# Coveo Platform 7.0

## EPiServer CMS 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.

Document part number: PM-120817-EN

Publication date: 1/3/2019

## Table of Contents

**1. EPiServer CMS Connector** 1

1.1 Features 1

**2. EPiServer CMS Connector Deployment Overview** 2

**3. EPiServer CMS Connector Requirements** 3

**4. Getting the EPiServer CMS Connector Files** 4

**5. Adding a Connector** 5

**6. Configuring the Coveo Web Service on an EPiServer CMS Server** 8

**7. Adding a User Identity** 11

**8. Configuring and Indexing an EPiServer CMS Source** 13

8.1 Configuring the EPiServer CMS Source to Crawl the File Manager 16

8.2 Creating and Using an EPiServer CMS Configuration File 18

8.3 Creating and Using an EPiServer CMS Mapping File 19

8.4 Modifying Hidden EPiServer CMS Source Parameters 21

8.5 Adding an Explicit Connector Parameter 22

# 1. EPiServer CMS Connector

Deprecated

The Coveo prototype connector for EPiServer CMS systems allows you to index the content of one or more EPiServer CMS servers, integrating that content into the Coveo unified index, and making it easily searchable by users.

**Note:** As of March 2, 2017, the EPiServer CMS connector is deprecated (see What Does Deprecated Mean for a Coveo Connector?).

## 1.1 Features

The following details the features available in the EPiServer CMS connector:

**Content Indexing**

The connector can retrieve and index exclusively the content of the following EPiServer CMS elements:

- Pages
- Documents
- Downloadable files available from the File Manager

### Limitations

This prototype connector does not support:

- Incremental refresh (but does support scheduled source refresh)
- Permissions

### What's Next?

Review the deployment process (see "EPiServer CMS Connector Deployment Overview" on page 2).

# 2. EPiServer CMS Connector Deployment Overview

The following procedure outlines the tasks needed to deploy the Coveo prototype connector for EPiServer CMS systems. The tasks indicate the order in which you must perform key configurations on both the EPiServer CMS and Coveo systems. When needed, the steps refer to a detailed procedure.

## To deploy the EPiServer CMS connector

1. Validate that your environment meets the requirements (see "EPiServer CMS Connector Requirements" on page 3).

2. Get and install the EPiServer CMS prototype connector files.

   The connector files are not distributed with the CES installation (see "Getting the EPiServer CMS Connector Files" on page 4).

3. On the EPiServer CMS system, configure the Coveo Web service.

   The connector uses Coveo Web service files that you must copy on the EPiServer CMS server (see "Configuring the Coveo Web Service on an EPiServer CMS Server" on page 8).

4. Configure the user identity.

   The connector needs an account to connect to the EPiServer CMS system and access the entire content that you want to index (see "Adding a User Identity" on page 11).

5. Configure and index the EPiServer CMS source.

   The connector needs to know details about the EPiServer CMS source to be able to index its content (see "Configuring and Indexing an EPiServer CMS Source" on page 13).

6. Optional tasks:

   a. Configure the connector to index the content of the EPiServer CMS File Manager.

      You can also index the content of the downloadable files stored in the EPiServer CMS File Manager by specifying which starting addresses to crawl (see "Configuring the EPiServer CMS Source to Crawl the File Manager" on page 16).

   b. Create and use a configuration file.

      You can instruct the connector to index linked files using a configuration file (see "Creating and Using an EPiServer CMS Configuration File" on page 18).

   c. Create and use a custom mapping file.

      You can customize how the connector maps metadata to Coveo fields using a custom mapping file (see "Creating and Using an EPiServer CMS Mapping File" on page 19).

   d. Add hidden source parameters.

      You can use a few hidden source parameters to fine tune how the connector operates (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

# 3. EPiServer CMS Connector Requirements

Your environment must meet the following requirements to be able to use the Coveo connector for EPiServer CMS systems:

- Coveo EPiServer CMS Connector package

  The files associated with the Coveo prototype connector for EPiServer CMS systems are not available with the default CES installation. Contact the Coveo Support to get the EPiServer CMS package.

- EPiServer CMS version 6

  The connector was developed and tested with EPiServer CMS version 6.1.

## What's Next?

Add the connector to CES (see "Getting the EPiServer CMS Connector Files" on page 4).

# 4. Getting the EPiServer CMS Connector Files

The Coveo prototype connector for EPiServer CMS systems is not available by default with the default CES installation. Because it is a prototype, the files associated with the connector are not distributed with the CES installation. Before being able to use the connector, you must get the EPiServer CMS files, copy associated files to your CES implementation, and let CES know where to find them.

## To add the EPiServer CMS connector to CES

1. Contact the Coveo Support , to get the EPiServer CMS files specifying if you are using a 32-bit or 64-bit server:
   - `Coveo.CES.CustomCrawlers.EpiServer.dll`
   - `Coveo.Connectors.EpiServer.dll`
   - `Coveo.Connectors.EpiServer.CoveoService.zip` that contains:
     - `CoveoService.asmx`
     - `Coveo.Connectors.EpiServer.CoveoService.dll`

       **Note:** For versions prior to CES 7.0.4897 (December 2012 monthly release), the file name was `CoveoService.dll`.
2. Copy the following files:
   - `Coveo.CES.CustomCrawlers.EpiServer.dll`
   - `Coveo.Connectors.EpiServer.dll`
3. On the Coveo Master server, paste the two files to the `[CES_Path]\Bin` folder.
4. Add the EPiServer CMS connector to the list of CES connectors using the following information in the **Modify Additional Connector** page (see "Adding a Connector" on page 5):
   - **Name**: `EPiServer`
   - **Assembly Path**: `Coveo.CES.CustomCrawlers.EpiServer.dll`
   - **Type Name**: `Coveo.CES.CustomCrawlers.EpiServer.EpiServerCrawler`
   - **Run in 64 bits**: On a 64-bit server, when you use the 64-bit version of the connector files, ensure to select this check box.
   - **Parameter**: Apart from required parameters (see "Adding a Connector" on page 5), you can also expose some hidden parameters (such as the optional `ConfigFile` and `MappingFile`) by adding them here. You can get the parameter types and names from the list of hidden parameters (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

## What's Next?

Configure the Web service on the EPiServer CMS server (see "Configuring the Coveo Web Service on an EPiServer CMS Server" on page 8).

# 5. Adding a Connector

A number of connectors are available out-of-the-box with the Coveo Platform. You can also add other custom or prototype connectors to your CES configuration to be able to index other repositories and systems.

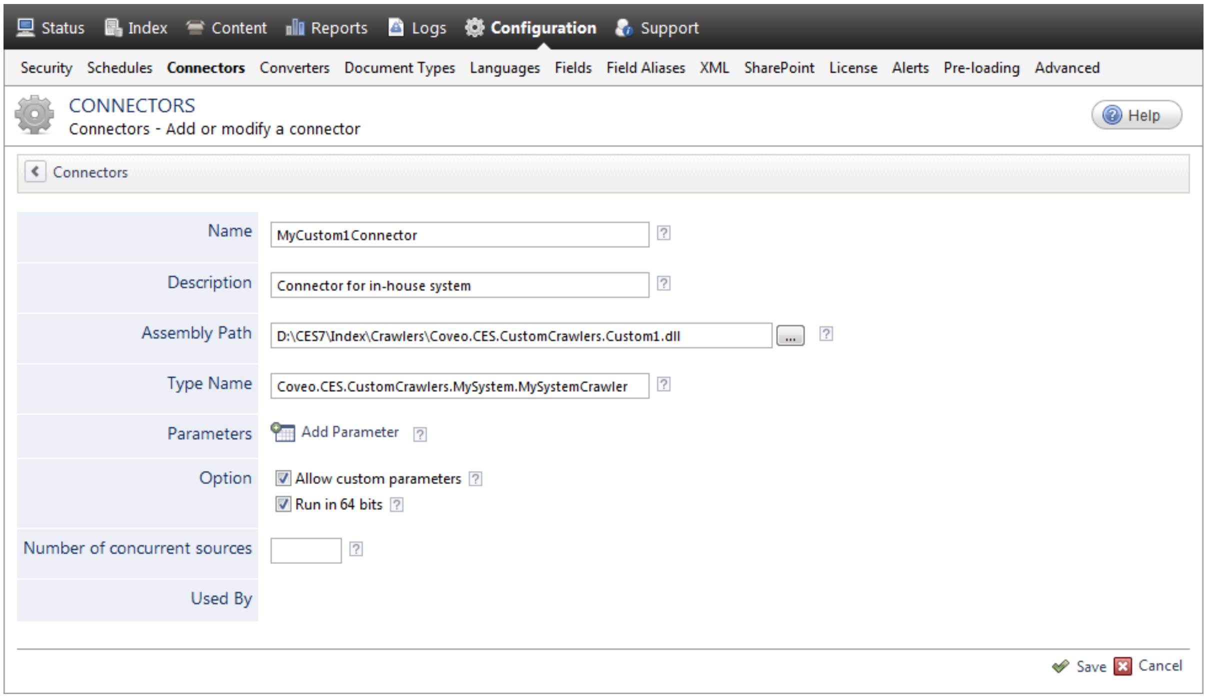
**Example:** Your in-house programmers, Coveo Professional Services, or a third party, developed a connector for your custom CMS system so that you can bring its content into your Coveo unified index.

**Note:** You can also contact the Coveo Professional Services to get assistance to develop your own connectors for example for in-house custom repositories/systems using the Open Connector API.

## To add an additional connector

1. Get the connector files, ensuring that they are compiled for your version and release of CES. Ask for the name of the class implementing the connector.
   - For a documented prototype connector, contact Coveo Support.
   - For a custom connector, contact whoever developed the connector (Coveo Professional Services, in-house, or third-party developer).
2. Using an administrator account, connect to the Coveo Master server.
3. Copy the connector files to the `[CES_Path]\Bin` folder.

   **Example:** On a 64-bit server, copy 64-bit versions of the connector files to the `C:\Program Files\Coveo Enterprise Search 7\Bin` folder. If you have a 32-bit version of the connector files, copy them to the `C:\Program Files\Coveo Enterprise Search 7\Bin\Win32` folder.
4. On the Coveo server, access the Administration Tool.
5. Select **Configuration** > **Connectors**.
6. In the **Connectors** page, the currently available connectors are listed. Click **Add**.
7. In the navigation panel on the left, click **Additional Connector**.
8. In the **Additional Connectors** page, click **Add**.
9. In the **Connectors** page, enter appropriate values for the following parameters:



CONNECTORS

Connectors - Add or modify a connector

| | |
|---|---|
| Name | MyCustom1Connector |
| Description | Connector for in-house system |
| Assembly Path | D:\CES7\Index\Crawlers\Coveo.CES.CustomCrawlers.Custom1.dll |
| Type Name | Coveo.CES.CustomCrawlers.MySystem.MySystemCrawler |
| Parameters | Add Parameter |
| Option | ☑ Allow custom parameters<br>☑ Run in 64 bits |
| Number of concurrent sources | |
| Used By | |

**Name**

A name of your choice to identify the additional connector.

**Description**

A description of the type of repository indexed by the connector.

**Assembly Path**

The full path of the additional connector assembly file.

**Example:** `C:\Program Files\Coveo Enterprise Search 7\Bin\Crawlers\Coveo.CES.CustomCrawlers.Custom1.dll`

**Note:** If the path is incomplete, CES searches for the assembly file in its `[CES_Path]\bin` directory (by default, `C:\Program Files\Coveo Enterprise Search 7\Bin`).

**Type Name**

The name of the class implementing the connector.

**Note:** The parameter is optional when only one class is inheriting from the `CustomCrawlers` class. When this is the case, you can leave the box empty.

**Example:** `Coveo.CES.CustomCrawlers.ODBC.ODBCCrawler`

**Parameters**

Allows you to define explicit parameters. Click **Add Parameter** to display the **Modify the parameters of the**

**additional connector** page (see "Adding an Explicit Connector Parameter" on page 22).

You must add the following parameters to get the **Security** section in the **Add Source** page for sources of this type and allow you to specify the security provider and user identity that the source will use.

| Type | Name | Label |
|---|---|---|
| Security Provider | `SecurityProvider` | Security Provider |
| User Identity | `UserIdentity` | Authentication |

**Options**

For 64-bit servers, select the **Run in 64 bits** check box to instantiate the connector in a 64-bit process.

**Number of concurrent sources**

Specify the maximum number of sources using this connector that can operate concurrently.

10. Click **Save**.

# 6. Configuring the Coveo Web Service on an EPiServer CMS Server

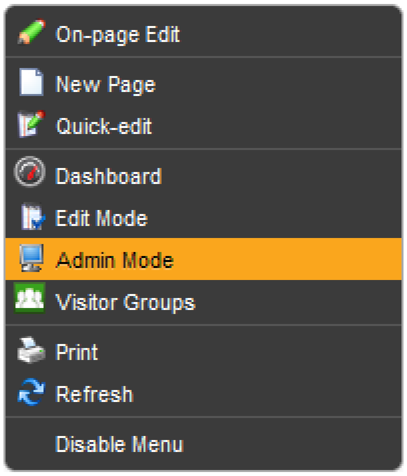
The Coveo connector for EPiServer CMS systems uses Web services that are inactive by default on an EPiServer CMS system. You must activate them for the account that you want to use to crawl the EPiServer CMS content. You must also deploy the Coveo Web service files on the EPiServer CMS server.

## To configure the Coveo Web service for an EPiServer CMS server

1. Activate the Web service:

   a. Using an administrator account, log in to the EPiServer CMS site.

   b. Right-click anywhere on the content, and then select **Admin Mode** in the contextual menu.



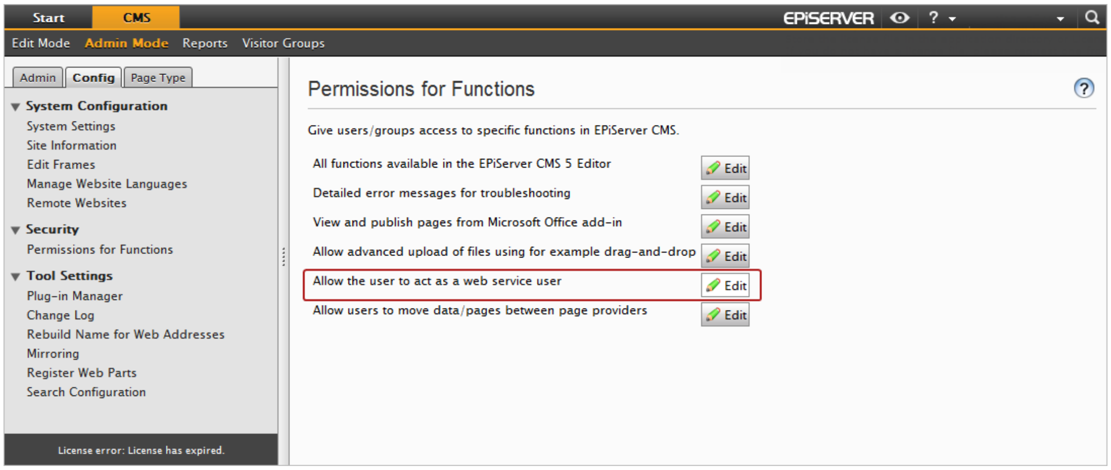
   c. In the panel on the left:

      i. Click the **Config** tab.

      ii. Under **Security**, click **Permissions for Functions**.

   d. In the **Permissions for Functions** panel on the right:

i. Click **Edit** next to **Allow the user to act as a web service user**.



Permissions for Functions

Give users/groups access to specific functions in EPiServer CMS.

| | |
|---|---|
| All functions available in the EPiServer CMS 5 Editor | Edit |
| Detailed error messages for troubleshooting | Edit |
| View and publish pages from Microsoft Office add-in | Edit |
| Allow advanced upload of files using for example drag-and-drop | Edit |
| Allow the user to act as a web service user | Edit |
| Allow users to move data/pages between page providers | Edit |

ii. Click **Add Users/Groups**.

e. In the **Add Users/Groups** dialog box, add the account that you selected to use to crawl the EPiServer CMS content, and then click **OK**.

f. Back in the **Permissions for Functions** panel on the right, click **Save**.

2. Deploy the Coveo Web service files on the EPiServer CMS server:

   a. From the EPiServer CMS package, copy the `CoveoService.asmx` file.

   b. Paste the file in the `WebServices` folder of the EPiServer CMS installation (not on the Website). This file basically points to the `CoveoService.dll`, which will be loaded in the Web services of the EPiServer CMS site.

   **Example:** When EPiServer CMS is installed using the default installation folder, the full path of the file is similar to `C:\Program File (x86)\EPiServer\CMS\6.1.379.0\Application\WebServices\CoveoService.asmx`.

   **Note:** You can save the `CoveoService.asmx` in another folder and use the `WebServiceUrl` hidden parameter to specify the folder (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

   c. From the EPiServer CMS package, copy the `CoveoService.dll` file.

   d. Paste the file in the `bin` folder of the EPiServer CMS website that you want to crawl.

   **Example:** When EPiServer CMS sites are installed using default installation folders, the full path to the file is similar to `C:\EPiServer\Sites\MyEPiServerSite\bin\CoveoService.dll`.

   e. On the EPiServer CMS server, reset IIS.

## What's Next?

The Coveo connector needs an account to connect to the EPiServer CMS system and access the entire content that you want to index (see "Adding a User Identity" on page 11).

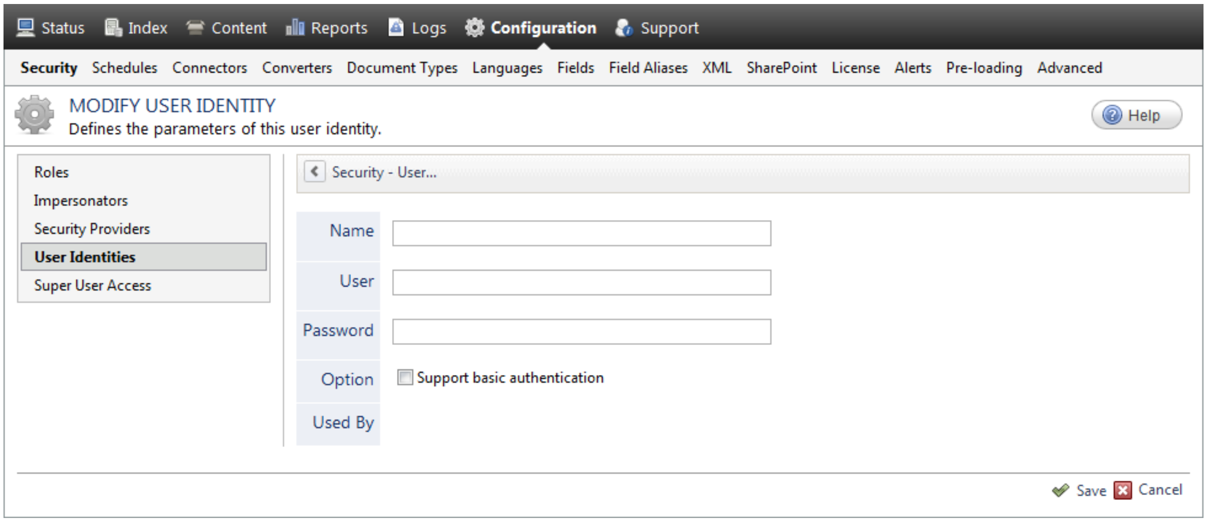
# 7. Adding a User Identity

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:



MODIFY USER IDENTITY

Defines the parameters of this user identity.

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.

# 8. Configuring and Indexing an EPiServer CMS Source

A source defines a set of configuration parameters for a specific EPiServer CMS server.

**Note:** In an environment with more than one EPiServer CMS servers, define one source for each EPiServer CMS server that you want to index.

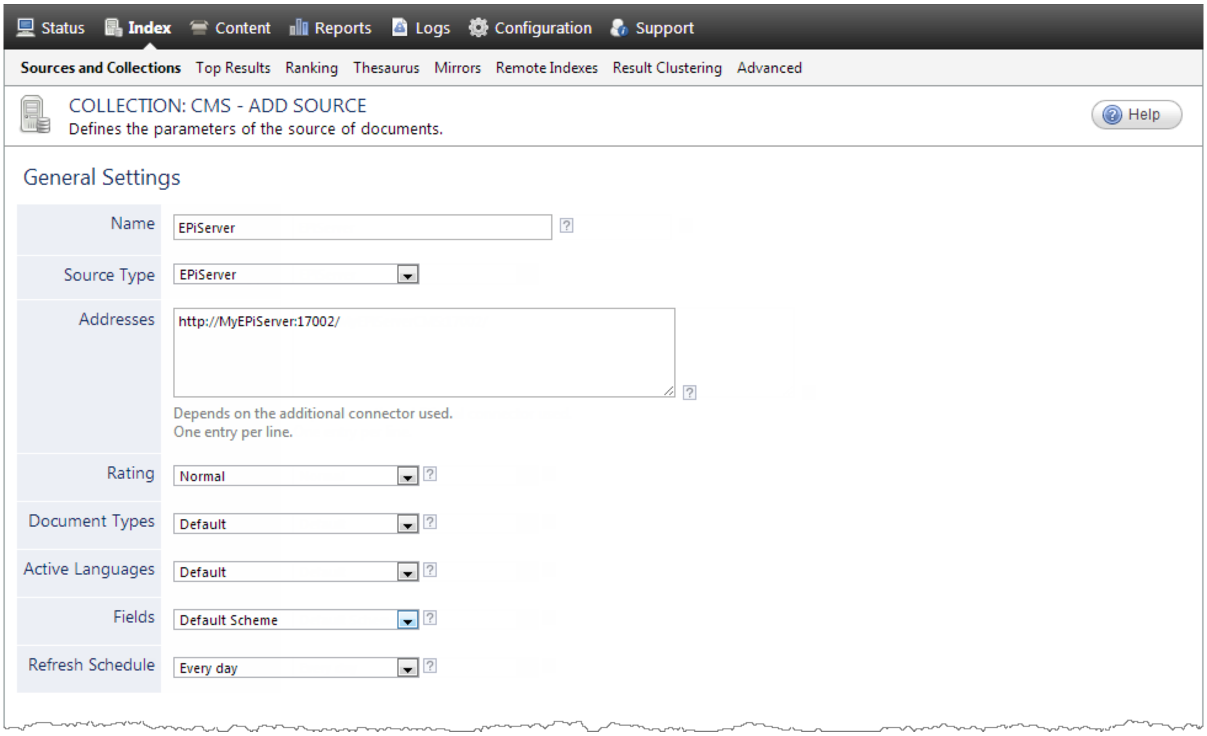
## To configure and index an EPiServer CMS source

1. On the Coveo server, access the Administration Tool.
2. Select **Index** > **Sources and Collections**.
3. In the **Sources** section, click **Add**.
4. 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.
5. In the **General Settings** section of the **Add Source** page:



COLLECTION: CMS - ADD SOURCE

Defines the parameters of the source of documents.

General Settings

| | |
|---|---|
| Name | EPiServer |
| Source Type | EPiServer |
| Addresses | http://MyEPiServer:17002/<br>Depends on the additional connector used.<br>One entry per line. |
| Rating | Normal |
| Document Types | Default |
| Active Languages | Default |
| Fields | Default Scheme |
| Refresh Schedule | Every day |

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

**Name**

A descriptive name of your choice for the connector source.

> **Example:** `EPiServer CMS`

**Source Type**

The connector used by this source. In this case, select **EPiServer**.

> **Note:** If you do not see the **EPiServer** option, ensure that you obtained the connector files and added the connector to CES (see "Getting the EPiServer CMS Connector Files" on page 4).

**Addresses**

The root address of the EPiServer CMS application in the `http://[EPiServer-host]:[port]/` form.

> **Example:** `http://MyEpiServer:17002/`

b. The following parameters generally do not need to be changed:

**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 a 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 legacy system sources.

**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.

**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.

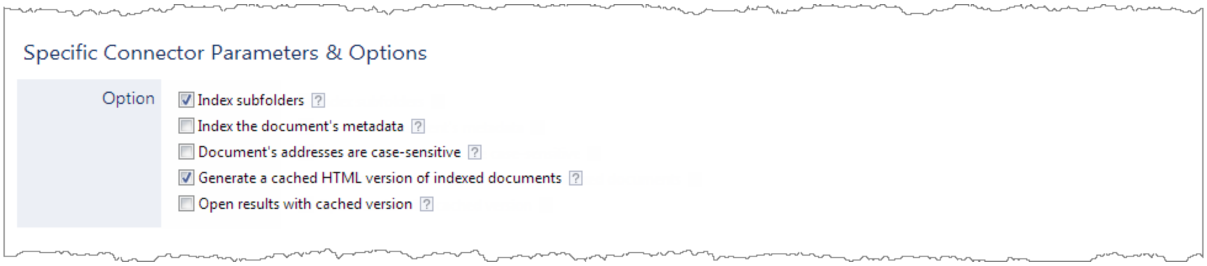
Because incremental refresh is not available for the EPiServer CMS prototype connector, ensure to select an appropriate source refresh schedule, as this is the only mechanism that ensures that the index content is kept up-to-date.

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

**Parameter**

Click **Add Parameter** when you want to show advanced source parameters (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

6. In the **Specific Connector Parameters & Options** section of the **Add Source** page, review the **Option** check boxes that generally do not need to be changed:



Specific Connector Parameters & Options

Option
- ☑ Index subfolders
- □ Index the document's metadata
- □ Document's addresses are case-sensitive
- ☑ Generate a cached HTML version of indexed documents
- □ Open results with cached version

**Index Subfolders**

Keep this check box selected (recommended). By doing so, all subfolders from the specified portal 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.

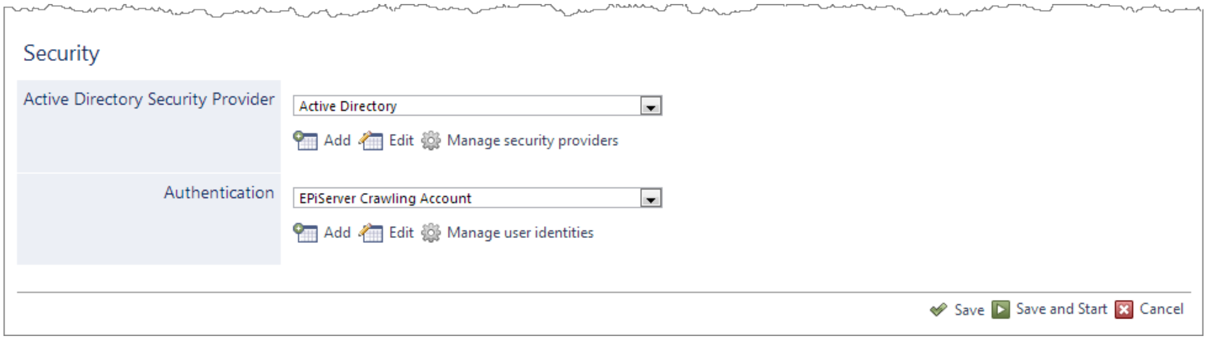
**Generate a cached HTML version of indexed documents**

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:



Security

Active Directory Security Provider: Active Directory

Authentication: EPiServer Crawling Account

   a. In the **Security Provider** drop-down list, select an Active Directory security provider.

   b. In the **Authentication** drop-down list, select the user identity that you created for the EPiServer source (see "Adding a User Identity" on page 11).

   c. Click **Save**.

8. In the navigation panel on the left, click **General**.

9. In the **General** page:

   a. Modify the **Title Selection Sequence** so that the **Use the filename** option is the first option at the top of the list.

   b. Click **Apply Changes**.

10. When you are ready to start indexing the EPiServer CMS source, click **Start**.

11. In the navigation panel on the left, click **Status**, and then validate that the indexing process runs without errors.

## 8.1 Configuring the EPiServer CMS Source to Crawl the File Manager

The File Manager is an EPiServer CMS section where all the files uploaded to the EPiServer CMS installation reside. It is a virtual file system hosted in the Web server. The Coveo connector can access these files through the Coveo Web service.

**Example:** You may want to crawl and index the content of the File Manager because it contains files that are not referenced from any pages so that users will be able to search for their content.

You instruct the connector to index the File Manager by adding the `FileManagerConfigurationString` hidden parameter to the EPiServer CMS source. The value of this parameter is the configuration string that contains the list of starting points to crawl in the File Manager.

Each starting point represents a virtual path provider, which gives access to the files it hosts. In the configuration string, each starting point is composed of two values:

- Virtual path provider `Name`
- Virtual path provider `VirtualPath`

You can find these values in the `episerver.config` file associated with the EPiServer CMS website that you are crawling.

**Note:** When you want to index only downloadable documents referred from links in pages, you should rather use an optional configuration file to instruct the connector to do so (see "Creating and Using an EPiServer CMS Configuration File" on page 18).

## To configure the EPiServer CMS source to crawl the File Manager

1. Identify the starting points that you want to crawl in the EPiServer CMS File Manager:

   a. Using a text editor, open the `episerver.config` configuration file for the EPiServer website that you are crawling.

      **Example:** For the *MyEPiServerSite* site the file can be in `C:\EPiServer\Sites\MyEPiServerSite\episerver.config`.

   b. In the configuration file, look for the `<virtualPath>` node.

   c. Inside this node, look for the `<providers>` node that contains `<add ...>` nodes, one for each starting point.

   d. For each starting point that you want to crawl note the values of the `Name` and `VirtualPath` attributes.

2. Build the configuration string specifying the starting points to crawl:

   a. For each starting point, directly concatenate the values of the `Name` and `VirtualPath` attributes.

   b. Separate starting point strings by a semicolon character.

**Example:** The default starting point *Documents* in EPiServer CMS has the following `Name=SiteDocuments` and `VirtualPath=~/Documents/` attribute values. The default starting point *Global Files* has the `Name=SiteGlobalFiles` and `VirtualPath=~/Globabl/` attribute values. The resulting string to crawl both starting points is: `SiteDocuments~/Documents;SiteGlobalFiles~/Global/`.

3. Add the `FileManagerConfigurationString` hidden parameter to the EPiServer CMS source and use the string built in the previous step for the parameter value (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

## 8.2 Creating and Using an EPiServer CMS Configuration File

You can optionally create and use a configuration file with an EPiServer CMS source. The purpose of the configuration file is to specify fields that are links to downloadable documents that you want to index. Without a configuration file, the connector crawls the content of the pages but not the content of downloadable documents linked from the pages.

**Note:** An alternative to using a configuration file to crawl downloadable documents is to crawl the EPiServer CMS File Manager content (see "Configuring the EPiServer CMS Source to Crawl the File Manager" on page 16).

### To create and use an EPiServer CMS configuration file

1. Using an administrator account, connect to the Coveo Master server.
2. Using a text editor:
   a. Create an XML file respecting the format illustrated in the following commented example.

```
<episerverdocumentconfig>
  <!-- This is the type of item. For now, this config only applies for type "page" -->
  <page>
    <!-- This is the type of pages, as defined within EPiServer -->
    <pagetype>
      <!-- The value of pagetype -->
      <string>[AlloyTech] Document</string>
    </pagetype>
    <fieldstodownload>
      <!-- this is the list of fields that the connector attempts to download as documents. --
>
      <fieldnames>
        <!-- This is the value of the metadata contained in the page that allows to retrieve
the document. -->
        <!-- Generally this takes the form of a virtual path in the file manager. -->
        <string>DocumentInternalPath</string>
      </fieldnames>
    </fieldstodownload>
  </page>
</episerverdocumentconfig>
```

   b. Save the file using a name of your choice in the `[Index_Path]\Config` folder.

   **Example:** `C:\CES7\Config\MyEPiServerCMSConfig.xml`

3. To instruct the source to use this configuration file, add the `ConfigFile` hidden parameter to the EPiServer

CMS source and use the filename and path where you saved the file as the value for the parameter (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

## 8.3 Creating and Using an EPiServer CMS Mapping File

The Coveo connector for EPiServer CMS system uses a built-in mapping to determine what metadata from your original documents are associated with fields for the documents in the Coveo index. The content of the built-in mapping file is presented below.

```
<?xml version="1.0" encoding="utf-8" ?>
<EpiServerMapping>
  <Mapping type="Default">
    <Fields>
      <Title>%[coveo_Title]</Title>
      <ModifiedDateUtc>%[coveo_DateModified]</ModifiedDateUtc>
      <CustomFields>
        <CustomField name="sysauthor">%[coveo_CreatedBy]</CustomField>
      </CustomFields>
    </Fields>
  </Mapping>
  <Mapping type="Page">
    <Fields>
      <ClickableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</ClickableUri>
      <PrintableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</PrintableUri>
      <Body>
        <html><body>
        %[MainBody]
        %[SecondaryBody]
        %[ThirdBody]
        </body></html>
      </Body>
      <CustomFields>
        <CustomField name="pagetype">%[coveo_TypeName]</CustomField>
      </CustomFields>
    </Fields>
  </Mapping>
  <Mapping type="Document">
    <Fields>
      <ClickableUri>%[coveo_ServerUrl]%[coveo_VirtualPath]</ClickableUri>
      <PrintableUri>%[coveo_ServerUrl]%[coveo_VirtualPath]</PrintableUri>
    </Fields>
  </Mapping>
</EpiServerMapping>
```

You can optionally create and use a custom mapping file to tailor the mapping to your needs.

### To create and use a custom EPiServer CMS mapping file

1. Using an administrator account, connect to the Coveo Master server.

2. Using a text editor:

   a. Create an XML file respecting the format illustrated in the following commented example.

```
<?xml version="1.0" encoding="utf-8" ?>
<EpiServerMapping>
  <!-- These fields will be applied to all the documents that have the mappings defined later.
-->
  <CommonMapping>
    <Fields>
```

```
      <Title>%[coveo_Title]</Title>
      <ModifiedDateUtc>%[coveo_DateModified]</ModifiedDateUtc>
      <CustomFields>
        <CustomField name="sysauthor">%[coveo_CreatedBy]</CustomField>
      </CustomFields>
    </Fields>
  </CommonMapping>
  <Mapping type="Default">
    <Fields>
      <Title>%[coveo_Title]</Title>
      <ModifiedDateUtc>%[coveo_DateModified]</ModifiedDateUtc>
      <CustomFields>
        <CustomField name="sysauthor">%[coveo_CreatedBy]</CustomField>
      </CustomFields>
    </Fields>
  </Mapping>
  <!-- It is valid to put the pagetype of an item to set metadata on. -->
  <Mapping type="[AlloyTech] News item">
    <Fields>
      <Title>Grosse Ventouse</Title>
      <ClickableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</ClickableUri>
      <PrintableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</PrintableUri>
      <Body>
        <html><body>
        %[MainBody]
        %[SecondaryBody]
        %[ThirdBody]
        </body></html>
      </Body>
      <CustomFields>
        <CustomField name="sysauthor">JoeBeton</CustomField>
        <CustomField name="pagetype">%[coveo_TypeName]</CustomField>
      </CustomFields>
    </Fields>
  </Mapping>
  <!-- This is the type of mapping defined by the connector. "Page" comprises all pages of
EPiServer that were not matched by their paged type, above. -->
  <Mapping type="Page">
    <Fields>
      <ClickableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</ClickableUri>
      <PrintableUri>%[coveo_ServerUrl]%[coveo_StaticLinkUrl]</PrintableUri>
      <Body>
        <html><body>
        %[MainBody]
        %[SecondaryBody]
        %[ThirdBody]
        </body></html>
      </Body>
      <CustomFields>
        <CustomField name="pagetype">%[coveo_TypeName]</CustomField>
      </CustomFields>
    </Fields>
  </Mapping>
  <!-- This is all the binary documents that will be retrieved from EPiServer, either from a
page or the file manager. -->
  <Mapping type="Document">
    <Fields>
      <ClickableUri>%[coveo_ServerUrl]%[coveo_VirtualPath]</ClickableUri>
      <PrintableUri>%[coveo_ServerUrl]%[coveo_VirtualPath]</PrintableUri>
    </Fields>
  </Mapping>
</EpiServerMapping>
```

b. Save the file using a name of your choice in the `[Index_Path]\Config` folder.

**Example:** `C:\CES7\Config\MyEPiServerCMSMapping.xml`

3. To instruct the connector to use this configuration file, add the `MappingFile` hidden parameter to the EPiServer CMS source and use the filename and path where you saved the file as the value for the parameter (see "Modifying Hidden EPiServer CMS Source Parameters" on page 21).

## 8.4 Modifying Hidden EPiServer CMS Source Parameters

The **Add Source** and **Source: ... General** pages of the Administration Tool present the parameters with which you can configure the connector for most 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 adding and setting values of hidden parameters only when they apply to one or more of your EPiServer CMS sources.

### Hidden EPiServer CMS source parameters

**ConfigFile (String)**

This parameter is the full path to a custom EPiServer CMS configuration file (see "Creating and Using an EPiServer CMS Configuration File" on page 18).

**MappingFile (String)**

This parameter is the full path to a custom EPiServer CMS mapping file (see "Creating and Using an EPiServer CMS Mapping File" on page 19).

**NumberOfRefreshThreads (Integer)**

This parameter is the number of refresh threads used by the source. The default value is 2.

**BatchSize (Integer)**

This parameter is the number of items to fetch per request made to the EPiServer CMS server. The default value is 50. The minimum value is 1. A small value forces the connector to make small but frequent queries to the server. A larger value leads to larger and less frequent queries.

**FileManagerConfigurationString (String)**

This parameter specifies the EPiServer File Manager content to crawl. The string is made up of the starting addresses to crawl separated by a semicolon character. Each starting address is composed of the values for the virtual path provider `Name` and `VirtualPath` attributes (see "Configuring the EPiServer CMS Source to Crawl the File Manager" on page 16).

**ShortCutTypesToIgnore (String)**

This parameter determines one or more types of content that the connector ignores to prevent indexing content more than once. The possible values are:

- `0` is a normal page.
- `1` is a shortcut to another page.
- `2` is an external link.

- `3` is only text instead of the page.
- `4` is a fetched data from another page.

The string is composed of one or more values separated by a semicolon character. The default value is `1;2;3`.

**WebServiceUrl (String)**

This is the full URL to point to the Coveo Web service deployed on the EPiServer CMS installation. The default value is `[StartingAddress]/WebServices/CoveoService.asmx` (see "Configuring the Coveo Web Service on an EPiServer CMS Server" on page 8).

**IgnoreDeletedPages (boolean)**

Whether to ignore pages that were deleted (Recycle Bin). The default value is `False`. This parameter is available with CES 7.0.4887+

### To modify hidden source parameters

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

1. Refer to "Adding an Explicit Connector Parameter" on page 22 to add a hidden source parameter.
2. For a new EPiServer CMS 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 EPiServer CMS 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 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, and then click **Apply Changes**.

## 8.5 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.

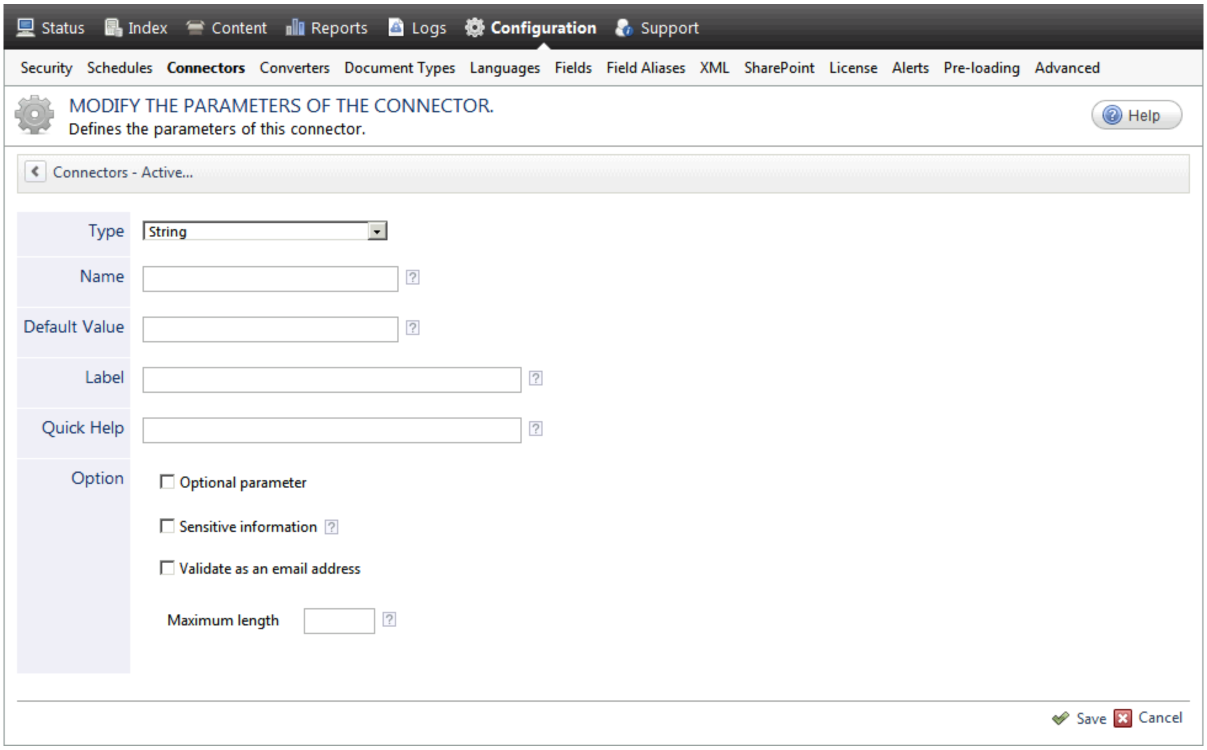
You can however choose to make one or more of these parameters appear in the **Add Source** and **Source: ... General** pages of the Administration Tool so that you can change their default value.

## 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:


Status Index Content Reports Logs Configuration Support
Security Schedules Connectors Converters Document Types Languages Fields Field Aliases XML SharePoint License Alerts Pre-loading Advanced
MODIFY THE PARAMETERS OF THE CONNECTOR.
Defines the parameters of this connector.
Help
Connectors - Active...
Save
Cancel

Type: String
Name:
Default Value:
Label:
Quick Help:
Option:
□ Optional parameter
□ Sensitive information
□ Validate as an email address
Maximum length:

   a. In the **Type** list, select the parameter type as specified in the parameter description.
   b. In the **Name** box, type the parameter name exactly as it appears in the parameter description. Parameter names are case sensitive.

c. 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.

d. 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.