Credentials” and click on Named Credentials.
Next, click on **New Named credential** button to create new credentials

Named Credentials

Help for this Page

A named credential specifies a callout endpoint and its required authentication parameters. When setting up callouts, avoid setting authentication parameters for each callout by referencing named credentials.

View: All Create New View

New Named Credential

14) In the fields, enter the following values:

- For **label/name** enter CallMeBack or whatever name you prefer
- In **URL**, put the org instance URL. For this, you must look at the current URL while in Classic mode, and copy from the https to “salesforce.com/” – something like this:
<https://xxx.my.salesforce.com/> (“xxx” will be different for you)
- Identity Type → Named Principal
- Authentication Protocol → OAuth 2.0

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- e. Authentication Provider → choose **OrgAuthentication_AuthProvider** (it was created in step 2)
- f. Scope → refresh_token full
- g. Start Authentication Flow on Save → True
- h. Generate Authorization Header → True
- i. Allow Merge Fields in HTTP Header → True
- j. Allow Merge Fields in HTTP Body → True

Below you can see a screenshot of what it might look like.

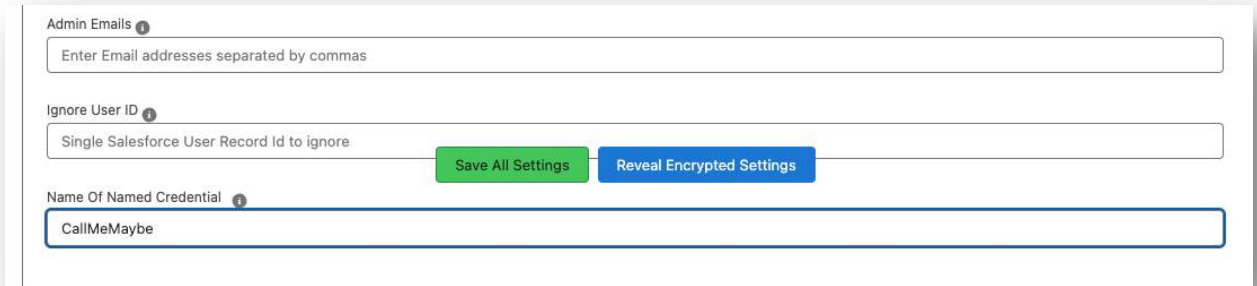
The screenshot shows the 'New Named Credential' configuration window in Salesforce. The window has a title bar 'New Named Credential' and a subtitle 'Specify the callout endpoint's URL and the authentication settings that are required for Salesforce to make callouts to the remote system.' Below the subtitle are 'Save' and 'Cancel' buttons. The form is divided into several sections:

- Label**: CallMeMaybe
- Name**: CallMeMaybe
- URL**: https://XXX.my.salesforce.com/
- Authentication** (expanded section):
 - Certificate**: (empty field with a search icon)
 - Identity Type**: Named Principal (dropdown menu)
 - Authentication Protocol**: OAuth 2.0 (dropdown menu)
 - Authentication Provider**: AName (dropdown menu with a search icon)
 - Scope**: refresh_token full
 - Authentication Status**: Pending
 - Start Authentication Flow on Save**: ☒
- Callout Options** (expanded section):
 - Generate Authorization Header**: ☒
 - Allow Merge Fields in HTTP Header**: ☒
 - Allow Merge Fields in HTTP Body**: ☒
 - Outbound Network Connection**: (empty field with a search icon)

At the bottom of the form are 'Save' and 'Cancel' buttons.

ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- 15) Click **Save**, and when you do, it will send you to a login page. Put your details. On successful login, it will ask for confirmation of granted rights on two different screens, and after you accept, it is all set.
- 16) Go to the configuration page again (Setup → Installed Packages → Address Verification → Configure) and on the first tab, scroll all the way down and enter the name of the new Named Credential you just created and then press the **Save All Settings** button.



- 17) Great Job! You should be all done.

Child Object Usage

Both Autocomplete Lightning components should work with any object inside of Salesforce. However, they are stand-alone, so if used with a child object, they will not automatically link you back to the parent object.

For example, if you are on an Account and click the “New” button from the Contact related list, it would allow you to add the Contact, but it would force you to also update the Account lookup field in order to connect it back to the Account. Not optimal.

In order to automatically link a child account, you would need to set up a custom Lightning component, as described below:

- 1) First, go to our GitHub code repository and in the “lightning” folder, you can find the **child_object_create_quick_action** code. It is also listed on the next page, though getting it from the repository ensures you have the most recent version.

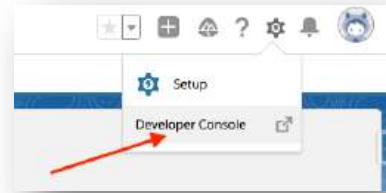


GitHub.com Address Verification Sample Code Repository

<https://github.com/ToAFinish/addressverification>

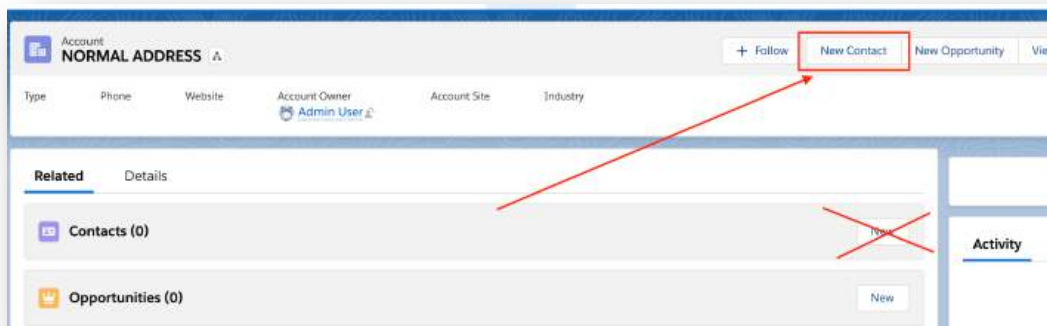
ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- 2) Once you have the code, click on the Setup “gears” icon and select **Developer Console**. This will open a new window with the Developer Console.
- 3) In the console, click File → New → Lightning Component, and then enter a name for the component that will tell you what object it is going to be used for, and press the **Submit** button.
- 4) Once you do that, you will be put in the “Component” element, with some default code. Delete that code, and replace it with the code you pulled from the GitHub repository, a version of which is below.
- 5) You will want to change the **OBJECT** highlighted below with the actual API name of the child object, and the **FIELD** highlighted below with the actual API name of the lookup field, pointing at the parent object. For example, for usage with a Contact, it should say “Contact-AccountId”. After updating these items, go to File → Save. Now you can close the Developer Console.



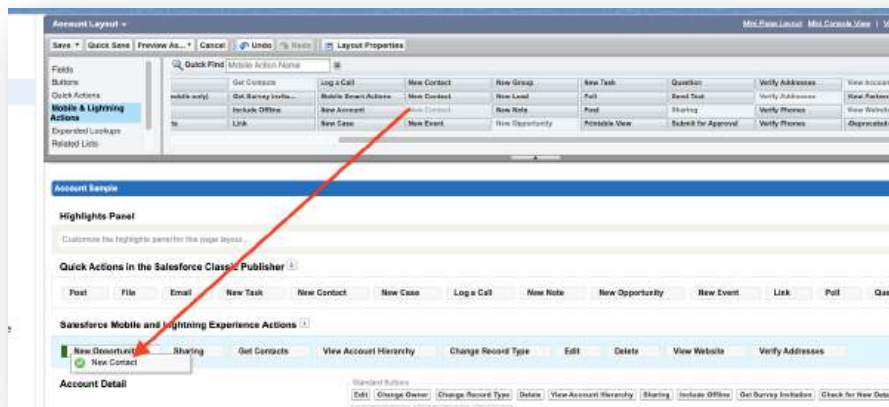
```
<aura:component implements="flexipage:availableForRecordHome,
force:lightningQuickActionWithoutHeader,force:hasRecordId,force:hasSObjectName">
  <aura:attribute name="sObjectName" type="String" access="global"
    description="The parent sObject Name"/>
  <aura:attribute name="recordId" type="String" access="global"
    description="The parent record Id"/>
  <smartystreets:AutoCompleteConnector sObjectName="{!v.sObjectName}"
    recordId="{!v.recordId}" currentObjectInfo="OBJECT-FIELD">
  </smartystreets:AutoCompleteConnector>
</aura:component>
```

- ❖ Now, you need to create the button that will launch the *AutoCompleteNew* component, linking to the parent record. Unfortunately, you cannot use this to override the “New” function for the child record, but rather must launch it from a button on the parent object. See below for a visual:



ADDRESS, EMAIL, PHONE VERIFICATION FOR SALESFORCE

- 6) Go to Setup → Object Manager and find the parent object of the record you will be creating the child object from. Click on it.
- 7) Now, click on the “Buttons, Links, and Actions” menu selection from the left, and when it comes up, press the **New Action** button.
 - a. Select **Lightning Component** for the “Action Type”
 - b. Select the component you just created for the “Lightning Component”
 - c. Enter **0px** for the “Height”
 - d. Enter whatever “Label” and “Name” you want and **Save**.
- 8) Go back to the same object, and this time click on the “Page Layouts” menu selection on the left. Find the layout you want to use this button with, which might be more than one if you are using multiple layouts for different purposes. Click on the Layout name.
 - a. From the “Mobile & Lightning Actions” menu item, drag your new Action into the “Salesforce Mobile and Lightning Experience Actions” section. Note that you might have to open this section first.



- b. Go down to the related list of the child object and edit the list to remove the “New” button. This is an Optional, but recommended step, to remove confusions.
 - c. Now save the layout, and if necessary, do the same with other layouts.
- 9) You are ready to test your new button. When you click it, the parent record should already be populated in the appropriate field.

Important Notes About Overriding New/Edit Functions

When you use our Lightning components, which are necessary only if you want to use the Autocomplete feature, several administrative notes are necessary. This applies to the **AutoCompleteNew** component for overriding the “New” function on an object, or the **AutoCompleteEdit** one for overriding the “Edit” function.



- ❖ Another important note is that any time the AutoCompleteEdit Lightning component is used to override the Edit function, Salesforce turns off the Inline Editing for that object. This is not something that our app does, but rather standard Salesforce behavior. Because of this, many of our customers prefer to create a custom Lightning Action that calls the AutoCompleteEdit component, rather than overriding the standard Edit function.
- ❖ The first time you use this component, it will take a bit longer than usual, as it builds a cache of the screen to use in future sessions. This is helpful because it will significantly reduce the load time going forward. However, some users may not have the rights necessary to read the object layout metadata. To do so, users must have these rights via their profile or a permission set:

1. Customize Application
2. API Enabled

If you have users who cannot have these rights, you can still use the Lightning component by first opening the Lightning component (either the **AutoCompleteNew** or the **AutoCompleteEdit**) as an Administrator, and then going to Setup → Installed Packages → Address Verification → Configure and go to the “More...” Tab. You will see a “Page Layout Sync”. Select the object you want to use with AutoComplete, and the Profile of the users who cannot have the necessary rights to build the cache themselves, and then press the “Sync Page Assignment” button. This will start an Apex Job that will copy the cache to their profiles.

Please note that you will have to do this each time a change is made to the layout!

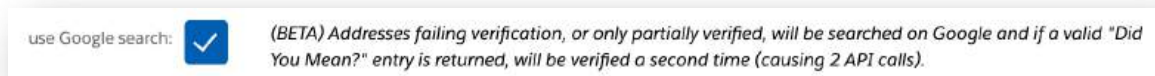
Advanced Topics

Address Verification App Extended

In this section you will learn about other areas where Address Verification can be employed to great effect. These uses of the app will extend the capabilities beyond general usage, and typically will require some more advanced configuration, and in some cases development.

Using Google Search to Improve Verifications

By turning on the Google Search function, you will be able to improve some verification results. You can find this function in the Configure page.



You will also need to add the following URL to your **Remote Site Settings** setup page:

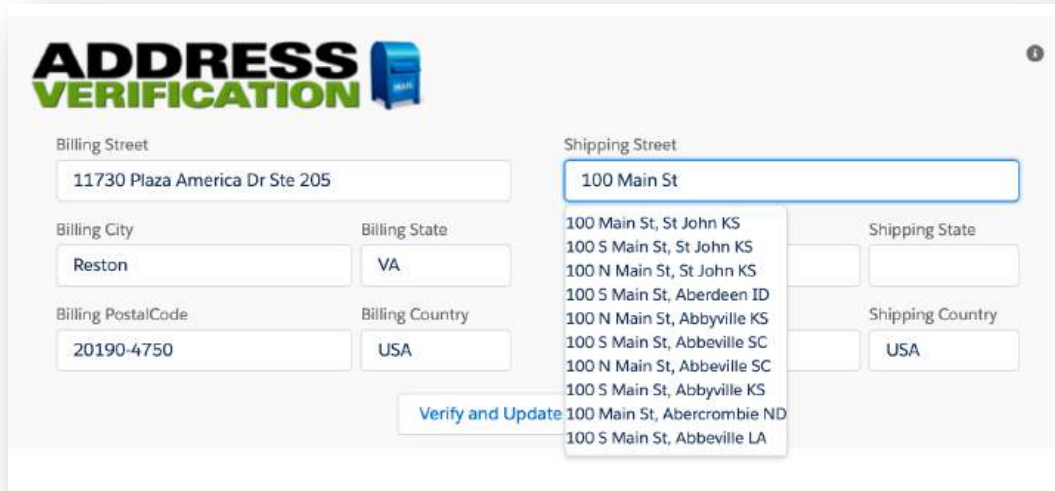
`http://www.google.com`

The way this function works is that it will first send the verification to Smarty and if it returns a bad address flag, then it will search Google for the address. If Google returns an alternate spelling of the address through their “Did you mean?” feature, a secondary verification call will be sent to Smarty with the spelling change that Google suggests. If this one is successful, then the address is verified successfully, if not this is noted in the footnotes.

Note that this function only works with manual verifications, and not with daily scheduled job verifications. It is best used in attempts to figure out a particularly difficult address, especially in situations where the local language might be affecting how an address is spelled.

Address Verification in a Flow

Newly released is the ability to verify one or more addresses in a step of a Flow. Either verifying an existing record, or verifying an address independently of any existing records.



The screenshot displays the ADDRESS VERIFICATION interface. It features two main sections: Billing and Shipping. The Billing section includes fields for Billing Street (11730 Plaza America Dr Ste 205), Billing City (Reston), Billing State (VA), Billing PostalCode (20190-4750), and Billing Country (USA). The Shipping section includes fields for Shipping Street (100 Main St), Shipping State, and Shipping Country (USA). A dropdown menu is open under the Shipping Street field, showing a list of suggested addresses: 100 Main St, St John KS; 100 S Main St, St John KS; 100 N Main St, St John KS; 100 S Main St, Aberdeen ID; 100 N Main St, Abbeville KS; 100 S Main St, Abbeville SC; 100 N Main St, Abbeville SC; 100 S Main St, Abbeville KS; 100 Main St, Abercrombie ND; and 100 S Main St, Abbeville LA. A 'Verify and Update' button is located at the bottom of the form.

All of the code to build the Lightning Components that work in the Flow, along with detailed instructions, can be found in the "flow" folder in our GitHub repository:

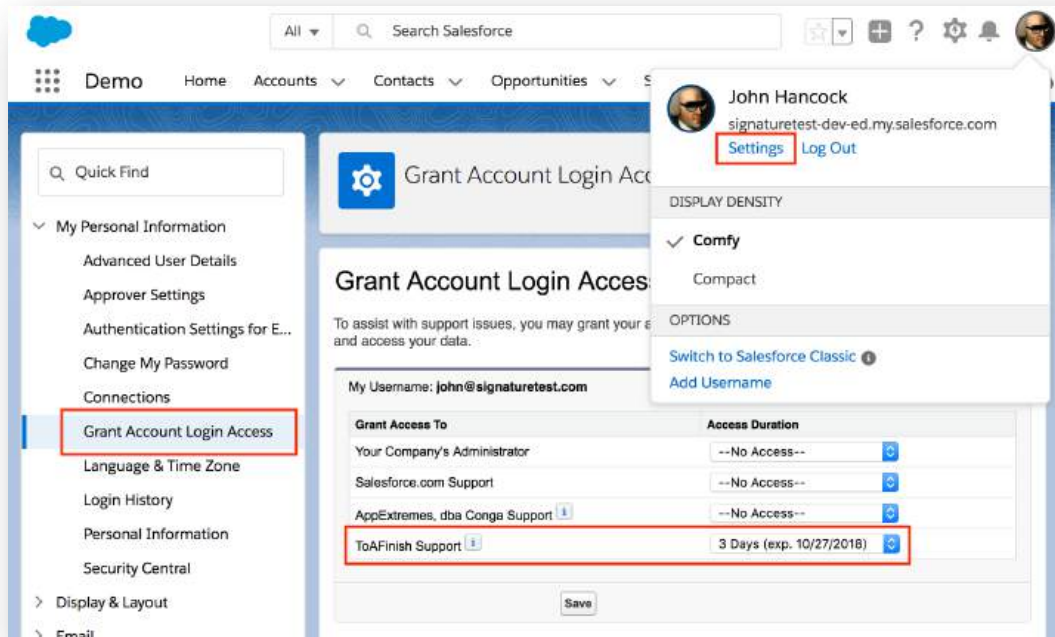


GitHub.com Address Verification Sample Code Repository

<https://github.com/ToAFinish/addressverification>

Feedback and Remote Access

Please send us your feedback to support@toafinish.com, and don't hesitate to email if you have questions about these instructions. When you do, you might want to give us remote access, as shown in the screenshot below, so that we can log in as you to see the problem.



We greatly enjoy being able to provide users with an app they will find useful. So, if you like the app, please leave us a positive review on the AppExchange:

<https://appexchange.salesforce.com/listingDetail?listingId=a0N300000059eEUEAY>

But if you run into issues, *please* don't leave them there for the world to see (unless we can't get it resolved). Rather, contact us at support@toafinish.com and we'll work together to make this app To A Finish™.

Thank you!

The To A Finish™ team.