



# IMPLEMENTATION GUIDE



ADDRESS VERIFICATION FOR SALESFORCE.COM

# Implementation Guide

---

© 2011-2019 To A Finish LLC  
12407 MoPac Expwy N • Suite 250  
Austin, TX 78758  
Phone 512.436.3464 • Fax 267.935.7773

---

# Table of Contents

<b><i>Introduction to this Implementation Guide</i></b>	<b>5</b>
<b>Introducing To A Finish</b>	<b>6</b>
<b>Introducing the Address Verification App</b>	<b>7</b>
<b>Before You Begin, Requirements</b>	<b>8</b>
<b><i>Initial Configuration</i></b>	<b>9</b>
<b>Signing Up for a SmartyStreets Account</b>	<b>10</b>
Enabling Autocomplete	12
Entering API Keys	13
International Verification Options	13
<b>Optional Settings</b>	<b>14</b>
Setting Up Default Addresses	15
Remote Site Entries	18
<b>Signing Up for BETA</b>	<b>18</b>
<b><i>Configuring Standard Objects</i></b>	<b>19</b>
<b>Accounts, Contacts &amp; Leads</b>	<b>20</b>
Turning on Field History	20
Modifying Layouts	22
<b><i>Configuring Custom Objects</i></b>	<b>24</b>
Creating Mandatory Fields	24
Creating Optional Fields	25
Creating International Fields	28
Creating Custom Address Records	29
<b>Lightning Setup</b>	<b>30</b>
Creating the Custom “Verify” Button	30
<b>Classic Setup</b>	<b>32</b>
Creating the Custom “Verify” Button	32
<b>Setup for All Scenarios</b>	<b>33</b>
Un-verifying Records On Change	33
Turning on Field History Tracking	36
Modifying the Layout	36
<b><i>Automation and Bulk Verification</i></b>	<b>37</b>

---

<b>Manual Bulk Verification</b>	<b>37</b>
List Verification for Standard Objects	37
List Verification for Custom Objects	40
<b>Automated Verification Options</b>	<b>41</b>
Standard Object Trigger Verifications	41
Address Pre-Cleaning Option	41
Custom Object Trigger Verifications	44
Custom Object Process Verifications	45
Daily Scheduled Verification Job	48
Turning Off All Triggers	49
Logging Verification Results	49
Advanced Options	49
<b>Setting Up Autocomplete</b>	<b>50</b>
Initial Configuration	51
Lightning Experience Configuration	52
Classic Configuration	52
Important Note for Using Custom Edit Pages	52
Add the "Create Edit Page" Link	53
Enabling the New Custom "Edit" Page for New Records	54
Enabling the New Custom "Edit" Page with Existing Records	55
<b>Autocomplete in a Flow</b>	<b>57</b>





Version 2019.03.26

## 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.

ICON KEY	
	Details You Can Skip
	Don't Miss This!
	Sample Code
	Danger!

**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.

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.

Most of the configuration, and all of the advanced material, has been moved to an online repository. You can find it using the link below:

GitHub.com Address Verification Repository

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

## 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 mostly 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. This is not a blanket statement about all consulting and app companies, rather a general statement of 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 us. Allow us to show you the difference of doing work To A Finish™.

You can find more about To A Finish at [www.toafinish.com](http://www.toafinish.com)



*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



Features a very simple way to cleanse addresses within your Salesforce or Force.com system. Automatically, or with the touch of a button, addresses on the current record are verified and cleansed. Works in standard or custom objects.

Try it for free for 30 days! (On the Annual pricing package only...)

This app works with the top-rated SmartyStreets Address Cleansing service. It can also work with various other APIs for data augmentation, including company, phone, and soon email verifications as well.

Address Verification for Salesforce provides buttons on the Account, Contact and Lead layouts which instantly verify and cleanse the standard addresses. Easy instructions for adding buttons for custom objects and/or addresses are provided in later sections of this manual.



If you use SmartyStreets, return statuses are included, along with demographic data like Latitude and Longitude, County Name, FIPs Code, and much more.

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 should work.

### **SmartyStreets Account Option**

SmartyStreets offers an address cleansing service, which our app uses. Please keep in mind that SmartyStreets.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 possibly recommended in a few cases:

- 1) SmartyStreets offers a free monthly plan with 250 verifications. If your needs are limited to only a few verifications each month, then it might make sense to use an account with them, rather than purchasing through us.

*\* It also makes sense to sign up for this free plan when testing the app.*

- 2) SmartyStreets offers an unlimited monthly or annual package. If you are verifying millions of records, this might be a better option for you than one of our packages, though you should carefully price it to make sure. Contact [sales@toafinish.com](mailto:sales@toafinish.com) to discuss your options.
- 3) SmartyStreets offers International Verifications (verifications for addresses outside of the United States). By contract, we cannot resell these verifications, so you will need a secondary account with them if you need International Verifications.





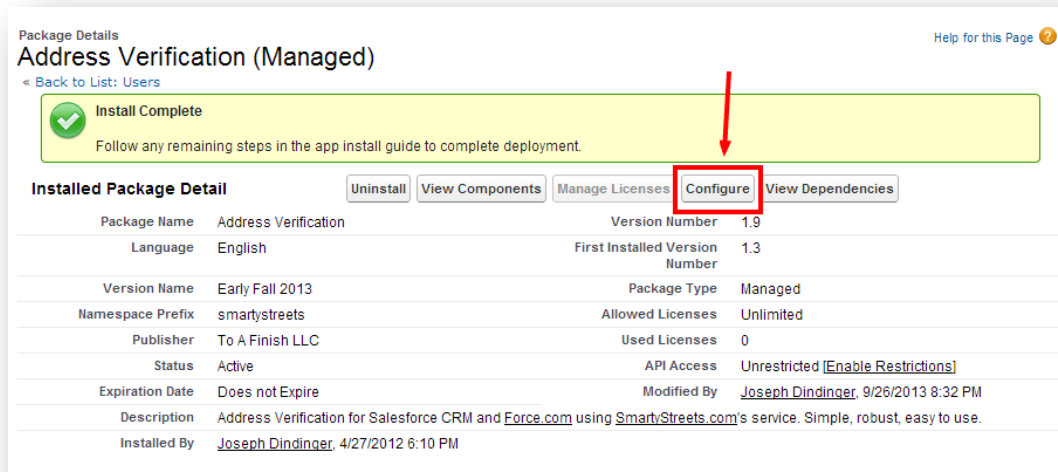
## Initial Configuration

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

The initial steps in the configuration are completed by clicking the “Configure” button on the Address Verification package after it has been installed and pasting in the Security Token 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.



Package Details  
Address Verification (Managed)

« Back to List: Users

Install Complete  
Follow any remaining steps in the app install guide to complete deployment.

Uninstall View Components Manage Licenses **Configure** View Dependencies

Package Name	Address Verification	Version Number	1.9
Language	English	First Installed Version Number	1.3
Version Name	Early Fall 2013	Package Type	Managed
Namespace Prefix	smartystreets	Allowed Licenses	Unlimited
Publisher	To A Finish LLC	Used Licenses	0
Status	Active	API Access	Unrestricted <a href="#">[Enable Restrictions]</a>
Expiration Date	Does not Expire	Modified By	Joseph Dindinger, 9/26/2013 8:32 PM
Description	Address Verification for Salesforce CRM and Force.com using SmartStreets.com's service. Simple, robust, easy to use.		
Installed By	Joseph Dindinger, 4/27/2012 6:10 PM		

## Signing Up for a SmartyStreets Account

The first step is to create an account with SmartyStreets. Setting up this account is free, quick and painless (only the first name, email address, and password are necessary) and you will be presented with a key, which is necessary for the following step.

### ADDRESS VERIFICATION SETUP

WELCOME   MANAGE ADDRESSES   AUTOMATION   BETA

#### SMARTYSTREETS ACCOUNT CONFIGURATION



You will need to log into SmartyStreets.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 SmartyStreets.com, click the button below to sign up for a FREE account with 250 API calls per month. Please note that International Address Verifications are now supported.

auth-id:  
Enter Auth Id

auth-token:  
Enter Auth token

autocomplete key (optional):  
Enter Key

verify international: ☐ + Note: You will need to contact SmartyStreets to get this feature enabled on your account. Click [this link](#) and you can apply for International verifications from there.

 [SmartyStreets.com Signup](#)  [Verify Settings](#)

This will take you to [www.smartystreets.com/pricing](http://www.smartystreets.com/pricing) and you can then click on the **Start Now** button by the free plan to get started.

*\* We recommend starting with the free plan and then either purchasing a plan through us in order to get a discount on the app, or purchasing a plan through SmartyStreets.*

## Checkout

Order Summary

US PLAN

**250**  
lookups per month for

**FREE**

How do SmartyStreets subscriptions work? [Find out here.](#)

Sign Up

First Name:

Phone Number:

Email Address:

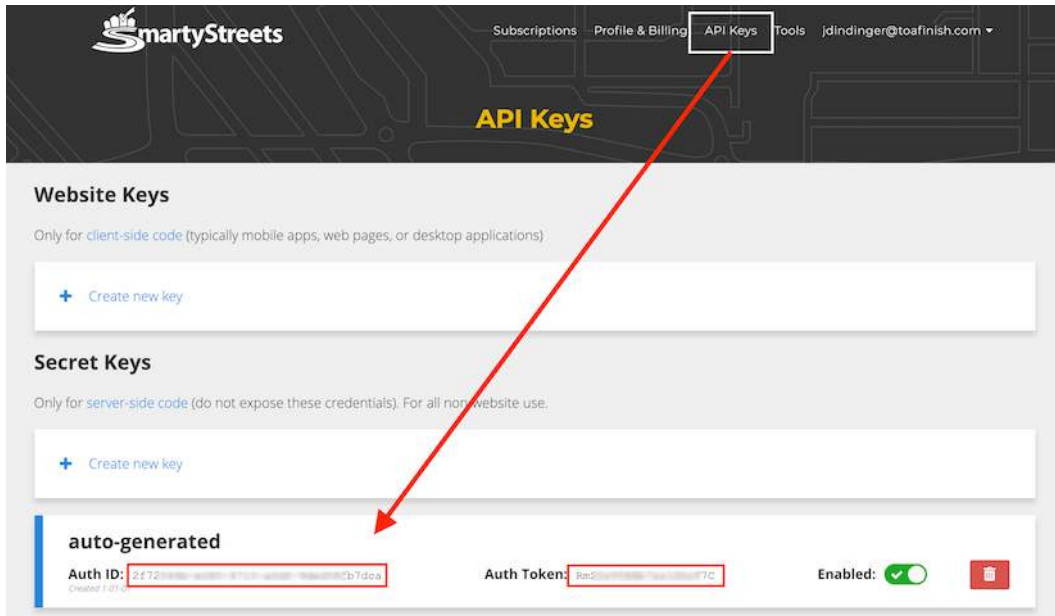
Password:

[Create Account](#)

## ADDRESS VERIFICATION FOR SALESFORCE.COM

Fill out the required fields and press the **Create Account** button.

You should be taken to the Account page. If not, press the **Account** button at the top to go there. Now, click on the **API Keys** menu option, and then under “Secret Keys”, you will see your keys. You can press the Create new key button if you do not wish to use the one that was created for you.



Copy the **Auth ID** and **Auth Token** to a text file. These are the codes you will need in the next step.

## Enabling Autocomplete

Using the Lightning Components for creating and updating records while verifying addresses needs some additional setup. As well, in Salesforce Classic, replacing standard Salesforce edit pages with custom Visualforce ones including an autocomplete+verify feature is available also available, but needs a significant amount of setup.

To use the autocomplete feature, you must first set up a Website Key in your SmartyStreets account. But to do that, you must know which Salesforce instance you are on. You will create the Website Key based on that information.

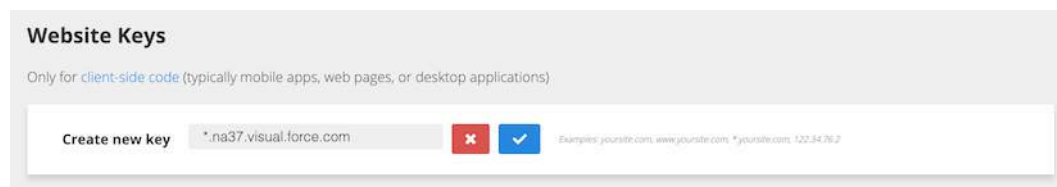
One way to figure it out is to go to: <https://status.salesforce.com/status> and use the "Search Instance or Domain" feature. It will allow you to enter your custom domain and then tell you which is your Salesforce instance. To know what to enter here, you need to copy the URL when you are in Salesforce and find the custom "domain" part of it. For example, in this URL we have highlighted the part that is the custom domain:

`https://yourcompany.lightning.force.com/lightning/page/home?0.source=alohaHeader`

Search for the domain and the search box will tell you which instance you are on, something like NA65 or NA37 or whatever.

Once you know this information, go back to your SmartyStreets "API Keys" page described in the last section and this time generate a website key. You must enter the following exactly under the "Domain or IP" field, where XXXX is your Salesforce instance, in this example on the screenshot we use NA37:

**\*.XXXX.visual.force.com**



**Website Keys**

Only for [client-side code](#) (typically mobile apps, web pages, or desktop applications)

Create new key  ✖ ✓ Examples: yoursite.com, www.yoursite.com, \*yoursite.com, 122.34.76.2

Once you generate the website key, save this one too in the text file in preparation for the next steps.

## Entering API Keys

The third step is simple, assuming that you completed the first step, and optionally the second, and you have the API Keys in a text file ready to be used. Switch back to the configuration page for the Address Verification app in Salesforce.com and simply paste them in the location provided. Then press the **Save All Settings** button *at the bottom of the page*, and you are done.

## ADDRESS VERIFICATION SETUP

WELCOME    MANAGE ADDRESSES    AUTOMATION    BETA

### SMARTYSTREETS ACCOUNT CONFIGURATION

You will need to log into SmartyStreets.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 SmartyStreets.com, click the button below to sign up for a FREE account with 250 API calls per month.

Please note that International Address Verifications are now supported.

auth-id:

auth-token:

autocomplete key (optional):

verify international: ☒ Note: You will need to contact SmartyStreets to get this feature enabled on your account. Click this [link](#) and you can apply for International verifications from there.

**SmartyStreets.com Signup**    **Verify Settings**

### OPTIONAL SETTINGS

All of these settings are optional and should only be used if you understand what they are doing. Please

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

## International Verification Options

You can turn ON or OFF international verifications here as well. Please realize that if you turn it on, then any time you have an international address, it will try to verify it like a US one, but using the International API from SmartyStreets.

In order for it to work, you need to purchase an International Plan from SmartyStreets and have them turn it on. However, even if you haven't turned it on with SmartyStreets, the app will attempt to verify the address, so in order to not see errors, you want to make sure SmartyStreets is set up for International verifications before turning it on.

Note, that we also have the ability to add an additional set of Authorization Credentials that will be used exclusively for International Verifications. This is helpful so that if you purchase your SmartyStreets plan through us, you can have a separate plan for International verifications. (SmartyStreets does not allow the resale of their International Plans.)

## Optional Settings

On the Setup Page's Welcome tab, you can also change these optional settings.

<b>fixed name for USA verifications</b>	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 <b>USA</b> , or <b>US</b> or <b>United States</b> in order for it to function correctly.
<b>default autocomplete country name</b>	The autocomplete function works only for addresses in the United States, so this setting should be left at USA or similar in order for it to work. If you put a different country here, it will restriction the autocomplete function from working currently. (For now.)
<b>only use states in these countries (use for country/state picklists only)</b>	When using Country/State picklists, Salesforce enforces the list of States that you set up. So, if SmartyStreets 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. 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: <b>US, GB, FR, ES</b> in this field.
<b>auto-fill country</b>	<p>If you are using international verifications, you will notice that the countries come back with the standardized 2-character ISO abbreviation. With this option checked, the country designations returned by SmartyStreets will be converted to their full Country names when they are verified. For example, CA would become Canada, and BA would become Bosnia and Herzegovina.</p> <p>Note that this will only work on new verifications; it will not clean up your existing data unless you re-verify it.</p>

## Setting Up Default Addresses

Now we must set up the addresses that the Address Verification app will support. Click on the "Manage Addresses" subtab in the Setup page, and then press the **Create Default Addresses** button in order to create the list of 5 or 6 default addresses. (There will be one additional if you have the "Person Accounts" feature turned on.)

Action	Name	Object	Active?	Street	City	State	Zip
<a href="#">Edit</a>   <a href="#">Del</a>	Account Billing	Account	✓	BillingStreet	BillingCity	BillingState	BillingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Account Shipping	Account	✓	ShippingStreet	ShippingCity	ShippingState	ShippingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Contact Mailing	Contact	✓	MailingStreet	MailingCity	MailingState	MailingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Contact Other	Contact	✓	OtherStreet	OtherCity	OtherState	OtherPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Lead	Lead	✓	Street	City	State	PostalCode

## 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:

Option	Notes about the field
<b>Name</b>	Can be anything you want
<b>Object</b>	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.
<b>Active?</b>	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.
<b>Street</b>	Required
<b>City</b>	Required
<b>State</b>	Required
<b>Zip Code</b>	Required
<b>Verified</b>	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.
<b>Last Verified</b>	Required. Date/Time field that keeps track of the last time the address was verified. (Even if it is no longer verified.)
<b>Return Code</b>	Required. This field will be populated with the result of the verification once it is attempted. Values will be one of these: Confirmed, Not Confirmed, Confirmed - By Dropping Secondary Info, Confirmed - Missing Secondary Info, or Not Submitted.
<b>Street2</b>	Second line of the address. If no field selected, then the 2 <sup>nd</sup> line of the address will be put on the end of the first Street field.
<b>Unit</b>	Apartment or Suite number
<b>County</b>	

## ADDRESS VERIFICATION FOR SALESFORCE.COM

Option	Notes about the field
<b>County FIPs Code</b>	
<b>Country</b>	
<b>Island</b>	This information is not from SmartyStreets, and only supports the Hawaiian and two sets of Spanish islands.
<b>Do Not Verify</b>	Checkbox field that can be used to ensure the particular record is not verified.
<b>Clean Street</b>	Populated with a stripped-down version of the street address, calculated manually, to be used in matching addresses against other records.
<b>Time Zone</b>	
<b>DST</b>	Daylight Savings?
<b>UTC Offset</b>	Number of hours from the UTC standard. Used along with DST to calculate the current address.
<b>Barcode</b>	Barcode value, used in a formula field to visually create a barcode.
<b>Urbanization</b>	Designation only for Puerto Rico
<b>RDI</b>	Residential Delivery Indicator (Residential vs Commercial)
<b>Latitude</b>	
<b>Longitude</b>	
<b>Record Type</b>	Accuracy of verification in descending order: (F)irm, (G)eneral Delivery, (H)igh-Rise, (P).O. Box, (R)ural Route, and (S)treet.
<b>Carrier Route</b>	
<b>Congressional District</b>	
<b>Is Active?</b>	From the USPS, currently delivering mail at this address.
<b>Is Vacant?</b>	From the USPS, have delivered mail, but marked vacant after nobody picked up for a certain amount of days.
<b>Verification Used</b>	Which verification service was used: SmartyStreets, Google, or UPS
<b>Footnotes</b>	Notes about the verification that was performed
<b>Internal Zip Code Database Fields</b>	
<b>Area Code(s)</b>	Not from SmartyStreets. Only available with the optional internal Zip Code Database. Valid Phone Area Code(s) for the zip code.
<b>Urban Area (Code)</b>	Not from SmartyStreets. Only available with the optional internal Zip Code Database.
<b>Urban Area Name</b>	Not from SmartyStreets. 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.
<b>International Only Fields</b>	
<b>Organization</b>	Seldom populated. The name of the recipient, firm, or company at this address
<b>Address Precision</b>	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
<b>Administrative Area</b>	The most common administrative division within a country (e.g. Region)



## ADDRESS VERIFICATION FOR SALESFORCE.COM

Option	Notes about the field
<b>Geocode Precision</b>	How precise is the geocoding of the address? Same values as the Address Precision.
<b>Premise</b>	Alphanumeric code pertaining to an individual location
<b>Premise Extra</b>	Extra information about the premise that is not necessarily authoritative but might still be useful
<b>Sub Building Number</b>	The alphanumeric component of the sub building
<b>Sub Building Type</b>	The leading sub-building type of the sub building
<b>Super Administrative Area</b>	The largest administrative division within a country
<b>Thoroughfare Name</b>	The name component of the thoroughfare
<b>Address Options</b>	
<b>Don't Overwrite Street1</b>	Will not overwrite the entered Street line with what SmartyStreets returns.
<b>Don't Overwrite City</b>	Will not overwrite the entered City with what SmartyStreets returns.
<b>Don't Overwrite State</b>	Will not overwrite the entered State with what SmartyStreets returns.
<b>Don't Overwrite Zip Code</b>	Will not overwrite the entered Postal/Zip Code with what SmartyStreets returns.
<b>Only 5 Digits on Zip Code</b>	Will only populate the first 5 digits of the Zip Code (not the additional 4 digits)
<b>Clean Address Prior to Verification</b>	With this option selected, the field indicated in the “Clean Street” (see above) will be populated with a
<b>Remove Additional Info from Street 1</b>	With this option selected, will try to remove anything that doesn’t look like it is part of the address from the Street 1 field prior to sending it to SmartyStreets.
<b>Clear all verified fields after address change</b>	Typically 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.
<b>Stop record creation on bad addresses (Autocomplete only)</b>	When using the AutoComplete feature, checking this box will remove the “Use as Is” button when an address is not able to be verified.
<b>Use Street 2 with Autocomplete</b>	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.
<b>Use Internal Zip Database (BETA)</b>	When this is selected, following fields will be populated IF they are configured above and are blank prior to the update/insert: City, State, County, Area Codes, County FIPs, Urban Area Code and Urban Area Name.
<b>Allow Verifications with no Street (BETA)</b>	Will use the SmartyStreets ZipCode 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.

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.

## 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. However, if you have been installed for a while, you might want to look to make sure you have these remote site settings set up.

In Salesforce, please navigate to Setup → Administration Setup → Security Controls → 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 SmartyStreets Autocomplete add:  
<https://us-autocomplete.api.smartystreets.com>
- 3) For SmartyStreets status in the Setup page add:  
<http://status.smartystreets.com>
- 4) For (Optional) Google Verifications add – do not do this unless we have turned on Google verifications for you:  
<https://maps.googleapis.com>
- 5) For (Optional) UPS Verifications add – do not do this unless we have turned on UPS verifications for you:  
<https://onlinetools.ups.com>

## Signing Up for BETA

In order to use new features which we release for early testing from time to time, please send an email to [support@toafinish.com](mailto:support@toafinish.com) with your Org ID, or fill out a request using the Contact Us form included in the app.

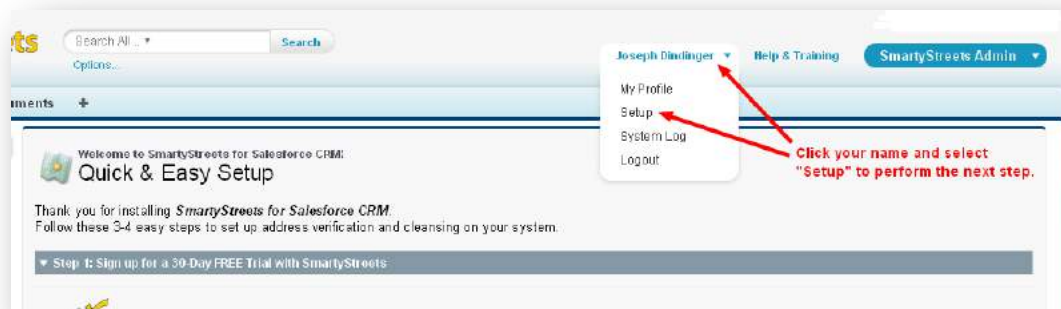
Once you are approved for BETA, we will contact you with information whenever a new feature is coming out and needs to be tested.

## Configuring Standard Objects

*This chapter will teach you how to configure the app to be used with standard objects like Accounts, Contacts and Leads.*

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. 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 CRM.



We will display screenshots for the steps for the Account as an example. The Contact and Lead steps will be similar.

## 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.

### Turning on Field History

The screenshot shows the Salesforce 'Account Fields' page. The left sidebar contains navigation links for 'Force.com Home', 'Personal Setup', and 'App Setup'. Under 'App Setup', 'Customize' is expanded, and 'Accounts' is selected, with 'Fields' highlighted. The main content area shows a table of 'Account Standard Fields'. A 'Set History Tracking' button is located above the table. Red arrows point from numbered instructions to specific elements: 1. 'Expand "/>

Home SmartyStreets Documents +

Expand All | Collapse All

Quick Find

Force.com Home

Personal Setup

- My Personal Information
- Email
- Import
- Desktop Integration
- My Chatter Settings

App Setup

- Customize
  - Tab Names and Labels
  - Home
  - Activities
  - Campaigns
  - Leads
  - Accounts
    - Fields
    - Related Lookup Filters
    - Validation Rules
    - Triggers
    - Partner Roles
    - Contact Roles

## Account Fields

This page allows you to specify the fields that can appear on the Account page. You can create up to 500 Account custom fields.

Note that deleting a custom field will delete any filters that use the custom field. It may also change the result of Assignment or Escalation Rules that rely on the custom field data.

Set History Tracking

### Account Standard Fields

Account Standard Fields Help ?

Action	Field Label	Field Name	Data Type	Controlling Field
	<u>Account Name</u>	Name	Name	
Edit	<u>Account Number</u>	AccountNumber	Text(40)	
Edit	<u>Account Owner</u>	Owner	Lookup(User)	
Edit	<u>Account Site</u>	Site	Text(80)	
Edit	<u>Annual Revenue</u>	AnnualRevenue	Currency(18, 0)	
	<u>Billing Address</u>	BillingAddress	Address	
	<u>Created By</u>	CreatedBy	Lookup(User)	
Edit	<u>Description</u>	Description	Long Text Area(32000)	
Edit	<u>Employees</u>	NumberOfEmployees	Number(8, 0)	
Edit	<u>Fax</u>	Fax	Fax	
Replace   Edit	<u>Industry</u>	Industry	Picklist	
	<u>Last Modified By</u>	LastModifiedBy	Lookup(User)	
Replace   Edit	<u>Ownership</u>	Ownership	Picklist	
Edit	<u>Parent Account</u>	Parent	Lookup(Account)	
Edit	<u>Phone</u>	Phone	Phone	
	<u>Rating</u>	Rating	Picklist	

1. Expand "Customize"
2. Expand "Accounts"
3. Click on "Fields"
4. Press the "Set History Tracking" button

## Account Field History

[Help for this Page](#) ?☒ Enable Account History

This page allows you to select the fields you want to track on the Account History related list. Whenever a user modifies any of the fields selected below, the old and new field values are added to the History related list as well as the date, time, nature of the change, and user making the change. Note that multi-select picklist and large text field values are tracked as edited; their old and new field values are not recorded.

Save Cancel

[Deselect all fields](#)

## Track old and new values

Account Name	<input type="checkbox"/>	Account Number	<input type="checkbox"/>
Account Owner	<input type="checkbox"/>	Account Site	<input type="checkbox"/>
Annual Revenue	<input type="checkbox"/>	Billing Address	<input checked="" type="checkbox"/>
Billing Barcode	<input type="checkbox"/>	Billing County	<input type="checkbox"/>
Billing County FIPS Code	<input type="checkbox"/>	Billing Last Verified	<input type="checkbox"/>
Billing Return Code	<input type="checkbox"/>	Billing Unit Number	<input type="checkbox"/>
Billing Urbanization	<input type="checkbox"/>	Billing Verified	<input type="checkbox"/>
Employees	<input type="checkbox"/>	Fax	<input type="checkbox"/>
Industry	<input type="checkbox"/>	Ownership	<input type="checkbox"/>
Parent Account	<input type="checkbox"/>	Phone	<input type="checkbox"/>
Rating	<input type="checkbox"/>	SIC Code	<input type="checkbox"/>
Shipping Address	<input checked="" type="checkbox"/>	Shipping Barcode	<input type="checkbox"/>
Shipping County	<input type="checkbox"/>	Shipping County FIPS Code	<input type="checkbox"/>

1. Press the "Enable Account History" checkbox
2. Select "Billing Address" and "Shipping Address"
3. Press the "Save" button

## Modifying Layouts

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

**Account Page Layout**

This page allows you to create different page layouts to display Account data. After creating page layouts, click the Page Layout Assignment button to control which page layout users see by default.

**Account Page Layouts**

Action	Page Layout Name	Created By	Modified By
<a href="#">Edit</a> <a href="#">Del</a>	Account Layout	Joseph Dindinger, 11/12/2011 6:21 PM	Joseph Dindinger, 11/12/2011 6:21 PM

**Steps:**

1. Expand "Customize"
2. Expand "Accounts"
3. Click on "Page Layouts"
4. Click "Edit" next to each layout (this screenshot only shows one, but typically there will be multiple layouts)

**Navigation Path:** Force.com Home > App Setup > Customize > Accounts > Page Layouts

**Account Layout**

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

**Fields**

Section	Field Name
Blank Space	Billing Address 5...
Account Name	Billing Barcode
Account Number	Billing County
Billing Address	Billing County Fl...
Account Owner	Billing Last Veri...
Account Site	Billing Return Code
Annual Revenue	Billing Unit Number
Billing Address	Billing Urbanization
Account Owner	Billing Verified
Account Name	Created By
Parent Account	Description
	Employees
	Annual Revenue

**Account Sample**

**Account Detail**

Standard Buttons: Edit Delete Sharing Include Offline Enable As Partner Custom Buttons

**Account Information** (Header visible on edit only)

Account Owner	Sample User	Phone	1-415-555-1212
Account Name	Sample Account Name	Fax	1-415-555-1212
Parent Account	Sample Account	Website	www.salesforce.com

**Additional Information**

Type	Sample Type	Employees	96,286
Industry	Sample Industry	Annual Revenue	\$123.45

**Description Information** (Header not visible)

Description	Sample Description
-------------	--------------------

**Address Information** (Header visible on detail only)

Billing Address	Suite 300, The Landmark @ One Market San Francisco, CA 94105 US	Shipping Address	Suite 300, The Landmark @ San Francisco, CA 94105 US
-----------------	---	------------------	--

**Billing Address Status**

Sample Billing Address Status

**System Information**

**Note:** You can move the "Status" field for Billing and Shipping addresses onto the layout, or any other of the address fields that might be useful. (This step is optional.)

## ADDRESS VERIFICATION FOR SALESFORCE.COM

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

Force.com Home

Personal Setup

- My Personal Information
- Email
- Import
- Desktop Integration
- My Chatter Settings

App Setup

- Customize
- Tab Names and Labels
- Home

Account Layout

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Fields

- Buttons
- Custom Links
- Related Lists

Quick Find Button Name

Delete Enable As Partner Submit for Approval

Disable Customer ... Get Contacts Verify Addresses

Disable Partner A... Include Offline

Edit Sharing

In the "Buttons" section, drag the "Verify Addresses" button to the Custom Buttons area.

Account Sample

Account Detail

Edit Delete Sharing Include Offline Enable As Partner Verify Addresses

Account Information (Header visible on edit only)

Phone: 1-415-555-4040

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

Installed Packages

- AppExchange Marketplace
- Critical Updates

Administration Setup

- Manage Users
- Company Profile
- Security Controls
- Communication Templates
- Translation Workbench
- Data Management
- Monitoring
- Mobile Administration
- Desktop Administration
- Email Administration
- Google Apps

Fields

- Buttons
- Custom Links
- Related Lists

Quick Find Related List Name

Account History Cases Contracts Opportunities

Activity History Contact Roles Notes & Attachments Partners

Approval History Contacts Offices

Assets Content Deliveries Open Activities

This list is not customizable

Offices

Office Name	Phone	City	State	Owner Alias
Sample Office Name	1-415-555-1212	Sample City	Sample State	Sample Owner Alias

Account History

This list is not customizable

Copyright © 2000-2011 salesforce.com, inc. All rights reserved. | [Privacy Statement](#) | [Security Statement](#) | [Terms of Use](#) | [508 Compliance](#)

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

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

### 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 4 for instructions on setting this up.

## 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.

### **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. Return values are: Confirmed, Not Confirmed, Confirmed - By Dropping Secondary Info, Confirmed - Missing Secondary Info, or Not Submitted



## Creating Optional Fields

Several other fields are not required, but recommended to be created for each address to be verified.

Sample Field Name	Field Type	Description
Street 2	Text(128)	Sometimes used to include 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.
Address Status	Formula (Text)	Use the following formula (substituting XXX with the API name of the "address return code" field name): <pre>If ( XXX = "Confirmed",   IMAGE("/img/msg_icons/confirm16.png", "Confirmed" ,16,16) &amp; " Confirmed",   If ( XXX = "Confirmed - By Dropping Secondary Info",     IMAGE("/img/msg_icons/warning16.png", "Confirmed - By Dropping Secondary Info" ,16,16) &amp; " Confirmed - By Dropping Secondary Info",     If ( XXX = "Confirmed - Missing Secondary Info",       IMAGE("/img/msg_icons/warning16.png", "Confirmed - Missing Secondary Info" ,16,16) &amp; " Confirmed - Missing Secondary Info",       If ( XXX = "Not Submitted",         IMAGE("/img/msg_icons/error16.png", "Not Submitted" ,16,16) &amp; " Not Submitted",         If ( XXX = "Not Confirmed",           IMAGE("/img/msg_icons/error16.png", "Not Confirmed" ,16,16) &amp; " Not Confirmed",           IMAGE("/img/msg_icons/inf16.png", "Note Yet Verified" ,16,16) &amp; " Not Yet Verified" ) ) ) ) )</pre>
Address Barcode	Text(32)	Shows the barcode returned on verification
Address Barcode Image	Formula (Text)	Use the following formula (substituting XXX with the API name of the "address barcode" field name): <pre>if ( not ( isblank ( XXX ) ),   IMAGE("https://www.barcodesinc.com/generator/image.php? code=" + XXX + "&amp;style=196&amp;type=C128B&amp;width=206&amp;height=50&amp;xres=1&amp;font= 3", "Address Barcode" ), "" )</pre>
Address County	Text(64)	Shows the county name returned on verification
Address County FIPs Code	Text(32)	Shows the FIPs code for the county returned on verification
Address Latitude	Number (13, 5)	Used for geocoding the Latitude of the address
Address Longitude	Number (13, 5)	Used for geocoding the longitude of the address.
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
Active?	Checkbox	Shows if the Post Office thinks this is an active address
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.

## ADDRESS VERIFICATION FOR SALESFORCE.COM

Sample Field Name	Field Type	Description
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.
Footnotes	Long Text Area	Notes on the details of the verification
RDI	Text(32)	Shows the Residential Delivery Indicator from the Post Office.
Record Type	Text(1)	Shows the returned accuracy of the verification.
Record Type Description	Formula (Text)	<code>If(Record_Type__c = "F", "Firm",  If(Record_Type__c = "G", "General Delivery",  If(Record_Type__c = "H", "High-Rise",  If(Record_Type__c = "P", "P.O. Box",  If(Record_Type__c = "R", "Rural Route",  If(Record_Type__c = "S", "Street", "" ) ) ) ) )</code>
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: SmartyStreets, Google Maps, UPS
UTC Offset	Number (16,2)	Used to indicate how many hours different from GMT/UTC time
Current Time	Formula (Text)	For the code for this field, please see code on the next page. Note that you will need to set up the UTC Offset and the Daylight Savings? fields as described above.
<b>Non-SmartyStreets Data</b>  These fields are populated by sources outside of SmartyStreets and are not guaranteed to be 100% accurate. If these fields are critical for you, please contact us at <a href="mailto:support@toafinish.com">support@toafinish.com</a> to discuss sources and how often they are updated.		
Island	Text (200)	This is for the name of the Island on which an address is located. Only the Hawaiian and both sets of Spanish islands are supported.
Area Codes	Text (32)	Phone area codes supported in this zip code
Urban Area Code	Text (10)	Code for the primary Urban Area for the zip code, sourced from the 2010 Census.
Urban Area Name	Text (255)	Primary Urban Area Name/Description for the Urban Area Code for the zip code. Must be used together with the Urban Area Code, as it is populated from the code.

## ADDRESS VERIFICATION FOR SALESFORCE.COM

**Current Time Formula** 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"
  )
)
```

## 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 Precision	Text (255)	How precise is the address? From best to worst: 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)
Geocode Precision	Text (255)	How precise is the geocoding of the address? From best to worst: DeliveryPoint — rooftop level Premise — property or building level Thoroughfare — street level Locality — city or town level AdministrativeArea — state or province level None — unknown address
Organization	Text (255)	The name of the recipient, firm, or company at this address
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

## 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 “Manage Addresses” tab and then click the “New Address” button.

WELCOME **MANAGE ADDRESSES** AUTOMATION BETA

SET UP DEFAULT FIELD MAPPING

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

[Create Default Records](#)

Address List

Action	Name	Object	Active?	Street	City	State	Zip
<a href="#">Edit</a>   <a href="#">Del</a>	Account Billing	Account	✓	BillingStreet	BillingCity	BillingState	BillingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Account Shipping	Account	✓	ShippingStreet	ShippingCity	ShippingState	ShippingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Contact Mailing	Contact	✓	MailingStreet	MailingCity	MailingState	MailingPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Contact Other	Contact	✓	OtherStreet	OtherCity	OtherState	OtherPostalCode
<a href="#">Edit</a>   <a href="#">Del</a>	Lead	Lead	✓	Street	City	State	PostalCode

[+ New Address](#) [+ Add Supported Object](#)

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.

Address Detail

[Save](#) [Cancel](#)

System Addresses Information

Name

Object  Active? ☒

Required Fields

Street	<input type="text" value="Street"/>	Verified	<input type="text" value="Address Verified"/>
City	<input type="text" value="City"/>	Last Verified	<input type="text" value="Address Last Verified"/>
State	<input type="text" value="State"/>	Return Code	<input type="text" value="Address Return Code"/>
Zip Code	<input type="text" value="Zip"/>		

Optional Fields

Street 2	<input type="text" value="Street 2"/>	BarCode	<input type="text" value="--None--"/>
Unit	<input type="text" value="--None--"/>	Urbanization	<input type="text" value="--None--"/>
Country	<input type="text" value="--None--"/>	County FIPS Code	<input type="text" value="--None--"/>

*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 on the prior page.*

## Lightning Setup

### Creating the Custom "Verify" Button

Please note that the following section will be very technical and perhaps difficult to follow if you do not yet have experience in Salesforce development. We provide code samples, but they must be tweaked and modified to fit your own custom object.

All of the JavaScript code referenced in the pages below can be found in the "lightning" folder in our GitHub repository:

GitHub.com Address Verification Sample Code Repository

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

1) Go to Setup -> Develop -> Apex Classes and press the "New" button.

\* Note: this must be done in a sandbox or dev org

2) In a new tab, go to our GitHub repository, in the Lightning folder, and open the "custom\_apex\_controller" entry. Paste the code (not the line numbers) into the box in Salesforce. You will need to replace all of the references to the custom **Office\_\_c** object with the object and field names of your custom object. Once this is done, Save. If it saves, then you are most likely safe to continue.



3) Now open the Developer Console

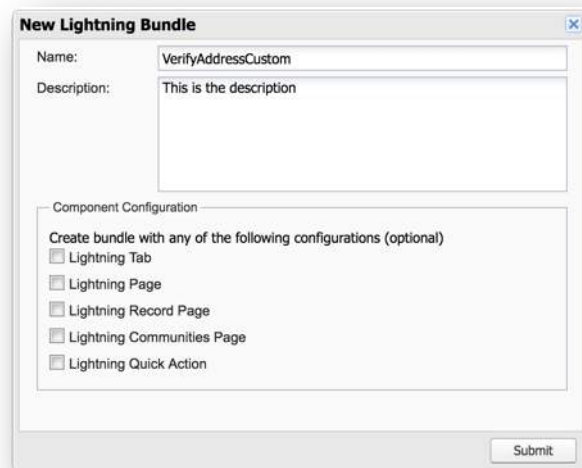
4) Click File -> New -> Lightning Component

5) Give the component a name and optional description. Click Submit.

6) Replace the default text in the new component with the code in the "custom\_lightning\_component" entry in our GitHub repository.

7) Now click on the **Controller** item to create a new controller for this bundle.

8) Replace the default text in the new controller with the code in the "custom\_lightning\_controller" entry in the Lightning folder in our GitHub repository. Again, updating the areas referencing the sample Office\_\_c custom object with your own custom object and fields.



## ADDRESS VERIFICATION FOR SALESFORCE.COM

*\*Note: To ensure that you have located all areas in the code that need attention, do a search for "\_\_c", which indicates a custom object or field. Replace the name of the field in the code with the name in your custom object.*

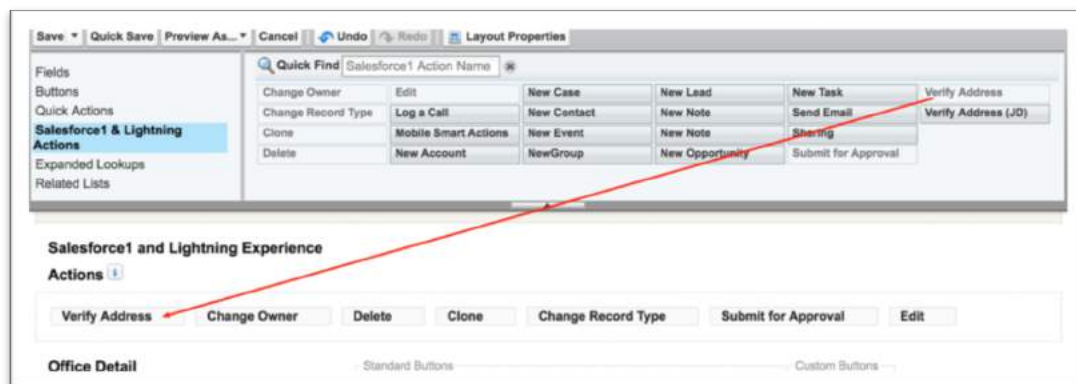
9) Go back to Setup -> Create -> Objects and click on the name of your custom object.

10) Scroll down to the "Buttons, Links, and Actions" section and click on the "New Action" button.

11) Select "Lightning Component" as the Action Type, and then select the component that you just created. Give it a name like "Verify Address" and save.

12) 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 "Salesforce Mobile and Lightning Experience Actions" menu at the top and find the Action you just created. Drag it down into the "Salesforce 1 and Lightning Actions" section.

*\*Note: if the "Salesforce Mobile and Lightning Experience Actions" section is closed, you might need to open it for updates.*



13) Now switch to Lightning and you can test the button. You can also test it on Salesforce Mobile.

## Classic Setup

### Creating the Custom “Verify” Button

A custom button must be created on the object to launch the address verification, and then added to the layout.

**App Setup**

- Customize
- Create
  - Apps
  - Custom Labels
  - Objects**
  - Packages
  - Report Types
  - Tabs
  - Workflow & Approvals
- Develop
- Deploy
  - Schema Builder *New!*
  - Installed Packages
  - AppExchange Marketplace
  - Critical Updates

**Administration Setup**

- Manage Users
- Company Profile
- Security Controls
- Communication Templates
- Translation Workbench
- Data Management
- Monitoring
- Mobile Administration
- Desktop Administration
- Email Administration
- Google Apps

**Standard Fields**

Action	Field Label	Field Name	Data Type	Controlling Field	Track History
	<u>Created By</u>	CreatedBy	Lookup(User)		<input type="checkbox"/>
	<u>Last Modified By</u>	LastModifiedBy	Lookup(User)		<input type="checkbox"/>
Edit	<u>Office Name</u>	Name	Text(80)		<input type="checkbox"/>
Edit	<u>Owner</u>	Owner	Lookup(User,Queue)		<input type="checkbox"/>

**Custom Fields & Relationships**

New Field Dependencies Set History Tracking

Action	Field Label	API Name	Data Type	Controlling Field	Modified By	Track History
Edit   Del	<u>Address Last Verified</u>	Address_Last_Verified__c	Date/Time		Joseph Dindinger, 11/14/2011 10:55 PM	<input type="checkbox"/>
Edit   Del	<u>Address Verified</u>	Address_Verified__c	Checkbox		Joseph Dindinger, 11/14/2011 10:56 PM	<input type="checkbox"/>
Edit   Del	<u>City</u>	City__c	Text(40)		Joseph Dindinger, 11/12/2011 6:31 PM	<input type="checkbox"/>
Edit   Del	<u>Country</u>	Country__c	Text(64)		Joseph Dindinger, 11/14/2011 10:53 PM	<input type="checkbox"/>
Edit   Del	<u>State</u>	State__c	Text(10)		Joseph Dindinger, 11/12/2011 6:31 PM	<input type="checkbox"/>
Edit   Del	<u>Street</u>	Street__c	Text Area(255)		Joseph Dindinger, 11/12/2011 6:30 PM	<input type="checkbox"/>
Edit   Del	<u>Street 2</u>	Street_2__c	Text(128)		Joseph Dindinger, 11/12/2011 6:30 PM	<input type="checkbox"/>
Edit   Del	<u>Zip</u>	Zip__c	Text(10)		Joseph Dindinger, 11/12/2011 6:32 PM	<input type="checkbox"/>

**Standard Buttons and Links**

Action	Label	Name	Overridden	Display	Modified By
Edit	Offices Tab	Tab	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	List	List	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	View	View	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	Edit	Edit	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	Delete	Delete	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	Clone	Clone	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	New	New	<input type="checkbox"/>	Standard Salesforce.com Page	
Edit	Accept	Accept	<input type="checkbox"/>	Standard Salesforce.com Page	

**Custom Buttons and Links**

New

No custom buttons or links defined

Click the "New" button under "Custom Buttons and Links" for the custom object you are setting up.

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

```
<apex:page standardController="Object_Name__c" sidebar="false">
  <smartystreets:VerifyAddresses />
</apex:page>
```

Replace the highlighted “Object\_Name\_\_c” with the API name of the custom or standard object.



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 “Detail Page Button” and have the behavior “Display in Existing Window without sidebar or header” and 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 layout, and you are good to go!



above.

A word about JavaScript buttons. We still provide the code for JavaScript buttons in our repository. These can be used in Classic Salesforce, but are no longer recommended by Salesforce. If you are currently using Classic, it is OK to stay with them as they provide a somewhat faster experience, but our recommendation is to move to either a Lightning Component or a Visualforce page as described

## Setup for All Scenarios

### 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.

We have two options: Trigger (must be coded in a Sandbox first) or Workflow rule.

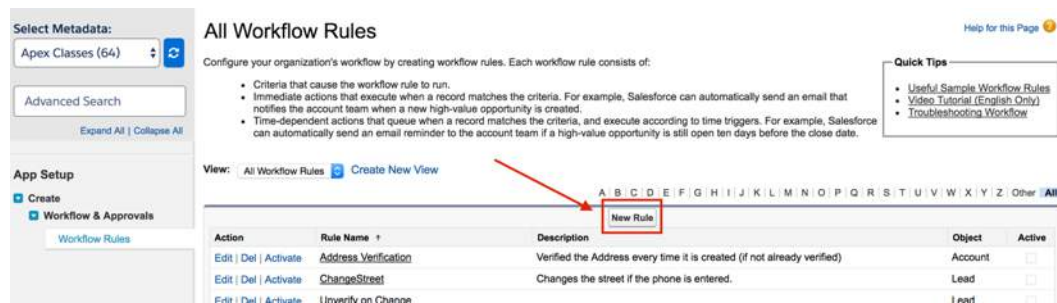
#### **Option 1: Flagging Addresses Un- Verified via Workflow Rule**

An easy way to mark a record as needing verification is via a workflow rule. This option might be preferable for the learned point-and-click administrator, or someone who is good at formulas and simply prefers configuration via clicks vs. code.

*\*Note: You could also use the newer Process Builder for this task, but we are not currently recommending it as the errors when these processes don't work are harder to manage. However, it appears that Processes have more of a future than Workflow Rules, so if you are already using them, please add this to your feature list.*

- 1) To build it with Workflow Rules, go to Setup → Create → Workflow & Approvals → Workflow Rules and click on the “New Rule” button to get started.

## ADDRESS VERIFICATION FOR SALESFORCE.COM



- 2) On the next screen select the **Object** where your custom address is, and then press the Next button.
- 3) Name the rule and change the **Evaluation Criteria** to “created, and every time it’s edited.” Then change the **Rule Criteria** to “formula evaluates to true” and now you need to create a formula using the **isChanged()** function to determine if any of the address fields were changed. See the example below.

The screenshot shows the 'Step 2: Configure Workflow Rule' page. The page has a header 'Step 2 of 3' and buttons for 'Previous', 'Save & Next', and 'Cancel'. Below the header is a section 'Edit Rule' with fields for 'Object' (Office), 'Rule Name' (Unverify), and 'Description' (This will unverify an office after changing the address). Below this is a section 'Evaluation Criteria' with radio buttons for 'created', 'created, and every time it's edited' (selected), and 'created, and any time it's edited to subsequently meet criteria'. Below this is a section 'Rule Criteria' with a dropdown for 'Run this rule if the following' (formula evaluates to true). Below this is a section 'Example' with a formula: `or ( isChanged ( Street__c ), isChanged ( City__c ), isChanged ( State__c ), isChanged ( Zip__c ) )`. Below the formula is a section 'Functions' with a dropdown for 'All Function Categories' and a list of functions: ABS, AND, BEGINS, BLANKVALUE, BR, CASE.

- 4) After making sure that your formula is correct, press the **Save & Next** button. Then, from the **Add Workflow Options** dropdown, select **Field Update**.

- 5) Enter a Name and Unique Name for the field update, and then select the **Field to Update**. This must be the checkbox field indicating if an address is verified, that is configured in the Setup page for this address. Next, select “False” in the **Checkbox Options** section, and Save.
- 6) On the next page click **Done** and then back at the main rule page, click the **Activate** button.

You are now ready to test this workflow rule by going to a custom record that is already verified and changing one of the address fields. When you save, the “Verified” flag field should be set to False.

#### **Option 2: Flagging Addresses Un- Verified via Trigger**

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.

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

This trigger unchecks the verified flag whenever the address is updated again. This is important to do so that the address does not say it is verified, when in fact, changes have been made to it, and the final version has never been verified.

Edit   Del	Street2	Street_2__c	Text(128)	Joseph Dindinger, 11/12/2011 6:30 PM	✓
Edit   Del	Zip	Zip__c	Text(10)	Joseph Dindinger, 11/12/2011 6:32 PM	✓

**Related Lookup Filters**

No related lookup filters defined.

**Validation Rules**

No validation rules defined.

New

Validation Rules Help ?

**Triggers**

No triggers defined

New

Triggers Help ?

Click the "New" button to create a new trigger for the custom object.

When you press the “New” button, a blank textbox will come up. Paste the Apex code that you can find in the “triggers” folder on our GitHub.com repository, linked to below. The code from a file called **resetAddressFlag** 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.

For example, in the example code you will pull from GitHub.com, you will want to make the following changes:

1. All instances of **Office\_\_c** will need to be replaced with the API Name of the custom object.
2. Four field names will possibly need to be updated, depending on the names of those fields within the custom object: **Street\_\_c**, **Zip\_\_c**, **City\_\_c**, and **State\_\_c**.



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.

- *Note: The next chapter explains how to verify in a list or automatically.*

## Turning on Field History Tracking

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

## Modifying the Layout

Modifying the layout for a custom object is exactly the same as what is done for a standard object.

## Automation and Bulk Verification

*What to do with existing records, or records that will be imported into Salesforce in bulk*

**I**n 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 addresses and in the next section, we get into the new Autocomplete feature, but here we discuss the back-up plan for all of this – what to do when a record is saved in Salesforce and is not perfectly clean.

### Manual Bulk Verification

If the amount 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.com list. We will step through how to do it for Leads, but the process is very similar for Accounts and Contacts, and even custom objects.

Go to Setup → Customize → Leads → Search Layouts and click **Edit** next to the “Leads List View.”

## ADDRESS VERIFICATION FOR SALESFORCE.COM

Home Chatter Files Campaigns Leads Accounts Contacts Opportunities Contracts Cases Solutions Products Reports Dashboards +

Leads

Select Metadata:  
Apex Classes (15)

Advanced Search

Expand All | Collapse All

App Setup

- Customize
- Leads
  - Fields
  - Related Lookup Filters
  - Validation Rules
  - Triggers
  - Page Layouts
  - Field Sets
  - Compact Layouts
  - Search Layouts**
  - Assignment Rules
  - Settings
  - Buttons, Links, and Actions
  - Lead Processes
  - Record Types
  - Limits

### Lead Search Layouts

Search layouts allow you to select the standard and custom fields that are displayed in the following search features:

- Search Results Columns for search and tagging. [View example](#)
- Lookup Dialogs that pop up when you click the magnifying glass on Lookup fields. [View example](#)
- Lookup Phone Dialogs that pop up when you click the magnifying glass on a SoftPhone dial pad. [View example](#)
- Recently viewed, modified, and created records lists displayed on tab home page. [View example](#)
- Search Results Filter Fields for search. [View example](#)

Search layouts also allow you to select the standard and custom buttons that are displayed in the following search features:

- Search Results for search and tagging.
- List Views for filtering records. [View example](#)

Search

Action	Layout	Columns Displayed	Buttons Displayed	Modified By
<a href="#">Edit</a>	Search Results	Name, Title, Phone, Company, Email, Lead Status, Owner Alias		Joseph Dindinger, 9/26/2011 6:38 PM
<a href="#">Edit</a>	Lookup Dialogs	Name, Company	N/A	Joseph Dindinger, 9/26/2011 6:38 PM
<a href="#">Edit</a>	Lookup Phone Dialogs	Name, Company, Phone, Mobile	N/A	Joseph Dindinger, 9/26/2011 6:38 PM
<a href="#">Edit</a>	Leads Tab	Name, Company, Phone	N/A	Joseph Dindinger, 9/26/2011 6:38 PM
<a href="#">Edit</a>	Leads List View	N/A	New, Accept, Change Status, Change Owner, Add to Campaign, Verify Selected Addresses	Dev Team3, 12/11/2014 4:00 AM
<a href="#">Edit</a>	Search Filter		N/A	Joseph Dindinger,

From the list of available buttons, select the one called “Verify Selected Addresses” and press the → arrow button. Next, press the **Save** button.

Custom Buttons

Available Buttons

--None--

Selected Buttons

Verify Selected Addresses

Add

Remove

Up

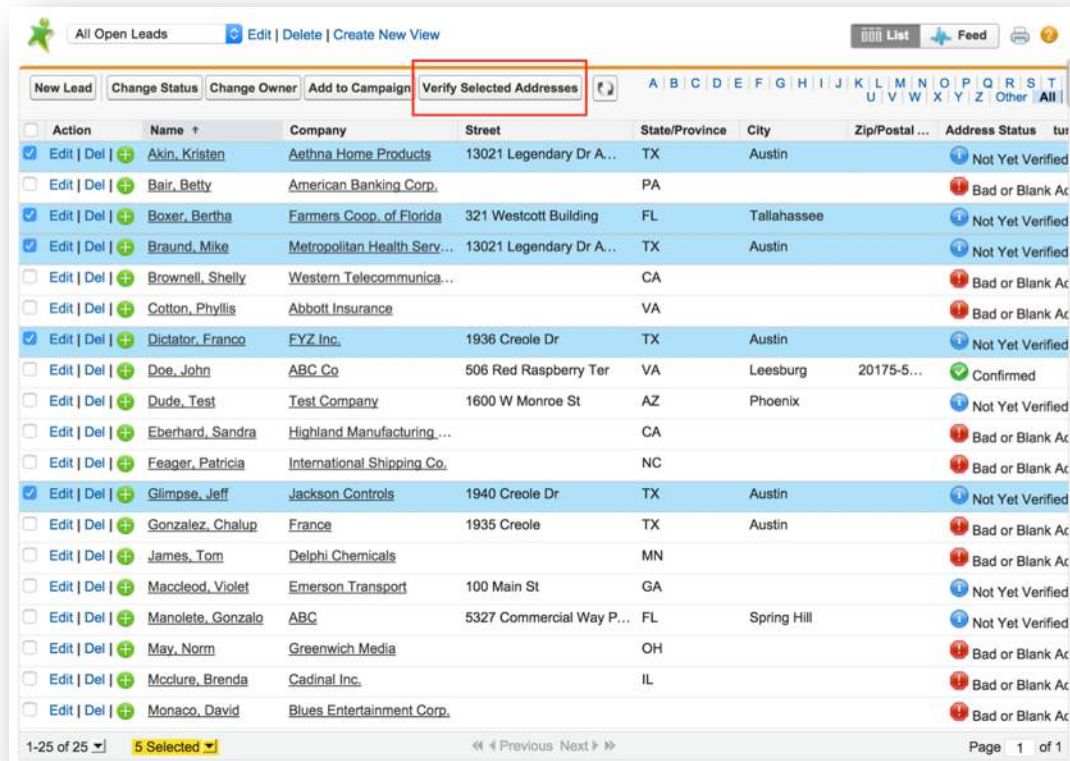
Down

Save

Cancel

The button will now be available from any Lead list view. You can select as many records as you want, up to 100 and then press the Verify Selected Addresses.

## ADDRESS VERIFICATION FOR SALESFORCE.COM



Please note several things when verifying multiple addresses at once:

- 1) You will use up a SmartyStreets.com Address Lookup for every address. So, even though you verify 100 at a time, you are still charged for 100.
- 2) If addresses are very long, sometimes Salesforce.com (not SmartyStreets) will choke on multiple verifications at once, and will show you an error. If that happens, please try selecting less addresses and trying again.

## List Verification for Custom Objects

A button can also be added for multiple-record address validation for custom objects. Please follow these steps and you should be able to add the button fairly easily.

First, create a simple Visualforce page by going to Setup → App Setup → Develop → 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!



above.

A word about JavaScript buttons. We still provide the code for JavaScript buttons in our repository. These can be used in Classic Salesforce, but are no longer recommended by Salesforce. If you are currently using Classic, it is OK to stay with them as they provide a somewhat faster experience, but our recommendation is to move to either a Lightning Component or a Visualforce page as described



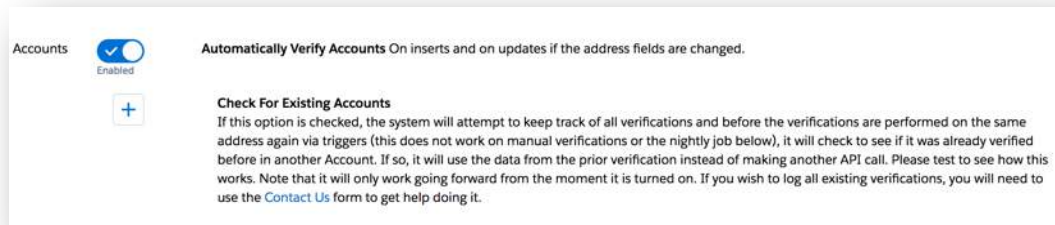
## Automated Verification Options

Depending on the amount 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.

### Standard Object Trigger Verifications

You can turn on automatic verifications via Trigger for the Lead, Account and Contact objects in Salesforce. This will automatically verify any addresses configured on these objects whenever they are imported or updated. This feature is easy to turn on:

- 1) Go to Setup → Installed Packages → and click on the Address Verification app. Once there, you can click the “Configure” button.
- 2) In the Setup page, click the AUTOMATION tab, then select one or more of the “Automatically Verify” fields and press the “Save Settings” button. You can select Accounts, Contacts or Leads.



This will now turn on the automated verification for these objects.

- ❖ 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 SmartyStreets, 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. If this timing is a showstopper, please see the section on the advanced Autocomplete feature, where we overcome this issue, but add a little complexity to the process.

### Address Pre-Cleaning Option

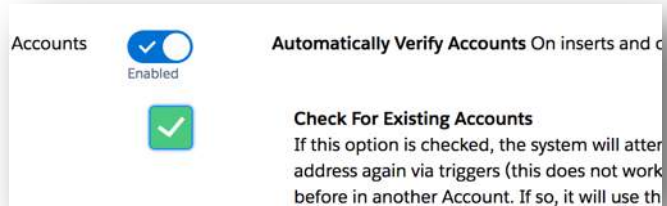
We are introducing a new feature which allows you to immediately verify addresses entered if they already exist in the system without having to make an API call and re-verify the same address again. This feature is very helpful if duplicate values are often entered into your org. For example, if multiple Leads with the same address are expected to be entered, you can often avoid re-validating the same address multiple times.

- ❖ Note that currently we have a few limitations for this new feature:
    - *This only works with triggers. Meaning when the verifications are automatic. If you click the “Verify Addresses” buttons or schedule a job to clean up all addresses, this feature is not used.*
    - *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.*
    - *Only certain “critical” fields are copied to the new object, it is not completely populated with data the way a regular verification would.*
    - *For organizations with a large amount of records, usually above 200,000 Accounts, Contacts or Leads, this may fail to work depending on several factors. If this is your case, please contact [support@toafinish.com](mailto: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 **MANAGE ADDRESSES** 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).

The screenshot shows the Salesforce Setup page for Address Verification. The 'Optional Fields' section is expanded, displaying a grid of fields for configuration. The 'Clean Street' field is highlighted with a red box, indicating it is the field to be configured for the address verification feature.

Field Name	Field Type
Street 2	--None--
Unit	Address Unit Number
County	Address County
Country	Country
Latitude	Address Latitude
Longitude	Address Longitude
Record Type	Address Record Type
Carrier Route	Address Carrier Route
Congressional District	Address Congressional District
Is Active	Address Active?
Footnotes	Address Footnotes
BarCode	Address Barcode
Urbanization	Address Urbanization
County FIPs Code	Address County FIPs Code
RDI	Address RDI
Do Not Verify	Address Do Not Verify
Time Zone	Address Time Zone
DST	Address Daylight Savings?
Is Vacant	Address Vacant?
Verification Used	Verification Used
Clean Street	Clean Street

- 2) Next, switch to the **AUTOMATION** tab in the Setup page, and check the appropriate “Check for Existing...” field. Press the **Save Settings** button to turn the feature on.



- 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](mailto:support@toafinish.com) email address. We will be happy to talk this through with you and help you to get your data cleaned. In a future release, we expect to make it easy to do, but for now, it is an administrative function that isn't easy.

For custom objects or custom addresses, see the next two sections on how to set up your own trigger or process to do this same function yourself.

## Custom Object Trigger Verifications

If you have a custom address outside of the Lead, Account and Contact objects and want automated address verification, you will need to either create a custom trigger, which is explained here, or else create a custom process, explained in the following section.

- ❖ Please note that triggers might be currently more reliable in the long run than processes, especially if bulk imports are intended for this object. It will be potentially more complicated to set up a trigger, but it might be worth it for the reliability achieved.

For triggers, first we must set up a couple helper classes. Note that this must be done in a sandbox, and then deployed into production. A production instance of Salesforce will not let you create classes or triggers directly, rather only through the sandbox process.

- 1) Go to Setup → Develop → Apex Classes and click the **New** button.
- 2) When you press the New button, a blank textbox will come up. Paste the Apex code that you can find in the “class” folder on our GitHub.com repository, linked to below. The code from a file called **ScheduleBatchOfficeAddressVerification**, but before saving, you must modify the code a little in order to customize it for your particular custom object:



### GitHub.com Address Verification Sample Code Repository

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

- a. You can change the name of the Class.
  - b. Replace “Office\_\_c” on line 7 with the API name of your object.
  - c. Save.
- 3) Once complete, go to Setup → Create → Objects and click on your custom object. Once there, scroll down to the “Triggers” area and press the **New** button.
  - 4) In the text area provided, fill in the code found in the “trigger” folder in our GitHub.com repository, see above. The file is called autoVerifyOfficeAddress and again, you will need to make several changes to the code before it will work:
    - a. You can change the name of the trigger
    - b. All instances of “Office\_\_c” will need to be changed to match the API name of your selected object.
    - c. All field names may need to be changed, depending on what your fields are called: Street\_\_c, City\_\_c, State\_\_c, Zip\_\_c, AddressVerified\_\_c

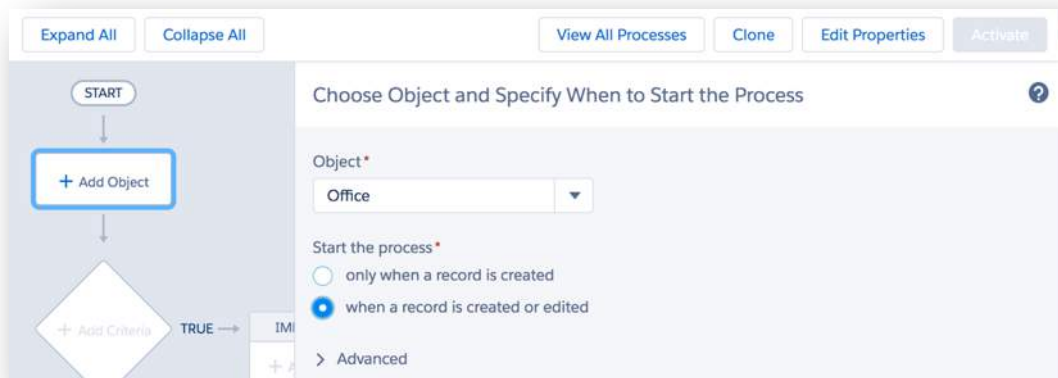
and DoNotVerify\_\_c are all found in the code and might need to be changed to the correct name.

- d. If you changed the name of the Apex Class in step 3 above, you will need to change it here too.
- 5) When finished, press the **Save** button, and the trigger is now ready to run every time a record is updated.

## Custom Object Process Verifications

If you have a custom address outside of the Lead, Account and Contact objects and want automated address verification, you will need to either create a custom trigger, which is explained in the section prior, or an easier solution is to build out a process to handle the verification.

- 1) Go to Setup → Create → Workflow & Approvals → Process Builder and press the **New** button.
- 2) Call the process what you will and then select “a record changes” under the **The Process Starts When** dropdown, and press the **Save** button.
- 3) Now click on the “Add Object” area and select your custom object under the **Object** field, and then select “when a record is created or edited” under the **Start the process** selection. Click **Save**.



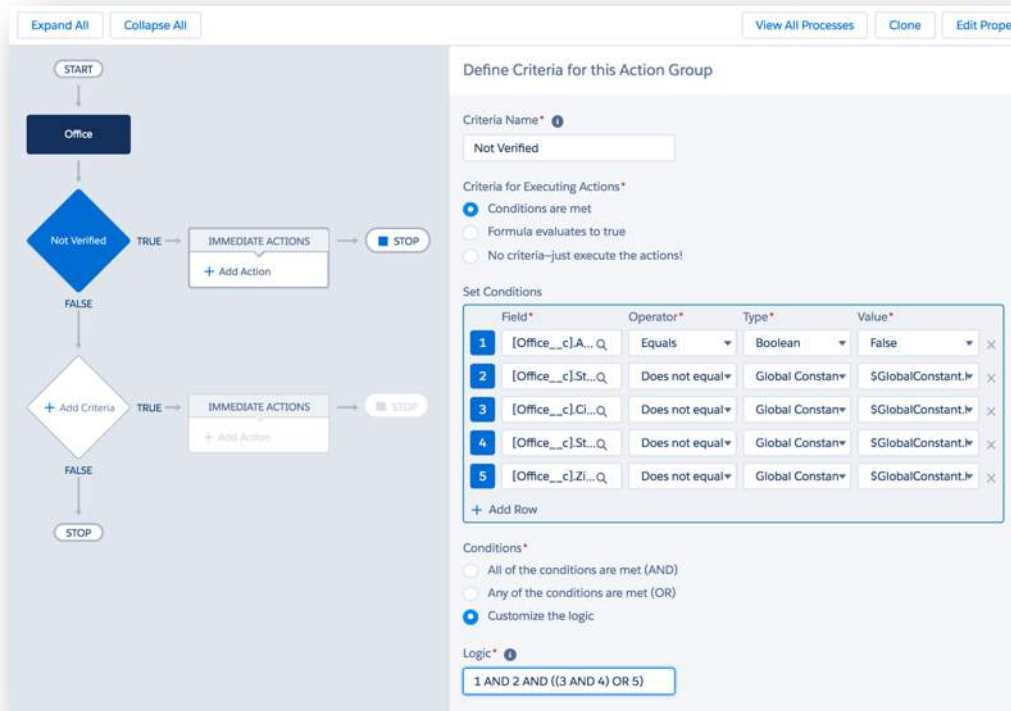
- 4) Now click the **Add Criteria** section on the left.
- 5) In the Add Criteria popup, you will need to fill out a few things. First, name it something like “Not Verified”. Next, you will need to add 5 conditions as shown below. Note that the field names will be different, depending on what you have called your fields:

## ADDRESS VERIFICATION FOR SALESFORCE.COM

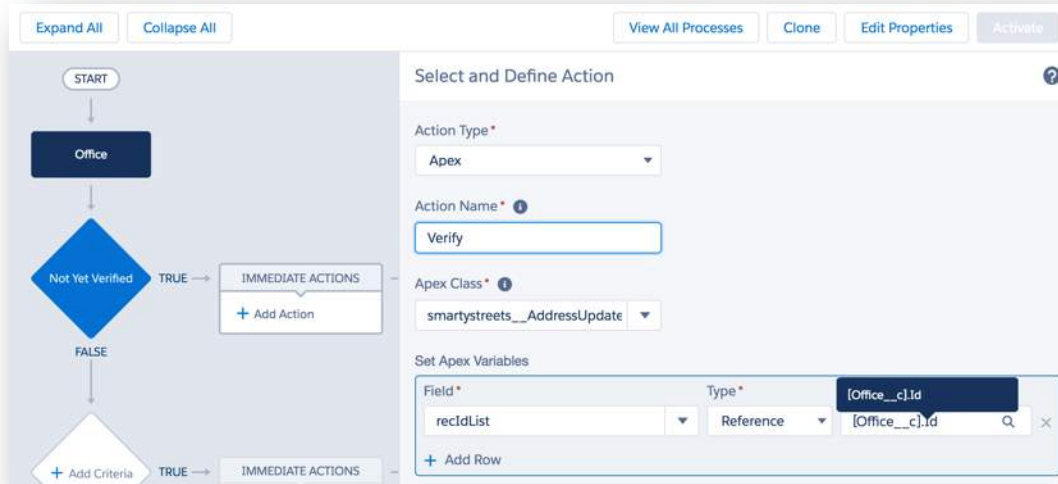
Field	Operator	Type	Value
AddressVerified__c	Equals	Boolean	False
Street__c	Does not equal	Global Constant	\$GlobalConstant.null
City__c	Does not equal	Global Constant	\$GlobalConstant.null
State__c	Does not equal	Global Constant	\$GlobalConstant.null
Zip__c	Does not equal	Global Constant	\$GlobalConstant.null

- 6) Under the **Conditions** section, select “Customize the logic” and then enter the following, and then press the **Save** button:

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



- 7) Now click the **Add Action** link under the “IMMEDIATE ACTIONS” area. In the popup, select “Apex” as the **Action Type**, then give it a name, and select “smartystreets\_\_AddressUpdateService” under the **Apex Class**.
- 8) Now, in the “Apex Variables” area, click the **Add Row** link. In the new row, select “recIdList” for the **Field**, “Reference” for the **Type**, and drill into the **Value** field and select the **Record Id** for your custom object.



- 9) Now press the **Activate** button at the top of the page and you are ready to start testing. This process should validate the record in your custom object whenever you create or update a record that hasn't been already verified.
- ❖ Please note that this same technique can be used to verify the standard objects (Leads, Contacts, Accounts) and could potentially also be used with different filters to meet specific business requirements – for example, if you want to verify an address only under certain conditions. Use the above steps as a guideline for how to set it up, but you are free to change it up to fit your process.

## Daily Scheduled Verification Job

If you do not want to have addresses verified via triggers or processes as shown in the sections above, you can also schedule a daily job that will verify all addresses that need it.

- 1) Go to Setup → Installed Packages → and click on the **Address Verification** app. Once inside, click on the **Configure** button to take you to the setup page.
- 2) Click on the **AUTOMATION** tab and scroll down to the “Daily Scheduled Job” section. Here is where you can turn on this job. Most of it is self-explanatory, but a couple notes:
  - a. 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.
  - b. You are able to also have an email sent to the current user when the job completes, IF it validated any addresses.
- 3) Press the **Save Settings** button to schedule the job. You can always press the **Delete Schedule Job** if you want it to stop. You are done!
- 4) If you prefer to run the job one time, you can press the **Run This Job Now** button.

The screenshot shows the 'DAILY SCHEDULED JOB' configuration interface. At the top, there's a header 'DAILY SCHEDULED JOB'. Below it, the 'job start time' is set to '18:00'. The 'email when job complete' checkbox is checked. A note explains that the 'expire validations > X days' setting allows re-verifying addresses previously verified over X days in the past, with 0 meaning never. Below this, there's a section for SOQL filters with examples: 'limit 100', 'where name like 'abc co%'' (with a note that 'where' is added), 'where state = 'TX' limit 10', and 'where firstname = 'Frank' order by lastname limit 5'. A link provides more information on SOQL queries. At the bottom, there are three sections for adding filters: 'verify accounts' with a '+' button and a 'Test' button; 'verify contacts' with a '+' button and a 'Test' button; and 'verify leads' with a '+' button and a 'Test' button. At the very bottom, there are three buttons: 'Delete Scheduled Job' (with a red 'x' icon), 'Save and Schedule Job' (with a green checkmark icon), and 'Run This Job Now' (with a blue play icon).



## Turning Off All Triggers

You can turn off the Account, Contact and Lead triggers installed with our app. 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 following automation in the process builder, which is able to do both of these functions:

- 1) Monitor address fields and whenever one of them is changed, uncheck (turn off) the appropriate “Address Verified” checkbox for the particular address, update the Address Return Code to “Changed” and perhaps also update the address footnotes. This will indicate that the address needs to be re-verified.
- 2) Set up desired records to automatically verify if criteria like address field population, and “Address Verified” field being false. See Custom Object Process Verifications in Chapter 4, for a description of how to set this up.

## 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](mailto: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.



<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 is provided free with the app, but needs to be configured. SmartyStreets does not charge you for API calls when you use this, but when you press the “Save” button, it then sends a regular verification call to SmartyStreets, and this one is charged. As with the previous steps, we need to switch to the “Setup” area.

The screenshot shows the 'New Lead: Regular' form in Salesforce. The 'Address Information' section is highlighted. The 'Street' field contains the text '415 Mission St, San' and has a dropdown menu open showing several suggestions: '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', and '415 Mission St, San Juan Capistrano CA'. To the right of the 'Street' field, there are input fields for 'City/Province', 'Country', and 'State'.

## Initial Configuration

- 1) Ensure that you have completed the **Enabling Autocomplete** section in Chapter 2 to set up the web key in the configuration page. This will open up the SmartyStreets 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” with the address field(s) inside of it.

The screenshot displays a Salesforce Lead record layout. At the top, there are tabs for 'Quick Actions', 'Mobile & Lightning Actions', and 'Expanded Lookups'. Below these are several fields: 'Address', 'Address Carrier R...', 'Address Current Time', 'Address Last Veri...', 'Address Premise', and 'Address Return Code'. The main section is titled 'Lead Information (Header visible on edit only)' and contains fields for 'Lead Owner' (Sample Text), 'Phone' (1-415-555-1212), 'Name' (Sarah Sample), 'Mobile' (1-415-555-1212), 'Company' (Sample Text), and 'Email' (sarah.sample@company.com). The 'Address Information' section is highlighted with a red box and contains the 'Address' field with the value 'Suite 300, The Landmark @ One Market, San Francisco, CA 94105, US'. Below this is the 'Description Information (Header visible on edit only)' section with a 'Description' field (Sample Text). At the bottom is the 'International Information' section with fields for 'Registered Organization' (Sample Text), 'Thoroughfare Name' (Sample Text), 'Super Administrative' (Sample Text), and 'Sub Building Number' (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 → Installed Packages → Address Verification → Configure, and click on the “Manage Addresses” tab. The address must be listed here. If not, then it must be added.
  - a. Also, the Country field, as well as the required fields must be populated on the address record. Note that only US addresses are supported at this time, but the Country field must still be available in order for it to work.
  - b. You can turn on the Street 2 field by populating this field and also turning on the “Use Street 2 with Autocomplete” feature at the bottom of the address configuration.
  - 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.

Once these steps are done, you are ready to continue, depending on if you are on Classic or Lightning.

## Lightning Experience Configuration

- 1) In Lightning, go to Setup -> Object Manager and then go to the object where you want to set Autocomplete up.
- 2) 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.
- 3) 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.
- 4) 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 options. 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.

## Classic Configuration

### Important Note for Using Custom Edit Pages

Please note that once you start using custom edit pages you are entering a realm where you will have more administration than is typically needed in a Salesforce environment. Usually if you want to make a change to the layout like add or remove a field, you just have to click the "Edit Layout" link and make the change. However, once you have implemented this, you will have two additional steps:



1. You will need to re-create the edit page by clicking on the "Create Edit Page" link (you might need to add this link to your layout first). This is shown in steps 7-11 below.
2. You will need to give users access to this page again by updating the security for it. This is shown in steps 12-14 below.

All of the JavaScript code referenced in the pages below can be found in the "js" folder in our GitHub repository:



GitHub.com Address Verification Sample Code Repository

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

### Add the "Create Edit Page" Link

2) The second thing that we need to do is create a JavaScript link that will be used to create our edit page. We will assume in these instructions that we are creating it for Leads, but the instructions can be carried out also for Contacts or Accounts, by tweaking them slightly.

3) Go to our GitHub repository. Click on the "create\_lead\_edit\_page" page and copy all of the code to the clipboard. Note that when copying the code, do not include the line numbers. Start where the code starts at the top of line 1 (with "{!REQUIRED...}") and go all the way to the last line.

*\* Note: If you are working on Accounts, there is one special requirement for Person Accounts. The regular "create\_account\_edit\_page" will not work for Person Accounts, so you will need to also install the "create\_person\_account\_edit\_page" in that case.*

4) Go back to Salesforce and go to Setup -> Customize -> Leads -> "Buttons, Links and Actions" and click on the "New Button or Link" button.

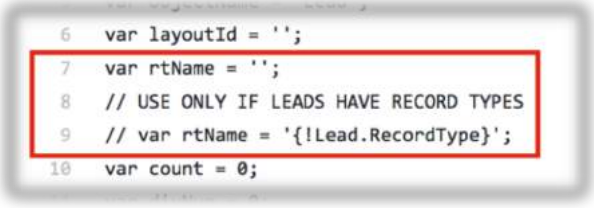
5) In the create button page, select the following:

- a) Label = "Create Edit Page" (you can change this if you wish)
- b) Display Type = "Detail Page Link" (you can change this to Detail Page Button if you wish)
- c) Behavior = "Execute JavaScript"
- d) Content Source = "OnClick JavaScript"

6) In the body of the button, past the code you copied to the clipboard in step #2. Save and press "OK".

You have now created the link and you can add it to the Lead Layout so that you can use it. Just one quick question, though...

Does the object you are wanting to set up (e.g. Leads) use Record Types? In order to check: go to Setup -> Customize -> Leads -> Record Types, do you see any record types listed? If so, you will need to comment line #7 and uncomment line #9. Save.



```

6  var layoutId = '';
7  var rtName = '';
8  // USE ONLY IF LEADS HAVE RECORD TYPES
9  // var rtName = '{!Lead.RecordType}';
10 var count = 0;

```

7) Go to Setup -> Customize -> Leads -> Page Layouts

8) IF you have more than one layout, you need to know which of the layouts are being used. Click on the "Page Layout Assignments" and make a note of which page layouts are used. *For each one, you will need to do steps 9-14.*

9) Click "Edit" next to the Page Layout you want to modify

10) Scroll down to the "Custom Links" section and click on the "Custom Links" list in the top toolkit. Drag down the "Create Edit Page" link into the "Custom Links" section. Save the Layout.

11) Now go to a record of the type you just updated the layout for (e.g. Leads) and you will see the "Create Edit Page" link. Click this link to create the edit page. You will see it open up. You can Save or click "Cancel" if you wish.

## Enabling the New Custom "Edit" Page for New Records

Now we need to do two steps in order to use the new page we created:

12) First, to enable the new custom New page on record edits or inserts, please do the following. Go to Setup -> Customize -> Leads -> "Buttons, Links and Actions" and click the **Edit** link on the "New" label.

13) Select the **AV Lead Override (New/Edit)** page and Save. Now, when you add a new Lead, the custom Edit page should display (if you have access to it through your profile).

*\* Note: You can use this as well with the Edit page overwrite, but if you do so, inline editing will no longer work. See the next sub-section for how to use the new custom Edit page with existing records.*

### **Enabling the New Custom “Edit” Page with Existing Records**

This step will allow you to use the autocomplete edit page with existing records. Note that you can use the previous section to overwrite the “Edit” page, if you want, but doing that will remove the Inline Editing function that many users find useful in Salesforce. By creating a custom Edit button as explained in this section, you can maintain inline editing, while overwriting the main Edit page.

14) Now we must create a button for our custom edit page. We will assume in these instructions that we are creating it for Leads, but the instructions can be carried out also for Contacts or Accounts, by tweaking them slightly.

15) Go back to Salesforce and go to Setup -> Customize -> Leads -> "Buttons, Links and Actions" and click on the "New Button or Link" button.

16) In the create button page, select the following:

- a) Label = " Edit (Autocomplete)" (you can change this if you wish)
- b) Display Type = "Detail Page Button"
- c) Behavior = "Display in existing window without sidebar or header"
- d) Content Source = "Visualforce Page"
- e) Content = “AV Lead Override (New/Edit)”

17) Go back to Setup -> Customize -> Leads -> Page Layouts

18) IF you have more than one layout, you need to know which of the layouts are being used. Click on the "Page Layout Assignments" and make a note of which page layouts are used. *For each one, you will need to do steps 19-21.*

19) Click "Edit" next to the Page Layout you want to modify

20) Scroll to the "Buttons" section and click on the "Buttons" list in the top toolkit. Drag down the "Edit" button into the "Custom Buttons" section. You may want to remove the standard “Edit” button as well. Save the Layout.

21) Now go to a record of the type you just updated the layout for (e.g. Leads) and you will see the custom "Edit" button. Test and make sure it takes you to the new page.

After testing as the System Administrator, now deploy to users.

22) We need to give everybody access to this new Edit page that was created. Go to Setup -> Build -> Develop -> VisualForce Pages.

23) Find a page with a name starting with "AV" and then named with the object name followed by the record type name, or "Standard" if no record types exist. For example, for Leads, it would be "AV Lead Standard". Click the "Security" link next to this page.

*\* Note: You might want to sort the list of VisualForce pages by the Last Modified Date so that the latest page created will be at the top.*

24) Move over as many profiles as you want from the "Available Profiles" to the "Enabled Profiles". If you only want some of your users to use the new edit page with auto-complete, then don't put their profile into the "Enabled Profiles". Save when done.

You will need to do the same steps 2-16 for the Account and Contact objects.

*Please note that autocomplete is not currently available for custom objects in Classic (it works in Lightning). However, we would be happy to work with you to create custom autocomplete pages for your own object, or integrate with other page solutions you might be using. We have some starter code that could help you.*



## Autocomplete 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.

**ADDRESS VERIFICATION**

Billing Street: 11730 Plaza America Dr Ste 205

Billing City: Reston | Billing State: VA

Billing PostalCode: 20190-4750 | Billing Country: USA

Shipping Street: 100 Main St

Shipping State: | Shipping Country: USA

Verify and Update

Shipping Street suggestions:

- 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
- 100 S Main St, Abbeville LA

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>

We hope you enjoy our Address Verification app for Salesforce! Please do let us know if you run into any issues by simply emailing [support@toafinish.com](mailto:support@toafinish.com) and asking any question. We're here to help!