Informatica® Cloud Application Integration
July 2024

Synchronize NetSuite Cases with Salesforce Cases

Informatica Cloud Application Integration Synchronize NetSuite Cases with Salesforce Cases
July 2024

See patents at https://www.informatica.com/legal/patents.html.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Publication Date: 2024-07-12

Table of Contents

Preface

Use *Synchronize NetSuite Cases with Salesforce Cases* to learn how to synchronize NetSuite cases with Salesforce cases. The process is based on REST and SOAP APIs. This guide assumes that you have an understanding of the NetSuite Connector and Salesforce Connector concepts.

CHAPTER 1

Synchronize NetSuite Cases with Salesforce Cases Recipe Overview

The Synchronize NetSuite Cases with Salesforce Cases recipe is based on REST and SOAP APIs.

When a case is created or updated in NetSuite, a process is triggered to synchronize NetSuite cases with Salesforce cases. The process is called by an HTTP request. The process checks for a matching case contact or case owner account in Salesforce. If the case contact account does not exist, the process creates an account and the dependent contacts in Salesforce. The process then searches for a matching case in Salesforce using the NetSuite ID and creates or updates the case based on the search results.

Example

Consider you are an admin for an outdoor and recreational gear retailer. Your sales team uses NetSuite to manage and maintain inventory items. The team also uses Salesforce to manage and maintain products and associated price book entries that they use to sell to customers. Currently, the sales team is performing these tasks manually. Every time there's an inventory item that must be updated, a sales rep first updates the item with relevant details in NetSuite. Next, the rep determines if a corresponding product exists in Salesforce. This process is time-consuming and highly error-prone, so the team wants to automate it.

As a solution, you need to integrate NetSuite and Salesforce to achieve near real-time integration. With this recipe, when the sales team invokes the process, it synchronizes all tasks created or updated in NetSuite with Salesforce cases.

CHAPTER 2

Prerequisites for configuring a script in NetSuite

To synchronize NetSuite cases with Salesforce cases, the following prerequisites must be met:

- Create a script in NetSuite.
  For more information about creating a script, see "Creating a script" on page 7.
- Create a NetSuite ID in the Salesforce Account, Contact, and Case entities.
  For more information about creating a NetSuite ID, see "Creating a NetSuite ID in the Salesforce Account, Contact, and Case entities" on page 11

Configuring a script in NetSuite to subscribe to Case object change

NetSuite connections in Application Integration perform CRUD operations, such as create, read, update, delete on NetSuite tables. When you create or update a **Case** entity, a web request is initiated in NetSuite. To synchronize NetSuite cases with Salesforce cases, you need to create a script in NetSuite.

To configure a script in NetSuite, perform the following steps:

Step 1: Create a script in NetSuite.

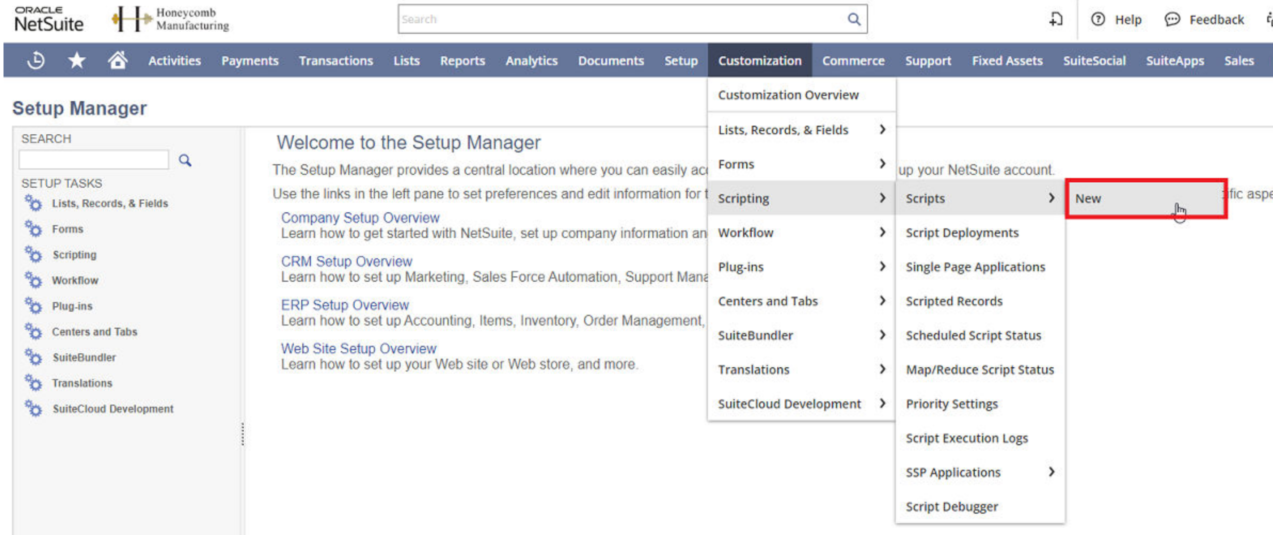
Step 2: Upload the script and create a script record.

Step 3: Deploy the script and apply it to a **Case** entity.

Creating a script

To create a script, perform the following steps:

1. Log in to the NetSuite organization.
2. Go to **Customization** > **Scripting** > **Scripts**, and then click **New**.



Welcome to the Setup Manager

The Setup Manager provides a central location where you can easily ac… up your NetSuite account.

Use the links in the left pane to set preferences and edit information for t… …fic aspe…

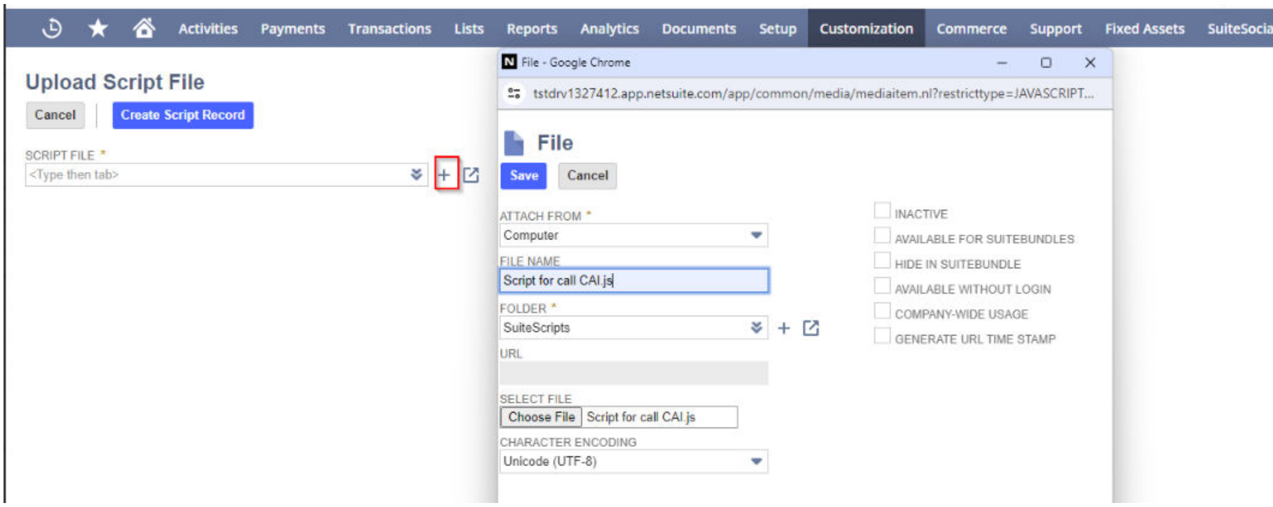
Company Setup Overview
Learn how to get started with NetSuite, set up company information an…

CRM Setup Overview
Learn how to set up Marketing, Sales Force Automation, Support Mana…

ERP Setup Overview
Learn how to set up Accounting, Items, Inventory, Order Management,…

Web Site Setup Overview
Learn how to set up your Web site or Web store, and more.

3. In the **Upload Script File** section, click the + icon to add a script file. Enter the file name and select the script file as shown in the following image:



SCRIPT FILE * <Type then tab>

tstdrv1327412.app.netsuite.com/app/common/media/mediaitem.nl?restricttype=JAVASCRIPT...

ATTACH FROM * Computer

FILE NAME Script for call CAI.js

FOLDER * SuiteScripts

URL

SELECT FILE Script for call CAI.js

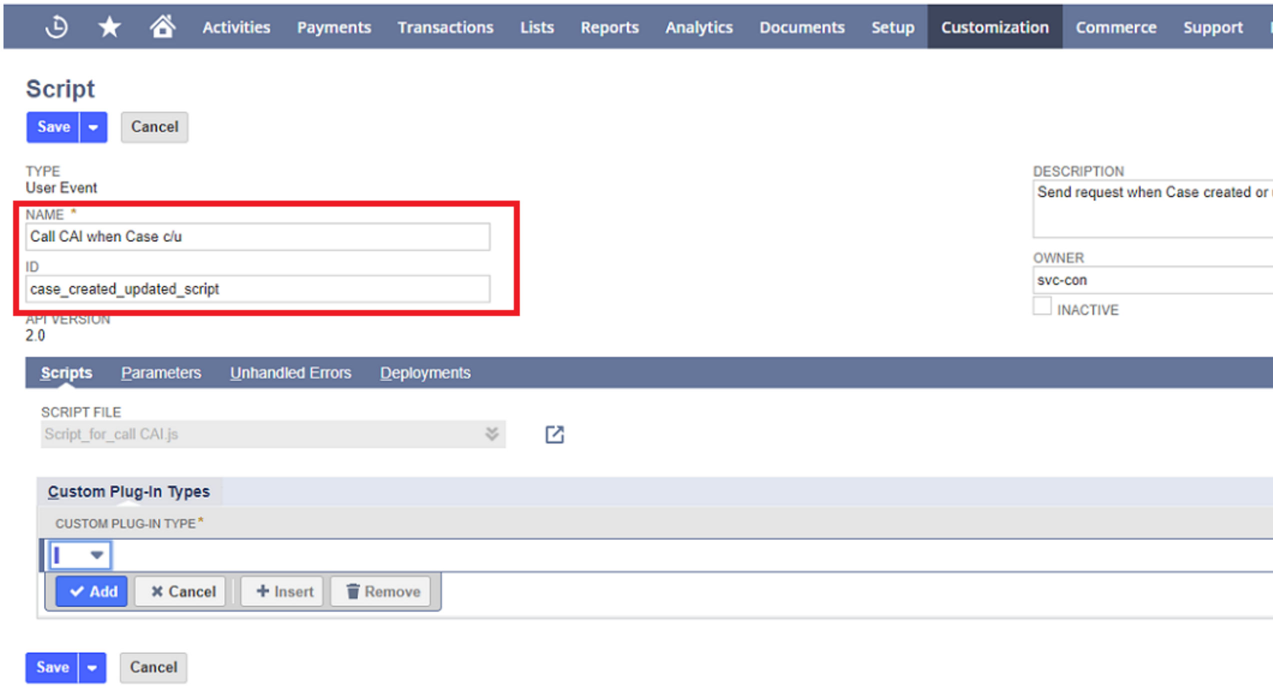
CHARACTER ENCODING Unicode (UTF-8)

□ INACTIVE
□ AVAILABLE FOR SUITEBUNDLES
□ HIDE IN SUITEBUNDLE
□ AVAILABLE WITHOUT LOGIN
□ COMPANY-WIDE USAGE
□ GENERATE URL TIME STAMP

Note: Use the `.js` extension for the script file name.

For more information about the script format, see "Sample script" on page 10.

4. Click **Save**.
5. After adding the script file in the **Upload Script File** section, click **Create Script Record**.
6. In the **Script** section, enter the NetSuite event name and ID as shown in the following image:



Script

TYPE
User Event

NAME *
Call CAI when Case c/u

ID
case_created_updated_script

API VERSION
2.0

DESCRIPTION
Send request when Case created or

OWNER
svc-con

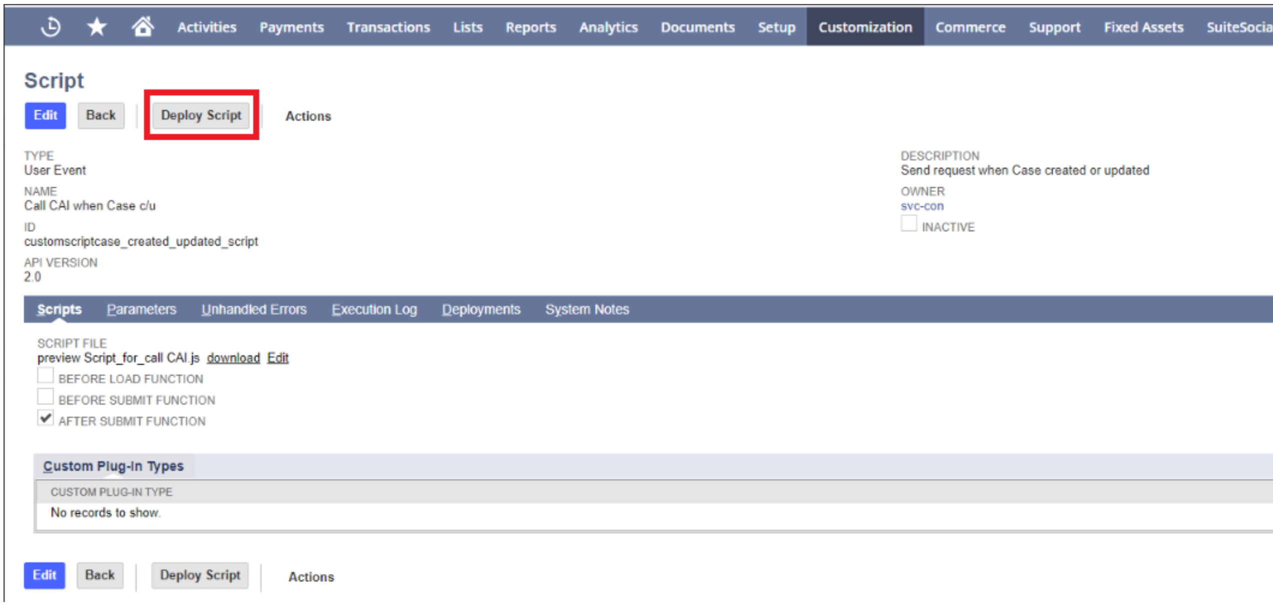
□ INACTIVE

SCRIPT FILE
Script_for_call CAI.js

CUSTOM PLUG-IN TYPE *

7. Click **Save**.

8. In the **Script** section, click **Deploy Script**.



Script

TYPE
User Event

NAME
Call CAI when Case c/u

ID
customscriptcase_created_updated_script

API VERSION
2.0

DESCRIPTION
Send request when Case created or updated

OWNER
svc-con

□ INACTIVE

SCRIPT FILE
preview Script_for_call CAI.js download Edit

□ BEFORE LOAD FUNCTION
□ BEFORE SUBMIT FUNCTION
☑ AFTER SUBMIT FUNCTION

CUSTOM PLUG-IN TYPE
No records to show.

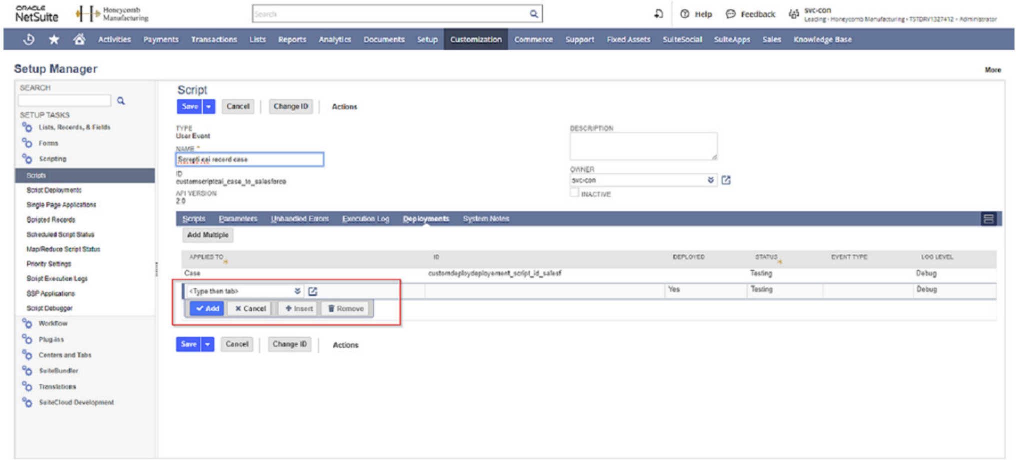
9. You can now apply the script to any **Case** entity. Enter the deployment script ID and select the **Status**.

10. Click **Save**.

    The scrip is successfully deployed.

    Important: The deployed script can be used for any entity in NetSuite, not just the **Case** entity.

    You can create and add multiple scripts and reuse them.



TYPE
User Event

NAME *
Screpti cai record case

ID
customscriptcai_case_to_salesforce

API VERSION
2.0

DESCRIPTION

OWNER
svc-con

□ INACTIVE

APPLIES TO	ID	DEPLOYED	STATUS	EVENT TYPE	LOG LEVEL
Case	customdeploydeployement_script_id_salesf		Testing		Debug
<Type then tab>		Yes	Testing		Debug

Sample script

You can create a script in NetSuite to automate business processes.

The following snippet is a sample script for NetSuite:

```
/**
 * @NApiVersion 2.x
 * @NScriptType UserEventScript
 */
define(['N/https', 'N/log'], function(https, log) {

    function afterSubmit(context) {
        if (context.type === context.UserEventType.CREATE || context.type ===
context.UserEventType.EDIT) {
            var newRecord = context.newRecord;
            var caseId = newRecord.id;
            var url = '{CAI process URL}?Input_Case_ID=' + caseId;
            try {
                var response = https.get({
                    url: url
                });
                log.debug({
                    title: 'GET Request Response',
                    details: 'Response: ' + response.body
                });
            } catch (e) {
                log.error({
                    title: 'Error Sending GET Request',
                    details: e.message
                });
            }
        }
    }

    return {
        afterSubmit: afterSubmit
    };
});
```

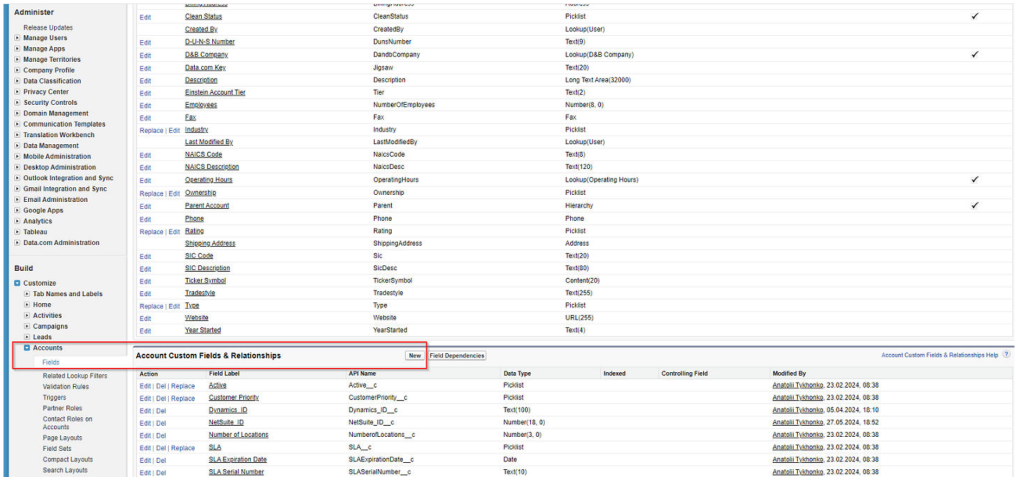
Note: You must include the process URL in the script as shown in the sample. Save the file with a `.js` extension.

Creating a NetSuite ID in the Salesforce Account, Contact, and Case entities

Provide the necessary details in the Account, Contact, and Case entities to create the NetSuite ID in Salesforce.

1. Go to **Setup** > **Build** > **Customize** > **Accounts** > **Fields**, and then click **New**.

   The following image shows the **Accounts** > **Fields** option:

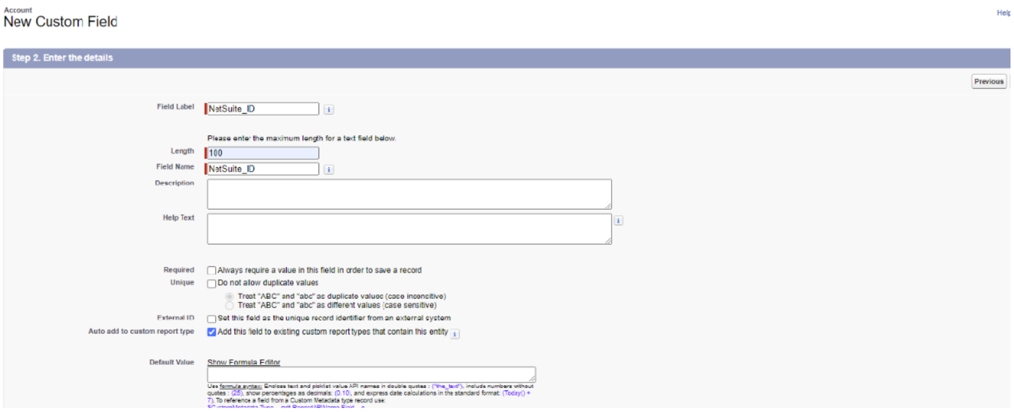


Edit	Clean Status	CleanStatus	Picklist	✓
	Created By	CreatedBy	Lookup(User)	
Edit	D-U-N-S Number	DunsNumber	Text(9)	
Edit	D&B Company	DandbCompany	Lookup(D&B Company)	✓
Edit	Data.com Key	Jigsaw	Text(20)	
Edit	Description	Description	Long Text Area(32000)	
Edit	Einstein Account Tier	Tier	Text(2)	
Edit	Employees	NumberOfEmployees	Number(8, 0)	
Edit	Fax	Fax	Fax	
Replace | Edit	Industry	Industry	Picklist	
	Last Modified By	LastModifiedBy	Lookup(User)	
Edit	NAICS Code	NaicsCode	Text(8)	
Edit	NAICS Description	NaicsDesc	Text(120)	
Edit	Operating Hours	OperatingHours	Lookup(Operating Hours)	✓
Replace | Edit	Ownership	Ownership	Picklist	
Edit	Parent Account	Parent	Hierarchy	✓
Edit	Phone	Phone	Phone	
Replace | Edit	Rating	Rating	Picklist	
	Shipping Address	ShippingAddress	Address	
Edit	SIC Code	Sic	Text(20)	
Edit	SIC Description	SicDesc	Text(80)	
Edit	Ticker Symbol	TickerSymbol	Content(20)	
Edit	Tradestyle	Tradestyle	Text(255)	
Replace | Edit	Type	Type	Picklist	
Edit	Website	Website	URL(255)	
Edit	Year Started	YearStarted	Text(4)	

Account Custom Fields & Relationships

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del | Replace	Active	Active__c	Picklist			Anatolii Tykhonko, 23.02.2024, 08:38
Edit | Del | Replace	Customer Priority	CustomerPriority__c	Picklist			Anatolii Tykhonko, 23.02.2024, 08:38
Edit | Del	Dynamics_ID	Dynamics_ID__c	Text(100)			Anatolii Tykhonko, 05.04.2024, 18:10
Edit | Del	NetSuite_ID	NetSuite_ID__c	Number(18, 0)			Anatolii Tykhonko, 27.05.2024, 18:52
Edit | Del	Number of Locations	NumberofLocations__c	Number(3, 0)			Anatolii Tykhonko, 23.02.2024, 08:38
Edit | Del | Replace	SLA	SLA__c	Picklist			Anatolii Tykhonko, 23.02.2024, 08:38
Edit | Del	SLA Expiration Date	SLAExpirationDate__c	Date			Anatolii Tykhonko, 23.02.2024, 08:38
Edit | Del	SLA Serial Number	SLASerialNumber__c	Text(10)			Anatolii Tykhonko, 23.02.2024, 08:38

2. In the **Choose Type Field** section, select **Text Area** and click **Next**.
3. Enter the details in the **Field Label**, **Length**, and **Field Name** fields.



Field Label: NetSuite_ID

Please enter the maximum length for a text field below.

Length: 100

Field Name: NetSuite_ID

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

○ Treat "ABC" and "abc" as duplicate values (case insensitive)
○ Treat "ABC" and "abc" as different values (case sensitive)

External ID: □ Set this field as the unique record identifier from an external system

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use:

   Save the field name for your future use.

4. Click **Next**. Don't make any change in the other pages.
5. Click **Save**.

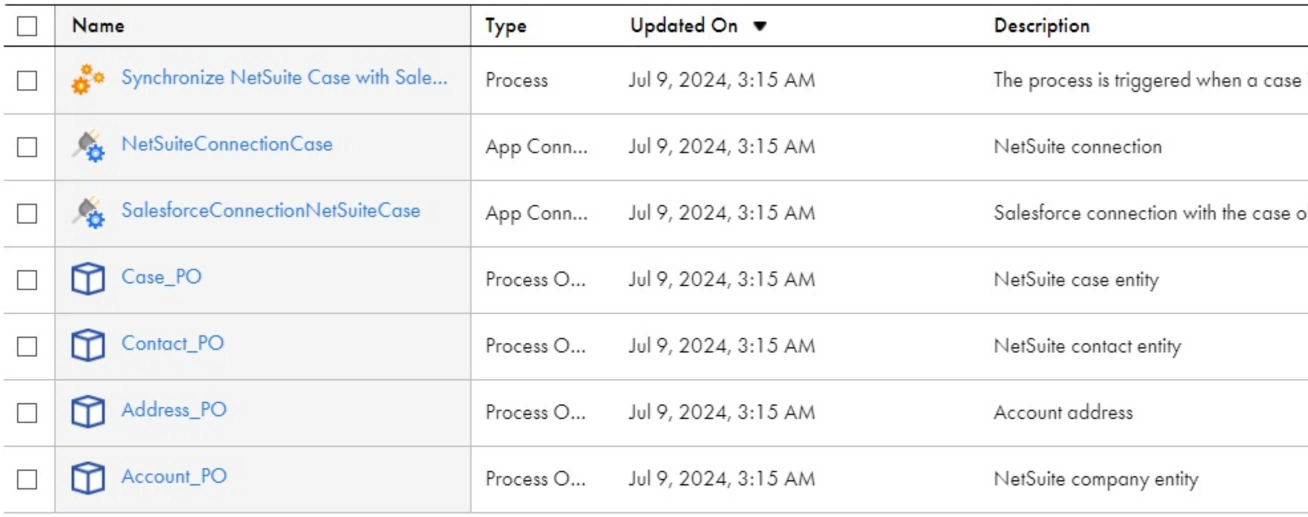
Follow the same steps for the **Contact** and **Case** entities.

CHAPTER 3

Synchronize NetSuite Cases with Salesforce cases recipe contents

The recipe contains multiple assets such as process objects, app connections, and process.

The following image shows the assets that the Synchronize NetSuite Cases with Salesforce cases recipe package contains:



□	Name	Type	Updated On ▼	Description
□	Synchronize NetSuite Case with Sale...	Process	Jul 9, 2024, 3:15 AM	The process is triggered when a case
□	NetSuiteConnectionCase	App Conn...	Jul 9, 2024, 3:15 AM	NetSuite connection
□	SalesforceConnectionNetSuiteCase	App Conn...	Jul 9, 2024, 3:15 AM	Salesforce connection with the case o
□	Case_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite case entity
□	Contact_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite contact entity
□	Address_PO	Process O...	Jul 9, 2024, 3:15 AM	Account address
□	Account_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite company entity

Synchronize NetSuite Cases with Salesforce Cases recipe assets

The following table lists the assets that the Synchronize NetSuite Cases with Salesforce Cases recipe package contains:

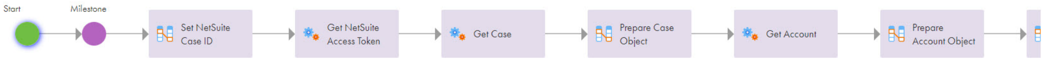
Asset Name	Asset Type	Description
Account_PO	Process object	Prepares a Salesforce user account from the NetSuite account.
Address_PO	Process object	Prepares a Salesforce user address from the NetSuite address.
Case_PO	Process object	Prepares a Salesforce case from the NetSuite case.
Contact_PO	Process object	Prepares a Salesforce contact from the NetSuite contact.
NetSuiteConnectionCase	App connection	Connects to the NetSuite connection.
SalesforceConnectionNetSuiteCase	App connection	Connects to the Salesforce connection with **Case** object filters.
Synchronize NetSuite Case with Salesforce Case	Process	Perform steps to synchronize data from NetSuite cases to Salesforce cases when a case is created or updated in NetSuite.

NetSuite to Salesforce (Case to Case) process

When a case is created or updated in NetSuite, the process is triggered.

The process checks for a matching case contact or case owner account in Salesforce. If the case contact account does not exist, the process creates a contact account and the dependent contacts in Salesforce. The process then searches for a matching case in Salesforce using the NetSuite ID and creates or updates the case based on the search results.

The following image shows the steps that the NetSuite to Salesforce (Case to Case) process contains:



The following table lists the steps that the NetSuite to Salesforce (Case to Case) process contains:

Step Name	Description
Start	The NetSuite script triggers the process and passes the Case ID for the created or updated Case in NetSuite.
Set NetSuite Case ID	Assigns the Input_Case_ID from the NetSuite case ID triggered using the script.
Get NetSuite Access Token	Gets an access token to authorize all the connection requests.
Get Case	Gets the case from the NetSuite Case ID.
Prepare Case Object	Parses the event and assigns values. You can get all the information about the **Case** such as **ID**, **Company**, **Contact**, **Email**, and so on.
Get Account	Gets the account from NetSuite.
Prepare Account Object	Prepares the account to use in Salesforce. Parses the event and assigns values.
Get Salesforce Account ID	Gets the Salesforce account ID.
Is Salesforce Account ID set	Checks if the account ID is already available in Salesforce. If the account ID is missing in Salesforce, creates a new account in Salesforce and gets the new account ID. Otherwise, updates the existing account ID.
Is Contact ID set	Checks if the contact ID from NetSuite is already available in Salesforce.
Is Salesforce Contact ID set	Searches for the contact in Salesforce. If the contact is missing in Salesforce, creates a new contact in Salesforce and gets the contact ID. Otherwise, updates the contact.
Search Case by NetSuite ID	Searches for the case in Salesforce
Is Salesforce Case present	If the case is missing in Salesforce, creates a new case in Salesforce. Otherwise, updates the case.
End	Ends the process.

CHAPTER 4

Using the Synchronize NetSuite Cases with Salesforce Cases recipe

To use the Synchronize NetSuite Cases with Salesforce Cases recipe, you must perform the following steps manually:

Step 1: Copy and access the recipe

Step 2: Configure and publish the NetSuite connection

Step 3: Configure and publish the Salesforce connection

Step 4: Configure and publish the process

Step 5: Test data synchronization from NetSuite cases to Salesforce cases

Copying and accessing the recipe content

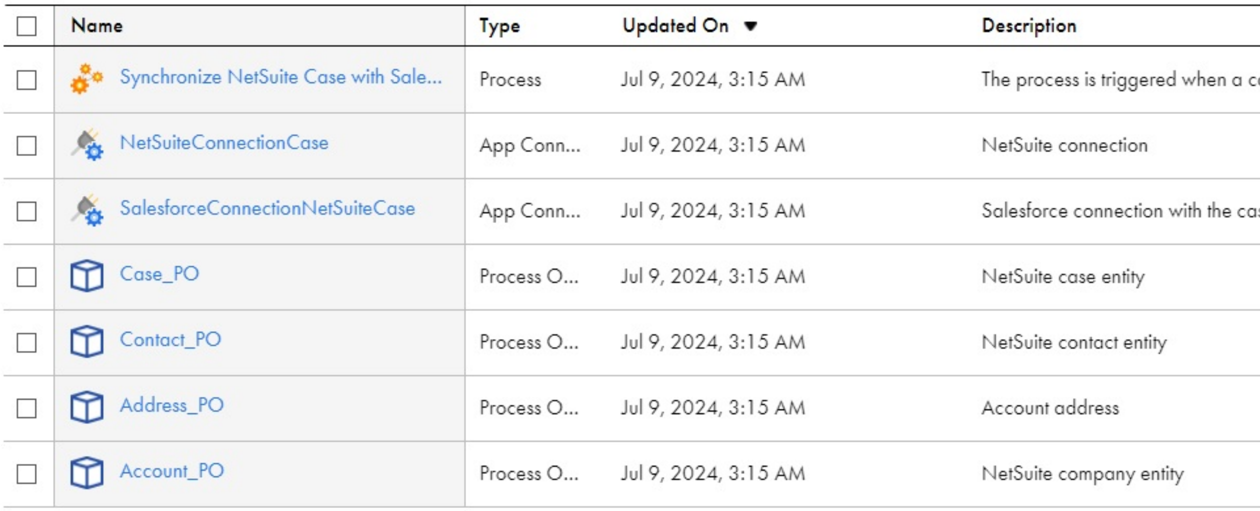
To copy and access the recipe content, perform the following steps:

1. Open the **Synchronize NetSuite Cases with Salesforce Cases** recipe and click **Use**.
2. Select the location where you want to copy the recipe, and then click **Continue**.
3. In the **Copying the recipe** dialog box, click **OK**.

   It might take some time for the recipe to get copied. You will receive a notification when the recipe is ready for use.
4. After the recipe is copied, click **Explore** to access the recipe content.

5. Navigate to the project or folder where you copied the recipe or enter the recipe name in the **Find** box.

   All the assets in the recipe are displayed as shown in the following image:



☐	Name	Type	Updated On ▼	Description
☐	Synchronize NetSuite Case with Sale...	Process	Jul 9, 2024, 3:15 AM	The process is triggered when a c
☐	NetSuiteConnectionCase	App Conn...	Jul 9, 2024, 3:15 AM	NetSuite connection
☐	SalesforceConnectionNetSuiteCase	App Conn...	Jul 9, 2024, 3:15 AM	Salesforce connection with the ca
☐	Case_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite case entity
☐	Contact_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite contact entity
☐	Address_PO	Process O...	Jul 9, 2024, 3:15 AM	Account address
☐	Account_PO	Process O...	Jul 9, 2024, 3:15 AM	NetSuite company entity

Configuring and publishing the NetSuite connection

To configure and publish the NetSuite connection, perform the following steps:

1. Open the **NetSuiteConnectionCase** connection.
2. In the **Type** field, select **NetSuite**.
3. From the **Run On** list, select Cloud Server or any Secure Agent.
4. In the **Connection Properties** section, enter values for the following properties:

Property	Description
Client ID	NetSuite OAuth 2.0 client ID to generate a valid access token. Enter the client ID that you generated from the **Integration** page in NetSuite.
Certificate ID	NetSuite certificate ID that you generated using **OAuth 2.0 Client Credentials**. NetSuite certificate ID that you generated under **Setup** > **Integration** > **Manage Authentication** > **OAuth 2.0 Client Credentials** (M2M) Setup in NetSuite.
Instance URL	NetSuite applications suite instance URL. Enter the instance URL in the following format: `https://[accountid].suitetalk.api.netsuite.com`
Certificate Private Key	NetSuite certificate private key. Enter the PKCS8 certificate as a Base64-encoded string in the following format: `-----BEGIN PRIVATE KEY----- ..... -----END PRIVATE KEY-----`

5. Save and publish the connection.

Configuring and publishing the Salesforce connection

To configure and publish the Salesforce connection, perform the following steps:

1. Open the **SalesforceConnectionCase** connection.
2. In the **Run On** field, select Cloud Server or any Secure Agent.
3. In the **Connection Properties** section, the **Object Filter** field value is **Case**.
4. In the **Authentication Type** field, select **OAuth** or **Password** as required.

   Based on the authentication type selected, perform one of the following steps:

   - For **Password** authentication:
     1. Enter values for the following properties:

        •User Name: Salesforce developer account user name.

        •Password: Salesforce developer account password.

        •Security Token: Salesforce security token.
   - For **OAuth** authentication, enter values for the following properties:

     - Authentication URL: Endpoint used to make OAuth authorization requests to Salesforce.
     - Token Request URL: Endpoint used to make OAuth token requests to Salesforce.
     - Session Duration: Number of minutes after which the OAuth token expires.
     - Authorize access: Click **Authorize** > enter Salesforce developer account credentials > click **Allow**.

     The OAuth authentication process starts. You can check the current authorization status in the **Authorization Status** property.
5. Save and publish the connection.

Configuring and publishing the process

1. Open the **NetSuite to Salesforce (Case to Case)** process. If you do not make changes to the process, you can publish it without opening the process.
2. On the **Start** tab of the Start step, select **Cloud Server** from the **Run On** list.
3. Optionally, you can change the tracing level from **Verbose** to **None** on the **Advanced** tab.
4. Save and publish the process.

Invoking the process

You must create a script to generate the GET request that calls the process.

To invoke a process, you must configure the script and include the process URL. Trigger the process, which is called by a web request from the NetSuite script when a **Case** entity is created or updated. The process takes the Case ID from the request, calls NetSuite to get all data from the **Case** entity. The process tries to find an account in Salesforce by NetSuite_ID based on the value of the account ID field. If a match is found, it updates the Salesforce account; if not, it creates a new account in Salesforce.

For **Contact** entities, the process follows the same steps as for **Account** entities, with the addition of creating an account to which the contact is associated. If no contact is specified, the process creates one with the first name (Primary Contact) and last name (Company Name) and uses data from the account.

The process attempts to find a Case in Salesforce by NetSuite_ID based on the value of the Case ID field. If a match is found, it updates the Salesforce Case; if not, it creates a new case in Salesforce.

Test data synchronization from NetSuite cases to Salesforce cases

When you invoke the process, the request searches for cases that were created or updated. The NetSuite script triggers the process, and the details are synchronized with the Salesforce cases.

The following table shows the fields that are synchronized between the NetSuite cases and the Salesforce cases for the **Case** entity:

NetSuite - Case	Salesforce - Case
Subject	Subject
Status	Status
Priority	Priority
Origin	Case Origin
Company	Account
Contact	Contact
Email	Web Email
Phone	Web Phone
ID	NetSuite_ID

The following table shows the fields that are synchronized between the NetSuite cases and the Salesforce cases for the **Account** entity:

NetSuite - Account	Salesforce - Account
Customer ID	Account Name
Phone	Web Phone
ID	NetSuite_ID

NetSuite - Account	Salesforce - Account
Web addresses	Web Name
Address	Address

For **Contact** entities, if the contact information is not available in NetSuite case, the process creates one with the first name (Primary Contact) and last name (Company Name) and uses data from **Account** in NetSuite. Enter the email and phone details. In a Salesforce Case, **Contact Email** and **Contact Phone** are populated from the **Contact** field. In NetSuite, these fields are populated from the contact or from the company if no contact is specified. The contact field is optional. Salesforce only allows these values to be populated from the contact, not from the account.

The following table shows the fields that are synchronized between the NetSuite cases and the Salesforce cases for the **Contact** entity:

NetSuite - Contact	Salesforce - Contact
Contact	First name + Last name
Main Phone (from account, if not present)	Phone
Email (from account, if not present)	Email
Job Title	Title
ID	NetSuite_ID
Fax (from account, if not present)	Fax

Rules and guidelines for using the Synchronize NetSuite Cases to Salesforce Cases recipe

Consider the following rules and guidelines when working with the Synchronize NetSuite Cases with Salesforce Cases recipe:

- You must first configure the connections in the recipe and publish them before opening or updating the process. Otherwise, the process will contain empty fields from the connections and will become invalid.
- Informatica recommends that you use the same names configured for the assets in the recipe. If you use the same asset names, you can publish all the assets and synchronize the data from the NetSuite cases with the Salesforce cases without any issue. However, if you change the names, you must ensure that you update the names in the related fields in other assets.
- If the tenant already contains connections with the same name as the connections added from the package, the process in the recipe becomes invalid. This is because the newly added connection name contains the suffix `-2`. For example, `<connection_name>-2`.
  In this case, you must manually reselect the connections with the new name and the event values in the next steps of the process wherever applicable.
- You must copy the process URL and paste it in NetSuite script. For more information, see Sample script.

- Ensure that all name variables with the **Input** parameter are the same in the NetSuite script.