



# **ForceGento**

## **(Magento Salesforce Connector)**

### **Installation Guide**

#### **Version 0.0.1**

#### **Table of Contents**

1. ForceGento Connector
2. Use Cases
3. Magento Package Code Installation
4. Generate Token Key in Salesforce
5. Install Salesforce Package
6. Contact Us form configuration setting in Magento
7. Salesforce Configuration in Magento
8. Mapping Magento and Salesforce Attributes
  1. Map Magento Contact Attribute to respective Salesforce Lead Fields
  2. Map Magento Customer Attribute to respective Salesforce Account Fields
  3. Map Magento Customer Attribute to respective Salesforce Contact Fields
  4. Map Magento Product Attribute to respective Salesforce Product Fields
  5. Map Magento Product Attribute to respective Salesforce PriceBook Fields
  6. Map Magento Order Attribute to respective Salesforce Opportunity Fields
  7. Map Magento Order Attribute to respective Salesforce PriceBook2 Fields
9. Sync Process
  1. Customer Configuration in Magento
  2. Order Configuration in Magento
10. Decode Source Guardian Code
11. Uninstall ForceGento Connect

#### **1. ForceGento Connector**

Magento is a Enterprise level ecommerce system which has several ecommerce features like Customers management, Orders management, Inventory management, Catalog management, Coupons and promotions, Newsletters and Reports etc. whereas Salesforce is CRM and has the capabilities to manage the customer interaction, call logs, marketing campaigns etc.

ForceGento Connect is an extension that's developed under the TechnoMile Development which helps to unite the Salesforce and Magento with generic feature which are needed for any business who works on CRM and Ecommerce independently.

ForceGento Connect sync the data from Magento to SalesForce and vice versa.

## 2. Use Cases

- Customer has Magento site & Salesforce Org
- Customer has Magento site but has no Salesforce Org
- Customer has no Magento site but has Salesforce Org

## 3. Magento Package Code Installation

1. Unzip the downloaded file (ForceGento\_Connector-0.0.1.tgz)
2. Read the installation document present in the Docs folder.
3. Once the required zip file is downloaded and unzipped, all related files would be available in your local machine.
4. Now it is required to follow instructions 4.1-4.5 mentioned below to copy the necessary files from your local machine to your Magento instance in the server box.
  1. Navigate to **app\code\community** and Place **ForceGento** folder.
  2. Navigate to **app\design\frontend\rwd** and Place **forcegento** folder.
  3. Navigate to **app\design\adminhtml\default\default\template** and Place **forcegento** folder.
  4. Navigate to **app\etc\modules** and Place **ForceGento\_Connector.xml** file.
  5. Navigate to **lib** and Place **soapclient** folder.

## 4. Generate Token Key in Salesforce

Note: To generate enterprise WSDL file follow the below Process

### Generate Enterprise WSDL file

- Go to Setup >> Develop >> API click on the link Generate Enterprise WSDL

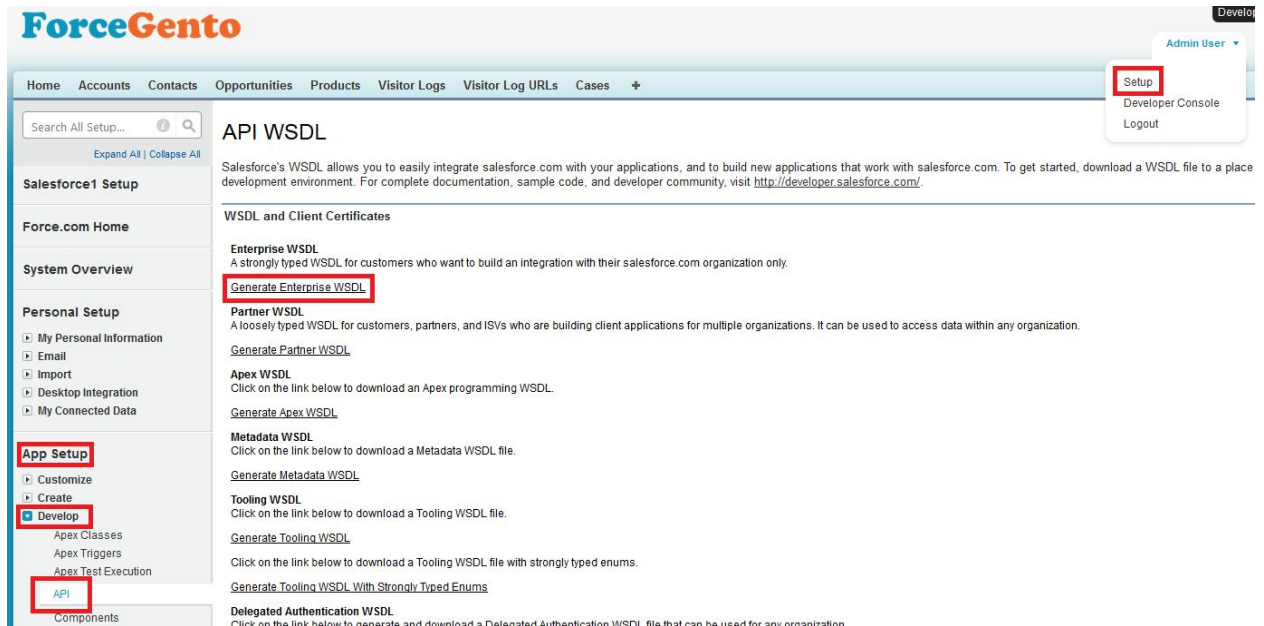
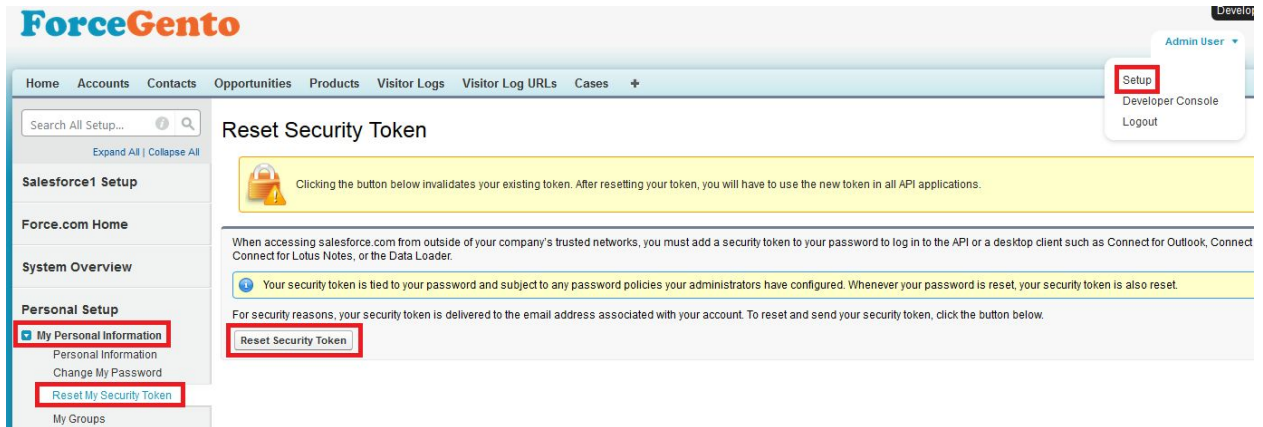


Fig: Generating Enterprise WSDL file

- Generate Token Key.  
Go to My Settings >> Personal >> Reset My Security Token >> click on the Reset Security
- Token button is shown below:



**Fig: Reset Security Token**

## 5. Install Salesforce Package

Copy link below and paste in your browser for installing Salesforce Package on your Salesforce Org.

<https://login.salesforce.com/packaging/installPackage.apexp?p0=04to000000003xMk>

## 6. Contact Us form configuration setting in Magento

**Create Custom Field for Contact Us form in Magento**

1. Login to Magento Admin panel.
2. Go to System >> Transactional Emails >> Add New Template
3. Select 'Contact Form' from Template drop down
4. Click on 'Load Template'.

The screenshot shows the 'New Email Template' page in Magento. At the top right, there are buttons for 'Back', 'Reset', 'Preview Template', and 'Save Template'. The 'Load default template' section has two dropdown menus: 'Template' (set to 'Contact Form') and 'Locale' (set to 'English (United States)'). Below these is a 'Load Template' button. The 'Template Information' section has three text input fields: 'Template Name' (set to 'Lead Contact Form'), 'Template Subject' (set to 'Lead Contact Form'), and 'Template Content' (containing a list of variables: Name, Company, E-mail, Telephone, and Comment). There is also an 'Insert Variable...' button.

**Fig: Add Contact Template**

5. Add “Company: {{var data.company}}” in the template content, to add a new Company field in the form.
6. Click on ‘Save Template’.

## 7. Salesforce Configuration in Magento

### Configure Salesforce API to allow data to sync from Magento to Salesforce

#### Configure Salesforce API Extension

1. Login to Magento Admin panel.
2. Go to System >> Configuration
3. In ‘ForceGento Connector API Extension’ section, click on ‘Salesforce API Configuration’.
4. Enable the API
5. Fill SalesForce username, password, token
6. Specify the path of WSDL file.
7. Select the client type (Enterprise, Partner)

**Fig: Salesforce API Configuration**

8. Save the configurations.
9. This will connect the magento app to SalesForce through SOAP.

10. Add to cart product and register for the License Key from here.  
<http://www.forcegento.com>

## 8. Mapping Magento and Salesforce Attributes

### Map Magento Contact Fields to respective Salesforce Lead Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Lead Mapping'.
2. Map the respective fields.

The screenshot displays the 'Salesforce Lead Mapping' configuration page. On the left is a sidebar with a 'Configuration' menu. Under 'GENERAL', options include General, Web, Design, Currency Setup, Store Email Addresses, Contacts, Reports, and Content Management. Under 'FORCEGENTO CONNECTOR API EXTENSION', options include Salesforce API Configuration, Order Configuration, Customer Configuration, and 'Salesforce Lead Mapping' (which is currently selected). Below this are 'Salesforce Account Mapping', 'Salesforce Product Mapping', and 'Salesforce Opportunity Mapping'. The main content area is titled 'Salesforce Lead Mapping' and contains a 'Mapping to Salesforce Leads' section. This section lists Magento fields on the left and corresponding Salesforce lead fields on the right, each with a dropdown menu and a '[STORE VIEW]' link. The mappings are: First Name to FirstName, Last Name to LastName, Email to Email, Telephone to Phone, Comment to Description, and Company to Company. A 'Save Config' button is located in the top right corner of the main area.

**Fig: Salesforce Lead Mapping**

3. Click 'Save Config' button.

### Map Magento Customer Fields to respective Salesforce Account Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Account Mapping'.
2. Map the respective fields.

Current Configuration Scope: Default Config [Manage Stores](#)

**Configuration**

- GENERAL
  - General
  - Web
  - Design
  - Currency Setup
  - Store Email Addresses
  - Contacts
  - Reports
  - Content Management
- FORCEGENTO CONNECTOR API EXTENSION
  - Salesforce API Configuration
  - Order Configuration
  - Customer Configuration
  - Salesforce Lead Mapping
  - Salesforce Account Mapping**
  - Salesforce Product Mapping
  - Salesforce Opportunity Mapping
- CATALOG
  - Catalog
  - Inventory
  - Google Sitemap
  - RSS Feeds

**Salesforce Account Mapping** [Save Config](#)

**Customer Mapping to Salesforce Accounts**

Name:	Name	[?] [STORE VIEW]
Type:	Type	[?] [STORE VIEW]
Billing Street:	BillingStreet	[?] [STORE VIEW]
Billing City:	BillingCity	[?] [STORE VIEW]
Billing State:	BillingState	[?] [STORE VIEW]
Billing Country:	BillingCountry	[?] [STORE VIEW]
ZipPostal Code:	BillingPostalCode	[?] [STORE VIEW]
Shipping Street:	ShippingStreet	[?] [STORE VIEW]
Shipping City:	ShippingCity	[?] [STORE VIEW]
Shipping State:	ShippingState	[?] [STORE VIEW]
Shipping Country:	ShippingCountry	[?] [STORE VIEW]
ZipPostal Code:	ShippingPostalCode	[?] [STORE VIEW]
Phone:	Phone	[?] [STORE VIEW]

**Fig: Salesforce Account Mapping**

3. Click 'Save Config' button.

## Map Magento Customer Fields to respective Salesforce Contact Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Account Mapping'.
2. Map the respective fields.

Current Configuration Scope: Default Config [Manage Stores](#)

**Configuration**

- GENERAL
  - General
  - Web
  - Design
  - Currency Setup
  - Store Email Addresses
  - Contacts
  - Reports
  - Content Management
- FORCEGENTO CONNECTOR API EXTENSION
  - Salesforce API Configuration
  - Order Configuration
  - Customer Configuration
  - Salesforce Lead Mapping
  - Salesforce Account Mapping**
  - Salesforce Product Mapping
  - Salesforce Opportunity Mapping

**Salesforce Account Mapping** [Save Config](#)

**Customer Mapping to Salesforce Accounts**

**Mapping to Salesforce Contact**

First Name:	FirstName	[?] [STORE VIEW]
Last Name:	LastName	[?] [STORE VIEW]
Email:	Email	[?] [STORE VIEW]
Phone:	HomePhone	[?] [STORE VIEW]

**Fig: Salesforce Contact Mapping**

3. Click 'Save Config' button.

## Map Magento Product Fields to respective Salesforce Product Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Product Mapping'.
2. Map the respective fields.

The screenshot shows the 'Salesforce Product Mapping' configuration page. On the left is a sidebar with a 'Configuration' menu. The 'FORCEGENTO CONNECTOR API EXTENSION' section is expanded, and 'Salesforce Product Mapping' is selected. The main area is titled 'Salesforce Product Mapping' and contains a 'Mapping to Salesforce Product2' table. The table has two columns: 'Name' and 'Value'. The 'Name' column lists various product attributes, and the 'Value' column shows the corresponding Salesforce Product2 field selected from a dropdown menu. A 'Save Config' button is located in the top right corner.

Name	Value
Name	[Name] [GLOBAL]
Product Model	[ProductCode] [GLOBAL]
Description	[Description] [GLOBAL]
Short Description	[ForceGento__Short_Description__c] [GLOBAL]
Product Type	[Family] [GLOBAL]
Weight	[ForceGento__Weight__c] [GLOBAL]
Tax Class	[ForceGento__Tax_Class__c] [GLOBAL]
Active	[IsActive] [GLOBAL]
Backorder	[ForceGento__Backorders__c] [GLOBAL]
Quantity	[ForceGento__Quantity__c] [GLOBAL]
Stock Availability	[ForceGento__Stock_Availability__c] [GLOBAL]

**Fig: Salesforce Product Mapping**

3. Click 'Save Config' button.

## Map Magento Product Fields to respective Salesforce PriceBook Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Product Mapping'.
2. Map the respective fields.



Current Configuration Scope: Default Config [Manage Stores](#)

**Configuration**

- GENERAL
  - General
  - Web
  - Design
  - Currency Setup
  - Store Email Addresses
  - Contacts
  - Reports
  - Content Management
- FORCEGENTO CONNECTOR API EXTENSION
  - Salesforce API Configuration
  - Order Configuration
  - Customer Configuration
  - Salesforce Lead Mapping
  - Salesforce Account Mapping
  - Salesforce Product Mapping**
  - Salesforce Opportunity Mapping

**Salesforce Product Mapping** [Save Config](#)

Mapping to Salesforce Product2

Mapping to Salesforce PricebookEntry

Price	UnitPrice ▲ Select Salesforce PricebookEntry Field 'UnitPrice'	[1] [GLOBAL]
Active	IsActive ▲ Select Salesforce PricebookEntry Field 'IsActive'	[2] [GLOBAL]

**Fig: Salesforce Pricebook Entry Mapping**

3. Click 'Save Config' button.

## Map Magento Order Fields to respective Salesforce Opportunity Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Opportunity Mapping'.
2. Map the respective fields.

Current Configuration Scope: Default Config [Manage Stores](#)

**Configuration**

- GENERAL
  - General
  - Web
  - Design
  - Currency Setup
  - Store Email Addresses
  - Contacts
  - Reports
  - Content Management
- FORCEGENTO CONNECTOR API EXTENSION
  - Salesforce API Configuration
  - Order Configuration
  - Customer Configuration
  - Salesforce Lead Mapping
  - Salesforce Account Mapping
  - Salesforce Product Mapping
  - Salesforce Opportunity Mapping**

**Salesforce Opportunity Mapping** [Save Config](#)

Order Mapping to Salesforce Opportunity

Name:	Name ▲ Select Salesforce Opportunity Field 'Name'	[1] [STORE VIEW]
Status:	StageName ▲ Select Salesforce Opportunity Field 'StageName'	[2] [STORE VIEW]
Amount:	Amount ▲ Select Salesforce Opportunity Field 'Amount'	[3] [STORE VIEW]
Total Qty:	TotalOpportunityQuantity ▲ Select Salesforce Opportunity Field 'TotalOpportunityQuantity'	[4] [STORE VIEW]
Web:	LeadSource ▲ Select Salesforce Opportunity Field 'LeadSource'	[5] [STORE VIEW]
Close date:	CloseDate ▲ Select Salesforce Opportunity Field 'CloseDate'	[6] [STORE VIEW]
Order Number:	ForceGento__OrderNumber__c ▲ Select Salesforce Opportunity Field 'ForceGento__OrderNumber__c'	[7] [STORE VIEW]

Order Product Mapping to Salesforce OpportunityLineItem

**Fig: Salesforce Opportunity Mapping**

3. Click 'Save Config' button.

## Map Magento Order Fields to respective Salesforce PriceBook2 Fields

1. In 'ForceGento Connector API Extension' section, click on 'Salesforce Opportunity Mapping'.

## 2. Map the respective fields.

The screenshot shows the 'Salesforce Opportunity Mapping' configuration page. On the left is a sidebar with a 'Configuration' menu. The 'FORCEGENTO CONNECTOR API EXTENSION' section is expanded, showing various configuration options. The 'Salesforce Opportunity Mapping' option is selected. The main content area is titled 'Salesforce Opportunity Mapping' and contains a 'Save Config' button in the top right corner. Below the title, there are two sections: 'Order Mapping to Salesforce Opportunity' and 'Order Product Mapping to Salesforce OpportunityLineItem'. The 'Order Product Mapping to Salesforce OpportunityLineItem' section is expanded, showing three fields: 'Quantity', 'Total Price', and 'Service Date'. Each field has a dropdown menu to select a Salesforce field. The 'Quantity' field is mapped to 'Select Salesforce OpportunityLineItem Field "Quantity"', 'Total Price' is mapped to 'Select Salesforce OpportunityLineItem Field "TotalPrice"', and 'Service Date' is mapped to 'Select Salesforce Opportunity Field "ServiceDate"'. Each field also has a '[STORE VIEW]' link.

**Fig: Salesforce Opportunity Line Item Mapping**

## 3. Click 'Save Config' button.

## 9. Sync Process

### Customer Configuration in Magento

1. In 'ForceGento Connector API Extension' section, click on Customer Configuration'.
2. Select the respective fields.

The screenshot shows the 'Customer Configuration' page. On the left is a sidebar with a 'Configuration' menu. The 'FORCEGENTO CONNECTOR API EXTENSION' section is expanded, showing various configuration options. The 'Customer Configuration' option is selected. The main content area is titled 'Customer Configuration' and contains a 'Save Config' button in the top right corner. Below the title, there are two sections: 'Contact' and 'Lead'. The 'Contact' section is expanded, showing two fields: 'Synchronize Customers' and 'Create as Lead'. 'Synchronize Customers' is set to 'Enable' and 'Create as Lead' is set to 'Disable'. The 'Lead' section is expanded, showing three fields: 'Synchronize Lead', 'Lead Source', and 'Lead Status'. 'Synchronize Lead' is set to 'Enable', 'Lead Source' is set to 'Web', and 'Lead Status' is set to 'Working - Contacted'. Each field has a dropdown menu to select a Salesforce field. The 'Synchronize Customers' field is mapped to 'Enable/Disable synchronization of Magento customers.', 'Create as Lead' is mapped to 'Create a Lead first. When customer converting by placing an order. Convert to Contact an Account', 'Synchronize Lead' is mapped to 'Enable/Disable synchronization of Magento Contact Us.', 'Lead Source' is mapped to 'Lead source to be set in salesforce when Lead created.', and 'Lead Status' is mapped to 'Lead status to be set in salesforce when Lead created.'.

**Fig: Customer Configuration**

3. Click 'Save Config' button.

## Order Configuration in Magento

1. In 'ForceGento Connector API Extension' section, click on Order Configuration'.
2. Select the respective fields.

Current Configuration Scope: Default Config Manage Stores

**Configuration**

- GENERAL
- General
- Web
- Design
- Currency Setup
- Store Email Addresses
- Contacts
- Reports
- Content Management
- FORCEGENTO CONNECTOR API EXTENSION**
  - Salesforce API Configuration
  - Order Configuration**
  - Customer Configuration
  - Salesforce Lead Mapping
  - Salesforce Account Mapping
  - Salesforce Product Mapping
  - Salesforce Opportunity Mapping

**Order Configuration**

**Default**

Synchronize Orders: Enable [GLOBAL]  
Enable/Disable synchronization of Magento orders.

Synchronize Products: Enable [GLOBAL]  
Enable/Disable synchronization of Magento products.

**Opportunity Customer Role**

Enable: Enable [GLOBAL]  
Enable Opportunity Customer Integration Role.

Opportunity Customer Role: Technical Buyer [GLOBAL]

**Opportunity Products**

Yes: Allows you to attach ordered items to Opportunity and Enables you to sync Magento products with Salesforce.  
No: Opportunity in Salesforce will still be created but products won't be passing along.

Enable: Enable [GLOBAL]

Default Pricebook: Standard Price Book [GLOBAL]  
Default pricebook to be used with products.

Save Config

**Fig: Customer Configuration**

3. Click 'Save Config' button.

## 10. Decode Source Guardian Code

- Check PHP Configuration in the server to see whether it has any Source Guardian modules or not?
- In order to check, you need to create a test file in the document root with the following script with the name **phpinfo.php**  
E.g.: `http://www.domainname.com/phpinfo.php`  

```
<? php
phpinfo ();
?>
```
- Search for **Source Guardian** from out. If the module exists, then skip next Step else proceed with next Step
- The Source Code of ForceGento Connector extension is encrypted. In regarding to that, you need a Source Guardian Loaders to load the ForceGento Connector connect successfully.

- Open the following link to download the specific Source Guardian Loader & follow the instructions  
<http://www.sourceguardian.com/ixeds/download.php>
- In order to detect the required loader and know how to install it, please enter the phpinfo link to the phpinfo () page that is running on your server.  
E.g.: <http://www.domainname.com/phpinfo.php>
- Download the required loader ixed.xx.lin on the Server
- Move the downloaded file ixed.xx.lin to the location mentioned in the steps given by Source Guardian
- Open your php.ini file from the location
- Restart the web server

## **11. Uninstall ForceGento Connect**

- Login into Magento Admin
- Go to System >> Configuration >> ForceGento Connector >> Extension Options.
- Select No from the drop down and Click on Save Config. Or
- Go to FTP Client.
- Go to Magento Folders.
- Follow the path app\etc\modules and select file ForceGento\_Connector.xml & Right Click the file to EDIT.
- Press Ctrl+F and change TRUE with FALSE
- Save the ForceGento\_Connector.xml.