Sitecore Connect for Salesforce CRM 2.1 Installation Guide

How to install and configure Sitecore Connect for Salesforce CRM 2.1

August 14, 2020

Table of Contents

1. Install on a content management or standalone server

This section covers how to install Sitecore Connect for Salesforce on your content management or standalone server.

1.1. Prerequisites

Before you install Sitecore Connect for Salesforce, you must have:

- Sitecore Experience Platform 9.1.
- Access to a Salesforce CRM instance and a user account on Salesforce - at a minimum, this account must have rights to read data from Salesforce. To write data to Salesforce, the account must also have rights to write data to Salesforce.
- Network connectivity - you must install on a Sitecore server that has network connectivity to your Salesforce instance. This can require firewall configuration, based on your network setup.

1.2. Install Sitecore installation packages

Sitecore Connect for Salesforce CRM consists of a number of Sitecore installation packages.

> **NOTE**
> Sitecore does not provide a single installation package because providing separate packages means that we can update individual components more easily.

To install Sitecore installation packages:

1. Download the following installation packages:
    - Data Exchange Framework (DEF) 2.1
    - Salesforce CRM Provider for Data Exchange Framework 2.1 (contains the Salesforce Provider for Data Exchange Framework package)
    - Sitecore Provider for Data Exchange Framework 2.1

- xConnect Provider for Data Exchange Framework 2.1 or 2.2

> **NOTE**
> Sitecore Connect for Salesforce Marketing Cloud 2.0 - Behavioral Data Exchange (SFMC-BDE) requires the version 2.2 of the xConnect Provider for DEF. If you plan on using that connector on the same server as the Sitecore Connect for Salesforce CRM connector, you must use version 2.2 of the xConnect Provider for DEF. You can find it on the Sitecore download site, on the SFMC-BDE 2.0 page.

- Sitecore Connect for Salesforce CRM 2.1 (contains the Connect for Salesforce package).

2. Install Data Exchange Framework.
3. Install Sitecore Provider for Data Exchange Framework.
4. Install xConnect Provider for Data Exchange Framework.
5. Install Salesforce Provider for Data Exchange Framework.
6. Install Connect for Salesforce.

1.3. Deploy the xConnect collection model

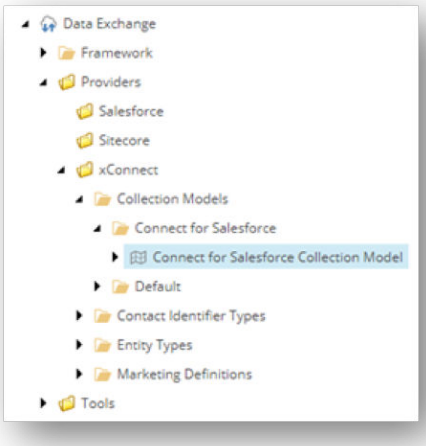
Sitecore Connect adds new facets to associate information from Salesforce with Sitecore contacts and interactions. You must deploy a collection model that defines these new facets to xConnect.

> **NOTE**
> For more information on how to deploy a custom collection model to xConnect, see the xConnect developer documentation.

To deploy the xConnect collection model:

1. In Sitecore, open the Content Editor.

2. Navigate to `sitecore/system/Settings/Data Exchange/Providers/xConnect/Collection Models/Connect for Salesforce/Connect for Salesforce Collection Model`.



3. On the ribbon, click **Convert Model to JSON**.

4. Your browser downloads a JSON file. Save this file to your local machine.
5. Deploy the JSON file to your xConnect Collection server, in the `\App_Data\Models` folder.
6. Deploy the JSON file to your xConnect indexing server. You have to copy the JSON file to two folders:
   - `\App_Data\Models\`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Models\`

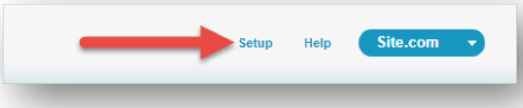
1.4. Configure the Salesforce Connected App

To make Salesforce Connect communicate with Salesforce, you must configure the Salesforce Connected app.

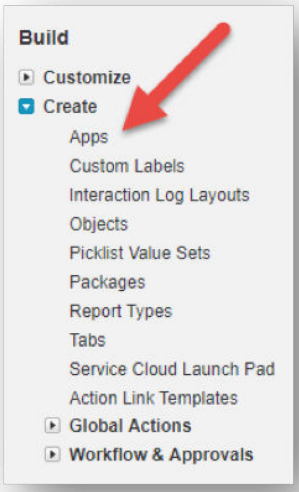
> **NOTE**
> These instructions assume you are using the Salesforce Classic user interface. The same configuration is available using the Salesforce Lightning user interface. We recommend you switch to the Salesforce Classic user interface temporarily to complete these steps.

To configure the Salesforce Connected app:

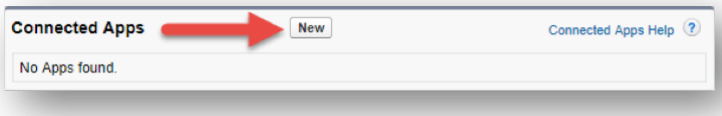
1. Log in to your Salesforce account.
2. On the top menu, click Setup.



3. In the left menu, click **Create** and then click **Apps**.



4. In the **Connected Apps** section, click **New**.

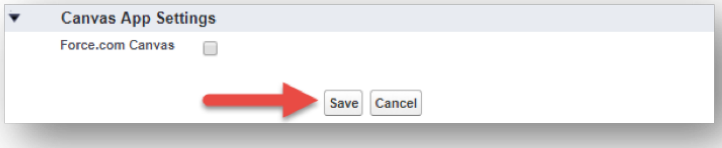


5. Enter the following values:

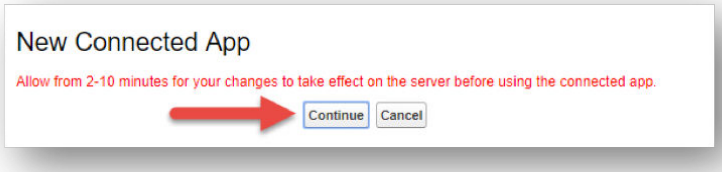
Field name	Value	Notes
Connected App Name	Sitecore Data Exchange App	The specific value does not matter.
API Name	Sitecore_Data_Exchange_App	Salesforce generates this value based on the app name. It is recommended that this value not be changed.
Contact Email	[your email address]	The specific value does not matter.
Enable OAuth Settings	Selected	
Callback URL	http://localhost	This value is not used, so the value you enter does not matter.
Selected OAuth Scopes	Access and manage your data (api)	

Field name	Value	Notes
Require Secret for Web Server Flow	Selected	

6. At the bottom of the screen, click **Save**.



7. Salesforce informs you that a new connected app was created. Click **Continue**.



1.5. Add a custom field in Salesforce (optional)

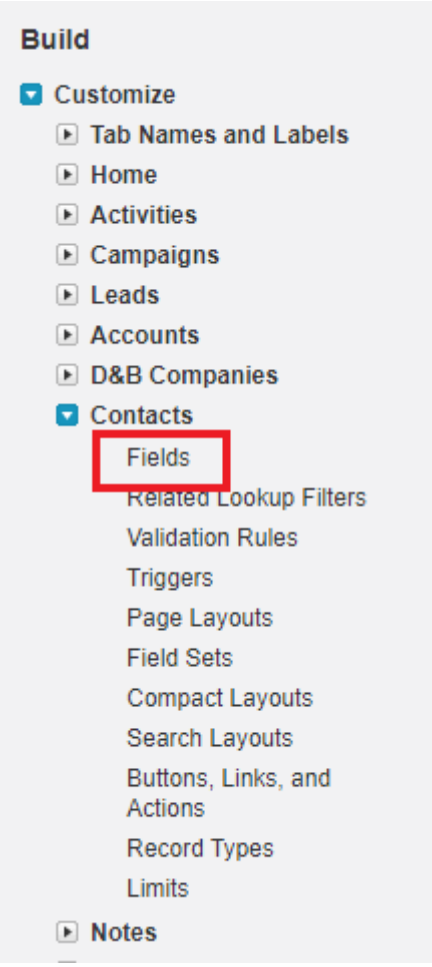
If you plan to write contact data from Sitecore to Salesforce, you must add a custom field in Salesforce. This field is used to associate the Salesforce contact with the corresponding Sitecore contact.

> **NOTE**
> These instructions assume you are using the Salesforce Classic user interface. The same configuration is available using the Salesforce Lightning user interface. We recommend you switch to the Salesforce Classic user interface temporarily to complete these steps.

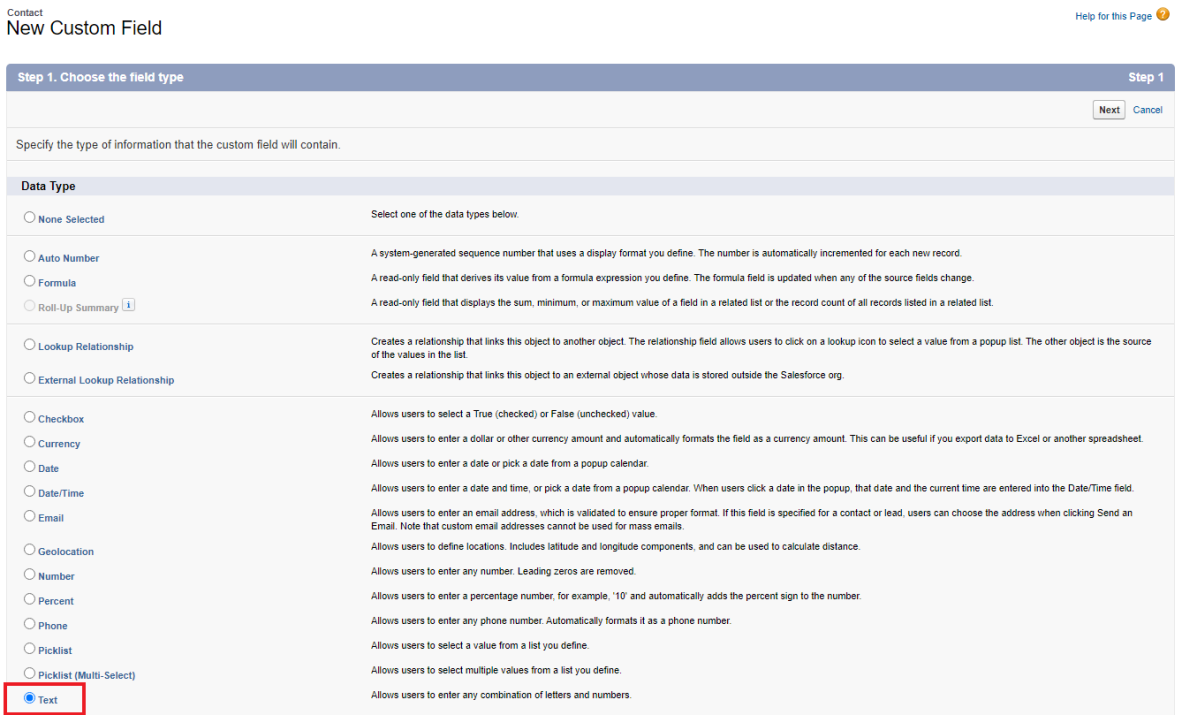
To add a custom field in Salesforce:

1. Log in to your Salesforce account. On the top menu, click **Setup**.

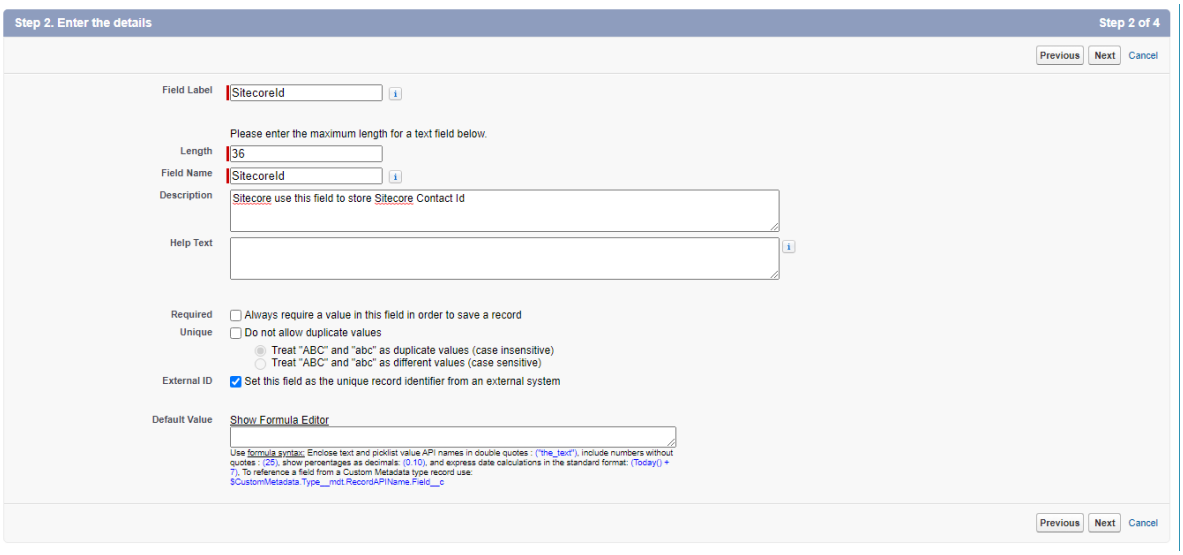
2. In the left menu, open the `Build/Customize/Contacts` node. Click **Fields**.



3. In the **Contact Custom Fields & Relationships** section, click **New**. In the **New Custom Field** dialog, select **Text** as the data type. Click **Next**.



4. In the **Enter the details** dialog, enter the following values:
   - **Data Type** - *Text*
   - **Field Label** - *Sitecore Id*
   - **Length** - *36*
   - **Field Name** - *SitecoreId*
   - **External ID** - selected



5. Click **Next**. In the **Establish field-level security** dialog, click **Next**. In the **Add to page layouts** dialog, click **Save**.

6. In the left menu, open the `Build/Customize/Activities` node. Click **Activity Custom Fields**.
7. In the **Activity Custom Fields** section, click **New**. In the **New Custom Field** dialog, select **Text** as the data type. Click **Next**.
8. In the **Enter the details** dialog, enter the following values:
   - **Data Type** - *Text*
   - **Field Label** - *SitecoreId*
   - **Length** - *36*
   - **Field Name** - *SitecoreId*
   - **Unique** - selected
   - **Treat "ABC" and "abc" as duplicate values (case insensitive)** - selected
   - **External ID** - selected
9. Click **Next**. In the **Establish field-level security** dialog, click **Next**. In the **Add to page layouts** dialog, click **Save**.

1.5.1. Include anonymous contacts

By default, synchronization does not include anonymous contacts. To enable synchronization of anonymous contacts, you must:

1. In the `wwwroot\<sitecoreInstanceName>_xconnect\App_Data\jobs\continuous\IndexWorker\App_data\config\sitecore\SearchIndexer\` folder, open the `sc.Xdb.Collection.IndexerSettings.xml` file.
2. Locate the `Sitecore\XConnect\SearchIndexer\Services\IndexerSettings\Options\IndexAnonymousContactData` node, and set it to *true*.
3. Save the configuration file.

2. Install on a content delivery server

The connector includes extra features that you can use on your Sitecore content delivery (CD) server, such as personalization rules or Forms integration. In order to use these features, you must install Sitecore Connect for Salesforce CRM on your CD server.

> **NOTE**
> You only need to install the connector on your CD server if you want to use the extra features.

2.1. Prerequisites

Before you install Sitecore Connect for Salesforce CRM on your CD server, you must have:

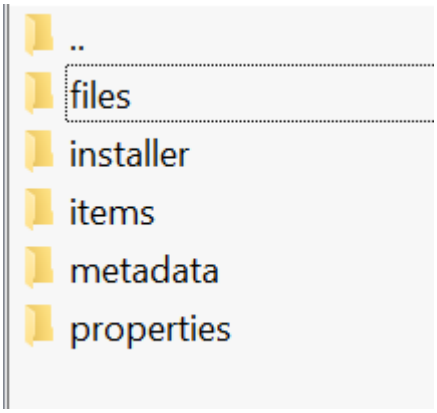
- Sitecore Experience Platform 9.1 or greater
- Data Exchange Framework 2.1.

2.2. Installation

The CD server does not have an interface to install Sitecore packages, so you must install the necessary files manually. To do so:

1. Download the following installation packages:
    - SQL Provider for Data Exchange Framework CD server
    - Sitecore Provider for Data Exchange Framework CD server
    - xConnect Provider for Data Exchange Framework CD server
    - Salesforce CRM Provider for Data Exchange Framework CD server
    - Sitecore Connect for Salesforce CRM CD server

2. Note that each installation package contains a `package.zip` file. The files in the `package.zip` files are in the following structure:



3. For each installation package, extract the files in the `files` folder into the website root folder of your CD server.

> **NOTE**
> When you extract the files you must preserve the folder structure from the `files` folder. For example, the `files` folder in the zip contains a subfolder named `bin`, and the files in this folder must go in the `<website>\bin` folder on your CD server.

4. Restart the CD server.

3. Upgrade a content management or standalone server

This section describes how to upgrade from Sitecore Connect for Salesforce 2.0.1 to 2.1.

3.1. Prerequisites

Before you upgrade Sitecore Connect for Salesforce, you must have the following:

- Sitecore Experience Platform 9.1 (or greater).
- Sitecore Connect for Salesforce 2.0.1 (installed).

3.2. Install Sitecore installation and update packages

Sitecore Connect for Salesforce consists of a number of Sitecore installation and update packages.

> **NOTE**
> Sitecore does not provide a single installation package because providing separate packages means that we can update individual components more easily.

To install Sitecore installation and update packages:

1. Download the following update packages:
   - Data Exchange Framework 2.1
   - Sitecore Provider for Data Exchange Framework 2.1
   - xConnect Provider for Data Exchange Framework 2.1
   - Salesforce CRM Provider for Data Exchange Framework 2.1, which contains the `Salesforce Provider for Data Exchange Framework 2.1.0.update` file
   - Sitecore Connect for Salesforce 2.1, which contains the `Connect for Salesforce 2.1,0.update` file
2. Download the following installation package:
   - Data Exchange Framework SDK 2.1
3. Install the update package for Data Exchange Framework 2.1.
4. Install the update package for Sitecore Provider for Data Exchange Framework 2.1.

5. Install the update package for xConnect Provider for Data Exchange Framework 2.1.
6. Install the update package for Salesforce Provider for Data Exchange Framework 2.1.
7. Install the update package for Connect for Salesforce 2.1.
8. Install Data Exchange Framework SDK 2.1.

> **NOTE**
> After you have installed the update packages, you must deploy the xConnect collection model.

3.3. Upgrade the existing tenants

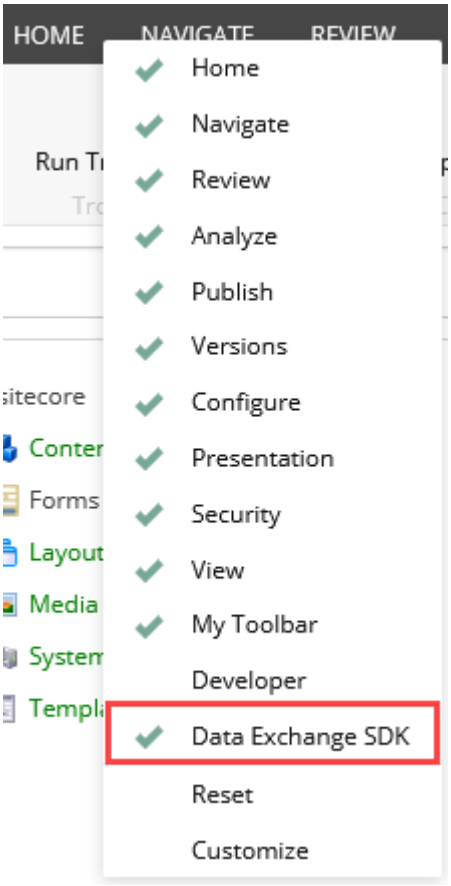
To take advantage of new and improved functionality, you must upgrade existing tenants.

> **NOTE**
> These steps involve using a tool to automatically change items in your tenant. The upgrade tool makes a backup of the tenant before making any changes. However, we recommend that you create your own backup before using the tool.

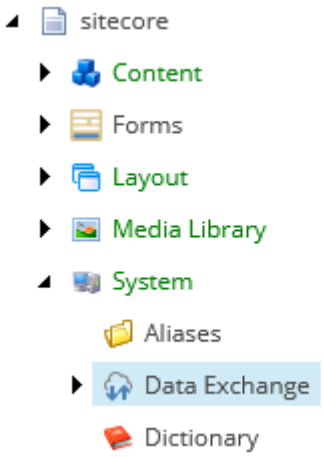
To upgrade the existing tenant:

1. Download the update definition file for Sitecore Connect for Salesforce CRM.
2. Copy the update definition file to the `...\App_Config\Sitecore\SalesforceConnect` folder on your Sitecore server.

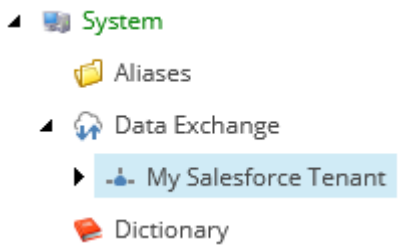
3. In the Content Editor, add the tab **DATA EXCHANGE SDK**.



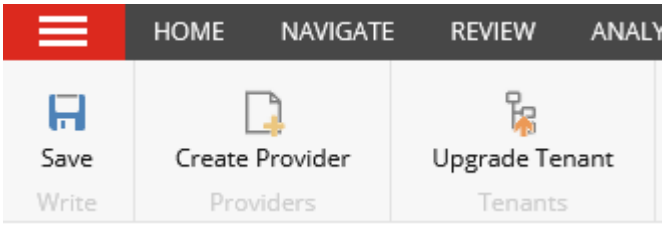
4. Navigate to `sitecore/system/Data Exchange`.



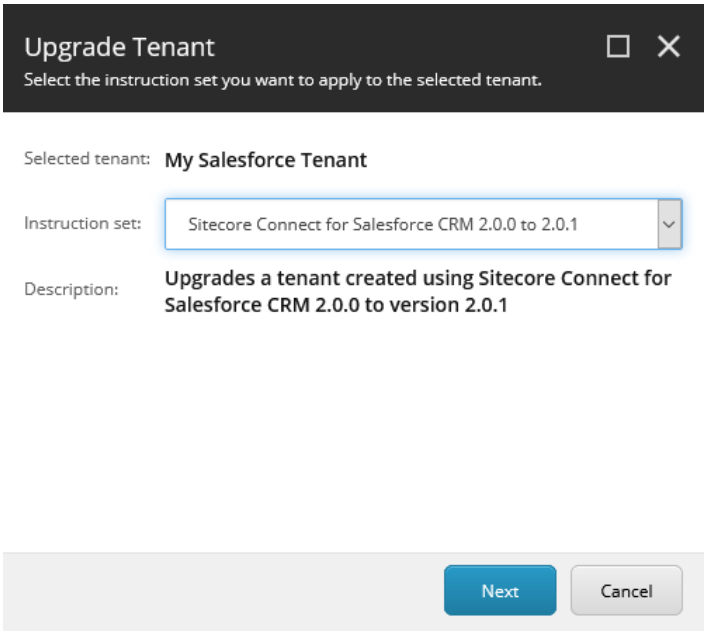
5. Select the tenant that you want to upgrade.



6. On the ribbon, on the **DATA EXCHANGE SDK** tab, click **Upgrade Tenant**.

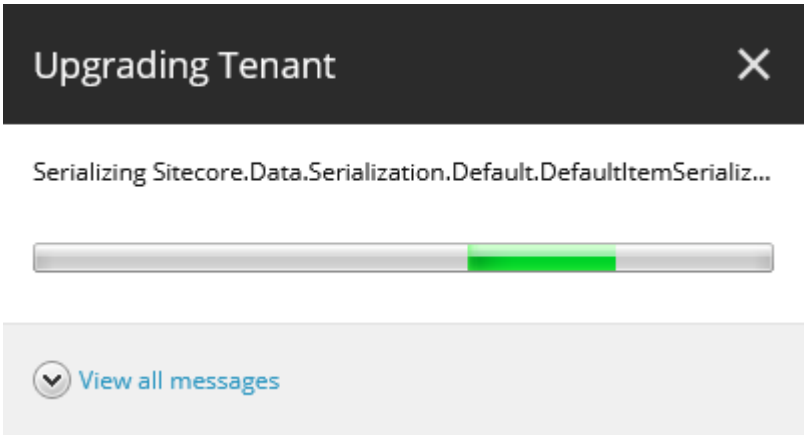


7. In the **Upgrade Tenant** dialog box, in the **Instruction set** field, click **Sitecore Connect for Salesforce 2.0.0 to 2.0.1**.

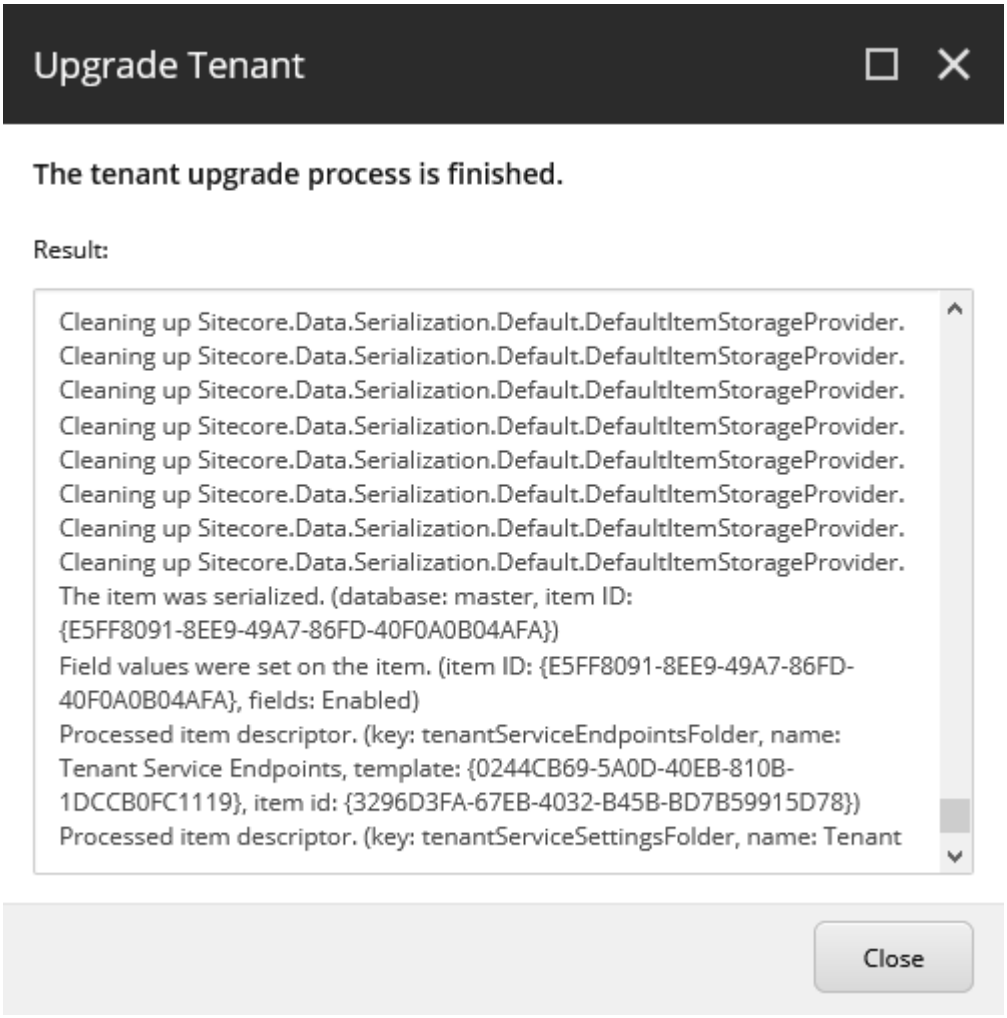


8. Click **Next**.
9. Read the information screen carefully. When you are ready to start the upgrade process, click **Start upgrade**.

A status box displays the progress.



10. When the upgrade process is finished, a summary of the upgrade process is displayed. Click **Close**.



NOTE
After the tenant is upgraded, you must manually re-enable the tenant before you can run any pipeline batches.

4. Upgrade a content delivery server

This section explains how to upgrade Sitecore Connect for Salesforce on your content delivery server.

4.1. Prerequisites

Before you upgrade Sitecore Connect for Salesforce, you must have the following:

- Sitecore Experience Platform 9.1 (or greater)
- Data Exchange Framework 2.1
- Sitecore Connect for Salesforce 2.0.1.

4.2. Installation

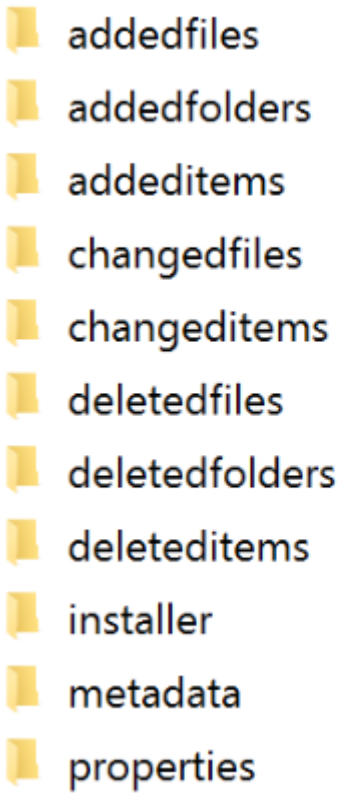
The CD server does not have an interface to install Sitecore packages, so you must update the files manually.

> **NOTE**
> For all of the following file operations you must preserve the folder structure from the zip file folders. For example, if the `changedfiles` folder in the zip file contains a `bin` subfolder, the files from this folder must go in the `<website>\bin` folder on your CD server.

To perform the upgrade:

1. Download the following update packages:
    - Update Package for Salesforce CRM Provider for Data Exchange Framework
    - Update Package for Sitecore Provider for Data Exchange Framework
    - Update Package for SQL Provider for Data Exchange Framework
    - Update Package for xConnect Provider for Data Exchange Framework
    - Update Package for Sitecore Connect for Salesforce CRM

2. Note that each update package file contains a `package.zip` file. The files in the `package.zip` file are in the following structure:



> **NOTE**
> The actual update package may not contain all of the folders shown in the image.

3. For each update package:
   - Add the folders in the `addedfolder` folder into the website root folder of your CD server.
   - Add the files in the `addedfiles` folder into the website root folder of your CD server.
   - Delete the folders in the `deletedfolder` folder from the website root folder of your CD server.
   - Delete the files in the `deletedfiles` folder from the website root folder of your CD server.
   - Add the files in the `changedfiles` folder into the website root folder of your CD server, overwriting the existing files.
4. Restart the CD server.

5. Configuration

This section covers basic configuration options for Sitecore Connect for Salesforce.

5.1. Get connection string values from Salesforce

Before you can configure the connection string, you must collect the following values from Salesforce:

- **User ID** - the ID that Sitecore uses to call the Salesforce API. This user ID does not have to have administrator rights, but it must have sufficient rights to perform the activities you expect to be able to from Sitecore. For example:
  - To have Salesforce contacts created in Sitecore, the user must have read-access on contacts and campaigns.
  - To push contact data from Sitecore into Salesforce, the user must have write-access on contacts.
- **Password** - the password for the Salesforce user that Sitecore uses to call the Salesforce API.

> **NOTE**
> The password must not contain the ; character (semicolon).

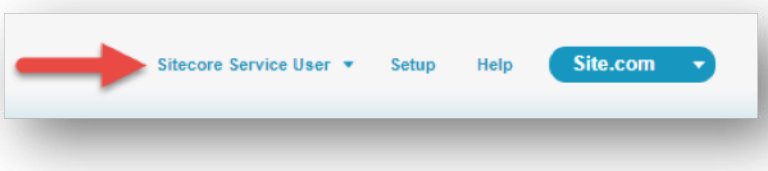
- **Security token**
- **Client ID and secret key**

5.1.1. Get the security token value

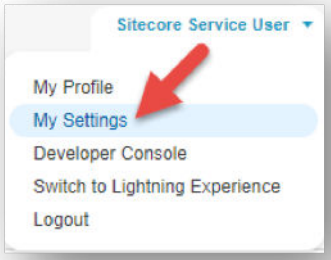
The security token value comes from the Salesforce user who used the connected app.

To get the security token:

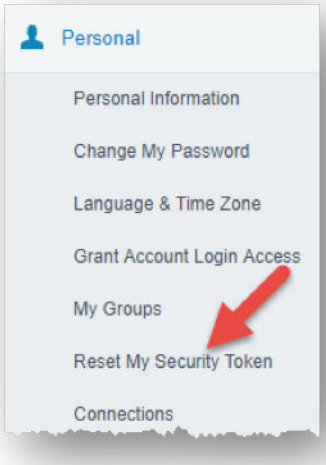
1. In Salesforce, log in with the user ID you plan to use in the connection string.
2. In the top menu, click the user name.



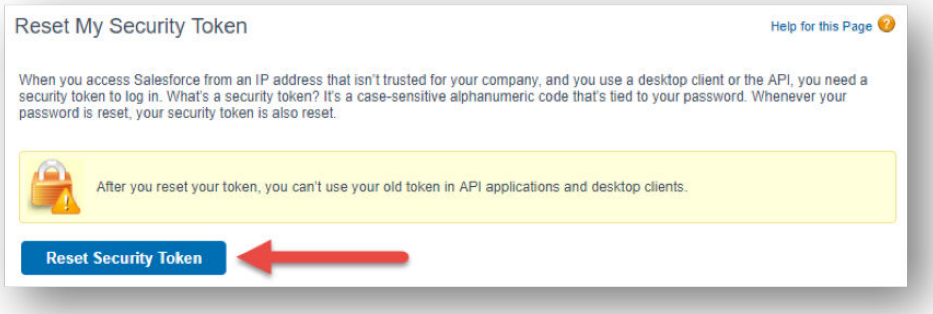
3. Click **My Settings**.



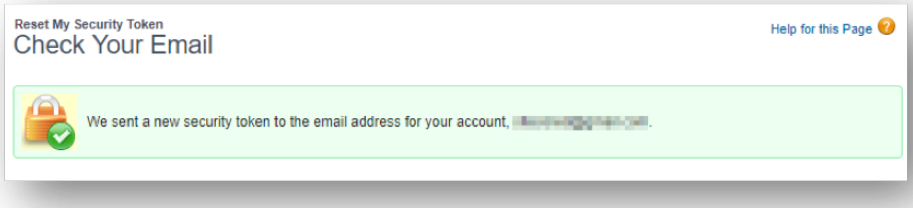
4. In the left menu, click **Personal**, **Reset My Security Token**.



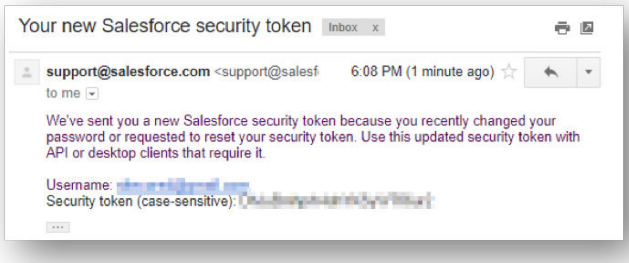
5. Read the warning on the screen carefully and then click **Reset Security Token**.



Salesforce informs you that the new security token will be emailed to you.



6. Check your inbox to find the email message with the new security token value.

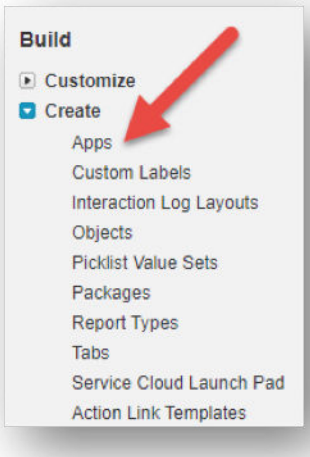


5.1.2. Get the client key and secret key values

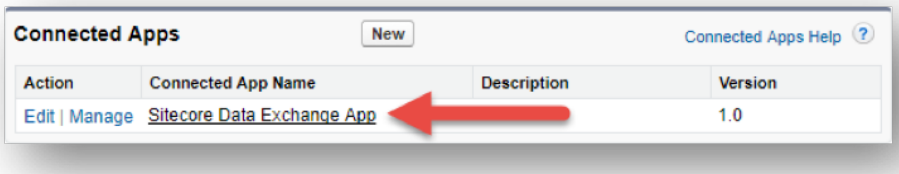
The client key and secret key values come from the Salesforce connected app.

To get these values:

1. In Salesforce, in the left menu, in the **Build** section, expand **Create** and the click **Apps**.



2. In the **Connected Apps** section, click the name of the connected app you created.



3. In the **API (Enable OAuth Settings)** section, copy the values for the **Consumer Key** and **Consumer Secret**.

> **NOTE**
> In the connection string, the Consumer Key value is called the client id, and the Consumer Secret value is called the secret key.

5.2. Add the connection string to Sitecore

When you have collected the connection string values from Salesforce, you must add the connection string to Sitecore.

To add the connection string to Sitecore:

- Add the following connection string to the `ConnectionStrings.config` file on your Sitecore server, replacing the values in brackets with the values you collected from Salesforce:

```
<add name="mysf" connectionString="user id=[user id];password=[password];client id=[client id];secret key=[secret key];security token=[security token]" />
```

The following is an example of what the connection string looks like after you have replaced the values in brackets:

```
<add name="mysf" connectionString="user id=someone@email.com;password=b;client id=GEH9zlTNB8o8BA45pAeDtC8W.DIqrAzuky2uffEEOwtHxIEhLzkmPwjz8KF_vzWY8dNIfurWHpfbZPGdtc3b;secret key=5468568999798354123;security token=g3ygFuNzGgm33YTfsM3WKG3AA" />
```

> **NOTE**
> If you connect to a Salesforce sandbox, you must add the parameter `sandbox=true` to the connection string.

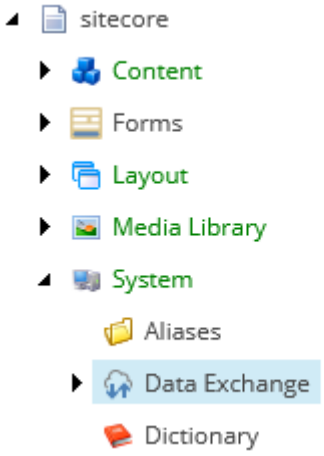
5.3. Create a tenant

A tenant is the basic unit of configuration in Sitecore Connect for Salesforce. All of the settings that determine how data between Salesforce and Sitecore is synchronized are contained in the tenant.

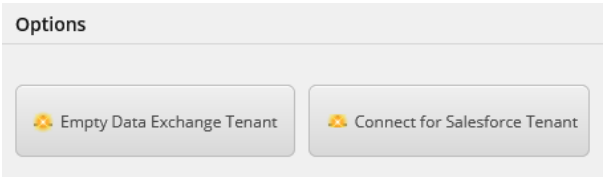
> **NOTE**
> Usually, a tenant represents a specific Salesforce instance. You can configure multiple tenants if you have multiple Salesforce instances you want to connect. It is also possible to configure a single tenant to connect to multiple Salesforce instances. However, that configuration is not covered in this guide.

To create a tenant:

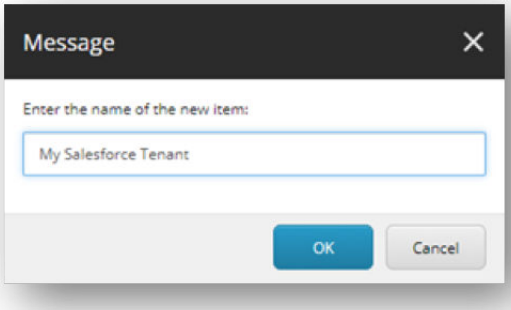
1. In the Content Editor, navigate to `sitecore/system/Data Exchange`.



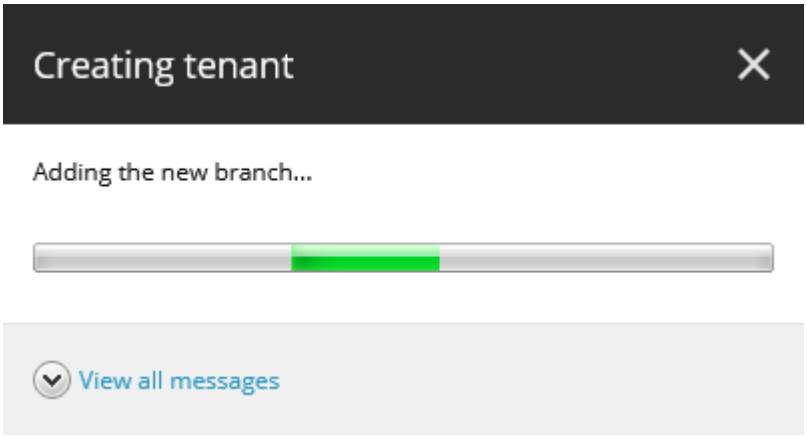
2. To add a new item, click **Connect for Salesforce Tenant**.



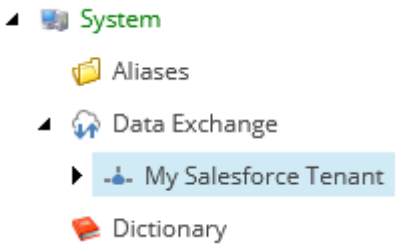
3. Enter a name for the tenant and click **OK**.



Creating the tenant takes some time. As it is created, a progress box appears.



After the tenant is created it appears under `sitecore/System/Data Exchange`.

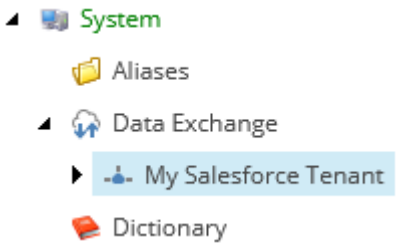


5.4. Enable the tenant

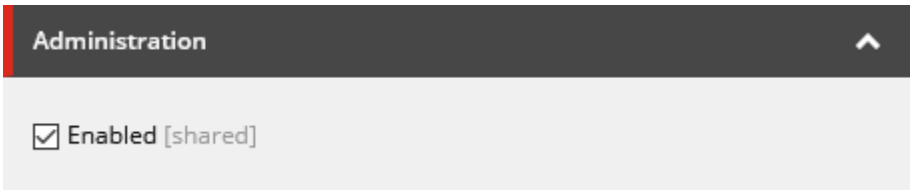
By default, the tenant is disabled. This is to ensure that no synchronization process is run until an administrator makes a conscious decision to enable the tenant.

To enable the tenant:

1. In the Content Editor, select your tenant.



2. On the **Content** tab, in the **Administration** section, select the **Enabled** check box.



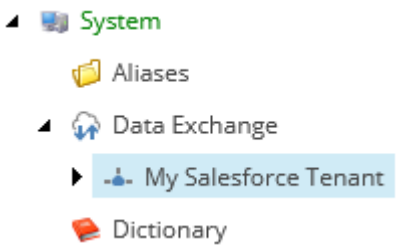
3. Save the item.

5.5. Configure endpoints

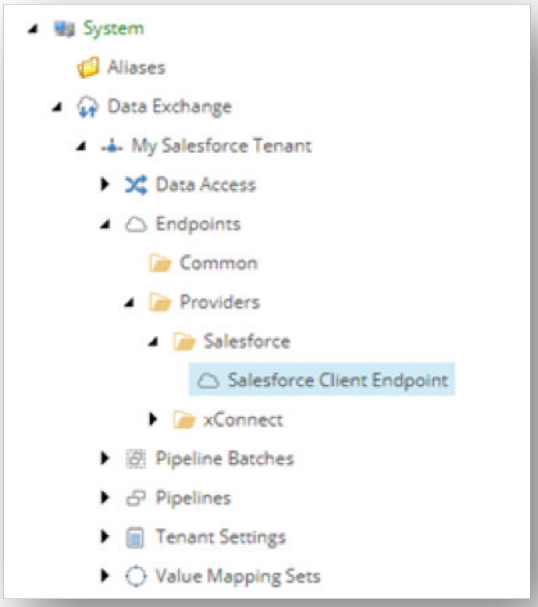
Endpoints are used to identify the source of data that is read and the target of data that is written. Sitecore Connect for Salesforce uses endpoints to represent Sitecore and a Salesforce instance.

To configure endpoints:

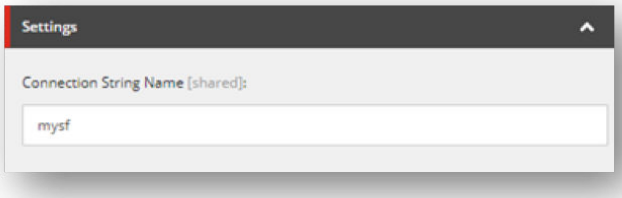
1. In the Content Editor, select your tenant.



2. Navigate to `Endpoints/Providers/Salesforce/Salesforce Client Endpoint`.

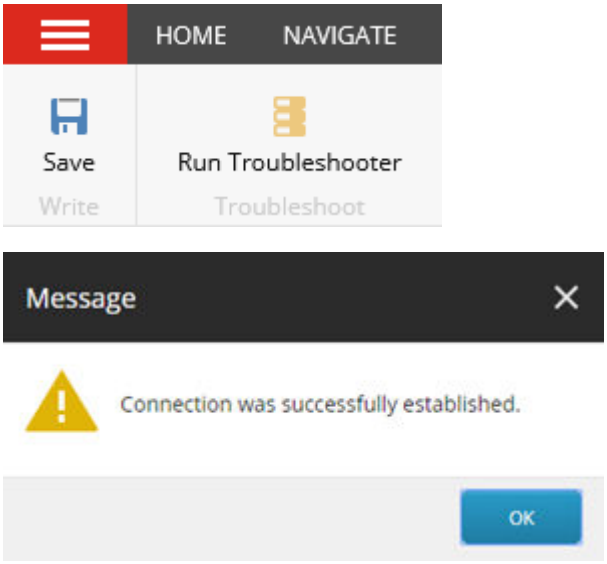


3. In the **Connection String Name** field, enter the name of the connection string that you added previously.

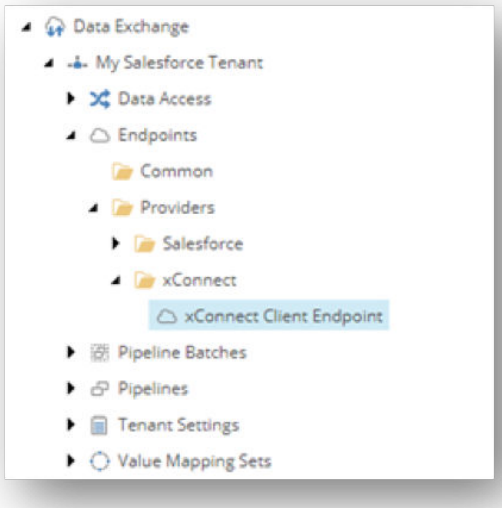


> **NOTE**
> If you enter a connection string that is not defined on your Sitecore server, a validation error is displayed.

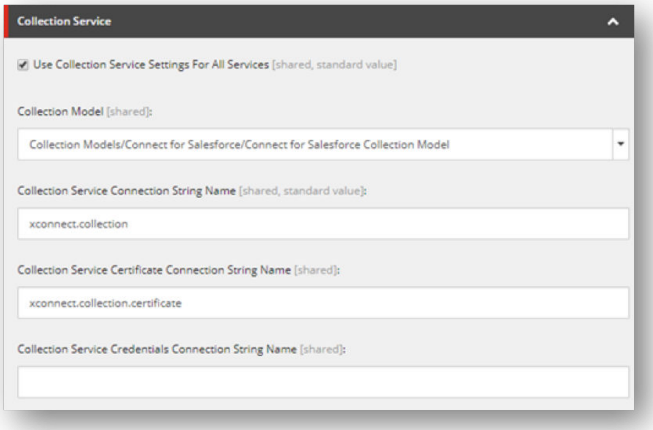
4. Save the item. On the ribbon, in the **Data Exchange** tab, click **Run Troubleshooter**, and in the message that appears to indicate that a connection could be established to your Salesforce instance, click **OK**.



5. Select your tenant and navigate to `Endpoints/Providers/xConnect/xConnect Client Endpoint`.



6. In the **Collection Service** section, enter values for the following fields:

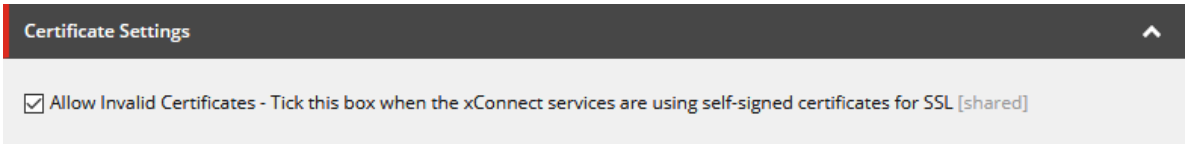


> **NOTE**
> All of these values are connection string names from the `ConnectionStrings.config` file on your Sitecore server.

Field name	When required	Description
Collection Service Connection String Name	Always required.	Connection string that identifies the xConnect collection service. By default the value is `xconnect.collection`.
Collection Service Certificate Connection String Name	Required if SSL is enabled on the xConnect server.	Connection string that identifies the certificate to use when establishing a secure connection to the xConnect server. By default the value is `xconnect.collection.certificate`.

Field name	When required	Description
Collection Service Credentials Connection String Name	Required if authentication is enabled on the xConnect server.	Connection string that specifies the credentials to use to connect to an xConnect server with authentication enabled. Authentication is not enabled on the xConnect server by default. For more information on supported formats for this connection string, see the Data Exchange Framework developer documentation.

7. If your xConnect server is using a self-signed certificate for SSL, select the **Allow Invalid Certificates** check box.



8. Save the item. Click **Run Troubleshooter**. A message appears indicating if a connection could be established. Click **OK** to close the message.
9. In the same tenant, navigate to `Endpoints/Providers/xConnect/xDB Reference Data Client Endpoint`.
10. In the **Settings** section, in the **Client Connection String** field, enter `xdb.referencedata.client`. In the **Client Certificate Connection String**, enter `xdb.referencedata.client.certificate`.
11. Save the item. Click **Run Troubleshooter**. A message appears indicating if a connection could be established. Click **OK** to close the message.

If you are planning to write contact data from Sitecore to Salesforce, you must add a custom field. This field is used to associate the Salesforce contact with the corresponding Sitecore contact.

5.6. Run pipeline batches

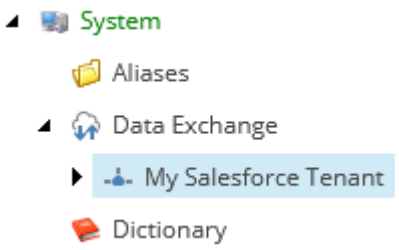
You execute the data synchronization by running the tenant pipeline batches. By default, the pipeline batches are not scheduled to run. For more information on how to run pipeline batches, please refer to the developer documentation for the Sitecore Data Exchange Framework.

5.7. Enable a Salesforce contact custom field (optional)

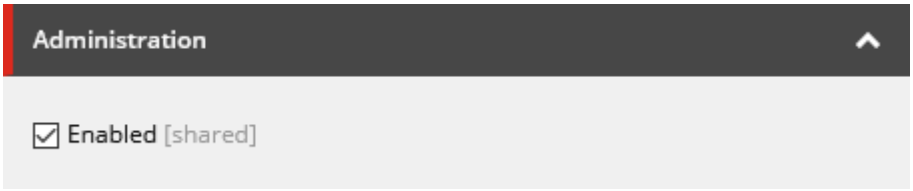
If you are planning to write contact data from Sitecore to Salesforce, you must add a custom field to Salesforce. You must enable this field on the tenant.

To enable a Salesforce contact custom field:

1. In the Content Editor, select your tenant.



2. Navigate to `Data Access/Value Accessor Sets/Providers/Salesforce/Salesforce Contact Fields/Sitecore Id on Salesforce Contact`.
3. In the **Administration** section, select the **Enabled** check box.



4. Save the item.

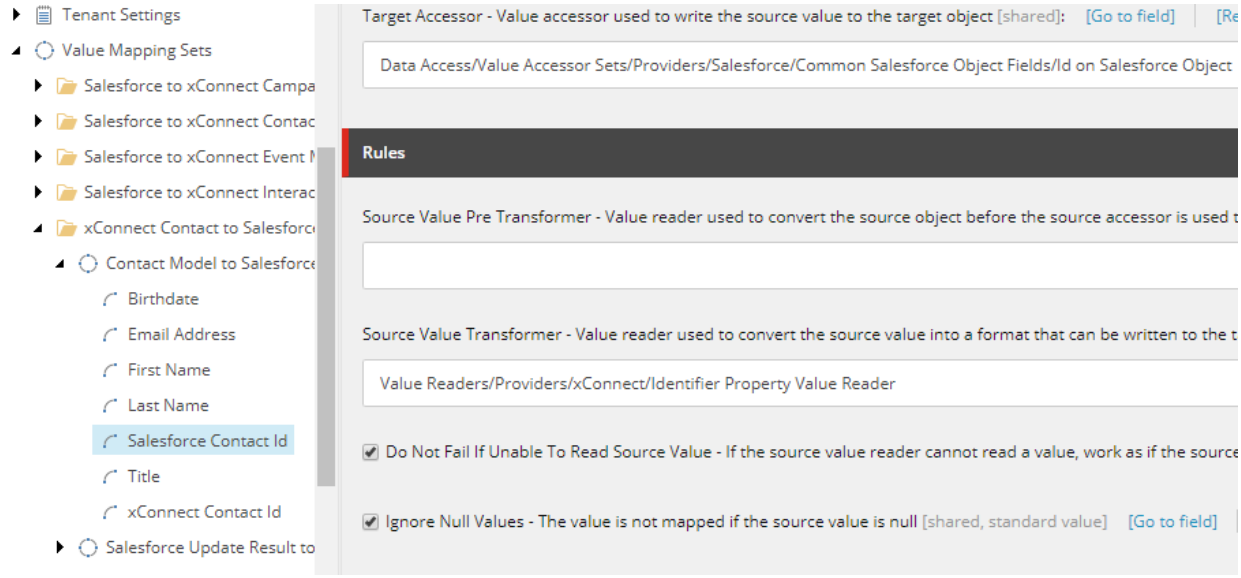
5.8. Prepare an xConnect to Salesforce synchronization

If you synchronize from xConnect to Salesforce CRM, the synchronization might fail the first time you run it, because contacts in xConnect have not yet been assigned a Salesforce Contact ID. To avoid this error, you must enable the *Do Not Fail If Unable To Read Source Value* setting before synchronizing.

To enable the *Do Not Fail* setting:

1. On the Sitecore Launchpad, open the Content Editor.
2. Navigate to the `/sitecore/system/Data Exchange/<tenant>/Value Mapping Sets/xConnect Contact to Salesforce Contact Mappings/Contact Model to Salesforce Contact/Salesforce Contact Id` node.

3. In the **Rules** section, select the **Do Not Fail If Unable To Read Source Value** check box:



5.9. Enable indexing for PII sensitive fields

If you synchronize from xConnect to Salesforce CRM, you must enable indexing for personally identifiable information (PII) sensitive fields. The `Read Contacts from xConnect` pipeline step searches **FirstName** and **Email** fields to identify which contacts to process. By default, xConnect does not index these fields for privacy reasons.

> **NOTE**
> Before enabling indexing on PII sensitive fields, we encourage you to review the security and data protection features of your chosen search provider (Azure Search or Solr).
>
> If you synchronize from Salesforce CRM to xConnect, you do not have to enable indexing for PII sensitive fields.

To enable indexing:

1. In the `wwwroot\<sitecoreInstanceName>_xconnect\App_Data\Config\Sitecore\SearchIndexer\` folder, open the `sc.Xdb.Collection.IndexerSettings.xml` file.
2. Locate the `Sitecore\XConnect\SearchIndexer\Services\IndexerSettings\Options\IndexPIISensitiveData` node, and set it to *true*:

```
<IndexerSettings>
    <Type>Sitecore.Xdb.Collection.Indexing.IndexerSettings, Sitecore.Xdb.Collection</Type>
    <LifeTime>Singleton</LifeTime>
    <Options>
        <!-- Indexer will split change set on chunks to improve memory consumption. Setting this option to 0, a negative value or removing the element completely, results in no splitting.-->
        <SplitRecordsThreshold>25000</SplitRecordsThreshold>
```

```
        <IndexPIISensitiveData>true</IndexPIISensitiveData>
        <IndexAnonymousContactData>false</IndexAnonymousContactData>
    </Options>
</IndexerSettings>
```

3. In the `wwwroot\<sitecoreInstanceName>_xconnect\App_Data\jobs\continuous\IndexWorker\App_data\config\sitecore\SearchIndexer\` folder, open the `sc.Xdb.Collection.IndexerSettings.xml` file.
4. Locate the `Sitecore\XConnect\SearchIndexer\Services\IndexerSettings\Options\IndexPIISensitiveData` node, and set it to *true*.