verifytreatment™

Salesforce Implementation Guide

Versions 2.0.0

Table of Contents

About VerifyTreatment

VerifyTreatment is a cloud-based health insurance verification platform that accelerates the admissions process. It searches and queries 500+ of the leading insurers in the U.S. ensuring immediate verification of benefits. VerifyTreatment makes the admissions process faster & easier to avoid lost admissions. It is designed for treatment centers to admit on the first call, improve patient relations, increase revenue and streamline workflow.

Application Summary

VerifyTreatment Health Insurance Verification is an AppExchange Application. It allows healthcare providers to seamlessly connect their CRM to 500+ insurance company databases to receive benefits & eligibility on-demand. The application verifies health insurance benefits directly from Salesforce - 24 hours a day, 7 days a week, including holidays. It is Available for Salesforce Lightning or Classic.

Features

- Accurately verify health insurance benefits with one click on any standard or custom objects in Salesforce.
- Gain access to 1000's of insurance company databases in one app.
- Seamlessly integrate electronic health insurance verification with your Salesforce CRM.
- Run reports & create dashboards with real-time insurance verification data.

Overview

This user manual provides step-by-step guidance on how to verify insurance benefits on a different object.

Specifications

By default, the "Verify Insurance" button is used on the "VOB Object" that comes with the managed package. To use the Verify Insurance button on any other standard or custom object (i.e. the controller object), please follow the steps below.

Controller Object - the parent object to the verification that contains the Verify Insurance button.

IMPORTANT NOTE: *The Verify Insurance button can only be placed on one object (either Standard or Custom object).*

For 2.0.0 Versions

Important Note: *If you are already using 1.38 or earlier versions of VerifyTreatment, and are updating to version 2.0.0 please follow the steps below..*

Salesforce Lightning or Classic Version:

1. From the Object Manager in Setup, select the **VOB** object with the description Linked to VerifyTreatment App.
2. In Page Layouts, select **Practice Group Layout**.
3. In **Related lists**, add the **Verification History** section to the page layout.
4. Save the page layout.

Integration of VerifyTreatment and Salesforce

Follow for both: Salesforce Classic and Lightning Experience.

To begin, you will start with creating the connection to Salesforce and VerifyTreatment with the users, payers, and facilities. This ensures that Salesforce and VerifyTreatment can communicate together with the proper information from your user and facilities, as well as the payers that you are receiving information from.

Authorize the user.

1. In the VerifyTreatment app, select the **Authorize** tab.

2. Select the button **Authorize User** and enter your Email and Password for Verify Tx.
3. Select the **Login** Button.
4. Select the **Allow** button. This ensures that salesforce can access your VerifyTreatment account.

Refresh Payer Name and Facility:

5. In the VerifyTreatment app, select the **PayerName/Facility** tab.
6. Select **Refresh Payer Name**.
7. After the payers are done loading, select the **Facility Name**.

The following provides steps on how to implement the "Verify Insurance" button on the Lead Object, however, you can use any other standard or custom object.

Map all required fields needed to successfully submit the verification (i.e the Patient First Name, Last name, Date of Birth, etc.). You can use existing fields for mapping, but the data type must match (e.g Date of Birth must be a data field). Below are all the required fields necessary for the field mapping. The controlling object must contain all required fields below.

Note - If you are using the VOB Object from the Verify Tx Package, you do not need to create these fields.

1. From the Object Manager in Setup, select the **Lead** Object.
2. In **Fields & Relationships**, locate the fields highlighted below to ensure that the data type matches what is in the table below. Example: Client Status should have the data type of Picklist with the values Male and Female.

Field Name	Field API Name	Data Type
Patient First Name	Patient_First_Name__c	Text (255)
Patient Last Name	Patient_Last_Name__c	Text (255)
Patient Birth Date	Patient_Birth_Date__c	Date
Patient Member Id	Patient_Member_Id__c	Text (255)
Client Status	Client_Status__c	Picklist Values

		• Prospect • Client • Discharged • Referred • Termed
Subscriber Relationship	Subscriber_Relationship__c	Text (255)
Gender	Gender__c	Picklist Values • Male • Female
VTX Facility	VTX_Facility__c	Lookup(VTX Facility)
VTX Payer Detail	VTX_Payer_Detail__c	Lookup(VTX Payer Detail)
Verification Status	Verification_Status__c	Rich Text Area (32768)
Vob Id	Vob_Id__c	Text (255)

After all these mappings are set, create a lookup field in the VerifyTreatment object and link it with the controller object.

1. From the Object Manager in Setup, select the **VerifyTreatment** object. It is an object from the **VerifyTreatment App** managed package.
2. In **Custom Fields & Relationships**, select **New**.
3. Pick the **Lookup Relationship** data type and **Next.**
4. Relate the object with the controller object**.** In our case, the controller object is **Lead**.
5. Name the Lookup Relationship (e.g. Lead Lookup).
6. Save the Lookup Relationship.

Field Name	**Field API Name**	**Data Type**
Lead Lookup	Lead_Lookup__c	Lookup (To Lead Object)

Next, create a Lookup field in the History object and link it with the controller object.

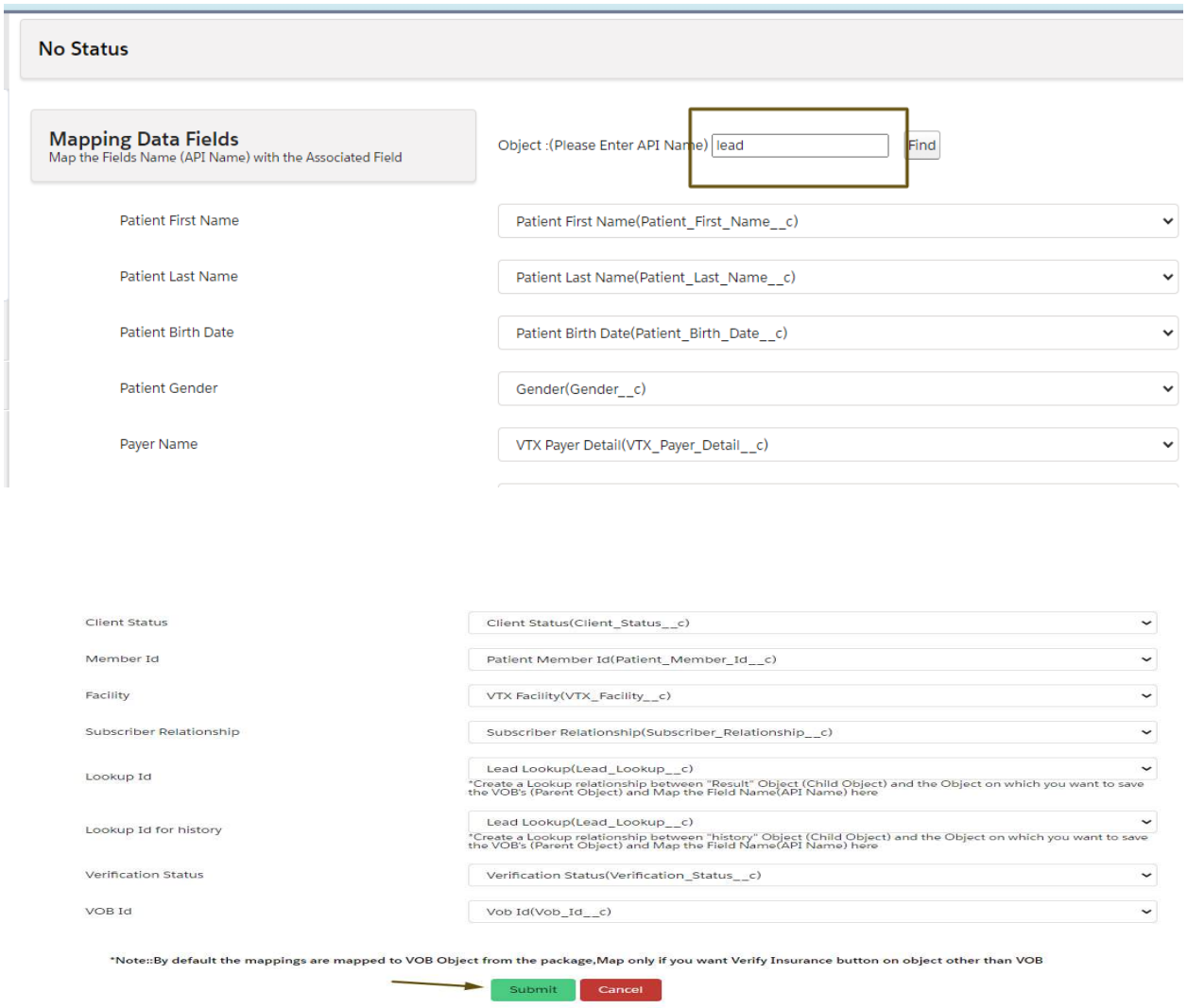
1. From the Object Manager in Setup, select the **Verification History** object. It is an object from the **VerifyTreatment App** managed package.
2. In **Custom Fields & Relationships**, select **New**.
3. Pick the **Lookup Relationship** data type and **Next.**
4. Relate the object on which you will be performing the **Verification of Benefits.** In our case, the object is **Lead**.
5. Name the Lookup Relationship. We are naming ours Lead Lookup.
6. Save the Lookup Relationship.

Field Name	Field API Name	Data Type
Lead Lookup	Lead_Lookup__c	Lookup (To Lead Object)

After all of the fields are created and properly configured, you need to map these fields in the Mappings tab so that the code in the background can make use of these fields. You need to map the API name of the field in the mappings tab.

1. In the VerifyTreatment app, select the **Mappings** tab.
2. Enter the **API** name into the Object and select find.
3. Map the fields from your object that you would like to pull from. For us this is the **Lead** object.
4. Select Submit.

Note: *By default, the Mappings are configured for VOB object.*


No Status
Mapping Data Fields
Map the Fields Name (API Name) with the Associated Field
Object :(Please Enter API Name) lead
Find
Patient First Name
Patient First Name(Patient_First_Name__c)
Patient Last Name
Patient Last Name(Patient_Last_Name__c)
Patient Birth Date
Patient Birth Date(Patient_Birth_Date__c)
Patient Gender
Gender(Gender__c)
Payer Name
VTX Payer Detail(VTX_Payer_Detail__c)
Client Status
Client Status(Client_Status__c)
Member Id
Patient Member Id(Patient_Member_Id__c)
Facility
VTX Facility(VTX_Facility__c)
Subscriber Relationship
Subscriber Relationship(Subscriber_Relationship__c)
Lookup Id
Lead Lookup(Lead_Lookup__c)
*Create a Lookup relationship between "Result" Object (Child Object) and the Object on which you want to save the VOB's (Parent Object) and Map the Field Name(API Name) here
Lookup Id for history
Lead Lookup(Lead_Lookup__c)
*Create a Lookup relationship between "history" Object (Child Object) and the Object on which you want to save the VOB's (Parent Object) and Map the Field Name(API Name) here
Verification Status
Verification Status(Verification_Status__c)
VOB Id
Vob Id(Vob_Id__c)
*Note::By default the mappings are mapped to VOB Object from the package,Map only if you want Verify Insurance button on object other than VOB
Submit
Cancel

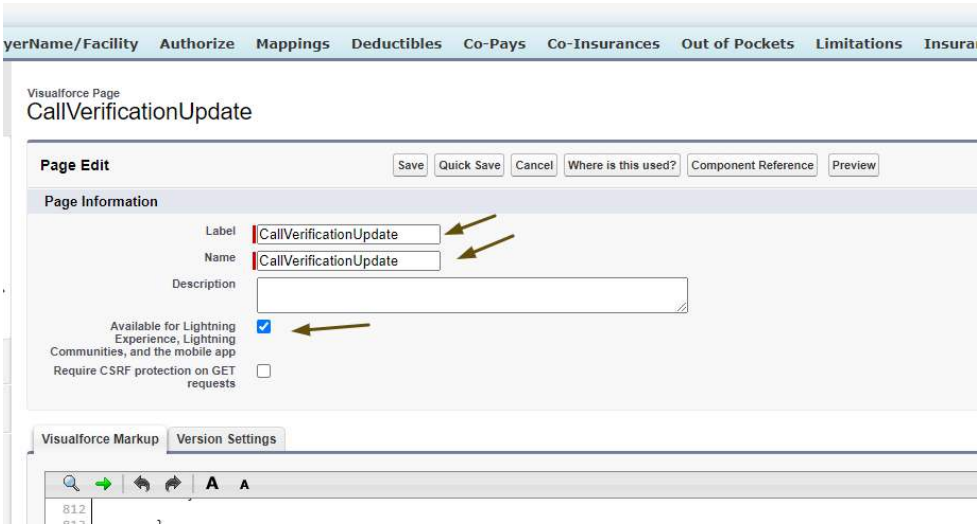
Associated Field	FieldAPI Name
Patient First Name	Patient First Name(Patient_First_Name__c)
Patient Last Name	Patient Last Name(Patient_Last_Name__c)
Patient Birth Date	Patient Birth Date(Patient_Birth_Date__c)
Patient Gender	Gender(Gender__c)
Payer Name	VTX Payer Detail(VTX_Payer_Detail__c)
Client Status	Client Status(Client_Status__c)
Member Id	Patient Member Id(Patient_Member_Id__c)
Facility	VTX Facility(VTX_Facility__c)
Subscriber Relationship	Subscriber Relationship(Subscriber_Relationship__c)
Lookup Id	Lead Lookup(Lead_Lookup__c)
Lookup Id for history	Lead Lookup(Lead_Lookup__c)
Verification Status	Verification Status(Verification_Status__c)
VOB Id	Vob Id(Vob_Id__c)

If all the fields are mapped correctly the status will change to Success Record Saved.

Finally, you will need to create a Custom Button in the Object on which you want to Verify Benefits. In our case, that object is the Lead, so we will create a custom button on the Lead Object and will place it into the layout. To do this, you will have to start with a visualforce page.

The steps to create the visualforce page are provided below:

1. From Setup, search **Visualforce Pages** in **Quick Find Box** and select the link.
2. Select **New**.
3. Label the button and ensure that the checkbox **Available for Lightning Experience, Lightning Communities, and the mobile app** is selected.



4. Copy code for visualforce page (from the link) and paste into the **Visualforce Markup** section.

   https://docs.google.com/document/d/1Huq9hgJKd9M93vToarNhSUY6LyD6eJUObh-smrwLrcY/edit

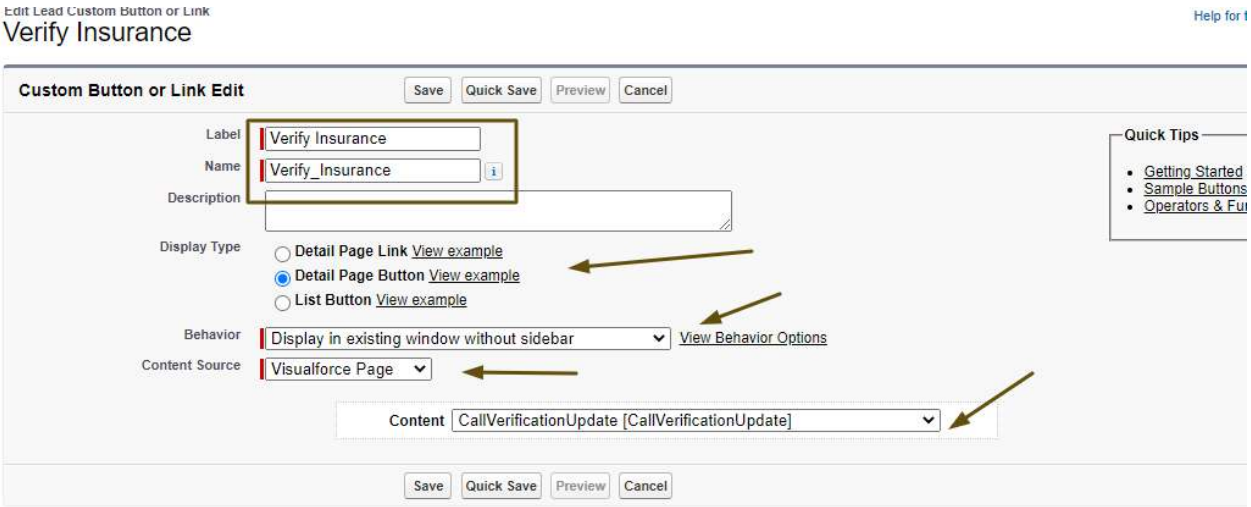
5. The controller on the visual force page must match the **API Name** of the object that you are placing the button on. For example we are using the Lead object, which means our standardController is Lead.

Code For VisualForce Page	
<apex:page	standardController="Lead"

6. Save the Page.

The Steps to Add the button are provided below:

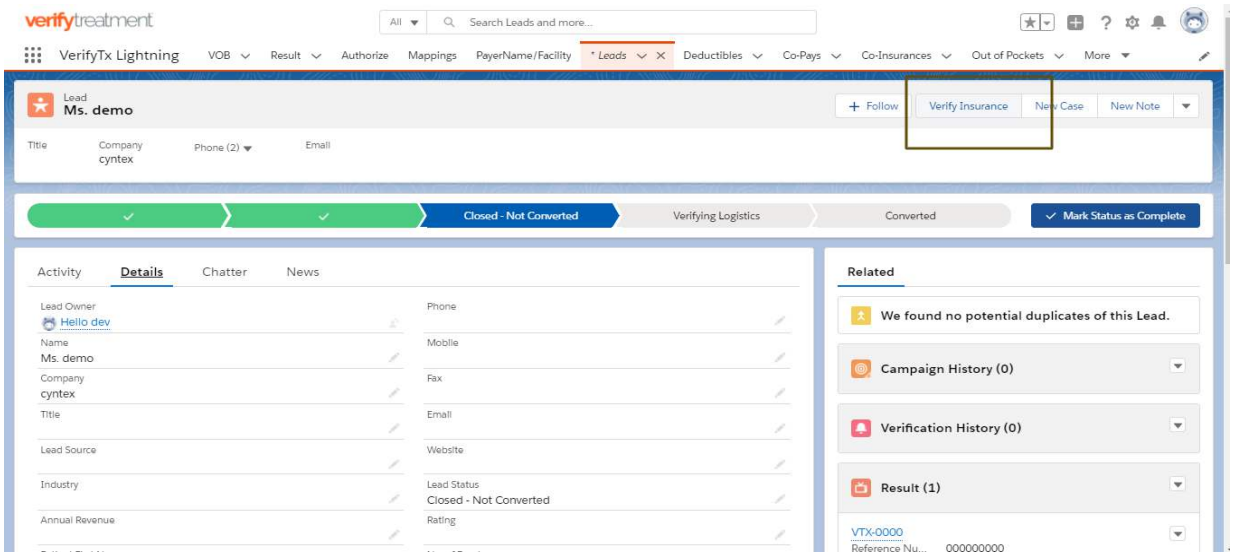
1. In the Object Manager in Setup, select the **Lead** object.
2. In **Buttons, Links, and Actions**, select **New Button or Link.**
3. Label the button, we are choosing **Verify Insurance.**
4. In Display type, select the **Detail Page button**.
5. In Behaviour, select **Display in existing window without sidebar.**
6. In Content Source, select **Visualforce Page.**
7. In Content, select the **visualforce** page you created. If you followed our use case it is called **CallVerificationUpdate**. See below.



8. Save the button.
9. In **Page Layouts**, select **Buttons**.
10. Drag your button onto the page. (**Verify Insurance** if you followed our use case).
11. Save the Page Layout.

Congratulations!

Now all you need to do is Create a New Lead record, populate the record, save, and "Verify Insurance" using the new button.



Salesforce Lightning Experience

If you have successfully completed the salesforce integration and would like the status update of the Client on another object, please follow the below steps.

1. From the Home Tab in Setup, enter **Apex Classes** and select the link.
2. Select the new button and paste the below code.

https://docs.google.com/document/d/1Huq9hgJKd9M93vToarNhSUY6LyD6eJUObh-smrwLrcY/edit

3. From the Home Tab in Setup, enter **Apex Triggers** and select the link.
4. Select **Developer Console** and create a new **trigger** on the object that you are using. We used **Lead**. Paste the below code in it.

https://docs.google.com/document/d/1Huq9hgJKd9M93vToarNhSUY6LyD6eJUObh-smrwLrcY/edit

Salesforce1 App

For using VerifyTreatment in Salesforce1 App, please follow below mentioned steps:

1. From the Object Manager in Setup, select the **VOB** object.
2. From Page Layouts, Select **Practice Group Layout.**
3. In **Mobile Cards**, remove **CallVerificationMethod**.

Please Note: If you are using 1.36 or an earlier version of the app and updating with the newer version, please perform the following steps before doing verification.

1. From the Object Manager in Setup, select the **VOB** object.
2. In **Page Layouts**, select your layout.
3. In **Mobile & Lightning Actions**, select the **Verify Insurance** button and add it to the **Page Layout** under the section **Salesforce Mobile and Lightning Experience Actions.**

Different Benefits in Different Objects

All of the different benefits are present in particular records and stored in different objects in salesforce. They are in the related section on the VerifyTreatment object that houses the results of the verification. These are named as Deductibles, OutOfPocket, PlanProductDetails, CoPay, CoInsurance.

History Capturing

History of each record that is verified and whose Status is client is captured in the History Object daily. The process of History Capturing is completed on 2 A. M. Daily and stored in the History object.