



---

## **Coveo Platform 7.0**

Salesforce Legacy Connector Guide

## Notice

The content in this document represents the current view of Coveo as of the date of publication. Because Coveo continually responds to changing market conditions, information in this document is subject to change without notice. For the latest documentation, visit our website at [www.coveo.com](http://www.coveo.com).

© Coveo Solutions Inc., 2013

Coveo is a trademark of Coveo Solutions Inc. This document is protected by intellectual property laws and is subject to all restrictions specified in the Coveo Customer Agreement.

Document part number: PM-120827-EN

Publication date: 1/3/2019

## Table of Contents

<b>1. Salesforce Legacy Connector</b> .....	<b>1</b>
1.1 Features .....	1
<b>2. Salesforce Legacy Connector Deployment Overview</b> .....	<b>3</b>
<b>3. Salesforce Legacy Connector Requirements</b> .....	<b>4</b>
<b>4. About the Default Salesforce Mapping File for the Legacy Connector</b> .....	<b>5</b>
4.1 Managing Permissions .....	5
4.2 Default Mapping File Content .....	6
<b>5. Creating and Using a Custom Salesforce Mapping File for the Legacy Connector</b> .....	<b>10</b>
5.1 Mapping file structure .....	11
5.2 Mapping child nodes .....	11
5.3 Mapping local expressions .....	13
5.4 Mapping foreign expressions .....	13
5.5 Using external resolvers .....	13
<b>6. Adding a User Identity</b> .....	<b>15</b>
<b>7. Configuring a Salesforce Security Provider for the Legacy Connector</b> .....	<b>17</b>
<b>8. Configuring and Indexing a Salesforce Source for the Legacy Connector</b> .....	<b>20</b>
8.1 Modifying Source Security Permissions .....	25
8.1.1 Modifying the permissions .....	25
8.1.2 Mapping the security permissions of a Web source .....	27
8.2 Modifying Hidden Salesforce Source Parameters for the Legacy Connector .....	28
8.3 Adding an Explicit Connector Parameter .....	30
<b>9. Adding Search Interfaces to a Search Hub With the .NET Interface Editor</b> .....	<b>33</b>



# 1. Salesforce Legacy Connector

**Deprecated**

**Important:**

- The Salesforce Legacy connector is removed from Coveo Enterprise Search 7.0 (CES 7) since the March 2016 release.
- The Legacy connector uses the Winter '12 API (version 23.0) of Salesforce; and since each Salesforce API version is guaranteed to be supported for three years, it is strongly recommended to migrate your Salesforce Legacy source(s) to one(s) that use(s) the second-generation connector for Salesforce . Contact [Coveo Support](#) for assistance.

The Coveo Salesforce Legacy connector allows you to crawl Salesforce content and bring it into the unified index, making the it easily searchable by end-users. The legacy connector uses a mapping file that defines the database object signatures as well as the objects and fields to index. You can modify the mapping file to customize the indexed content and easily add new object types and custom fields to optimally index your Salesforce database.

**Note:** The Salesforce Legacy connector is now only available to existing users with a CES instance that originally used this connector. The improved replacement solution is Coveo for Salesforce (see [Coveo for Salesforce](#)).

## 1.1 Features

**Incremental refresh**

The incremental refresh feature refreshes the content of the index based on the modification date of the objects in the Salesforce environment. If an item is modified, the incremental refresh feature refreshes the item automatically.

**Customizable object mapping**

With the mapping file, you can select which Salesforce objects to index. You can decide to index additional custom Salesforce objects and fields, or remove existing ones by just modifying the mapping file (see "[Creating and Using a Custom Salesforce Mapping File for the Legacy Connector](#)" on page 10).

**Metadata**

All the fields of the retrieved Salesforce objects are indexed as metadata. You can therefore customize CES based on that metadata via custom fields.

### Connector Feature History

Coveo Platform version	Date	Features
7.0.5785	August 2013	With the introduction of the second generation Salesforce connector, this connector becomes the Salesforce Legacy connector.

## What's Next?

Review the deployment process for the Salesforce legacy connector (see "[Salesforce Legacy Connector Deployment Overview](#)" on page 3).

## 2. Salesforce Legacy Connector Deployment Overview

### Deprecated

The following procedure outlines the steps needed to deploy the Salesforce Legacy connector. The steps indicate the order in which you must perform key CES configurations. When needed, the step refers to a detailed procedure.

**Note:** It is strongly recommended to rather use the second generation Salesforce connector.

To deploy the Salesforce legacy connector

1. Validate that your environment meets the requirements (see ["Salesforce Legacy Connector Requirements" on page 4](#)).

2. Configure the user identity.

The Coveo legacy connector needs an account to connect to your Salesforce environment and access the entire content that you want to index. Create a CES user identity that contains the credentials of a Salesforce account that has read access to all the Salesforce content that you want to index (see ["Adding a User Identity" on page 15](#)).

3. Optionally customize the default mapping file to adapt the Salesforce content to index (see ["About the Default Salesforce Mapping File for the Legacy Connector" on page 5](#) and ["Creating and Using a Custom Salesforce Mapping File for the Legacy Connector" on page 10](#)).

4. Configure and index the Salesforce source.

The Coveo legacy connector needs to know details about the Salesforce database to be able to index the desired content (see ["Configuring and Indexing a Salesforce Source for the Legacy Connector" on page 20](#)).

5. Add the out-of-the-box or custom Salesforce search interface to a search hub (see ["Adding Search Interfaces to a Search Hub With the .NET Interface Editor" on page 33](#)) and then validate that allowed users can search the Salesforce content.

6. Optionally, if you encounter issues, consider adding and modifying default values of hidden source parameters (see ["Modifying Hidden Salesforce Source Parameters for the Legacy Connector" on page 28](#)).

## 3. Salesforce Legacy Connector Requirements

### Deprecated

Your environment must meet the following requirements to be able to use the Coveo legacy connector for Salesforce repositories:

- Coveo license including the Salesforce Legacy connector

Your original Coveo license must have included support for the Salesforce legacy connector to be able to use this connector.

- An active Salesforce account to a Salesforce environment

This account requires a read permission on all the types of content crawled by the connector.

### What's Next?

Create a CES user identity for Salesforce (see ["Adding a User Identity" on page 15](#)).

## 4. About the Default Salesforce Mapping File for the Legacy Connector

### Deprecated

The Coveo legacy connector for Salesforce embeds an XML mapping file that is appropriate for many Salesforce setups. This default mapping file specifies to index the basic Salesforce database object types and the corresponding fields.

The Salesforce object types indexed with the default mapping file are:

- Account
- Contact
- Lead
- Opportunity
- Case
- Event
- Task
- Case Comment
- Email Message
- Document
- Attachment

### 4.1 Managing Permissions

The permissions associated to Salesforce content are incorporated into the unified index. When a user makes a query for Salesforce content, the index knows which content this user has access to and only returns query results for content this specific user has access to.

#### Using the default mapping file

The default mapping file includes a security binding mapping (`<AllowedUsers>`) that grants access to all users that are members of the *Salesforce Users* CES custom security group. This mapping is in the `<CommonMapping>` section of the mapping file and therefore applies to all indexed Salesforce types of content.

#### Using a custom mapping file

You can customize the mapping file to implement more specific early-binding security mappings, like creating independent custom security groups for each Salesforce type of content (see "[Creating and Using a Custom Salesforce Mapping File for the Legacy Connector](#)" on page 10).

**Example:** You may want that only users from your Sales department can search the *Opportunity* type of Salesforce content and only users from your Customer Support department can search the *Case* type of Salesforce content.

## 4.2 Default Mapping File Content

The following XML code presents the content of the default mapping file.

```
<?xml version="1.0" encoding="Windows-1252" ?>
<SalesForce>
  <CommonMapping>
    <Fields>
      <ContentType>binarydata</ContentType>
      <syssfcreatedby>%[CreatedBy.Name]</syssfcreatedby>
      <syssfcreatedbyid>%[CreatedBy.Id]</syssfcreatedbyid>
      <syssfcreateddate>%[CreatedDate]</syssfcreateddate>
      <sysauthor>%[LastModifiedBy.Name]</sysauthor>
      <sysdate>%[SystemModstamp]</sysdate>
      <syssfid>%[Id]</syssfid>
      <syssfowner>%[OwnerId.Name]</syssfowner>
      <syssfownerid>%[OwnerId]</syssfownerid>
    </Fields>
    <AllowedUsers>
      <AllowedUser type="CustomGroup" allowed="true">
        <Name>Salesforce Users</Name>
        <Server></Server>
      </AllowedUser>
    </AllowedUsers>
  </CommonMapping>
  <Mapping type="Account">
    <Fields>
      <Title>%[Name]</Title>
      <Body>%[Name]
Owner: %[OwnerId.Name]
Type: %[Type]
Industry: %[Industry]
Description:
%[Description]
      </Body>
      <sysfiletype>SFAccount</sysfiletype>
      <syssfbillingstreet>%[BillingStreet]</syssfbillingstreet>
      <syssfbillingcity>%[BillingCity]</syssfbillingcity>
      <syssfbillingstate>%[BillingState]</syssfbillingstate>
      <syssfbillingcountry>%[BillingCountry]</syssfbillingcountry>
      <syssfbillingpostalcode>%[BillingPostalCode]</syssfbillingpostalcode>
      <syssfshippingstreet>%[ShippingStreet]</syssfshippingstreet>
      <syssfshippingcity>%[ShippingCity]</syssfshippingcity>
      <syssfshippingstate>%[ShippingState]</syssfshippingstate>
      <syssfshippingcountry>%[ShippingCountry]</syssfshippingcountry>
      <syssfshippingpostalcode>%[ShippingPostalCode]</syssfshippingpostalcode>
      <sysworkphone>%[Phone]</sysworkphone>
      <syssfxfax>%[Fax]</syssfxfax>
      <syssfwebsite>%[Website]</syssfwebsite>
      <syssfcurrency>%[CurrencyIsoCode]</syssfcurrency>
      <syssfdescription>%[Description]</syssfdescription>
      <syssfcountry>%[BillingCountry]</syssfcountry>
      <syssfindustry>%[Industry]</syssfindustry>
    </Fields>
  </Mapping>
  <Mapping type="Contact">
    <Fields>
      <Title>%[FirstName] %[LastName] (%[Title])</Title>
      <Body>%[FirstName] %[LastName]
%[AccountId.Name]
Title: %[Title]
Account: %[AccountId.Name]
    </Body>
  </Fields>
  </Mapping>
</SalesForce>
```

```

Email: %[Email]
Phone: %[Phone]
Address:
%[MailingStreet]
%[MailingCity], %[MailingState]
%[MailingCountry]
</Body>
<sysfiletype>SFContact</sysfiletype>
<sysffirstname>%[FirstName]</sysffirstname>
<sysfflastname>%[LastName]</sysfflastname>
<sysffstreet>%[MailingStreet]</sysffstreet>
<sysffcity>%[MailingCity]</sysffcity>
<sysffstate>%[MailingState]</sysffstate>
<sysffcountry>%[MailingCountry]</sysffcountry>
<sysffpostalcode>%[MailingPostalCode]</sysffpostalcode>
<sysworkemail>%[Email]</sysworkemail>
<sysworkphone>%[Phone]</sysworkphone>
<syscompany>%[AccountId.Name]</syscompany>
<sysffaccount>%[AccountId.Name]</sysffaccount>
<sysffaccountid>%[AccountId]</sysffaccountid>
</Fields>
</Mapping>
<Mapping type="Lead">
<Fields>
<Title>%[Company] ([FirstName] [LastName])</Title>
<Body>%[FirstName] [LastName]
Title: %[Title]
Company: %[Company]
Email: %[Email]
Phone: %[Phone]
Address:
%[Street]
%[City], %[State]
%[Country]
Status: %[Status]</Body>
<sysfiletype>SFLead</sysfiletype>
<sysffirstname>%[FirstName]</sysffirstname>
<sysfflastname>%[LastName]</sysfflastname>
<syscompany>%[Company]</syscompany>
<sysffcity>%[City]</sysffcity>
<sysworkemail>%[Email]</sysworkemail>
<sysffindustry>%[Industry]</sysffindustry>
<sysffleadsource>%[LeadSource]</sysffleadsource>
<sysworkphone>%[Phone]</sysworkphone>
<sysffcountry>%[Country]</sysffcountry>
<sysffstate>%[State]</sysffstate>
<sysffcity>%[City]</sysffcity>
<sysffstreet>%[Street]</sysffstreet>
<sysffpostalcode>%[PostalCode]</sysffpostalcode>
<sysffleadstatus>%[Status]</sysffleadstatus>
</Fields>
</Mapping>
<Mapping type="Opportunity">
<CacheTables>
<Table name="OpportunityLineItem">
<IndexedField name="OpportunityId" />
</Table>
<Table name="PricebookEntry" />
<Table name="Product2" />
</CacheTables>
<Fields>
<Title>%[AccountId.Name] [%[Name]]</Title>
<Body>%[Name]</Body>
<sysfiletype>SFOpportunity</sysfiletype>
<sysffaccount>%[AccountId.Name]</sysffaccount>
<sysffaccountid>%[AccountId]</sysffaccountid>
<sysffcountry>%[AccountId.BillingCountry]</sysffcountry>
<sysffstagename>%[StageName]</sysffstagename>
<sysffamount>%[Amount]</sysffamount>
    
```

```

<sysffcurrency>%[CurrencyIsoCode]</sysffcurrency>
<sysffopportunitytype>%[Type]</sysffopportunitytype>
<sysffclosedate>%[CloseDate]</sysffclosedate>
<sysffclosequarter>%[FiscalYear]-Q#[FiscalQuarter]</sysffclosequarter>
<sysffleadsource>%[LeadSource]</sysffleadsource>
<sysffname>%[Name]</sysffname>
</Fields>
</Mapping>
<Mapping type="Case">
<Fields>
<ContentType>binarydata</ContentType>
<Title>%[Subject] (%[CaseNumber])</Title>
<Body>
AssetId : %[AssetId]
CaseNumber : %[CaseNumber]
ClosedDate : %[ClosedDate]
Contact : %[ContactId.Name]
CurrencyIsoCode : %[CurrencyIsoCode]
IsClosed : %[IsClosed]
IsEscalated : %[IsEscalated]
Origin : %[Origin]
Priority : %[Priority]
Reason : %[Reason]
Status : %[Status]
Subject : %[Subject]
SuppliedCompany : %[SuppliedCompany]
SuppliedEmail : %[SuppliedEmail]
SuppliedName : %[SuppliedName]
SuppliedPhone : %[SuppliedPhone]
Type : %[Type]
Description :
%[Description]
</Body>
<sysfiletype>SFCase</sysfiletype>
</Fields>
</Mapping>
<Mapping type="Event">
<Fields>
<Title>%[Subject]</Title>
<Body>%[Description]</Body>
<sysfiletype>SFEEvent</sysfiletype>
<sysffduration>%[DurationInMinutes]</sysffduration>
<sysfflocation>%[Location]</sysfflocation>
</Fields>
</Mapping>
<Mapping type="Task">
<Fields>
<Title>%[Subject]</Title>
<Body>%[Description]</Body>
<sysfiletype>SFTask</sysfiletype>
<sysffstatus>%[Status]</sysffstatus>
<sysffpriority>%[Priority]</sysffpriority>
</Fields>
</Mapping>
<Mapping type="CaseComment">
<Fields>
<Body>%[CommentBody]</Body>
<sysfiletype>SFCaseComment</sysfiletype>
<sysffcseid>%[ParentId]</sysffcseid>
</Fields>
</Mapping>
<Mapping type="EmailMessage">
<Fields>
<Title>%[Subject]</Title>
<Body>%[TextBody]</Body>
<sysfiletype>SFEmailMessage</sysfiletype>
<sysffcseid>%[ParentId]</sysffcseid>
<sysdate>%[MessageDate]</sysdate>
<sysrecipients>%[ToAddress]</sysrecipients>
    
```

```

<sysfrom>%[FromAddress]</sysfrom>
<syscc>%[CcAddress]</syscc>
<sysbcc>%[BccAddress]</sysbcc>
<sysconversationsubjectid>%[ParentId]</sysconversationsubjectid>
<sysmailbox>Salesforce.com</sysmailbox>
</Fields>
</Mapping>
<Mapping type="Document">
  <Fields>
    <clickableuri>https://na4.salesforce.com/servlet/servlet.FileDownload?file=%[Id]</clickableuri>
    <filename>%[Name]</filename>
    <body>%[Body]</body>
    <Title>%[Name]</Title>
  </Fields>
</Mapping>
<Mapping type="Attachment">
  <Fields>
    <clickableuri>https://na4.salesforce.com/servlet/servlet.FileDownload?file=%[Id]</clickableuri>
    <filename>%[Name]</filename>
    <body>%[Body]</body>
    <Title>%[Name]</Title>
  </Fields>
</Mapping>
</SalesForce>
    
```

## 5. Creating and Using a Custom Salesforce Mapping File for the Legacy Connector

You can create a custom mapping file to modify the behavior of the Salesforce legacy connector to adapt to your Salesforce database.

**Tip:** When you want to index only a subset of Salesforce types that are defined in the default mapping file, you do not have to create a custom mapping file. You can simply specify the Salesforce types to index in the **Mapping Types** field for the Salesforce source (see ["Configuring and Indexing a Salesforce Source for the Legacy Connector" on page 20](#)).

To create and use a custom mapping file

1. Copy the content of the default mapping file (see ["About the Default Salesforce Mapping File for the Legacy Connector" on page 5](#)).
2. Using an administrator account, connect to the Coveo Master server.
3. Using a text editor:
  - a. In an empty file, paste the content of the default mapping file.
  - b. Edit the content of the mapping file to suit your needs, referring to the following subsections for details on how to modify the file:
    - ["Mapping file structure" on page 11](#)
    - ["Mapping child nodes" on page 11](#)
    - ["Mapping security" on page 12](#)
    - ["Mapping local expressions" on page 13](#)
    - ["Mapping foreign expressions" on page 13](#)
    - ["Using external resolvers" on page 13](#)
  - c. Save the custom mapping file on the Coveo Master server.

**Example:** C:\CES7\Config\SalesforceMappingFile.xml

4. Configure the connector to use the custom mapping file in the Salesforce source:
  - When you create a Salesforce source (see ["Configuring and Indexing a Salesforce Source for the Legacy Connector" on page 20](#)).OR
  - For an existing Salesforce source:
    - a. On the Coveo server, access the Administration Tool.
    - b. Select **Index > Sources and Collections**.

- c. Under **Collections**, select the collection containing the Salesforce source.
- d. Under **Sources**, select the Salesforce source.
- e. In the navigation panel on the left, select **General**.
- f. In the **Mapping File** parameter, enter the full path to your custom mapping file.
- g. Click **Apply Changes**.

5. Rebuild the source.

## 5.1 Mapping file structure

The mapping file can be divided into three sections:

- **CommonMapping**: All the settings applied to all the mappings.
- **Mapping**: Individual mappings for each object type to index.

You can add mappings for custom Salesforce types not included in the default mapping file simply by copying and editing a mapping section for an existing type.

- **ExternalResolvers**: When you need to apply a sophisticated mapping, you can specify an external assembly type to call to resolve it.

## 5.2 Mapping child nodes

`<Fields>`

Fields for this mapping. From this point, you either define a system or custom field:

- System fields

System fields can be used directly under the `<fields>` node. They are mapped to an existing system field.

**Example:** The `<uri>` is mapped directly to the system field `URI` for this type.

- Custom fields

Custom fields can be mapped to a custom metadata name. The metadata is filled with the string content.

**Example:** The `<CustomField name="SFLocation">%[Location]</CustomField>` node maps the field named `Location` on the Salesforce object to the metadata `SFLocation` on the document in the unified index.

`<AllowedUsers>`

- **Type of users:** `Windows, CustomGroup, CustomUser, ExternalGroup, ExternalUser, WindowSid`
- **Name:** Name of the user
- **Server:** Usually references to the domain

## Mapping security

You can configure the Salesforce legacy connector by adding or removing allowed groups and users to the mapping file for each object types. The permissions are therefore incorporated in the index. Using this method is recommended as CES does not have to fetch the security permissions at query time for each query. However, security permissions can only be set for each object type. This means that individual objects of the same type will always have the same security permissions.

**Example:** The following mapping code gives to the Windows group `Sales` from the `MyOrganization` domain access to all the index types as the `CommonMapping` section applies to all the indexed mappings.

```
<CommonMapping>
  <AllowedUsers>
    <AllowedUser type="Windows" allowed="true">
      <Name>Sales</Name>
      <Server>MyOrganization</Server>
    </AllowedUser>
  </AllowedUsers>
</CommonMapping>
```

You can specify different security permissions for each mapping type by adding an `AllowedUsers` child node to every mapping node.

**Example:** The following mapping code gives access to the `Account` type to the CES custom security group `Salesforce Account Users` from the `MyOrganization` domain.

```
<Mapping type="Account">
  <AllowedUsers>
    <AllowedUser type="CustomGroup" allowed="true">
      <Name>Salesforce Account Users</Name>
      <Server>MyOrganization</Server>
    </AllowedUser>
  <Fields>
    <Title>%[Subject]</Title>
    <Body>%[Description]</Body>
    <CustomFields>
      <CustomField name="SFOwner">%[OwnerId.FirstName][OwnerId.LastName]</CustomField>
      <CustomField name="SFOwnerID">%[OwnerId]</CustomField>
      <CustomField name="SFStatus">%[Status]</CustomField>
      <CustomField name="SFPriority">%[Priority]</CustomField>
    </CustomFields>
  </Fields>
</Mapping>
```

You can also directly set Active Directory users or groups in the mapping file. This method however requires that you modify the mapping file and rebuild the source each time that you add or remove users or groups.

**Example:** The following mapping code gives access to the `Task` type to the Active Directory group `Sales` from the `MyOrganization` domain.

```
<Mapping type="Tasks">
  <AllowedUsers>
    <AllowedUser type="Windows" allowed="true">
      <Name>Sales</Name>
      <Server>MyOrganization</Server>
    </AllowedUser>
  </AllowedUsers>
  <Fields>
    <Title>%[Subject]</Title>
    <Body>%[Description]</Body>
    <CustomFields>
      <CustomField name="SFOwner">%[OwnerId.FirstName]%[OwnerId.LastName]</CustomField>
      <CustomField name="SFOwnerID">%[OwnerId]</CustomField>
      <CustomField name="SFStatus">%[Status]</CustomField>
      <CustomField name="SFPriority">%[Priority]</CustomField>
    </CustomFields>
  </Fields>
</Mapping>
```

### 5.3 Mapping local expressions

The Salesforce legacy connector uses a specific mapping expression to represent a value on a Salesforce object. Occurrences of the expression `%[field]` are replaced by the value of the field specified by the name in the brackets for this Salesforce object.

**Example:** The expression `<ModifiedDate>%[LastModifiedDate]</ModifiedDate>` replaces the string `%[LastModifiedDate]` by the value of the `LastModifiedDate` field for each Salesforce object. The value fetched is then assigned to the system field `ModifiedDate` on each document in the unified index.

### 5.4 Mapping foreign expressions

As some objects can reference to others in Salesforce, you can fetch the value of a foreign field in your mappings, as long as the foreign type has the requested field.

**Example:** The expression `<CustomField name="SFOwner">%[OwnerId.FirstName] %[OwnerId.LastName]</CustomField>` retrieves the specific object pointed by `OwnerId` and retrieves the value of the fields `FirstName` and `LastName` on that object. Then the resulting string is a concatenation of both results. An example of the result for this expression could be *John Smith*.

**Important:** Foreign expressions can dramatically slow down the indexing process, as a remote object has to be queried for every foreign expression. Coveo has a local cache for each field value; however the performance decreases every time it has to fetch field values on a new object. Use them carefully.

### 5.5 Using external resolvers

When declaring an external resolver, you need to provide a unique name as well as a fully qualified type name to call.

```
<SalesForce>
  <ExternalResolvers>
    <ExternalResolver name="Resolvername" type="FQTypeName"/>
  </ExternalResolvers>
</SalesForce>
```

```
</ExternalResolvers>
</SalesForce>
```

**Note:** For more information on fully qualified type names, refer to the Microsoft document [Specifying Fully Qualified Type Names](#).

Once the external resolver has been defined, you can use it in any mapping using the following syntax: @ [resolvername].

Parameters can be defined to pass to the external type. In this case, you must specify an additional bracket node with parameters separated by a semicolon (;): @ [resolvername] [param1;param2].

As the external resolvers are called last, fields to resolve (such as %[field]) can be specified as parameters and will be resolved before the external resolver is called.

### External resolver type limitations

For the external type to be used with the external resolver system, you must write a DLL that implements the `IMappingResolver` interface defined as follows:

```
namespace Coveo.CES.CustomCrawlers.SalesForce
{
    /**
     * //*****
     * /// <summary>
     * /// the Interface the callback functions in the mapping file must implement.
     * /// </summary>
     * //*****
     * public interface IMappingResolver
     * {
     *     string Resolve(string[] p_Parameters,
     *                   Record p_CurrentRecord,
     *                   MappingUtilities p_Uilities);
     * }
}
```

The parameters are passed as an array of string in `p_Parameters`. The current record being parsed is passed as well as `p_CurrentRecord`. Finally, the connector cached records, connection, disk cache, etc. can be accessed using the mapping utilities. The string returned by the resolve function will be used as the mapping value.

All these classes and interfaces are in the Salesforce legacy connector assembly, the external assemblies will need to add the Salesforce connector assembly as a reference.

## 6. Adding a User Identity

A user identity is a set of credentials for a given repository or system that you enter once in CES and can then associate with one or more sources or security providers.

A user identity typically holds the credentials of an account that has read access to all the repository items that you want to index. It is a best practice to create an account to be used exclusively by the Coveo processes and for which the password does not change. If the password of this account changes in the repository, you must also change it in the CES user identity.

To add a user identity

1. On the Coveo server, access the Administration Tool.
2. In the Administration Tool, select **Configuration > Security**.
3. In the navigation panel on the left, click **User Identities**.
4. In the **User Identities** page, click **Add**.
5. In the **Modify User Identity** page:

- a. In the **Name** box, enter a name of your choice to describe the account that you selected or created in the repository to allow CES to access the repository.

**Note:** This name appears only in the Coveo Administration Tool, in the **Authentication** or **User Identity** drop-down lists, when you respectively define a source or a security provider.

- b. In the **User** box, enter the username for the account that you selected or created to crawl the repository content that you want to index.
- c. In the **Password** box, enter the password for the account.
- d. In the **Options** section, the **Support basic authentication** check box is deprecated and not applicable for

most types of repositories. You should select it only when you need to allow CES to send the username and password as unencrypted text.

- e. Click **Save**.

**Important:** When you use Firefox to access the Administration Tool and it proposes to remember the password for the user identity that you just created, select to never remember the password for this site to prevent issues with automatic filling of username and password fields within the Coveo Administration Tool.

## 7. Configuring a Salesforce Security Provider for the Legacy Connector

### Deprecated

The Coveo legacy connector needs to resolve mappings between users and groups from your Salesforce environment and Microsoft Active Directory.

Once the security provider is created, you must associate it to the Salesforce source. The security provider is invoked every time a user performs a query and validates with Salesforce if the user performing the query has the right to view each document in the query results. This process can be slow; however it allows individual security permissions for each document instead of having permissions for each object type. Users are required to enter their account information for Salesforce sources in the Search Interface when using this security method.

**Note:** You can get familiar with how Coveo components deal with permissions on documents both at indexing and query time.

To configure a Salesforce security provider for the legacy connector

1. On the Coveo server, access the Administration Tool.
2. Select **Configuration > Security**.
3. In the **Security** page, in the navigation panel on the left, click **Security Providers**.
4. In the **Security Providers** page, click **Add** to create a new security provider.
5. In the **Modify Security Provider** page:

- a. Configure the following required parameters:

#### Name

Choose a meaningful name to identify the security provider.

**Example:** `Salesforce Security Provider (Legacy Connector)`

#### Security Provider Type

In the drop-down list, select **Salesforce (Legacy) (x64)**.

**Note:** `CES 7.0.5639- (July 2013)` The Salesforce Legacy connector appears in the list as **Salesforce (x64)**.

#### User Identity

In the drop-down list, select the user identity that you created for this source.

- b. Revise the default value for the following parameters that generally do not need to be changed:

#### Salesforce Login Url

The URL used to login to Salesforce Web services. The default value is `https://login.salesforce.com/`.

**Salesforce API Version**

The API version of Salesforce Web services to connect to, specified in the ##.# format. The default value is 23.0.

**Timeout Time**

The communication timeout allowed for any request sent to the Salesforce Web services. The default value is 100 seconds.

- c. In the **Parameters** section, in rare cases the [Coveo Support](#) could instruct you to click **Add Parameters** to specify other security provider parameter names and values that could help to troubleshoot security provider issues.
- d. Leave the **Allow Complex Identities** option cleared as it does not apply to this type of security provider.
- e. Click **Apply Changes**.

**What's Next?**

Create you Salesforce source (see "[Configuring and Indexing a Salesforce Source for the Legacy Connector](#)" on [page 20](#)).

## 8. Configuring and Indexing a Salesforce Source for the Legacy Connector

### Deprecated

A source defines a set of configuration parameters for a specific Salesforce database.

**Note:** When your organization has access to more than one Salesforce environment, you must define one source for each Salesforce environment that you want to index.

To configure and index a Salesforce source for the legacy connector

1. On the Coveo server, access the Administration Tool.
2. Select **Index > Sources and Collections**.
3. In the **Collections** section:
  - a. Select an existing collection in which you want to add the new source.OR
  - b. Click **Add** to create a new collection.
4. In the **Sources** section, click **Add**.

The **Add Source** page that appears is organized in three sections.
5. In the **General Settings** section of the **Add Source** page:

a. Enter the appropriate value for the following required parameters:

**Name**

A descriptive name of your choice for this legacy connector source.

**Example:** Salesforce site (Legacy connector)

**Source Type**

**CES 7.0.7814+ (August 2015)** The connector used by this source. In this case, select **Salesforce (deprecated)**.

- Notes:**
- If you do not see **Salesforce (deprecated)** in the **Source Type** list, ensure that your current environment meets the requirements (see "[Salesforce Legacy Connector Requirements](#)" on page 4).
  - **CES 7.0.7711– (June 2015)** The Salesforce Legacy connector appears in the list as **Salesforce (Legacy) (x64)**.
  - **CES 7.0.5639– (July 2013 )** The Salesforce Legacy connector appears in the list as **Salesforce (x64)**.
  - **CES 7.0.5785+ (August 2013)** The **Salesforce (x64)** type corresponds to the source to be used with the second generation Salesforce connector.

### Addresses

Enter the Salesforce Website URL: `http://www.salesforce.com/`

### Refresh Schedule

Time interval at which the index is automatically refreshed to keep the index content up-to-date. By default, the **Every day** option instructs CES to refresh the source everyday at 12 AM.

**Note:** You can create new or modify existing source refresh schedules.

- b. Review the value for the following parameters that often do not need to be modified:

### Rating

Change this value only when you want to globally change the rating associated with all items in this source relative to the rating to other sources.

**Example:** When this source replaces a legacy system, you may want to set this parameter to **High**, so that in the search interface, results from this source appear earlier in the list compared to those from the source for the legacy system.

### Document Types

If you defined a custom document type set for this source, select it.

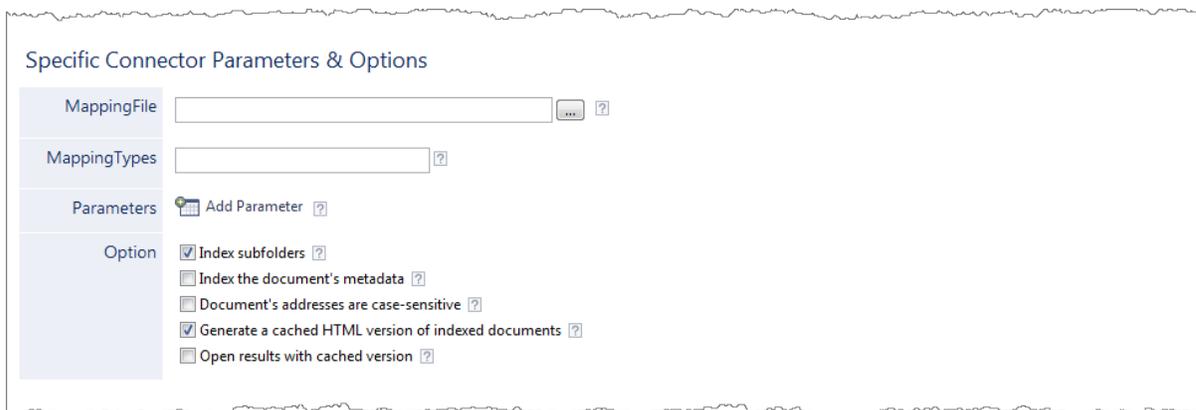
### Active Languages

If you defined custom active language sets, ensure to select the most appropriate for this source.

### Fields

If you defined custom field sets, ensure to select the most appropriate for this source.

- 6. In the **Specific Connector Parameters & Options** section of the **Add Source** page:



- a. Consider using the following optional parameters:

### Mapping File

Leave this box empty to use the default mapping file that instructs the connector to index basic

Salesforce elements and use the *Salesforce Users* CES custom security group to manage permissions (see ["About the Default Salesforce Mapping File for the Legacy Connector" on page 5](#)).

When you choose to create a custom mapping file, enter the absolute full path pointing to your custom mapping file (see ["Creating and Using a Custom Salesforce Mapping File for the Legacy Connector" on page 10](#)).

**Example:** C:\CES7\Config\SalesforceMappingFile.xml

### Mapping Types

Leave this box empty to instruct the connector to index all the mapping types defined in the default or custom mapping file.

This parameter is useful to easily index only a subset of mapping types among the ones defined in the mapping file. Enter each type to index separated by semicolons (see ["About the Default Salesforce Mapping File for the Legacy Connector" on page 5](#)).

**Example:** Enter: `Event;Task;Case`. The content for the other types will not be indexed.

- b. In the **Parameters** section, click **Add Parameter** when you want to show and configure advanced hidden source parameters (see ["Modifying Hidden Salesforce Source Parameters for the Legacy Connector" on page 28](#)).
- c. In the **Option** section, the state of check boxes generally does not need to be changed:

### Index Subfolders

Keep this check box selected (recommended). By doing so, all subfolders from the specified starting address are indexed.

### Index the document's metadata

When selected, CES indexes all the document metadata, even metadata that are not associated with a field. The orphan metadata are added to the body of the document so that they can be searched using free text queries.

When cleared (default), only the values of system and custom fields that have the **Free Text Queries** attribute selected will be searchable without using a field query.

**Example:** A document has two metadata:

- #LastEditedBy containing the value `Hector Smith`
- Department containing the value `RH`

In CES, the custom field `CorpDepartment` is bound to the metadata `Department` and its **Free Text Queries** attribute is selected.

When the **Index the document's metadata** option is cleared, searching for `RH` returns the document because a field is indexing this value. Searching for `hector` does not return the document because no field is indexing this value.

When the **Index the document's metadata** option is selected, searching for `hector` also returns the document because CES indexed orphan metadata.

### Document's addresses are case-sensitive

Leave the check box cleared. This parameter needs to be checked only in rare cases for systems in which distinct documents may have the same name but different casing.

### Generate a cached HTML version of indexed documents

When you select this check box (recommended), at indexing time, CES creates HTML versions of indexed documents. In the search interfaces, users can then more rapidly review the content by clicking the Quick View link rather than opening the original document with the original application. Consider clearing this check box only when you do not want to use Quick View links or to save resources when building the source.

### Open results with cached version

Leave this check box cleared (recommended) so that in the search interfaces, the main search result link opens the original document with the original application. Consider selecting this check box only when you do not want users to be able to open the original document but only see the HTML version of the document as a Quick View. In this case, you must also select **Generate a cached HTML version of indexed documents**.

7. In the **Security** section of the **Add Source** page:



- a. In the **Security Provider** drop-down list, select the security provider that you created for this source (see ["Configuring a Salesforce Security Provider for the Legacy Connector" on page 17](#)).
  - b. In the **Authentication** drop-down list, select the Salesforce user identity that you created.
  - c. Click **Save and Start** to save the source configuration and start indexing this source.
8. Validate that the source building process is executed without errors:
- In the navigation panel on the left, click **Status**, and then validate that the indexing proceeds without errors.
- OR
- Open the CES Console to monitor the source building activities.

## What's Next?

Set an incremental refresh schedule for your source.

## 8.1 Modifying Source Security Permissions

Three levels of security exist in CES index :

- Collection-level security
- Source-level security
- Document-level security

Source-level permissions determine which users have access to a source. By default, sources can be accessed by all users who have access to the parent collection. You can however override these permissions. Even if a user has access to a source, document-level permissions are required to display its content.

**Note:** Source-level permissions are not indexed for **Web Pages** sources; however, if Web files are stored locally (i.e., on the same network as CES), it is possible to associate file server permissions to them.

This topic contains the following sections:

- ["Modifying the permissions" on page 25](#)
- ["Mapping the security permissions of a Web source" on page 27](#)

### 8.1.1 Modifying the permissions

1. On the Coveo server, access the Administration Tool.
2. In the Administration Tool, select **Index > Sources and Collections**.
3. In the **Sources and Collections** page:
  - a. In the **Collections** section, select the collection the source that you want to modify.
  - b. In the **Sources** section, select the source that you want to modify.

- c. In the navigation panel on the left, select **Permissions**.
4. In the **Permissions** page, in the **Permissions** section, select one of the following options:

#### **Index security permissions**

Grants access to all users having the appropriate collection permissions. Document-level security is indexed.

#### **Specify the security permissions to index**

Grants access only to users whose accounts are entered in the **Allowed Users** box. Document-level permissions are not indexed.

**Important:** Because document-level permissions are not indexed, a user added to the **Allowed Users** list that do not have access to a document in the original repository will be able to view its excerpt, summary and Quick View from the search results.

#### **Index security permissions and specify additional security permissions to index**

Indexes document-level permissions and grants additional access to users whose accounts are entered in the **Allowed Users** box and denies access to users whose accounts are entered in the **Denied Users** box.

**Important:** The accounts entered in the **Allowed Users** box override document-level permissions. This means that even users who do not have access to a document are able to view its search result excerpt, summary, and Quick View.

#### **Late Binding**

When it is not possible to gather permission information at indexing time, as a last option to support secure search results, select **Late Binding** to identify which documents a user is allowed to see at query time. With this method, query response time is very slow the first time a query is performed, but much faster on next occurrences.

5. When you selected **Specify the security permissions to index** or **Index security permissions and specify additional security permissions to index**, for each user to which you want to modify the source permissions:
  - To grant access to the source, in the **Allowed Users** section:
    - i. Enter the account name of the user or group.
 

**Example:** For an Active Directory account, enter the name in the `domain\username` form.
    - ii. Select if the entered name is a **User** or a **Group**.
    - iii. Select the security provider in which this user or group is defined.
    - iv. Click **Add**.
  - To revoke access to the source for an account that is listed in the **Allowed Users** box, select the account, and then click **Remove**.
  - To revoke access to the source for an account that is not listed in the **Allowed Users** box, in the **Denied**

**Users** section:

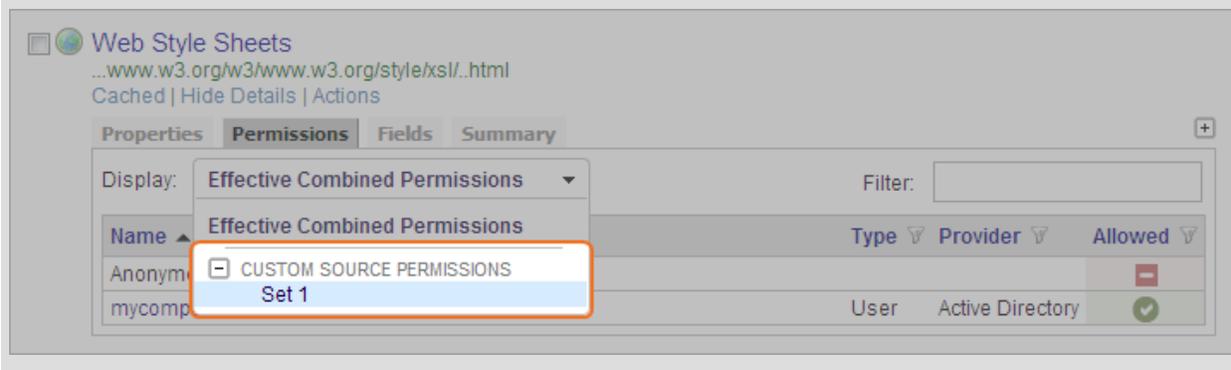
- i. Enter the account name of the user or group.

**Example:** For an Active Directory account, enter the name in the `domain\username` form.

- ii. Select if the entered name is a **User** or a **Group**.
- iii. Select the security provider in which this user or group is defined.
- iv. Click **Add**.

- 6. Click **Apply Changes**.

**Tip:** **CES 7.0.5388+ (April 2013)** When you add source level permissions, these permissions are automatically assigned to a **Custom Source Permissions** level that is visible from the Index Browser.



### 8.1.2 Mapping the security permissions of a Web source

You can map Web Page sources with local files to indexing of document-level security permissions for these sources.

**Example:** If `http://www.coveo.com` is mapped with its equivalent folder on `\\CoveoServer\WebPage\`, the permissions granted to the files in the `\\CoveoServer\WebPage\` folder are also indexed for the **Web Pages** sources.

- 1. On the Coveo server, access the Administration Tool.
- 2. In the **Permissions** page corresponding to the source is displayed, click **Add**.
- 3. In the **Edit Web File Security** page:
  - a. In the **Web Address** box, enter the address of the Web source.

**Example:** `http://www.coveo.com`

- b. In the **Network File Path** box, enter the path of the folder containing the Web files.

**Example:** \\CoveoServer\WebPage\.

- c. Click **Save**.

## 8.2 Modifying Hidden Salesforce Source Parameters for the Legacy Connector

### Deprecated

The **Add Source** and **Source: ... General** pages of the Administration Tool present the parameters with which you can configure the legacy connector for most Salesforce setups. More advanced and more rarely used parameters are hidden. You can choose to make one or more of these parameters appear in the **Add Source** and **Source: ... General** pages of the Administration Tool so that you can change their default value. Consider changing values of hidden parameters only when you encounter time out error messages or performance issues.

**Note:** The Salesforce legacy connector uses the open connector framework and can therefore be modified using custom parameters such as `liveindexingdelay` (see [Connector API Home](#)).

The following list describes the available advanced hidden parameters for Salesforce sources. The parameter type (integer, string,...) appears between parentheses following the parameter name.

#### Timeout (String)

The communication timeout allowed for any request sent to the Salesforce Web services. The default value is 100 seconds.

#### NbRefreshThreads (string)

The number of threads used to handle the indexing operation of the Salesforce repository. The default value is 1.

#### FilterExpression (string)

The list of conditions to add to any query sent to Salesforce. These conditions are appended to the `WHERE` clause of the Salesforce Object Query Language (SOQL) statement sent via the Web services. There is no default value.

**Example:** `ParentId='5007000000JQpLnAAL'` indexes all items that have the specified parent ID.

#### CustomLoginUri (string)

The URI used to login to Salesforce Web services. The default value is `https://login.salesforce.com/`.

#### CustomLoginUriVersion (string)

The API version of Salesforce Web services to connect to, specified in the `##.#` format. The default value is `23.0`.

#### CustomBaseUri (string)

The base URI used for indexed documents. In other words, this forces the `Uri` and `ClickableUri` properties to have the specified base URI. The default value is `https://na1.salesforce.com/`.

### FixXmlNamespace (Boolean)

Whether empty XML namespaces received in Web service responses must be fixed before processing. The default value is `False`.

### QueryLimit (Integer)

The maximum number of results returned by a query to Salesforce. The default value is 5000 items.

### QueryBatchSize (Integer)

The maximum number of results contained in a query result object (`QueryMore`). The default value is 200 items.

### ForeignCacheSize (Integer)

The B-tree cache size used by the foreign cache. The default value is 3145728 bytes.

### ForeignPageSize (Integer)

The B-tree page size used by the foreign cache. The default value is 65536 bytes.

### PrefetchCacheSize (Integer)

The B-tree cache size used by the prefetch cache. The default value is 1000 items.

### ExcludeConvertedLeads (Boolean)

When `True`, leads that are converted will not be indexed. The default value is `True`.

### ExcludeDeletedItems (Boolean)

When `True`, items that are marked as "IsDeleted" will not be indexed. The default value is `True`.

Use the following procedure only when you want to modify one or more of the above hidden source parameters.

### To modify hidden Salesforce source parameters

1. Refer to ["Adding an Explicit Connector Parameter" on page 30](#) to add one or more Salesforce hidden source parameters.
2. For a new Salesforce source, access the **Add Source** page of the Administration Tool to modify the value of the newly added advanced parameter:
  - a. Select **Index > Sources and Collections**.
  - b. Under **Collections**, select the collection in which you want to add the source.
  - c. Under **Sources**, click **Add**.
  - d. In the **Add Source** page, edit the newly added advanced parameter value.
3. For an existing Salesforce source, access the **Source: ... General** page of the Administration Tool to modify the value of the newly added advanced parameter:
  - a. Select **Index > Sources and Collections**.
  - b. Under **Collections**, select the collection containing the source you want to modify.

- c. Under **Sources**, click the existing Salesforce source in which you want to modify the newly added advanced parameter.
- d. In the **Source: ... General** page, edit the newly added advanced parameter value.

### 8.3 Adding an Explicit Connector Parameter

Connector parameters applying to all sources indexed using this connector are called explicit parameters.

When you create or configure a source, the Coveo Enterprise Search (CES) 7.0 Administration Tool presents parameters with which you can configure the connector for most setups. For many connectors, more advanced and more rarely used parameters also exist but are hidden by default. CES then uses the default value associated with each of these hidden parameters.

You can however choose to make one or more of these parameters appear in the **Add Source** and **Source: ... General** pages of the Administration Tool so that you can change their default value.

To add an explicit connector parameter

1. On the Coveo server, access the Administration Tool.
2. Select **Configuration > Connectors**.
3. In the list on the **Connectors** page, select the connector for which you want to show advanced hidden parameters.
4. In the **Parameters** section of the selected connector page, click **Add Parameter** for each hidden parameter that you want to modify.

**Note:** The **Add Parameter** button is present only when hidden parameters are available for the selected connector.

5. In the **Modify the parameters of the connector** page:

The screenshot shows the 'Configuration' tab in the Coveo Administration Tool. The main heading is 'MODIFY THE PARAMETERS OF THE CONNECTOR. Defines the parameters of this connector.' Below this is a breadcrumb 'Connectors - Active...'. The form contains the following fields:

- Type:** A dropdown menu currently set to 'String'.
- Name:** A text input field with a help icon.
- Default Value:** A text input field with a help icon.
- Label:** A text input field with a help icon.
- Quick Help:** A text input field with a help icon.
- Option:** A section containing:
  - Optional parameter
  - Sensitive information [?](#)
  - Validate as an email address
  - Maximum length:  [?](#)

At the bottom right of the form are 'Save' and 'Cancel' buttons.

- In the **Type** list, select the parameter type as specified in the parameter description.
- In the **Name** box, type the parameter name exactly as it appears in the parameter description. Parameter names are case sensitive.
- In the **Default Value** box, enter the default value specified in the parameter description.

**Important:** Do not set the value that you want to use for a specific source. The value that you enter here will be used for all sources defined using this connector so it must be set to the recommended default value. You will be able to change the value for each source later, in the **Add Source** and **Source: ... General** pages of the Administration Tool.

- In the **Label** box, enter the label that you want to see for this parameter.

**Example:** To easily link the label to the hidden parameter, you can simply use the parameter name, and if applicable, insert spaces between concatenated words. For the **BatchSize** hidden parameter, enter `Batch Size` for the label.

**Note:** To create multilingual labels and quick help messages, use the following syntax: `<@ln>text</@>`, where *ln* is replaced by the language initials—the languages of the Administration Tool are English (en) and French (fr).

**Example:** `<@fr>Chemin d'accès du fichier de configuration</@><@en>Configuration File Path</@>` is a label which is displayed differently in the French and English versions of the Administration Tool.

**Tip:** The language of the Administration Tool can be modified by pressing the following key combination: Ctrl+Alt+Page Up.

- e. Optionally, in **Quick Help**, enter the help text that you want to see for this parameter when clicking the question mark button  that will appear beside the parameter value.

**Tip:** Copy and paste key elements of the parameter description.

- f. When **Predefined values** is selected in the **Type** parameter, in the **Value** box that appears, enter the parameter values that you want to see available in the drop-down parameter that will appear in the Administration Tool interface. Enter one value per line. The entered values must exactly match the values listed in the hidden parameter description.
- g. Select the **Optional parameter** check box when you want to identify this parameter as an optional parameter. When cleared, CES does not allow you to save changes when the parameter is empty. This parameter does not appear for **Boolean** and **Predefined values** parameter types.
- h. Select the **Sensitive information** check box for password or other sensitive parameter so that, in the Administration Tool pages where the parameter appears, the typed characters appear as dots to mask them. This parameter appears only for the **String** type.

**Example:** When you select the **Sensitive information** check box for a parameter, the characters typed appear as follows in the text box:



- i. Select the **Validate as an email address** check box when you want CES to validate that the text string that a user enters in this parameter respects the format of a valid email address. This parameter appears only for the **String** type.
  - j. In the **Maximum length** box, enter the maximum number of characters for the string. This parameter appears only for the **String** type. When you enter 0, the length of the string is not limited.
  - k. Click **Save**.
6. Back in the **Connector** page, click **Apply Changes**.

The hidden parameter now appears in the **Add Source** and **Source: ... General** pages of the Administration Tool for the selected source. You can change the parameter value from these pages. Refer to the documentation for each connector for details.

**Note:** When you want to modify a hidden source parameter, you must first delete it, and then redefine it with the modified values.

## 9. Adding Search Interfaces to a Search Hub With the .NET Interface Editor

A search hub contains one or more Coveo .NET Front-End search interfaces. As a Coveo administrator, you can add search interfaces to a search hub.

For each .NET search interface, you can specify if the search interface can receive or send results from/to other search interfaces of the search hub through mini queries. The mini queries appear as mini-results. You can also assign audiences to each search interface so that only members of the selected audiences have access to the search interface.

To add a search interface with the .NET Interface Editor

1. Access the Coveo .NET Front-End Interface Editor.
2. Select **Search Hubs > Content**.
3. On the menu, in the **Current Hub** drop-down list, select the hub to which you want to add a search interface.
4. In the panel on the left, select the hub version to which you want to add a search interface (**Standard Version**, **Mobile Version**, or **Outlook Version**).
5. Click **Add**.
6. Under **Edit Hub Content**:

- a. In the **Interface** drop-down list, select the search interface that you want to add to the search hub.
- b. Select the **Enable mini queries from this search interface** check box to allow this interface to provide search highlights to produce mini-results in other search interfaces.
- c. Select the **Enable mini queries results in this search interface** check box to allow displaying search

highlights as mini-results from other search interfaces.

- d. Select the **Overridable by Per Uri settings or query string arguments** check box in cases where you want to be able to change the search interface by passing the search interface name via a query or using per URI settings. When selected, at loading time, a search interface looks at the query string and uses the specified search interface rather than its normal configuration.

**Note:** This override is a legacy feature that was used to switch between search interfaces before search hubs were introduced. In SharePoint, it can still be useful to switch between search interfaces as a function of the SharePoint site.

- e. When you want to restrict the access to this search interface to a specific group of users, in the **Available Audiences** list, select one or more audiences that are allowed to access this interface, and then click  to move one or more selected audiences to the **Selected Audiences** list.
- f. Click **OK**.