Sitecore Connect for Sitecore CMP 3.0 Installation Guide

A guide to installing Sitecore Connect for Sitecore CMP 3.0

September 14, 2020

Table of Contents

1. Prerequisites 3
   1.1. Prepare install on Azure 3
2. Installation 4
   2.1. Upgrading from version 2.0 to 3.0 4
   2.2. Troubleshoot installation errors 4
      2.2.1. Could not load file or assembly 'System.Diagnostics.DiagnosticSource' 4
      2.2.2. Could not load file or assembly 'Microsoft.IdentityModel.Clients.ActiveDirectory' 5
      2.2.3. Could not load file or assembly 'Polly' 5
3. Configuration 6
   3.1. Create a user 6
   3.2. Create an action 6
   3.3. Create a trigger 7
   3.4. Add the connection strings 8
4. Configure content and mappings 10
   4.1. Buckets 10
   4.2. Templates 11
   4.3. Entity mapping 12
   4.4. Field mapping 13
   4.5. Create a relation field mapping 15
      4.5.1. Change the relation field mapping delimiter 16
   4.6. Create a tag field mapping 16
      4.6.1. Change the tag field mapping delimiter 17
5. Taxonomy provider 18
   5.1. Use a custom taxonomy provider 18
6. Problem solving 20

1. Prerequisites

Before installing the Sitecore Connect™ for the Sitecore Content Marketing Platform (CMP) module, you must have the following:

- Sitecore Experience Platform 9.2, 9.3, or 10.0
- Sitecore Content Hub 3.4.

1.1. Prepare install on Azure

Before you install CMP on Azure, do the following:

1. Navigate to the `Website\Bin` folder. Check if it contains the `System.Diagnostics.DiagnosticSource.dll` file.

    > **NOTE**
    > If the file does *not* exist, you can proceed with the installation without going through the following steps.

2. If the DLL file exists, open the `web.config` file, and verify that the file contains a `bindingRedirect` tag. If the file does not contain the tag, you must add it to the file. For example:

    ```
    <dependentAssembly>
        <assemblyIdentity
            name="System.Diagnostics.DiagnosticSource"
            publicKeyToken="cc7b13ffcd2ddd51"
            culture="neutral"/>
        <bindingRedirect oldVersion="0.0.0.0-4.0.2.1" newVersion="4.0.2.0"/>
    </dependentAssembly>
    ```

3. When you install CMP, the Installation Wizard may ask if you want to overwrite the existing `System.Diagnostics.DiagnosticSource.dll` file. Do *not* overwrite the DLL file. Follow the prompts to keep the existing file.

2. Installation

To install the Sitecore Connect™ for Sitecore CMP module:

1. Download the Sitecore Connect™ for Sitecore CMP module installation package from the Sitecore download page.
2. On the Sitecore Launchpad, click **Control Panel**.
3. To open the Installation Wizard, in the **Control Panel**, in the **Administration** section, click **Install a package**.
4. Click **Upload package** and follow the steps to upload the CMP package to Sitecore.
5. Click **Next** and follow the steps to install the package.

2.1. Upgrading from version 2.0 to 3.0

To upgrade an existing Sitecore CMP 2.0 installation to Sitecore CMP 3.0, install the Sitecore CMP 3.0 installation package as described in the installation instructions.

When the Installation Wizard finishes, it reports that it encountered potential problems. This is expected behavior. You do not need to take any further actions.

2.2. Troubleshoot installation errors

When the installation finishes, one of the following error messages can appear. Follow the instructions to correct the error.

2.2.1. Could not load file or assembly 'System.Diagnostics.DiagnosticSource'

If you get the *Could not load file or assembly 'System.Diagnostics.DiagnosticSource, Version=4.0.2.0'* exception, do the following:

1. In the `web.config` file, add the following *bindingRedirect* tag:

```
<dependentAssembly>
    <assemblyIdentity name="System.Diagnostics.DiagnosticSource"
        publicKeyToken="cc7b13ffcd2ddd51"
        culture="neutral"/>
    <bindingRedirect oldVersion="0.0.0.0-4.0.2.1" newVersion="4.0.2.1"/>
</dependentAssembly>
```

2. Run the installation again.

2.2.2. Could not load file or assembly 'Microsoft.IdentityModel.Clients.ActiveDirectory'

If you get the *Could not load file or assembly 'Microsoft.IdentityModel.Clients.ActiveDirectory'* exception:

1. In the `web.config` file, check that the following bindings exist:

```
<dependentAssembly>
  <assemblyIdentity
    name="Microsoft.IdentityModel.Clients.ActiveDirectory"
    publicKeyToken="31bf3856ad364e35"
    culture="neutral"
    xmlns="urn:schemas-microsoft-com:asm.v1" />
  <bindingRedirect
    oldVersion="0.0.0.0-3.19.3.10102"
    newVersion="3.17.2.31801"
    xmlns="urn:schemas-microsoft-com:asm.v1" />
</dependentAssembly>

<dependentAssembly>
  <assemblyIdentity
    name="Microsoft.IdentityModel.Clients.ActiveDirectory.Platform"
    publicKeyToken="31bf3856ad364e35"
    culture="neutral"
    xmlns="urn:schemas-microsoft-com:asm.v1" />
  <bindingRedirect
    oldVersion="0.0.0.0-3.19.3.10102"
    newVersion="3.17.2.31801"
    xmlns="urn:schemas-microsoft-com:asm.v1" />
</dependentAssembly>
```

2. Add any missing bindings to the `web.config` file.

3. Run the installation again.

2.2.3. Could not load file or assembly 'Polly'

If you get the *Could not load file or assembly 'Polly'* exception, please read this KB article to resolve the issue.

3. Configuration

This section describes the following configuration steps for the Sitecore Connect™ for Sitecore CMP module:

- Create a Sitecore Content Hub user
- Create a Sitecore Content Hub action
- Create a Sitecore Content Hub trigger
- Add the connection strings
- Configure content and field mappings.

3.1. Create a user

You must create a Sitecore Content Hub user:

1. Open the Sitecore Content Hub, and on the ribbon, click **Manage**.
2. On the **Users** tab, click **Add User**.
3. In the **Username** field, enter the user name, and click **Save**.
4. To assign the user to a group, on the **User groups** tab, click **Add to User group**, select the check box, and then click **Select**.

8. To add the user group to the Policy combination, in the **Policy combination** panel, click **Add item**.
9. Click **Save**.
10. On the **Details** tab, click **Edit profile**.
11. In the Email field, enter an email address, and click **Save**.
12. Click **Reset Password,** and click **Yes** to confirm. A password reset link is sent to the email address previously provided. If the email does not show up, please check your spam filter.

3.2. Create an action

Actions of type *Azure service bus* can connect to an Azure Service Bus and post a message when a certain trigger is fired.

To create an action:

1. In the Sitecore Content Hub, click **Manage**.

2. Click **Actions**.
3. Click **New action**.
4. Enter, for example, the following values:

- **Name**: *Sitecore*
- **Type**: *CMP Azure Service Bus*
- **Connection string**: enter the connection string for the service bus here. The connection string must to be a connection string to the entire azure service bus.
- **Destination type**: *Topic*
- **Destination**: *hub_out*

13. To test the connection to the Azure Service Bus, click **Test Connection**.
14. To save the new action, click **Save**.

3.3. Create a trigger

The Sitecore Content Hub provides users with a Trigger framework, which allows them to configure triggers using the user interface. A trigger is fired when a certain event occurs – with certain conditions – and can initiate a set of actions.

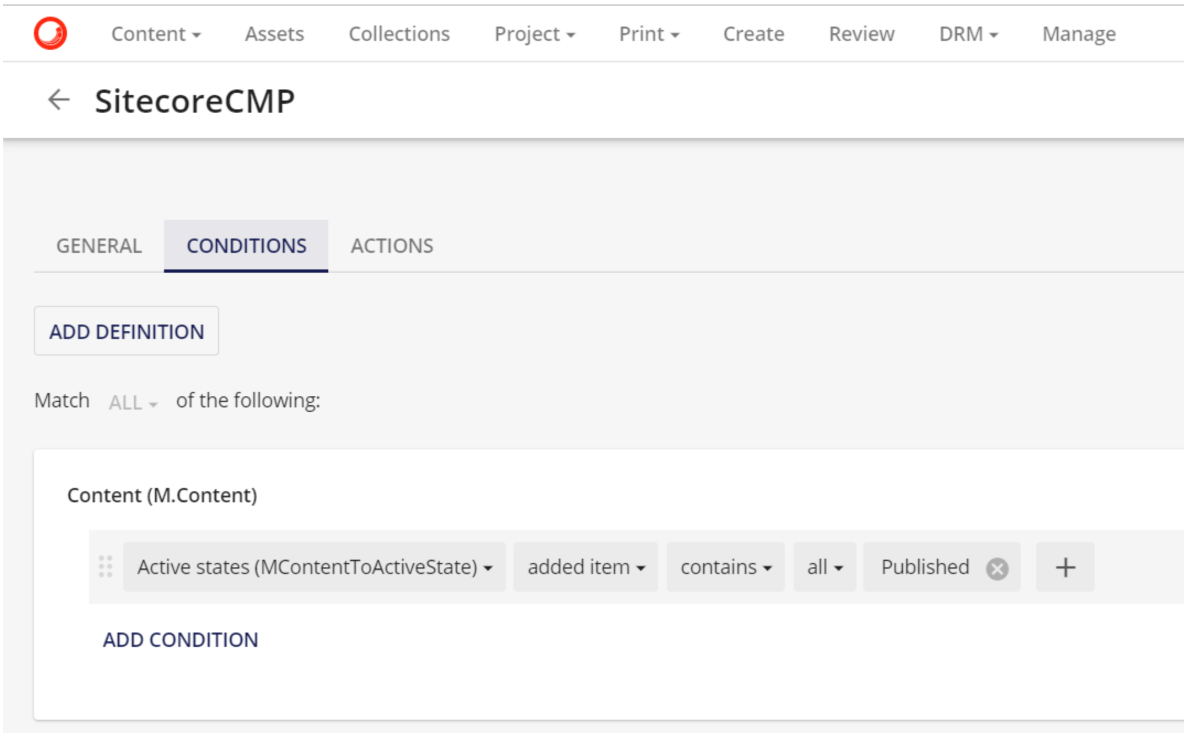
To create a trigger:

1. Open the Sitecore Content Hub. On the ribbon, click **Manage**.
15. Click **Triggers**, and click **New trigger**.
16. On the **General** tab, enter the following values:

Field	Value
Enter a name	*CMP content modification*
Enter a description	*Post to Azure Service Bus after content change*
Objective	*Entity modification*
Execution type	*In background*

17. On the **Conditions** tab, click **Add definition**, and in the drop-down list, select *Content*. This adds a new item to the list of conditions.

18. Add the following definitions:



Match ALL of the following:

Content (M.Content)

Active states (MContentToActiveState) | added item | contains | all | Published

This triggers the content import when an item is moved to the *Approved for publication* workflow state.

> **NOTE**
> For a complete overview of capabilities, see the Sitecore Content Hub documentation.

19. On the **Actions** tab, add the action you created previously (*CMP content modification*).
20. Click **Save and close**. Confirm to activate the trigger.

> **NOTE**
> You can also activate the trigger after you have completed all the configuration settings in Content Hub and in Sitecore. At this point, the trigger might start executing before the configuration is completed.

3.4. Add the connection strings

After you installed the package, open the `C:\inetpub\<site>\App_Config\ConnectionStrings.config` file and add the following connection strings:

- `<add name="CMP.ContentHub" connectionString="ClientId={client_id};ClientSecret={client_secret};UserName={username};Password={password};URI={uri};" />`

- `<add name="CMP.ServiceBusEntityPathIn" connectionString="{Azure Service Bus connection string with incoming topic}" />`
- `<add name="CMP.ServiceBusSubscription" connectionString="{Subscription name}" />`
- `<add name="CMP.ServiceBusEntityPathOut" connectionString="{Azure Service Bus connection string with outcoming topic}" />`

In the connection strings, insert the following values:

- **Client id** and **Client Secret**: these settings must correspond to the settings of the LogicApp in the OAuth Clients in the Sitecore Content Hub. To verify these settings: open the Sitecore Content Hub and on the ribbon, click **Manage**, click **OAuth Clients**, click **LogicApp**, and click **Edit**.
- **User Name** and **Password** - these fields must correspond to the values in the Sitecore Content Hub for the user that you created.
- **URI** - set the URI to Sitecore Content Hub
- **Incoming topic** - set the connection string for the Azure Service Bus and set the name for the topic of the messages posted by the Sitecore Content Hub, posted after the trigger is fired.
- **Subscription name** - set the name of the subscription.
- **Outgoing topic** - set the name of the topic of the messages posted to the Azure Service Bus after content is imported to Sitecore.

> **NOTE**
> You must restart the Sitecore website after you set or adjust the connection string.

4. Configure content and mappings

To import content, you must first create these items:

- Bucket(s) - specify where to store imported content.
- Template(s) - define the structure of imported content items.
- Entity mapping item(s) and Field mapping item(s) - determine how imported content is imported and where it is stored.

You can optionally add these items to control how relation content from Content Hub is processed:

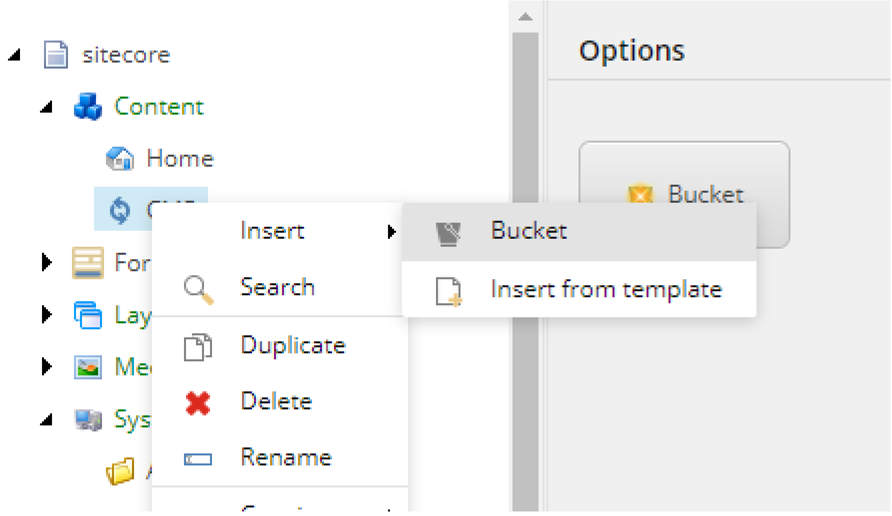
- Relation field mapping item(s) - map relation content to Sitecore fields.
- Tag field mapping item(s) - map relation content to Sitecore tags.
- Custom taxonomy provider - by default, tag field mapping stores the relation content value as a tag in Sitecore tag repositories. You can change this by specifying a custom taxonomy provider.

4.1. Buckets

All the imported content is stored in Sitecore buckets. For example, all blog posts are stored in a bucket called *Blogs*.

To create a bucket:

1. Right-click the **CMP** folder, click **Insert**, and click **Bucket**.



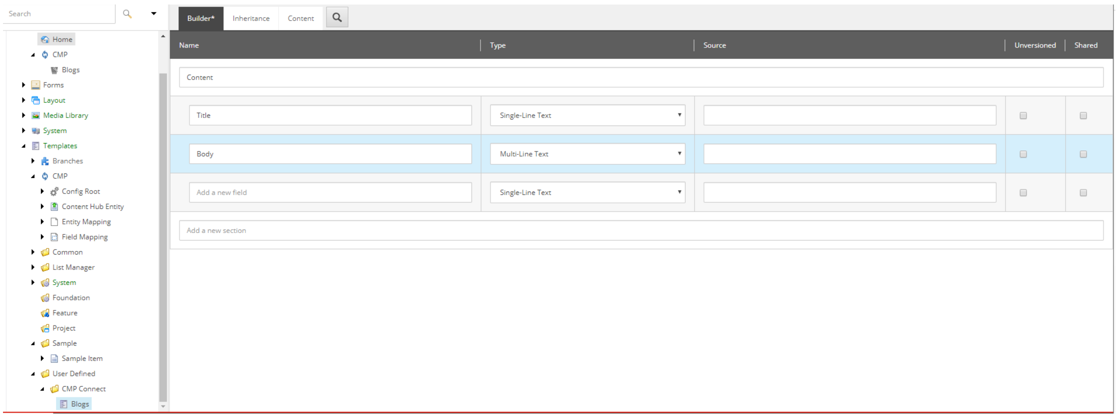
2. Enter a name for the bucket, and click **OK**.

4.2. Templates

The structure of the imported content is defined by the Sitecore templates. You must make sure that each template inherits the *Content Hub Entity* template. This template contains the **EntityIdentifier** field.

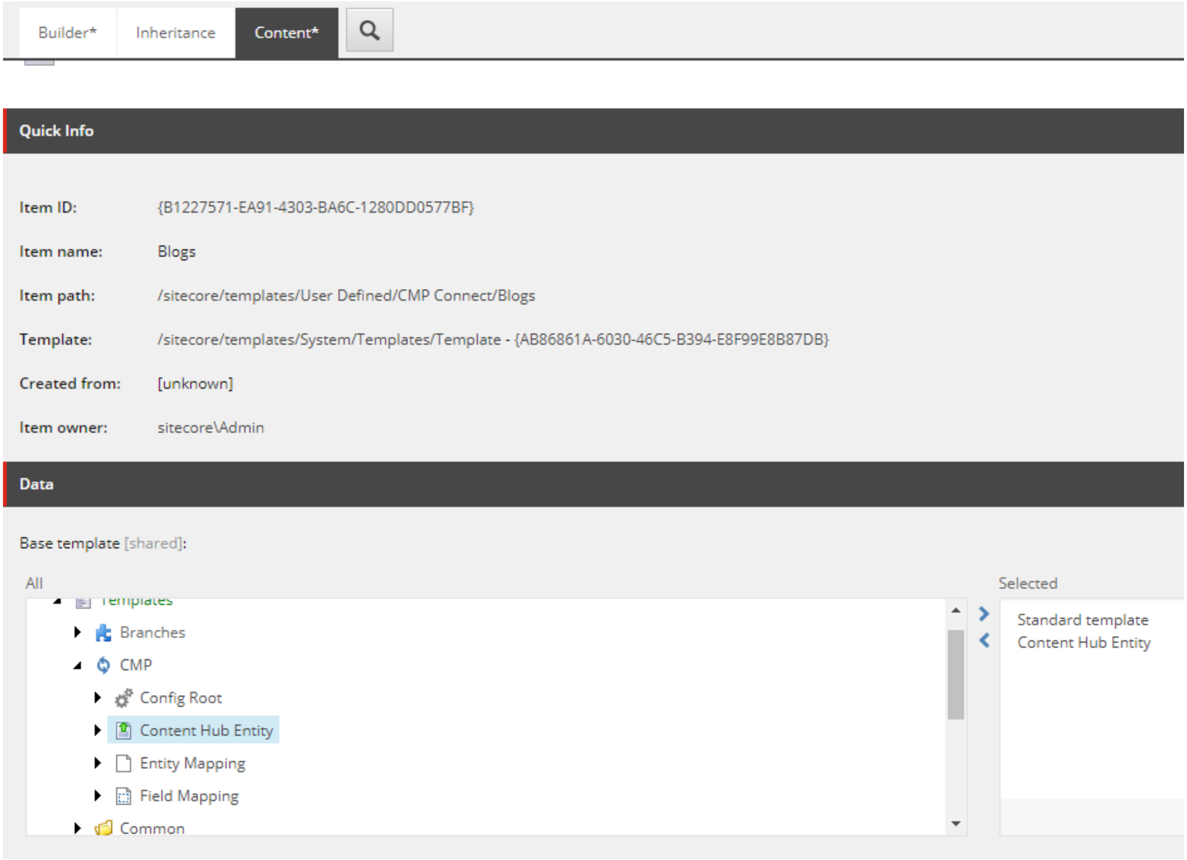
To create a template:

1. Navigate to a *CMP* folder, for example, `templates\User Defined\CMP Connect`. Create a template.
2. Add all the necessary fields. For example, for a blog post, add a **Title** Single-Line Text field and a **Body** Multi-Line Text field.



Name	Type	Source	Unversioned	Shared
Content				
Title	Single-Line Text		□	□
Body	Multi-Line Text		□	□
Add a new field	Single-Line Text		□	□
Add a new section				

3. In the **Data** section, in the **Base template** field, click **Content Hub Entity** and then click the right arrow to add it to the **Selected** list.



Quick Info

Item ID: {B1227571-EA91-4303-BA6C-1280DD0577BF}

Item name: Blogs

Item path: /sitecore/templates/User Defined/CMP Connect/Blogs

Template: /sitecore/templates/System/Templates/Template - {AB86861A-6030-46C5-B394-E8F99E8B87DB}

Created from: [unknown]

Item owner: sitecore\Admin

Data

Base template [shared]:

All

Selected

Standard template
Content Hub Entity

4. To save the template, click **Save**.

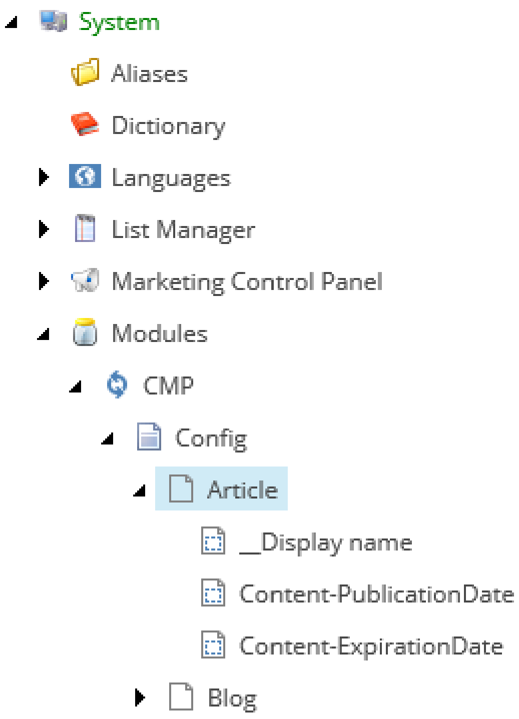
4.3. Entity mapping

Entity Mapping items and *Field Mapping* items define how content is imported and in what bucket it is stored.

To create an entity mapping item:

1. Navigate to `/sitecore/system/Modules/CMP`, right-click **Config**, click **Insert**, and click **Entity Mapping**. Enter the name for the new item, for example *Article*. By default, every entity

mapping comes with default field mappers for the display name, the publication date, and the expiration date.



2. In the **Data** section, fill in the following fields:

Field	Content
Content Type Id	enter the content type used in Content Hub
Bucket	enter the item bucket to keep Sitecore items created from M-Entities of a given content type id
Template	enter a template to be used to create the Sitecore items. The template must contain the fields that are used to store the imported content

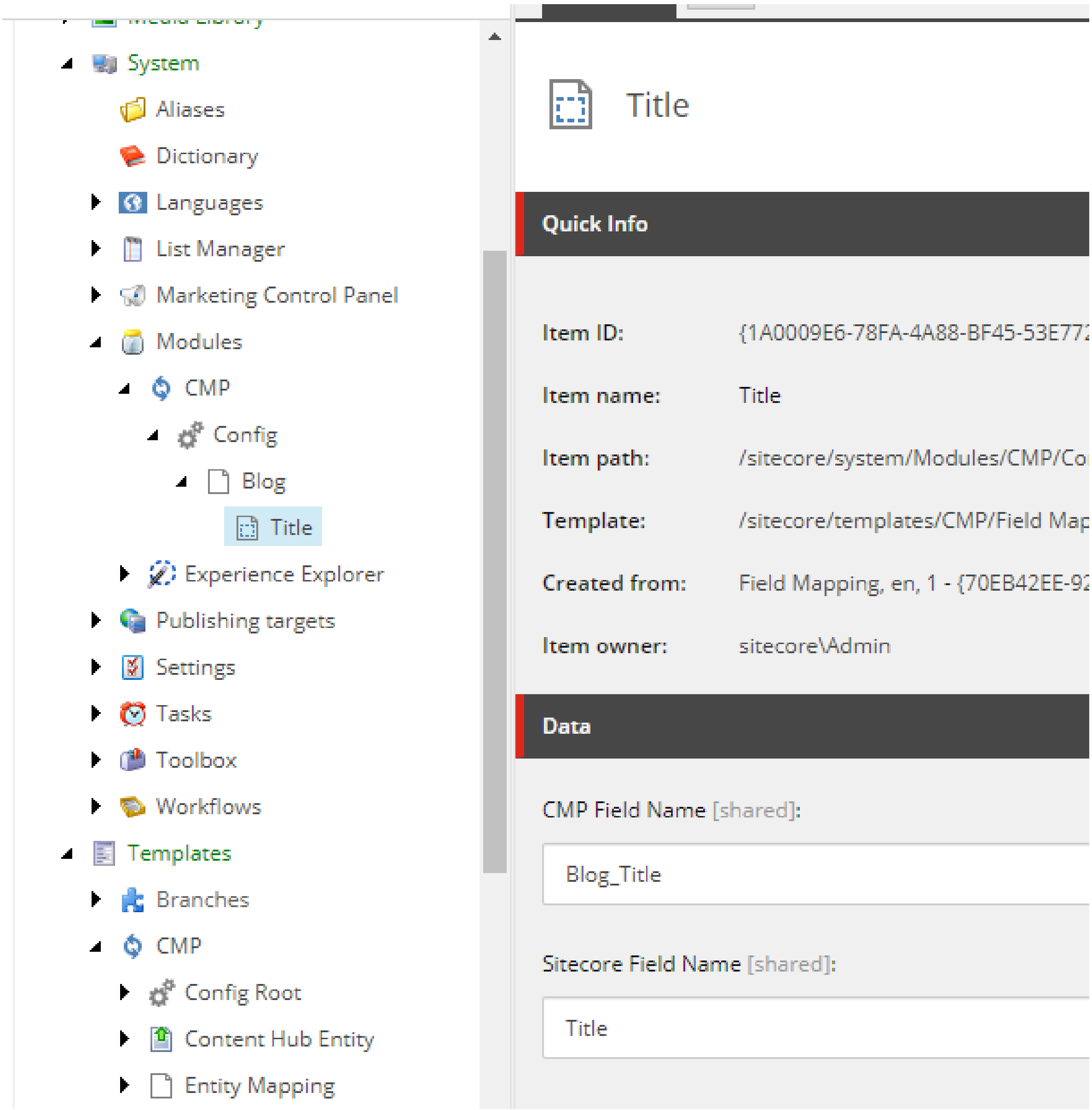
4.4. Field mapping

To map imported content to the Sitecore item structure, you must add *Field Mapping* items to each *Entity Mapping* item. You must create a *Field Mapping* item for each field that you want to import.

To create the *Field Mapping* item:

1. Right-click the *Entity Mapping* item that you created earlier, click **Insert**, and then click **Field Mapping**.
2. Enter a name, for example, *Title*, and click **OK**.
3. On the **Content** tab, in the **Data** section, fill in the following fields:

Field	Content
CMP Field Name	the name of the field in the Content Hub
Sitecore Field Name	the field name as defined in the template you are using



Title

Quick Info

Item ID: {1A0009E6-78FA-4A88-BF45-53E77

Item name: Title

Item path: /sitecore/system/Modules/CMP/Co

Template: /sitecore/templates/CMP/Field Map

Created from: Field Mapping, en, 1 - {70EB42EE-92

Item owner: sitecore\Admin

Data

CMP Field Name [shared]: Blog_Title

Sitecore Field Name [shared]: Title

You can add as many *Field Mapping* items as you require.

To verify the entity type and structure in the Content Hub:

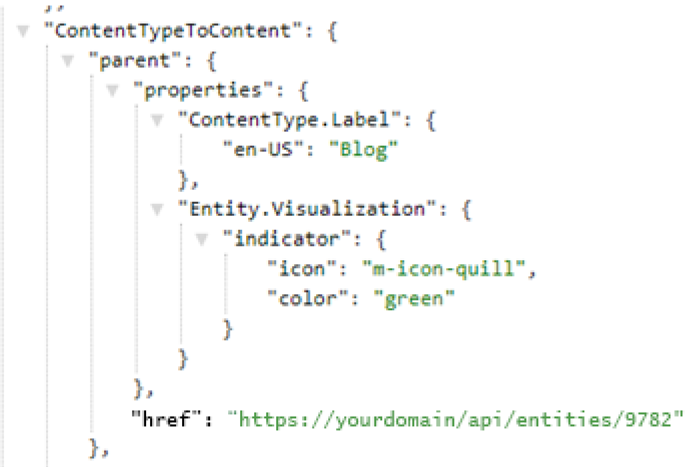
1. Create a new item in the Content Hub or edit an existing one, for example, a blog post.
2. Copy the URL, for example, `https://yourdomain/en-us/content/detail/9782`, and paste it in a new browser tab.

NOTE

In the example URLs, replace *9782* with the ID of the entity item you have selected.

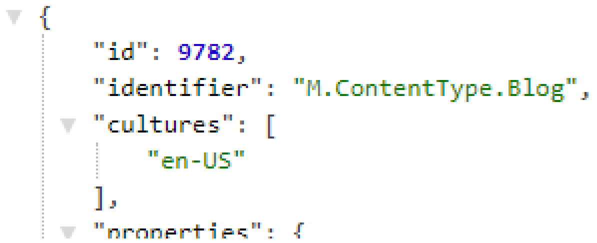
3. Change the URL to `https://yourdomain/api/entities/9782` and press enter. The entity structure is now shown in JSON.

4. To find the entity’s content type, find the **ContentTypeToContent** relation and follow the link (`"href": "https://yourdomain/api/entities/9782"`).



```
"ContentTypeToContent": {
    "parent": {
        "properties": {
            "ContentType.Label": {
                "en-US": "Blog"
            },
            "Entity.Visualization": {
                "indicator": {
                    "icon": "m-icon-quill",
                    "color": "green"
                }
            }
        },
        "href": "https://yourdomain/api/entities/9782"
    },
```

Now the content type is shown, in this example `M.ContentType.Blog`.



```
{
    "id": 9782,
    "identifier": "M.ContentType.Blog",
    "cultures": [
        "en-US"
    ],
    "properties": {
```

4.5. Create a relation field mapping

Relation field mapping lets you map the taxonomy fields from any content in Content Hub to a Sitecore field. If the content belongs to multiple taxonomy values, the mapping separates the values using the configured delimiter. The default delimiter is a comma (,).

To create a relation field mapping:

1. Right-click the *Entity Mapping* item that you created earlier, click **Insert**, and then click **Relation Field Mapping**.

2. Enter a name, for example, *Audiences*, and click **OK** .

3. In the **Data** section, fill in the following fields:

Field	Content
CMP Relation	The relation name of the Content Hub entity.
CMP Field Name	The name of the property in the Content Hub relation entity.
Sitecore Field Name	The name of the content field where Sitecore stores the taxonomy values, as defined in the template you are using.

4.5.1. Change the relation field mapping delimiter

You can change the delimiter Sitecore uses to separate values when it stores multiple taxonomy values.

To change the delimiter:

1. On your Sitecore instance, open the `App_Config\Modules\Sitecore.Connector.CMP.config` file.
2. In the `configuration/sitecore/settings` node, set the value of `Sitecore.RelationFieldMappingSeparator` to the delimiter you want to use. For example, `<setting name="Sitecore.RelationFieldMappingSeparator" value=", " />`.

4.6. Create a tag field mapping

Tag field mapping lets you map the taxonomy fields from any content in Content Hub to the tag repositories in Sitecore. If the content belongs to multiple taxonomy values, the mapping separates the values using the configured delimiter.

To create a tag field mapping:

1. Right-click the *Entity Mapping* item that you created earlier, click **Insert**, and then click **Tag Field Mapping**.
2. Enter a name, for example, *Tag Taxonomy*. Click **OK**.
3. In the **Data** section, fill in the following fields:

Field	Content
CMP Relation	The relation name of the Content Hub entity.
CMP Field Name	The name of the property in the Content Hub relation entity.
Enable / Disable to store Non-Leaf Node	Choose one of the following: • Enable Non-Leaf Node (Default) • Disable Non-Leaf Node
Store Tag Name Option As	Choose one of the following: • Full Path Name (Default) • Node Name

Field	Content
Custom Taxonomy Provider Name	The name of a Sitecore custom taxonomy provider. If you select a custom taxonomy provider, Sitecore stores the taxonomy in the custom provider instead of in the default taxonomy provider. Leave this field blank to use the default taxonomy provider. **NOTE** See Build a custom taxonomy provider in the Sitecore developer documentation for information on how to create custom taxonomy providers.

For more information on how the tag field mapping settings work, refer to the Sitecore developer documentation.

4.6.1. Change the tag field mapping delimiter

You can change the delimiter Sitecore uses to separate values when it stores multiple values in a tag. To do so:

1. On your Sitecore instance, open the `App_Config\Modules\Sitecore.Connector.CMP.config` file.

3. In the `configuration/sitecore/settings` node, change the setting `Sitecore.NodeNameSeparator` to the delimiter you want to use. For example, `<setting name="Sitecore.NodeNameSeparator" value="." \>`.

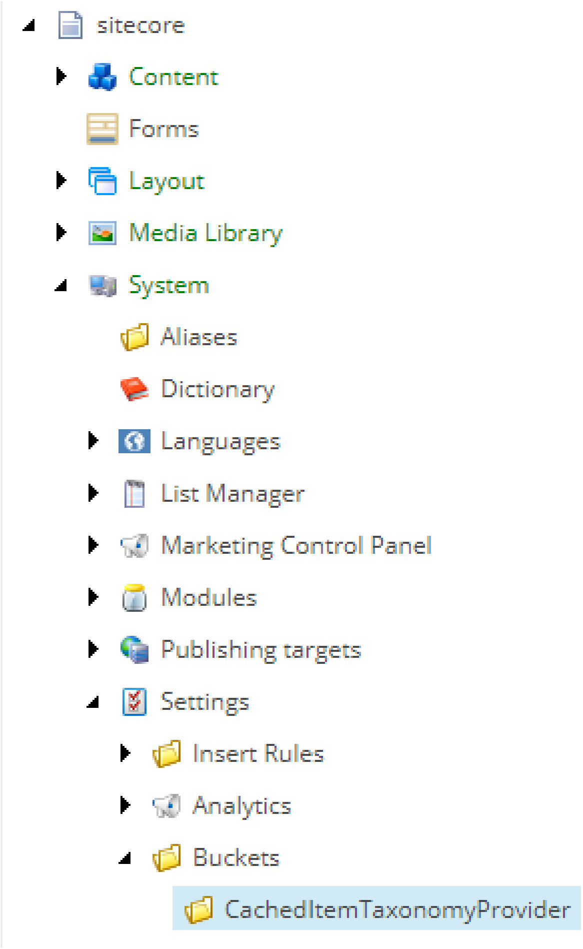
4. If you use special characters as a separator, you must allow those characters on the Sitecore level. You do this by using a patch file to change the `ItemNameValidation` setting. For example:

```
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
    <sitecore>
        <settings>
            <setting name="ItemNameValidation">
                <patch:attribute name="value">^.*$</patch:attribute>
            </setting>
        </settings>
    </sitecore>
</configuration>
```

> **IMPORTANT**
>
> This example allows *all* characters to be used. It is best practice to only allow the characters you need.

5. Taxonomy provider

When you set up the retrieval of taxonomy fields from any content in Content Hub, and synchronize the field values to the tag repositories in Sitecore, by default, Sitecore stores the taxonomy in `/sitecore/System/Settings/Buckets/CachedItemTaxonomyProvider`.



```
sitecore
  Content
  Forms
  Layout
  Media Library
  System
    Aliases
    Dictionary
    Languages
    List Manager
    Marketing Control Panel
    Modules
    Publishing targets
    Settings
      Insert Rules
      Analytics
      Buckets
        CachedItemTaxonomyProvider
```

5.1. Use a custom taxonomy provider

By default, taxonomies are stored in the *CachedItemTaxonomyProvider* provider.

If you have created a custom taxonomy provider, you can store the taxonomy there instead of in the default provider:

1. On your Sitecore instance, open the `[your Sitecore Instance]\App_Config\Modules\Sitecore.Connector.CMP\ Sitecore.Connector.CMP.config` file.
2. Change the `Sitecore.DefaultTaxonomyProvider` setting to the name of your custom taxonomy provider. For example:

```
<setting name="Sitecore.DefaultTaxonomyProvider" value="MyCustomTaxonomyProvider" />
```

> **NOTE**
> Do not leave the `Sitecore.DefaultTaxonomyProvider` setting empty.

3. Save the config file.

6. Problem solving

To help find solutions for errors or problems, you can do the following:

- Check the log in the Sitecore Content Hub for the created trigger. Trigger the event that fires the trigger and then check the log as follows:
  - On the menu ribbon, click **Manage** .
  - Click **Actions**.
  - Click the **Auditing** tab.
- Check posted messages in the Azure Service Bus.
- Check the Sitecore log file and scan for any CMP related errors.