

# **CMSImport PRO**

## **User manual**

**Version 2.3**



<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
<b>2</b>	<b>INSTALLATION.....</b>	<b>5</b>
<b>3</b>	<b>IMPORT CONTENT .....</b>	<b>6</b>
3.1	Introduction .....	6
3.2	Select Datasource Type .....	7
3.3	Select datasource .....	7
3.3.1	SQL Server .....	7
3.3.2	CSV File.....	8
3.3.3	XML File.....	8
3.3.4	Access database .....	9
3.3.5	Excel .....	9
3.3.6	RSS Feed .....	10
3.3.7	BlogML.....	10
3.3.8	WordPress .....	11
3.4	Select destination and document type .....	11
3.5	Create mapping .....	12
3.6	Confirm .....	14
3.7	Import.....	15
3.8	Save Import steps.....	15
3.9	Create a child definition .....	16
3.9.1	Select datasource .....	16
3.9.2	Specify content import options .....	17
3.10	End result.....	17
<b>4</b>	<b>IMPORT MEMBERS.....</b>	<b>19</b>
4.1	Introduction .....	19
4.2	Select Datasource Type .....	19
4.3	Select Datasource .....	19
4.3.1	SQL Server .....	20
4.3.2	CSV File.....	20
4.3.3	XML File.....	20
4.4	Select member type and role.....	21
4.5	Create Mapping .....	22
4.6	Confirm .....	22
4.7	Import.....	23
4.8	Save Import steps.....	24
<b>5</b>	<b>RELATED MEDIA IMPORT.....</b>	<b>25</b>
5.1	Rich text editor.....	26
5.2	Media picker .....	27
5.3	UComponents Multi Node Tree Picker (MNTP) .....	27
5.4	Digibiz Advanced Media Picker .....	28
5.5	Upload field.....	28
<b>6</b>	<b>SCHEDULE IMPORTS .....</b>	<b>30</b>
6.1	Scheduled task log .....	31

<b>7</b>	<b>IMPORT DASHBOARD .....</b>	<b>32</b>
<b>8</b>	<b>SETTINGS .....</b>	<b>33</b>
8.1	Login credential settings.....	33
8.2	Scheduler result settings .....	33
8.3	Media settings.....	33
8.4	Data adapters .....	34
8.5	Configure Import Dashboard .....	35
<b>9</b>	<b>EXTEND CMSIMPORT.....</b>	<b>36</b>
9.1	Setting up Visual Studio .....	36
9.2	What's in CMSImport.extensions .....	36
9.2.1	DataAdapter.....	36
9.2.2	FieldAdapter .....	36
9.2.3	Events.....	37
9.2.3.1	RecordImporting .....	37
9.2.3.2	RecordImported .....	37
9.2.3.3	Importing.....	37
9.2.3.4	Imported.....	38
9.3	Samples.....	38
9.4	Create a DataAdapter for Product Categories .....	38
9.4.1	Create the class.....	39
9.4.2	Specify an alias.....	39
9.4.3	Add UI .....	39
9.4.4	Update DataAdapter values .....	39
9.4.5	Validate .....	40
9.4.6	GetData .....	40
9.4.7	End result.....	41
9.5	Create a DataAdapter for Products. ....	41
9.6	Using a FieldAdapter to map text values against a Dropdownlist DataType .....	42
<b>10</b>	<b>MANUAL INSTALLATION/CONFIGURATION .....</b>	<b>45</b>
10.1	Manual installation of files .....	45
10.2	Manual configuration of Database.....	45
10.3	Manual configuration of the language files.....	47
10.4	Manual configuration of the scheduled task handler .....	47
<b>11</b>	<b>TROUBLESHOOTING .....</b>	<b>49</b>
11.1	I don't see the CMSImport package in my developer section .....	49
11.2	I don't see my column names when importing from a CSV file.....	49
11.3	I get weird column names when importing from a CSV file.....	49
11.4	I get an error not supported error when using access or excel files.....	49
11.5	Email is not send when importing a member .....	49
11.6	I get an Invalid License exception. ....	49

# 1 Introduction

CMSImport PRO helps you import content or members from any datasource into Umbraco. The following datasources are supported by default<sup>1</sup>:

- Access Database
- BlogML
- CSV
- Excel File
- RSS Feed
- SQL Server
- WordPress
- XML

CMSImport PRO allows you to save wizard steps so you can run the import later or even schedule it for a certain date and time. When you re-run an import already existing records will be updated and only new records will be added. When media is imported references in content or member data will be updated automatically.

And best of all with CMSImport Pro it's possible to import complete content structures also! This allows you to import a complete product catalog (Categories and Products), blogposts + comments, or any structure you want to import.

CMSImport PRO is optimized for **uBlogsy**, **Blog4Umbraco**, **TeaCommerce** and **Uwebshop**.

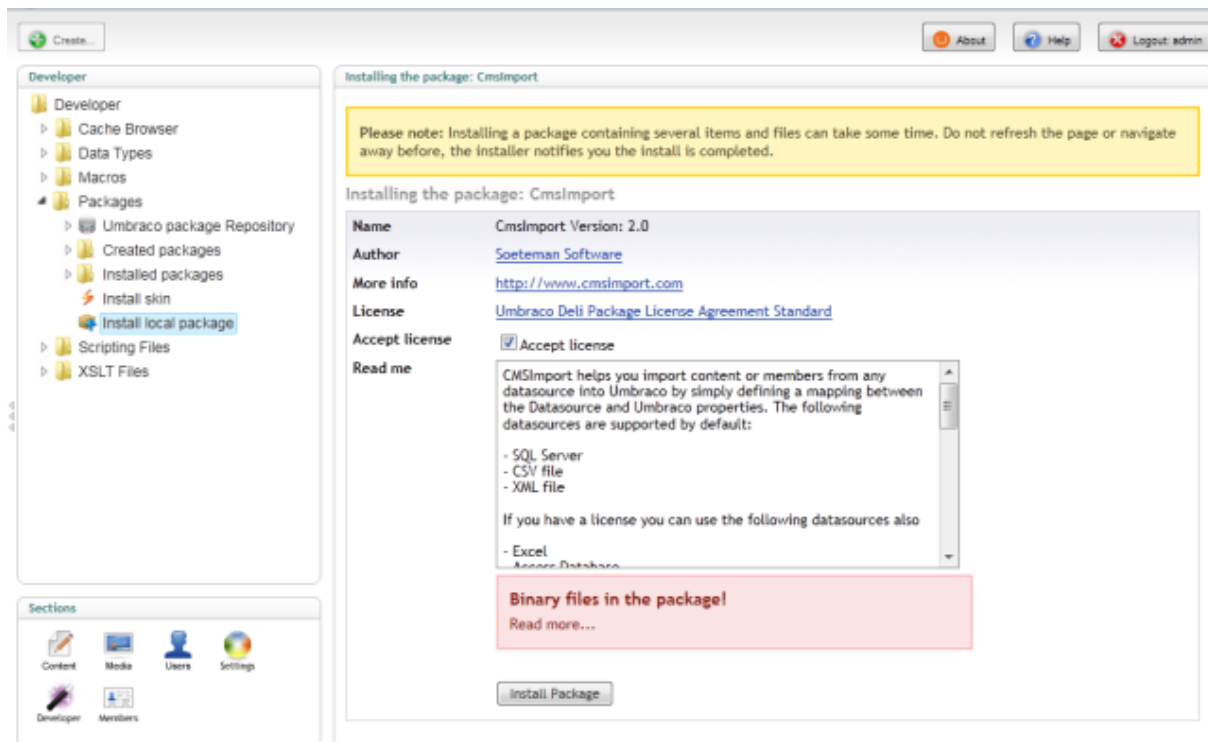
This document describes PRO features only. The free edition is limited in functionality.

---

<sup>1</sup> Access and Excel file requires you to install an extra driver from Microsoft

## 2 Installation

Install the CMSImport PRO package using the Umbraco package installer.



Make sure that the installer has modify rights on the following folders:

- /bin
- /config
- /umbraco

The installer also needs rights to create tables in the database.

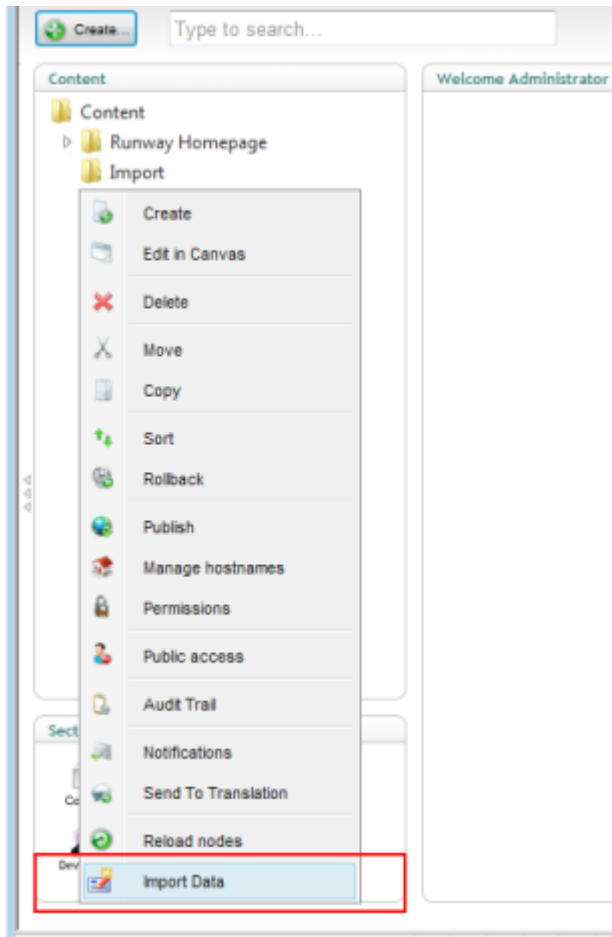
If for some reason you can't give the installer these rights, you can install the package manually. Read the chapter manual configuration how to do this.

Once the package is installed you have an extra folder in your developer section called CMSImport. You might need a page refresh, or even logout and in again to see this folder.

**When you want to use the PRO functionality add the license file to the bin folder of your Umbraco install.**

## 3 Import Content

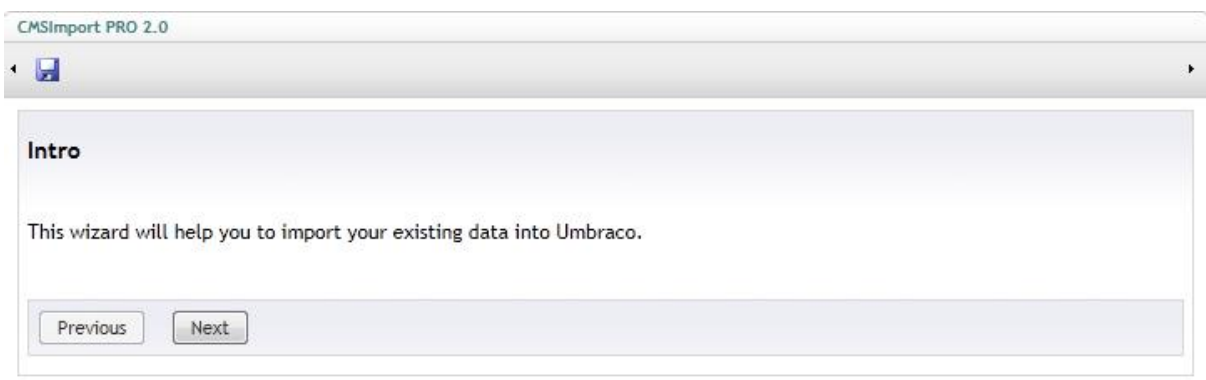
You can start CMSImport by right clicking on the content node and select the import data option (only visible for administrators).



Or you can go to the developer section, open CMSImport and select Start Import. The following wizard will open.

### 3.1 Introduction

When the wizard starts it will start with an introduction.



## 3.2 Select Datasource Type

In this step you are asked to specify a datasource type and how the data should be imported (content/members).



The following datasource types are supported by default<sup>2</sup>:

- Access database
- BlogML
- CSV
- Excel File
- RSS Feed
- SQL Server
- WordPress
- XML

It is possible to create extra datasource types. See chapter extend CMSImport for more information.

## 3.3 Select datasource

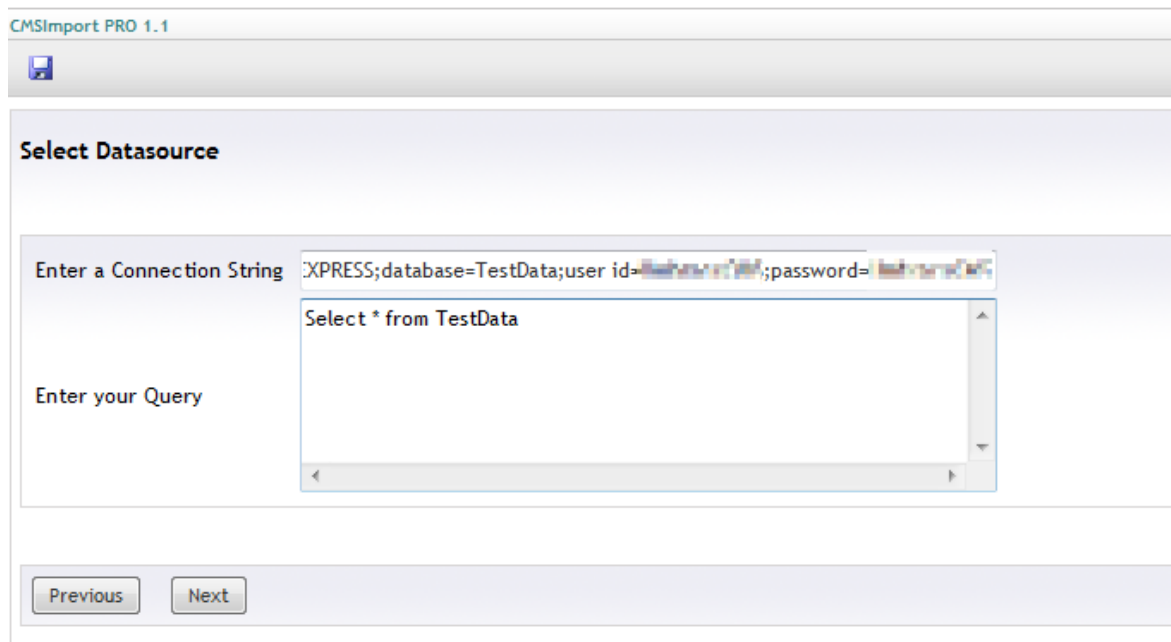
In the select datasource step you need to provide the information for the selected datasource Type.

### 3.3.1 SQL Server

In case of SQL server you need to provide the Connection string and the query that you want to use to retrieve the data.

---

<sup>2</sup> Access and Excel file requires you to install an extra driver from Microsoft



CMSImport PRO 1.1

**Select Datasource**

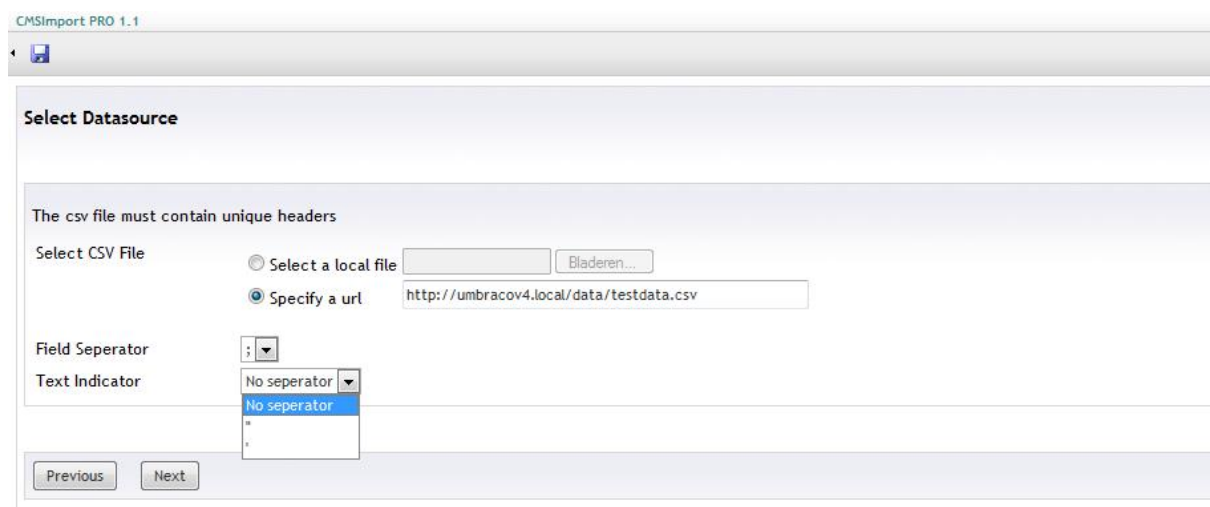
Enter a Connection String: XPRESS;database=TestData;user id=;password=

Enter your Query: Select \* from TestData

Previous Next

### 3.3.2 CSV File

In case of a CSV file you need to select a CSV File. You also need to provide the field separator and text indicator. A field separator is the character that is used to separate the columns. The text indicator is a character that indicates a text string. The default settings are based on an export from Microsoft Excel.



CMSImport PRO 1.1

**Select Datasource**

The csv file must contain unique headers

Select CSV File

☐ Select a local file  Bladeren...

☒ Specify a url

Field Separator:

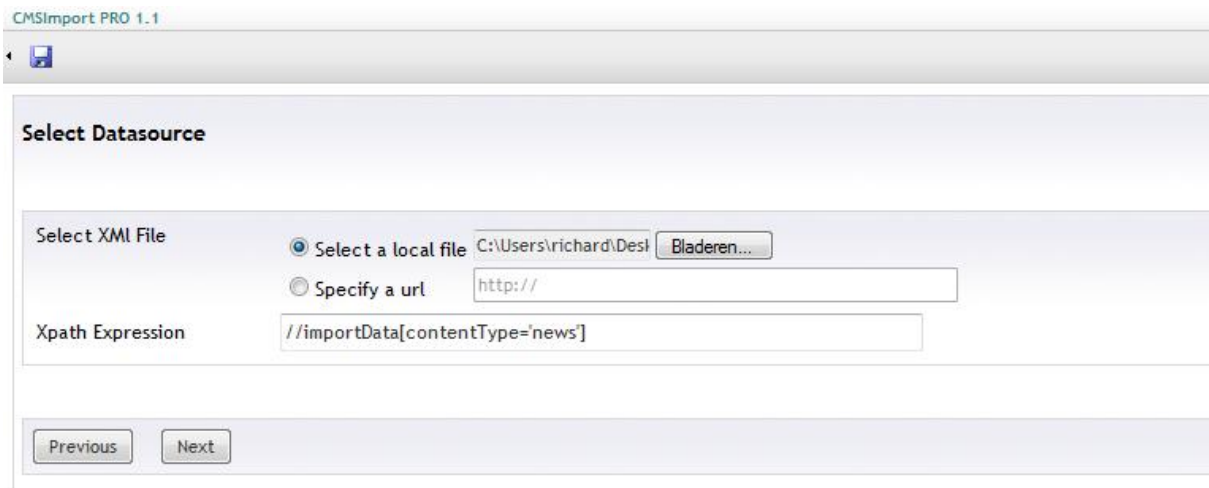
Text Indicator:

Previous Next

### 3.3.3 XML File

In case of an XML file you need to select the xml file. Optional you can specify an xpath expression.





CMSImport PRO 1.1

**Select Datasource**

Select XML File

☒ Select a local file C:\Users\richard\Desktop Bladeren...

☐ Specify a url http://

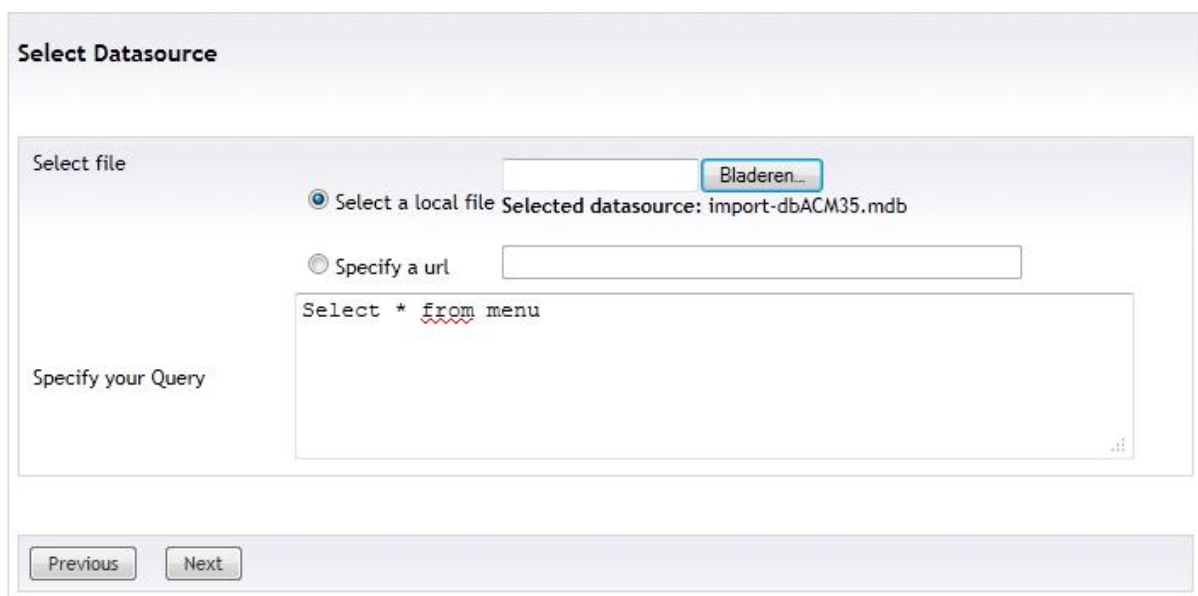
Xpath Expression //importData[contentType='news']

Previous Next

When using xml you can only use elements, attributes are not supported at the moment. All elements that you want to map later must be under the first child node of the xml document.

### 3.3.4 Access database

In case of an access database you can upload a mdb file, or point to a location and you can specify the query to execute



**Select Datasource**

Select file

☒ Select a local file Selected datasource: import-dbACM35.mdb Bladeren...

☐ Specify a url

Specify your Query

Select \* from menu

Previous Next

### 3.3.5 Excel

In case of an Excel sheet you can upload a xls(x) file, or point to a location. When you **click next** you can select the worksheet from the pull-down menu

**Select Datasource**

Select file

☒ Select a local file

☐ Specify a url

Select a worksheet

ProductCategories ▼

- ProductCategories
- Products
- ProductVariant

### 3.3.6 RSS Feed

In case of an rss file you can point to a location, or upload an rss file.

**Select Datasource**

Select a rss file

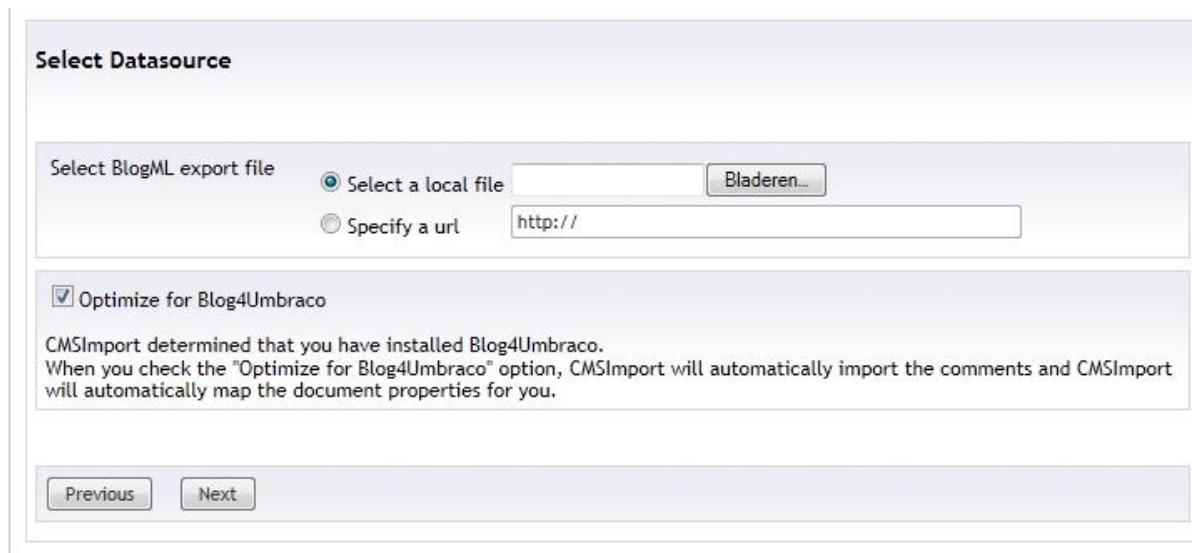
☐ Select a local file

☒ Specify a url

### 3.3.7 BlogML

In case of BlogML you can upload a BlogML file, or point to a location. When you have uBlogsy, Blog4Umbraco or the Blog starterkit installed you can specify that you want to optimize for either uBlogsy /Blog4Umbraco/Blog starter kit. When you enable that option CMSImport will automatically import comments and store it in the correct comments table also.

Don't select this option when you just want to import the blogposts and blogcomments based on an Umbraco documenttype .



**Select Datasource**

Select BlogML export file

☒ Select a local file  Bladeren...

☐ Specify a url

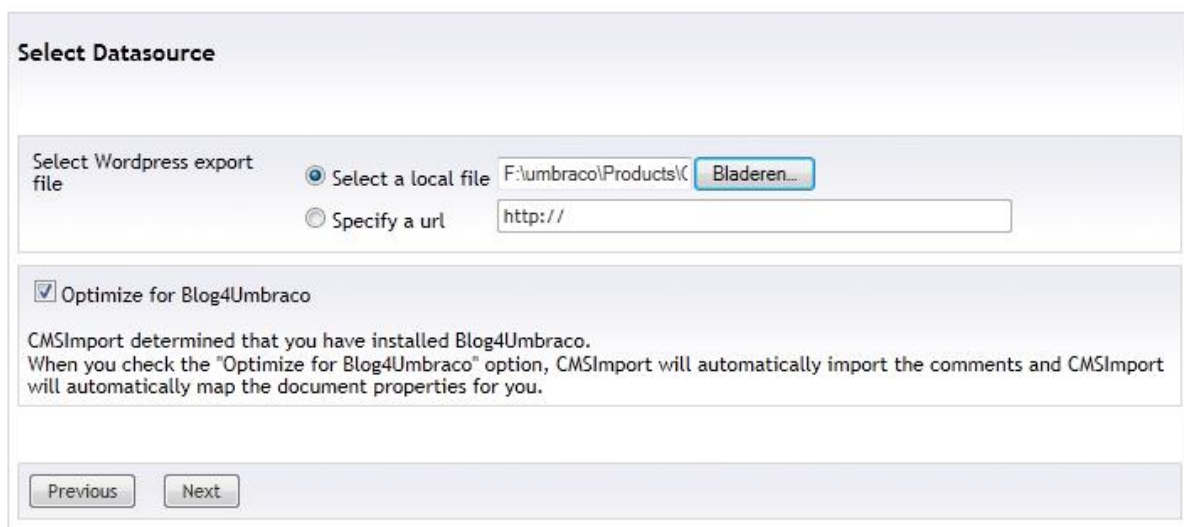
☒ Optimize for Blog4Umbraco

CMSImport determined that you have installed Blog4Umbraco.  
When you check the "Optimize for Blog4Umbraco" option, CMSImport will automatically import the comments and CMSImport will automatically map the document properties for you.

### 3.3.8 WordPress

In case of Wordpress you can upload a Wordpress file, or point to a location. When you have uBlogsy, Blog4Umbraco or the Blog starterkit installed you can specify that you want to optimize for either uBlogsy /Blog4Umbraco/Blog starter kit. When you enable that option CMSImport will automatically import comments and store it in the correct comments table also.

Don't select this option when you just want to import the blogposts and blogcomments based on an Umbraco documenttype.



**Select Datasource**

Select Wordpress export file

☒ Select a local file F:\umbraco\Products\K Bladeren...

☐ Specify a url

☒ Optimize for Blog4Umbraco

CMSImport determined that you have installed Blog4Umbraco.  
When you check the "Optimize for Blog4Umbraco" option, CMSImport will automatically import the comments and CMSImport will automatically map the document properties for you.

## 3.4 Select destination and document type

In this step you need to specify the location where you want to store the imported documents. This is prefilled if you started the import from the context menu in content. You can also specify the document type and you select the auto publish checkbox, when checked items are published automatically.

The "When the item already exists" and the primary key option are needed for content updates. With the "When the item already exists" option you specify what to do when an item is already

imported. Possible options are skip and update record. With the primary key you specify the key in the datasource. This field will be used to determine if an item is already imported.

To disable content updates uncheck the “Enable content updates” option. **Only do this when you don’t have a primary key in your datasource.** No relation between imported data and Umbraco document is stored so even when you run the import for the second time data will be imported as new records.

CMSImport can maintain the structure for you when you import content. This is normally be done using parent/child import definitions but sometimes you import content with a recursive foreign key to itself, for example when you import Product categories as in the example below. Some categories have a relation to a parent category. By selecting the recursive import option you can specify the key to its parent, in this case ParentProductcategoryID.

**Select Destination and DocumentType**

Location: Product Catalog [Delete](#) [Choose...](#)

DocumentType: ProductCategory

Auto publish: ☒

Enable content updates: ☒  
Only uncheck this option when you don't have a primary key in your datasource.

When the record already exists: Update record

Select primary key in datasource: ProductCategoryID

Enable recursive imports: ☒

Select foreign key for recursive import: ParentProductCategoryID

[Previous](#) [Next](#)

Some of the options might be disabled, or you might have a few extra options when run this step as a child import. See section structured import for more information

### 3.5 Create mapping

In this step you can specify the mapping between the fields from the data source and the properties of the Umbraco document type.

#### Quick tip:

When your fieldnames from the datasource are the same as the alias of the document property CMSImport will automatically map this field.

When you click on the green plus sign you can set advanced settings for that particular datatype. It should be at least possible to specify a default value for each datatype. Below you see the advanced settings for a datetime datatype.

**Create Mapping**

**Generic Properties**

<b>Name</b>	<input type="text" value="Name"/>	
<b>Publish at</b>	<input type="text" value="--Ignore--"/>	
<b>Unpublish at</b>	<input type="text" value="--Ignore--"/>	
<b>Create Date</b>	<input type="text" value="--Ignore--"/>	

**Date format**

Preview: 10/06/2011 13:30

**Default value**

[Clear Date](#)

Previous

Next

Every datatype has its own options. Using CMSImport PRO it's also possible to import media related to content the only requirement is that media is stored in the root of the website. Whenever you map against a media picker or rich text editor screen it's possible for you to specify a media location and it's even possible to specify a default value;

See chapter "Related media import" for more information.

Create Date	--ignore--	+
<b>Document property</b>		
<b>ProductNumber</b> Alias: productNumber Tab: ProductInfo	productCode	+
<b>Color</b> Alias: color Tab: ProductInfo	--ignore--	+
<b>ListPrice</b> Alias: listPrice Tab: ProductInfo	ListPrice	+
<b>Tags</b> Alias: tags Tab: ProductInfo	--ignore--	+
<b>Product Image</b> Alias: productImage Tab: ProductInfo	imagefileName	✖
<div> <b>Media location</b>            Import <a href="#">Delete</a> <a href="#">Choose...</a>    <b>Default value</b>  <a href="#">Choose...</a> </div>		

### 3.6 Confirm

In this step you can validate the selected options one more time. When you click next the import will start.

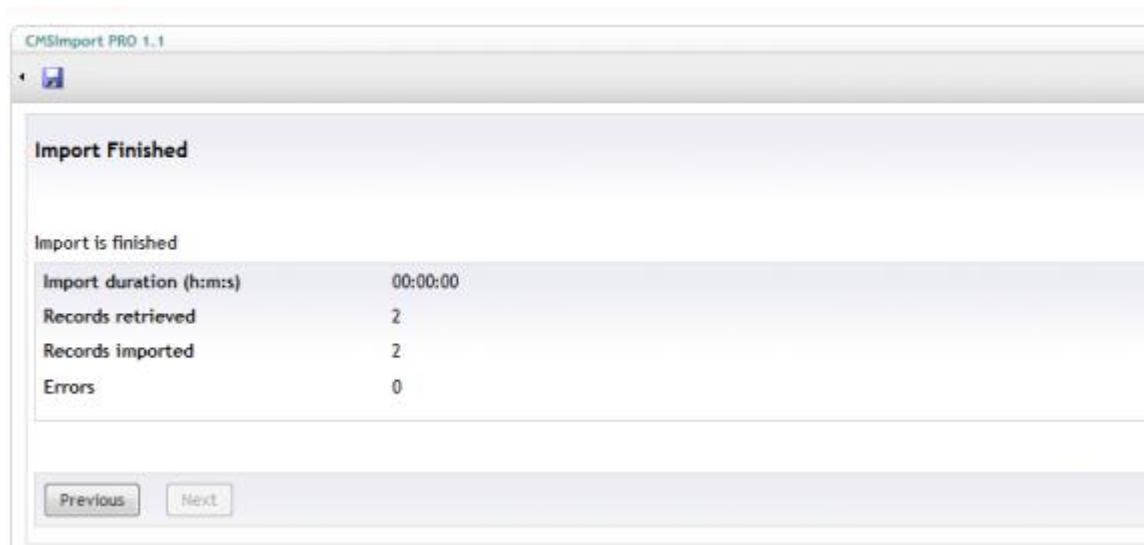
**Confirm**

Please validate your input and press Next to import the data.

<b>DataSource Type</b>	Excel file
<b>Datasource</b>	Provider=Microsoft.ACE.OLEDB.12.0;Data Source=F:\sites\UmbracoV45PRO\data\import\56953205-1edc-4121-9960-ec58105de06b_Products.xlsx;Extended Properties="Excel 12.0 Xml;HDR=YES";
<b>Data command</b>	ProductCategories\$
<b>Data Options</b>	
<b>Document location</b>	Product Catalog
<b>Document Type</b>	ProductCategory
<b>Autopublish</b>	True
<b>Content update options</b>	
<b>Action when the item already exists</b>	Update record
<b>Primary key</b>	ProductCategoryID
<b>Enable recursive imports</b>	True
<b>Foreign key for</b>	ParentProductCategoryID

### 3.7 Import

When you click next in this step the import starts. When the import is finished it will report what it did. If there were any errors it will also report the errors.



**Quick tip:**

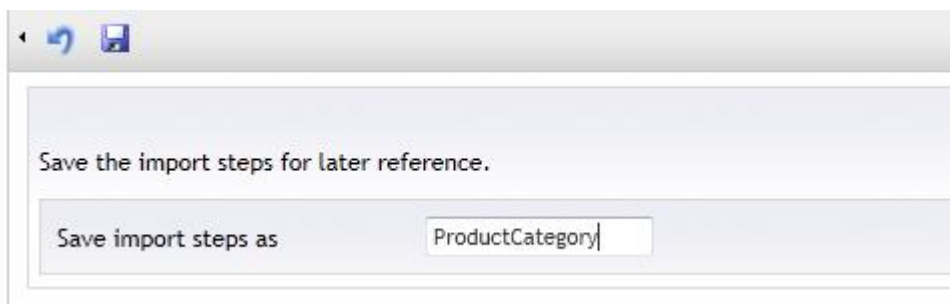
In case of test imports it might be useful to install the Content Maintenance Dashboard package also to bulk publish/unpublish and delete content nodes.

You can download the package via:

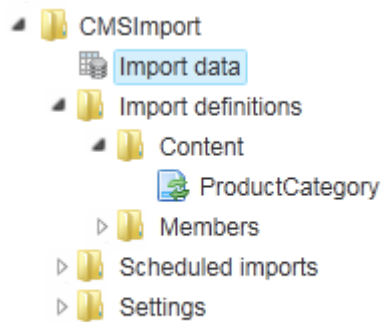
<http://our.umbraco.org/projects/developer-tools/content-maintenance-dashboard-package>

### 3.8 Save Import steps



When you click save, you can specify a name and when hitting the save button again the import steps are saved for later use.



Saved imports are stored as Import definitions and can be found in the Import definitions tree.

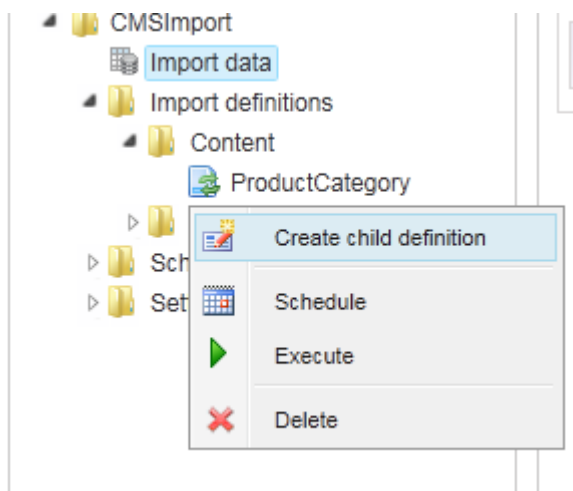


Possible icons for saved content items

-  Standard import definition
-  Recursive import definition

### 3.9 Create a child definition

When you want to use structured imports its possible to create a child definition. In the previous example we were importing Product categories. Now we can import products for those categories. Right click on the parent definition (in this case ProductCategory) and select **Create child definition**

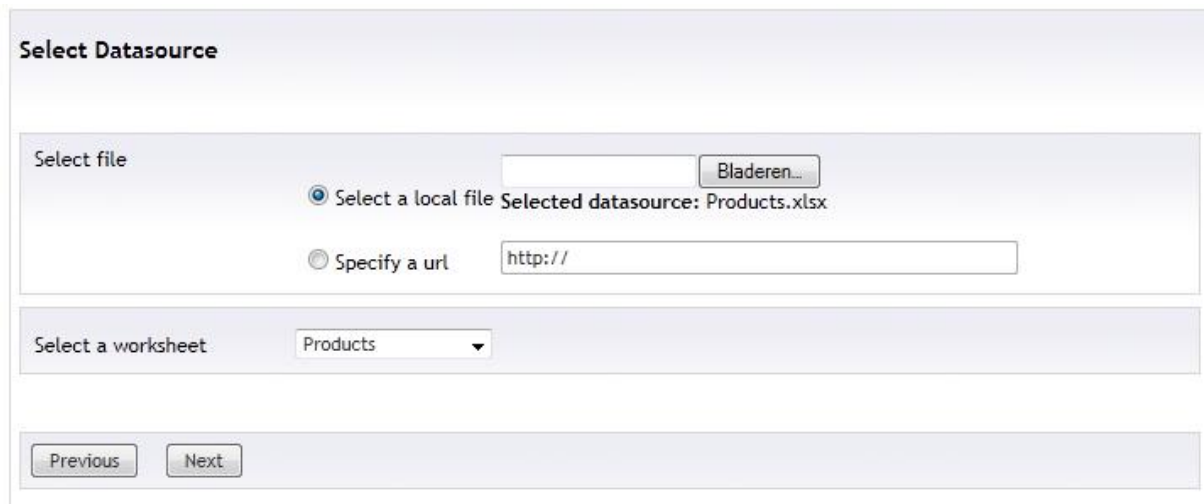


This will start the import wizard again, with a few small changes.

#### 3.9.1 Select datasource

When selecting the same datasource type as the parent it will automatically select the same datasource as its parent.





**Select Datasource**

Select file

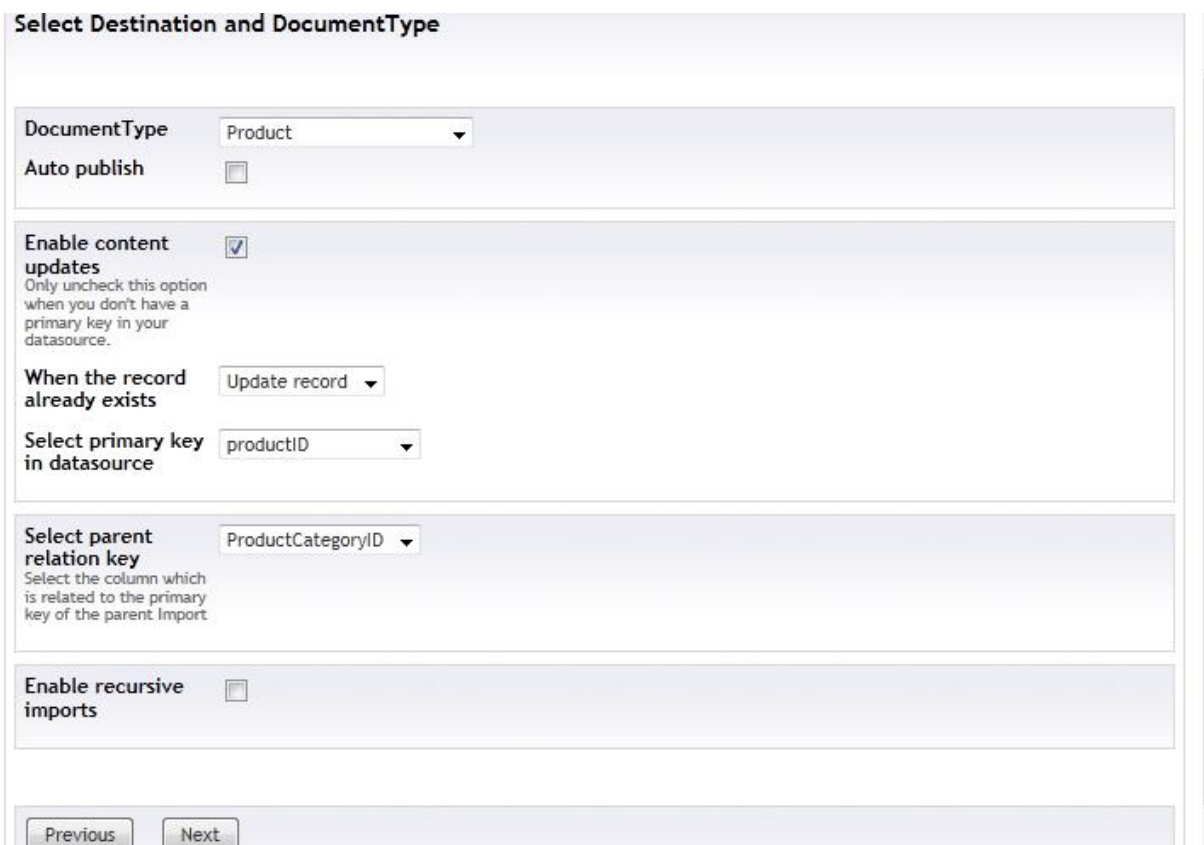
☒ Select a local file Selected datasource: Products.xlsx

☐ Specify a url

Select a worksheet

### 3.9.2 Specify content import options

When you set the content import options you don't need to specify the location since that will be determined based on the parent record. Instead you specify the **parent relation key**. In this case we want to add a relation based on category id so we specify ProductCategoryId as the relation. All other steps are the same.



**Select Destination and DocumentType**

DocumentType

Auto publish ☐

Enable content updates ☒  
Only uncheck this option when you don't have a primary key in your datasource.

When the record already exists

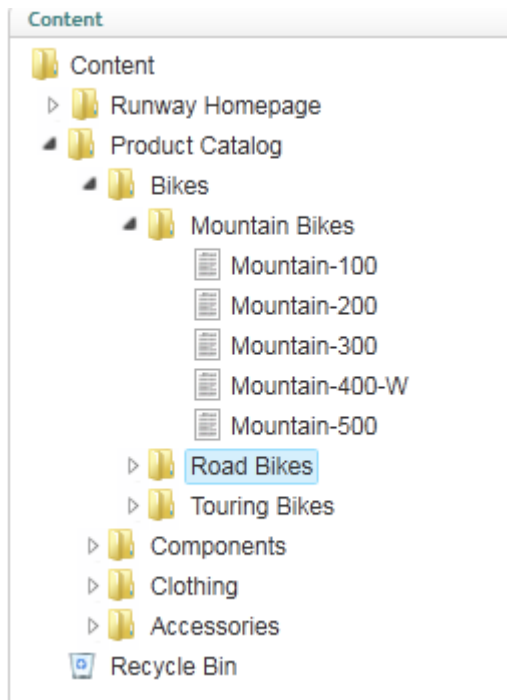
Select primary key in datasource

Select parent relation key   
Select the column which is related to the primary key of the parent Import

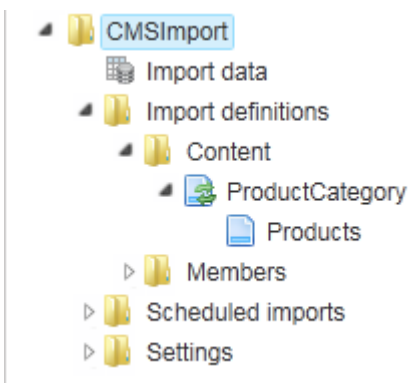
Enable recursive imports ☐

## 3.10 End result

When importing these two definitions the content tree is filled with categories and products.



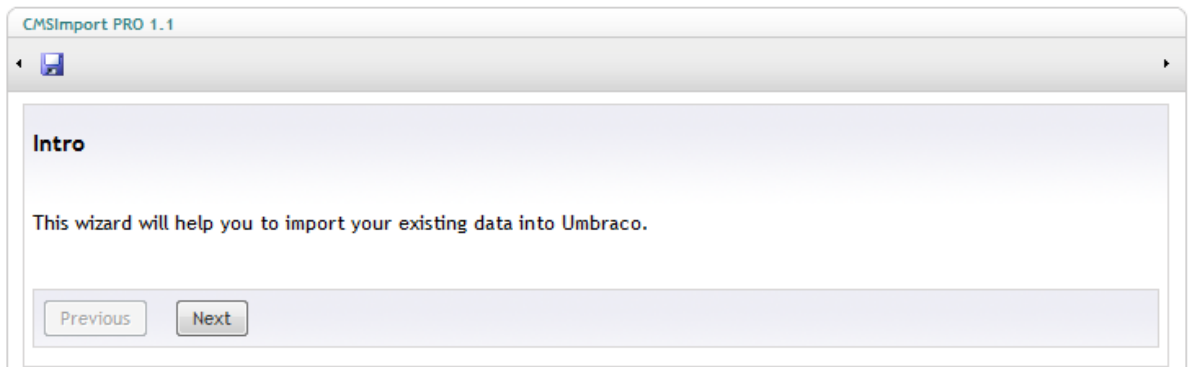
The import definition tree contains two items ProductCategory and Products as the child import definition



## 4 Import members

### 4.1 Introduction

When the wizard starts it will start with an introduction



### 4.2 Select Datasource Type

In this step you are asked to specify a datasource type and how the data should be imported (content/members).



The following datasource types are supported by default:

- Access database
- CSV
- Excel File
- SQL Server
- XML

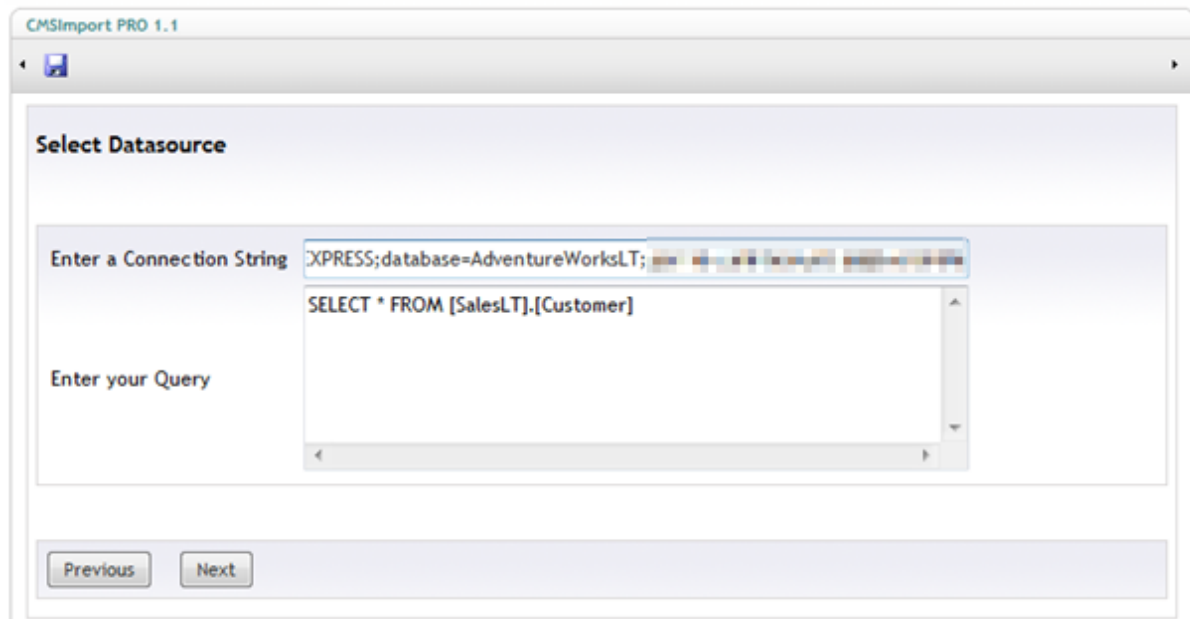
It is possible to create extra Datasource Types. See chapter extend CMSImport for more information.

### 4.3 Select Datasource

In the select Datasource step you need to provide the information for the selected datasource type.

#### 4.3.1 SQL Server

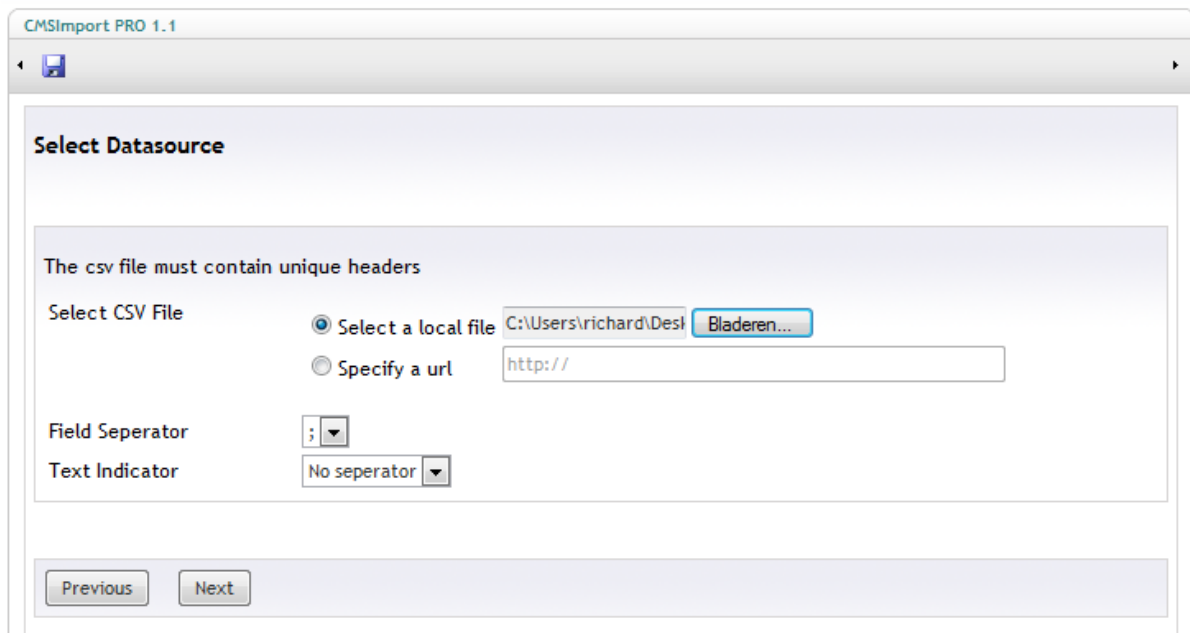
In case of SQL server you need to provide the Connection string and the query that you want to use to retrieve the data.



The screenshot shows the 'Select Datasource' window in CMSImport PRO 1.1. It has a title bar with the application name and a standard Windows window control set. The main area is titled 'Select Datasource'. Below the title, there are two input fields: 'Enter a Connection String' and 'Enter your Query'. The 'Enter a Connection String' field contains the text 'XPRESS;database=AdventureWorksLT;'. The 'Enter your Query' field contains the text 'SELECT \* FROM [SalesLT].[Customer]'. At the bottom of the window, there are two buttons: 'Previous' and 'Next'.

#### 4.3.2 CSV File

In case of a CSV file you need to select a CSV file. You also need to provide the field separator and Text Indicator. A field separator is the character that is used to separate the columns. The text indicator is a character that indicates a text string. The default settings are based on an export from Microsoft Excel.



The screenshot shows the 'Select Datasource' window in CMSImport PRO 1.1. It has a title bar with the application name and a standard Windows window control set. The main area is titled 'Select Datasource'. Below the title, there is a message: 'The csv file must contain unique headers'. Below this message, there are two radio buttons: 'Select a local file' (which is selected) and 'Specify a url'. The 'Select a local file' radio button is followed by a text field containing 'C:\Users\richard\Dest' and a button labeled 'Bladeren...'. The 'Specify a url' radio button is followed by a text field containing 'http://'. Below these radio buttons, there are two dropdown menus: 'Field Separator' (which is set to ';') and 'Text Indicator' (which is set to 'No separator'). At the bottom of the window, there are two buttons: 'Previous' and 'Next'.

#### 4.3.3 XML File

In case of an XML file you need to select the xml File. Optional you can specify an xpath expression.

**Select Datasource**

Select XML File

☐ Select a local file  Bladeren...

☒ Specify a url

Xpath Expression

When using xml you can only use elements, attributes are not supported at the moment. All elements that you want to map later must be under the first child node of the xml document.

#### 4.4 Select member type and role

In this step you can select the member type and assign one or more roles. With the “When the item already exists” option you specify what to do when an item is already imported. Possible options are skip and update record.

When the “Automatic generate password” option is checked a password is automatically generated for the imported member. When the “Send credentials via mail” is checked an email with login credentials is send to the imported member. You can edit the email template , check chapter settings on how to do this.

**Select Member type and roles**

Member type

Automatic assign role(s)

☒ Customer

☒ Company

When member exists

Automatic generate password ☒

Send credentials via mail ☐

Import content related media ☐

Make sure all folders containing the media are stored in the root of this umbraco environment! Read the manual for a detailed description.

## 4.5 Create Mapping

In this step you can specify the mapping between the fields from the data source and the properties of the Umbraco member type.

When you click on the green plus sign you can set advanced settings for that particular datatype. It should be at least possible to specify a default value for each datatype.

**Create Mapping**

**Generic Properties**

Member name	name	+
Member login	email	+
Member email	email	+

**Document property**

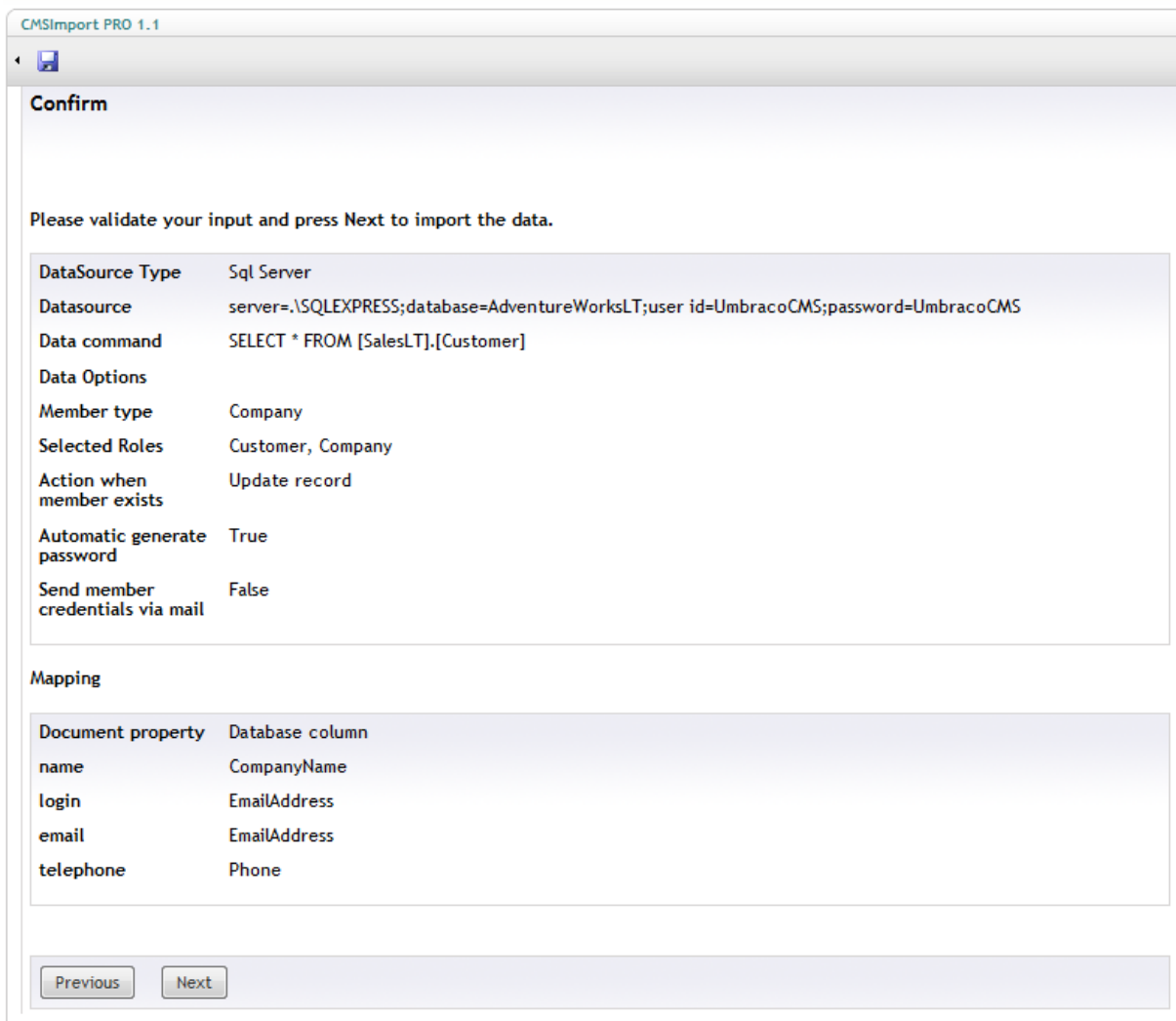
Department Alias: department Tab: Properties	Afdeling	+
Country Alias: country Tab: Properties	Country	+

### Quick tip:

When your fieldnames from the datasource are the same as the alias of the document property CMSImport will automatically map this field.

## 4.6 Confirm

In this step you can validate the selected options one more time. When you click next the import will start.



**Confirm**

Please validate your input and press Next to import the data.

DataSource Type	Sql Server
Datasource	server=.\SQLEXPRESS;database=AdventureWorksLT;user id=UmbracoCMS;password=UmbracoCMS
Data command	SELECT * FROM [SalesLT].[Customer]
Data Options	
Member type	Company
Selected Roles	Customer, Company
Action when member exists	Update record
Automatic generate password	True
Send member credentials via mail	False

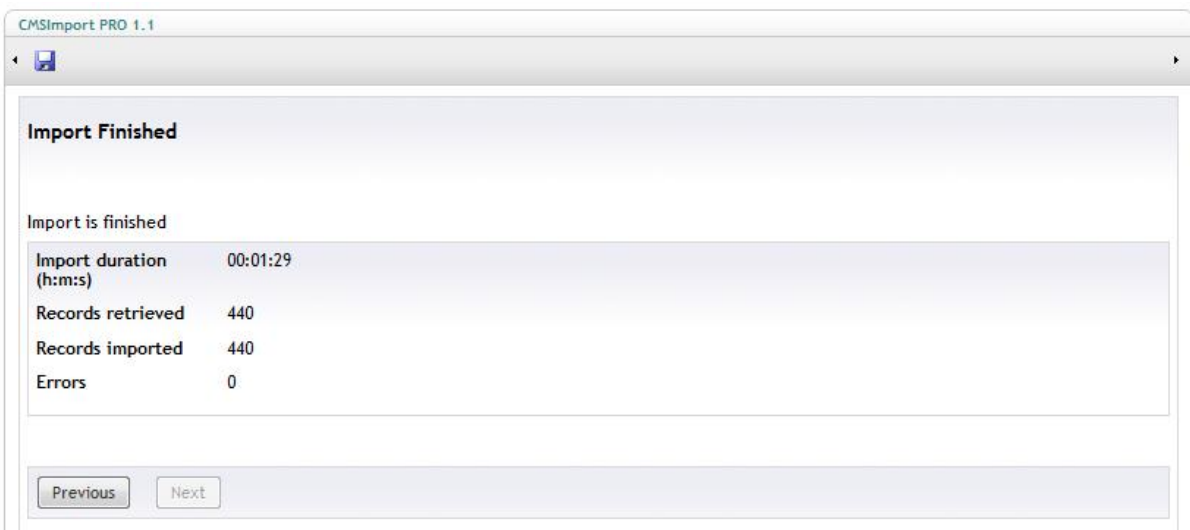
**Mapping**

Document property	Database column
name	CompanyName
login	EmailAddress
email	EmailAddress
telephone	Phone

Previous Next

## 4.7 Import

When you click next in this step the import starts. When the import is finished it will report what it did. If there were any errors it will also report the errors.



**Import Finished**

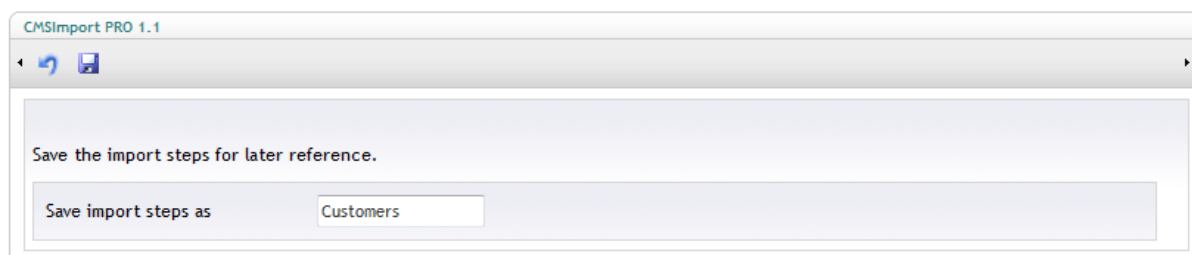
Import is finished

Import duration (h:m:s)	00:01:29
Records retrieved	440
Records imported	440
Errors	0

Previous Next

## 4.8 Save Import steps

When you click save, you can specify a name and when hitting the save button again the import steps are saved for later use.



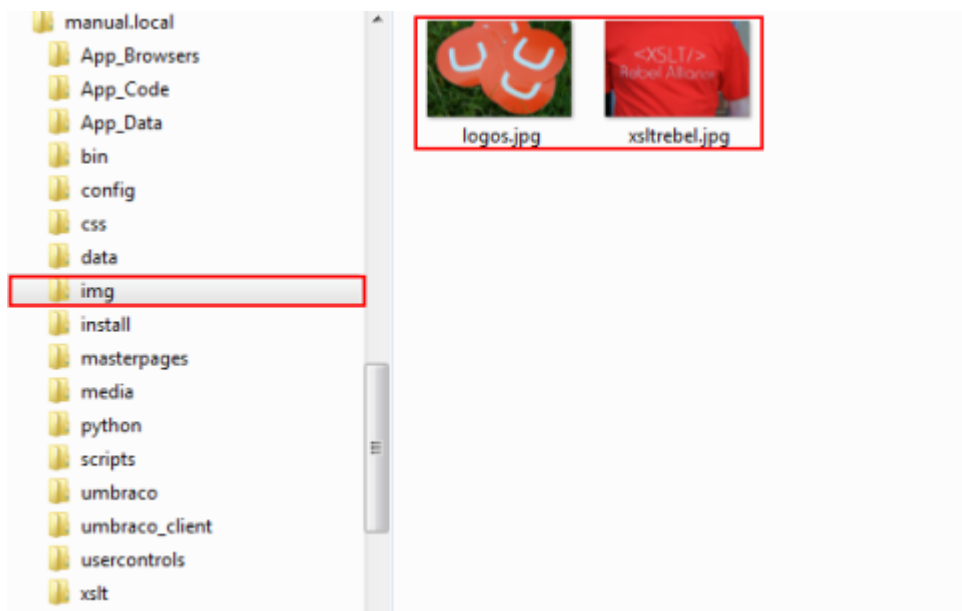


## 5 Related media import

CMSImport can import media also. This isn't a separate import process but integrated in content or media import. When CMSImport finds a reference to a relative path it will try to get the item and convert it to a media item, or store it in the media folder in case of an upload field.

**The only required thing is that the original media folder is copied to the root of your Umbraco folder.**

In the example below the img folder of the original site containing two images is stored in the Umbraco root.



In the Import wizard you can specify that you want to import media items also. Check the option “Import content related media” on the advanced setting of the property

**Create Mapping**

**Generic Properties**


Name	--ignore--	+
Publish at	--ignore--	+
Unpublish at	--ignore--	+
Create Date	--ignore--	+

Document property	Database column	
<b>Body Text</b> Alias: bodyText Tab: Content	--ignore--	+ 1
<b>Hide in navigation</b> Alias: umbracoNavHide Tab: Properties	--ignore--	+
<b>Download</b> Alias: mediaPicker Tab: Content	--ignore--	+ 2
<b>Upload</b> Alias: upload Tab: Content	--ignore--	+ 3

Previous Next

## 5.1 Rich text editor

When you map against a Richtext datatype you can specify the import options as shown below. When a reference to an image is found in the content, CMSImport will create a media Item and update the image source to the new Media item.

Document property	Database column
<b>Body Text</b> Alias: bodyText Tab: Content	--ignore-- 

**Import Images?**

By enabling Image import, images stored with a relative url will be imported automatically into Umbraco and the reference will be updated. Make sure all folders containing the images are stored in the root of this umbraco environment! Read the manual for a detailed description.

☒ Enable Image import

**Select image location**

Import [Delete](#) [Choose...](#)

**Import files?**

By enabling Image import, images stored with a relative url will be imported automatically into Umbraco and the reference will be updated. The following extensions are supported: .doc, .docx, .pdf, .ppt, .pptx, .rar, .xls, .xlsx, .zip

You can add/remove extensions in the settings folder of CMSImport. Make sure all folders containing the files are stored in the root of this umbraco environment! Read the manual for a detailed description.

☒ Enable file import

**Select file location**

Import [Delete](#) [Choose...](#)

## 5.2 Media picker

When you map against a mediapicker datatype you can specify the import options as shown below and CMSImport will create a media item and store the Id of the media item .

Download  
Alias:mediaPicker  
Tab:Content

Image ✕

Media location  
Import [Delete](#) [Choose...](#)  
  
Default value  
Public file [Delete](#) [Choose...](#)

## 5.3 Multi Node Tree Picker (MNTTP)

When you map against the Multi Node Tree Picker you can specify the import options as shown below and CMSImport will create a media item and store the Id of the media item. **Only datatypes configured for Media can be used.**

MediaMntp  
Alias:mediaMntp  
Tab:Content

Image ✕

Media stored with a relative url will be imported automatically into Umbraco and the reference will be updated. Make sure all folders containing the images are stored in the root of this umbraco environment! Read the manual for a detailed description.

Use a comma separated string of images if you want to assign more than 1 image example: img/image1,img/image2

Media location  
Specify a location where to store the imported media items  
  
img [Delete](#) [Choose...](#)  
  
Default value  
Pick a default media item that will be used in case the assigned value is null  
  
no\_image\_available\_small [Delete](#) [Choose...](#)

When the import process is finished you'll see that the media items are imported




**TIP** Using the MNTTP datatype you can specify a comma separated list of images you want to import.

## 5.4 Digibiz Advanced Media Picker

When you map against the Digibiz Advanced Media Picker (DAMP)<sup>3</sup> you can specify the import options as shown below and CMSImport will create a media item and store the Id of the media item. **Only the default Image and File media types are supported.**

DAMP Classic  
Alias: dampClassic  
Tab: Photos

Image


Media stored with a relative url will be imported automatically into Umbraco and the reference will be updated. Make sure all folders containing the images are stored in the root of this umbraco environment! Read the manual for a detailed description.

Use a comma separated string of images if you want to assign more than 1 image example: img/image1,img/image2

**Media location**  
Specify a location where to store the imported media items

**Public** [Delete](#) [Choose...](#)

**Default value**  
Pick a default media item that will be used in case the assigned value is null

[Choose...](#)

When the import process is finished you'll see that the media items are imported

DAMP Classic
[Choose...](#)
[Create...](#)


[xsltrebel](#)
[Edit](#)
[Open](#)
[Pixlr](#)
[Remove](#)


[logos](#)
[Edit](#)
[Open](#)
[Pixlr](#)
[Remove](#)

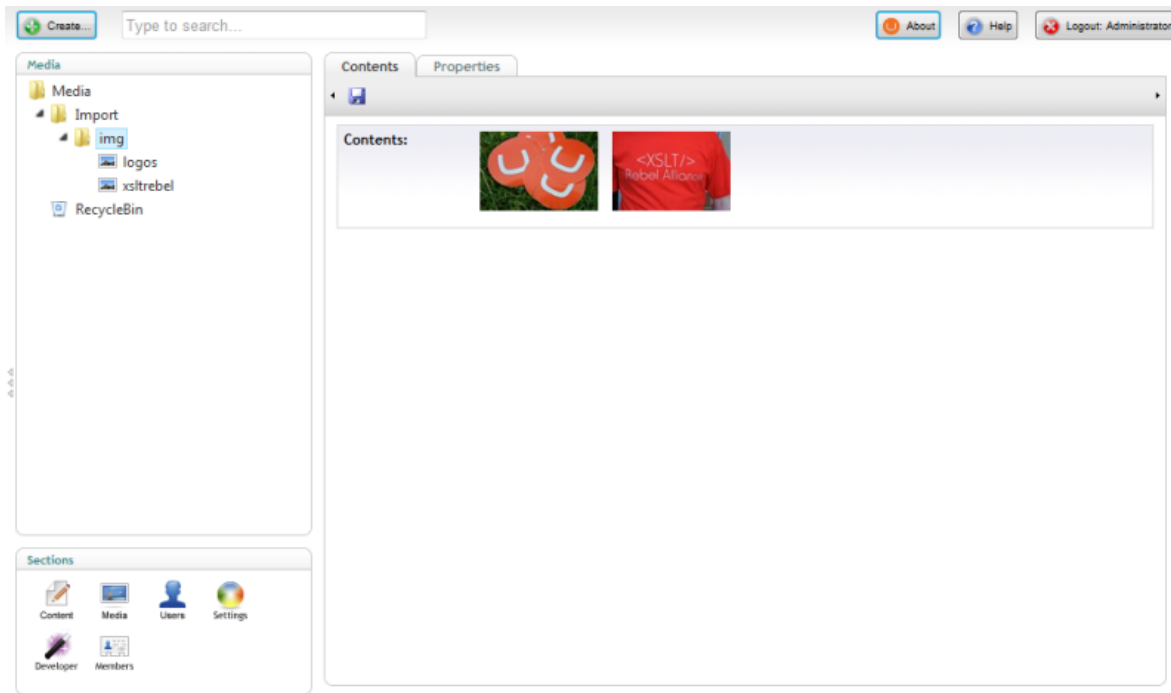
**TIP** Using the DAMP V2 datatype you can specify a comma separated list of images you want to import.

## 5.5 Upload field

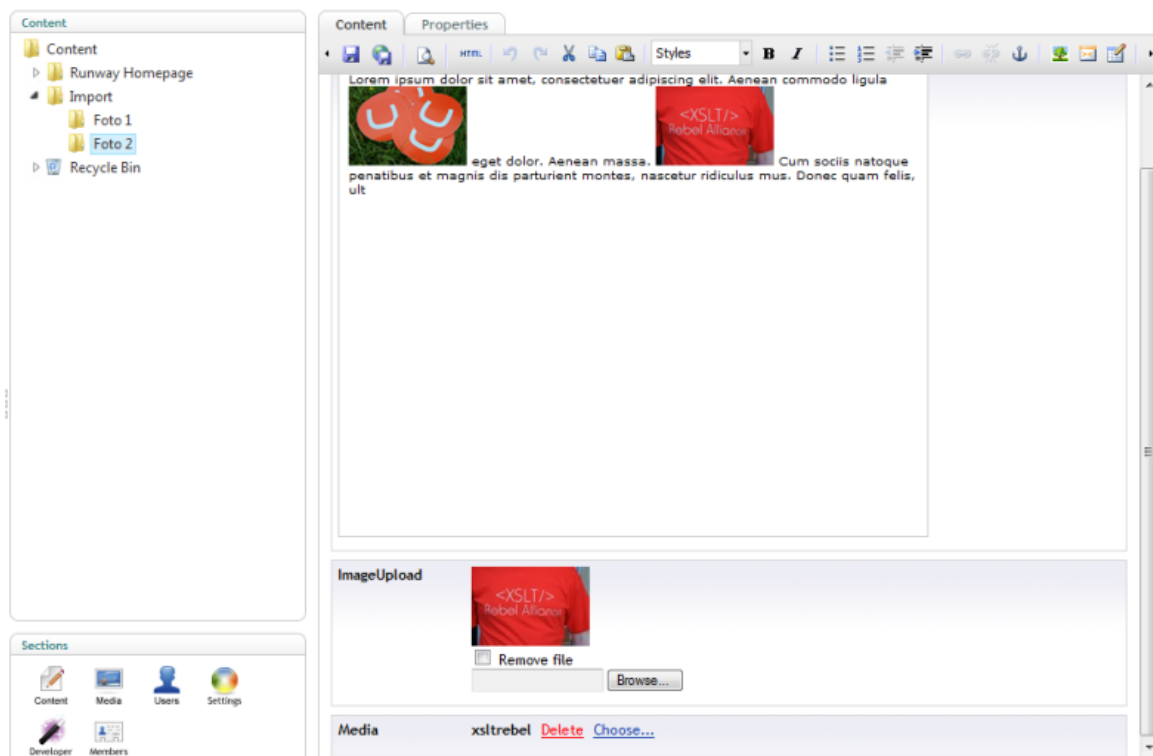
When an image (could also be a file) reference is mapped against an upload field. CMSImport will store the image in the Umbraco Media folder and update the reference in the Upload field

When the import process is finished you'll see that the media items are imported

<sup>3</sup> Damp version 2.x **only**



And when you open an imported item in the content section you'll see that all the references are updated to the imported media items



Currently this import process will work for the following datatypes:

- Upload field
- Media Picker
- TinyMCE
- Multi Node Tree Picker (Media only)
- DAMP Version 2.0

## 6 Schedule Imports

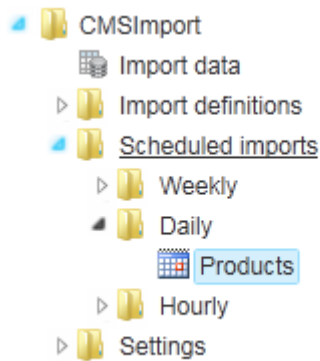
With CMSImport it's possible to run imports for on a certain day/time. When you right click on a saved import item (content/member) and click schedule the following screen will show up.

The screenshot shows the 'Schedule Imports' dialog box in CMSImport PRO 1.1. The dialog has a title bar with the text 'CMSImport PRO 1.1' and a standard window icon. The main area contains several fields and options:

- Saved import name:** A text field containing 'CMSImport manual'.
- Scheduled task name:** A text field containing 'Daily import records for CMSImport manual'.
- Notify emailadress:** A text field containing 'manual@cmsimport.com'.
- Run task as:** A dropdown menu with 'Administrator' selected.
- Execute every:** Three radio buttons: 'Week' (selected), 'Day', and 'Hour'.
- Days:** A list of days with checkboxes: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. All checkboxes are currently unchecked.
- Time:** Two dropdown menus showing '0' and '0' separated by a colon, indicating a time of 0:00.

You can schedule imports to run every week on certain days/time, every day on a certain time or every hour.

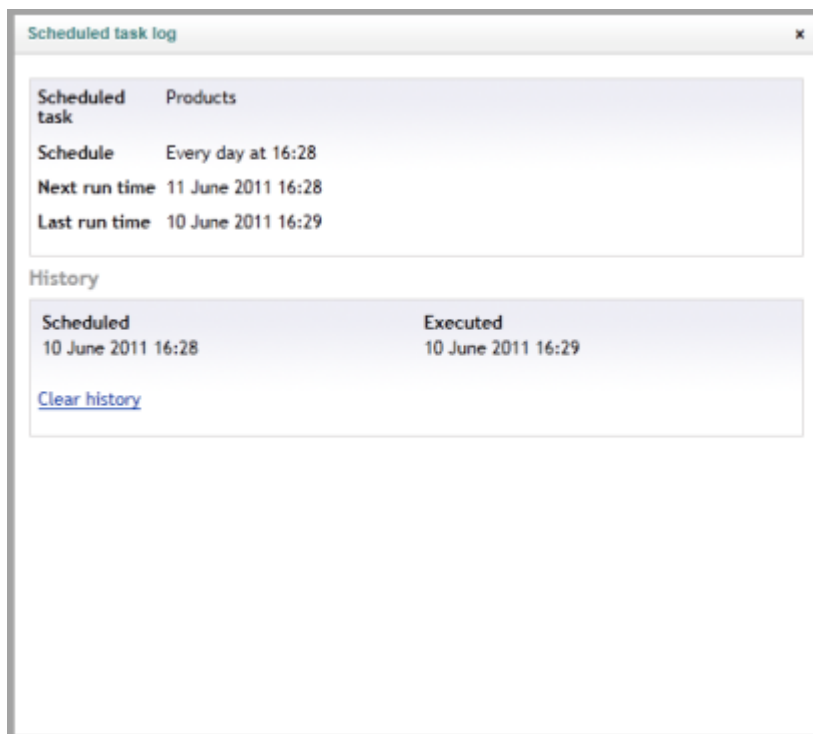
You can specify an email address to receive a notification when the import is finished and you can specify a user that will be assigned as creator for imported documents/members. When you click save the item will run on the selected day/time.



When the import is finished you will receive a notification email on the specified email address.

## 6.1 Scheduled task log

When you right click on a scheduled task you can view when the task did execute. As you already can see in the image below the scheduled time can be a little different from the real executed time. This can be caused by the website not being up at the given time. The scheduler will always execute the task when the site is back up and did not run the task. This dialog gives some insight when that happened.



## 7 Import Dashboard

Using the Import Dashboard you can allow content editors to upload data and media items(optional)without giving the user access to the Developer section. This is ideal when you want your users to upload a new product catalog, or member information. The user can't modify the import definition using this Dashboard, **only upload data**.

The screenshot shows the CMSImport Import Dashboard. On the left is a sidebar with a 'Content' section containing 'Simple website', 'Blog', and 'Recycle Bin'. Below it is a 'Sections' section with icons for 'Content', 'Media', 'Users', 'Settings', 'Developer', 'Members', and 'Shop'. The main area has a top navigation bar with 'Get Started', 'Last Edits', 'Change Password', 'Import Data' (selected), 'Comments', and 'Blog'. Below the navigation bar is a heading 'Import content or members into Umbraco using this simple tool.' followed by a list of instructions: 'Select what you want to import', 'Upload the file you want to import', 'Optional upload a zipfile with the media you want to import', and 'Select Import and wait until the import is finished.' Below this is a form with three main sections: 1. 'What do you want to import?' with a dropdown menu showing 'ProductCatalog' and a red '1' next to it. 2. 'Upload data file' with a text input field and a 'Bladeren...' button, with a red '2' next to it. 3. 'Upload media file' with a text input field and a 'Bladeren...' button, with a red '3' next to it. At the bottom of the form is an 'Import' button.

1. The user can select an import definition(saved in the CMSImport tree) to import. Only Import definitions that are enabled to import from the Dashboard can be selected. If there is only one import definition that can be selected this option will not be shown on the form
2. Using this upload field Users can upload their data file.
3. When the import definition allows the import of media, a separate upload field for a media file will be shown. The user can zip their folder with media that needs to be imported and use this upload field to import their media

When the datasource is uploaded CMSImport will first inspect if the datasource is still correct. Then it will import data.

See settings chapter how to configure the Import dashboard



## 8 Settings

Here you can modify the following settings

### 8.1 Login credential settings

These settings will be used when an email with login credentials is send to an imported member.

You can specify:

- The from email address
- The email subject
- The email body

In the email body you can use the following placeholders that will be replaced with real values when sending the email:

Snippet	Description
[#loginname]	The loginname of the imported member
[#password]	The password (NOT ENCRYPTED) of the imported member
[#email]	The Email address of the imported member
[#{property alias}]	This will replace the property alias with the imported value

### 8.2 Scheduler result settings

These settings will be used when an email is send to inform a user that a scheduled task is finished.

You can specify:

- The from email address
- The email subject
- The email body

In the email body you can use the following placeholders that will be replaced with real values when sending the email:

Snippet	Description
[#Taskname]	The name of the scheduled task
[#Duration]	The duration of the import process
[#RecordCount]	The amount of records in the datasource
[#RecordsAdded]	The amount of records added during the import process
[#RecordsUpdated]	The amount of records updated during the import process
[#RecordsSkippedCount]	The amount of records skipped during the import process
[#ErrorCount]	The amount of errors during the import process
[#Errors]	The error descriptions that occurred during the import process

### 8.3 Media settings

In this section you can specify which file extensions CMSImport needs to parse in case of media import mapped against a Rich Text editor. Only the selected file extensions will be picked up to import. You can also specify some domains url's that needs to be picked up. Url's starting with that

domain will also be picked up. You still need to make sure files are located in the root of your Umbraco install.

**Additional media settings for fields mapped against a Richtext property.**

**Allowed extensions for file import**  
Specify file extensions for files that must be imported when mapped against a Richtext property, other file extensions will be ignored.  
  
When mapped against a normal media picker all file extensions will be picked up.

[Add extension](#)

>>

<<

doc  
docx  
pdf  
ppt  
pptx  
rar  
xls  
xlsx

**Parse media for the following domain urls**  
CMSImport can only import files that are stored in the root folder of Umbraco by default. Sometimes a full url is specified in a field that is mapped against a Richtext property. This setting allows you to specify domain urls, so url's containing these domain urls will be converted to relative paths, example  
http://www.cmsimport.com/images  
/screenshot.png will be converted to /images/screenshot.png when you specify the domain url  
http://www.cmsimport.com/

[Add domain url](#)

>>

<<

http://elpadrinodkfileswordpre  
http://www.soetemansoftware.nl

## 8.4 Data adapters

CMSImport comes with a lot of DataAdapters you might not want to use. In this section you can disable the items you don't want to use.

**CMSImport PRO 2.0**

Below you can enable or disable the different Data Adapters.  
If you enable the Office data adapters make sure you have the [Microsoft Data Connectivity Components](#) installed.

**General**  
☒ Csv  
☒ Rss Feed  
☒ Sql Server  
☒ Xml

**Office**  
☒ Access database  
☒ Excel file

**Blog**  
☒ BlogML Comments  
☒ Wordpress blogposts  
☒ Wordpress Comments  
  
☒ BlogML blogposts

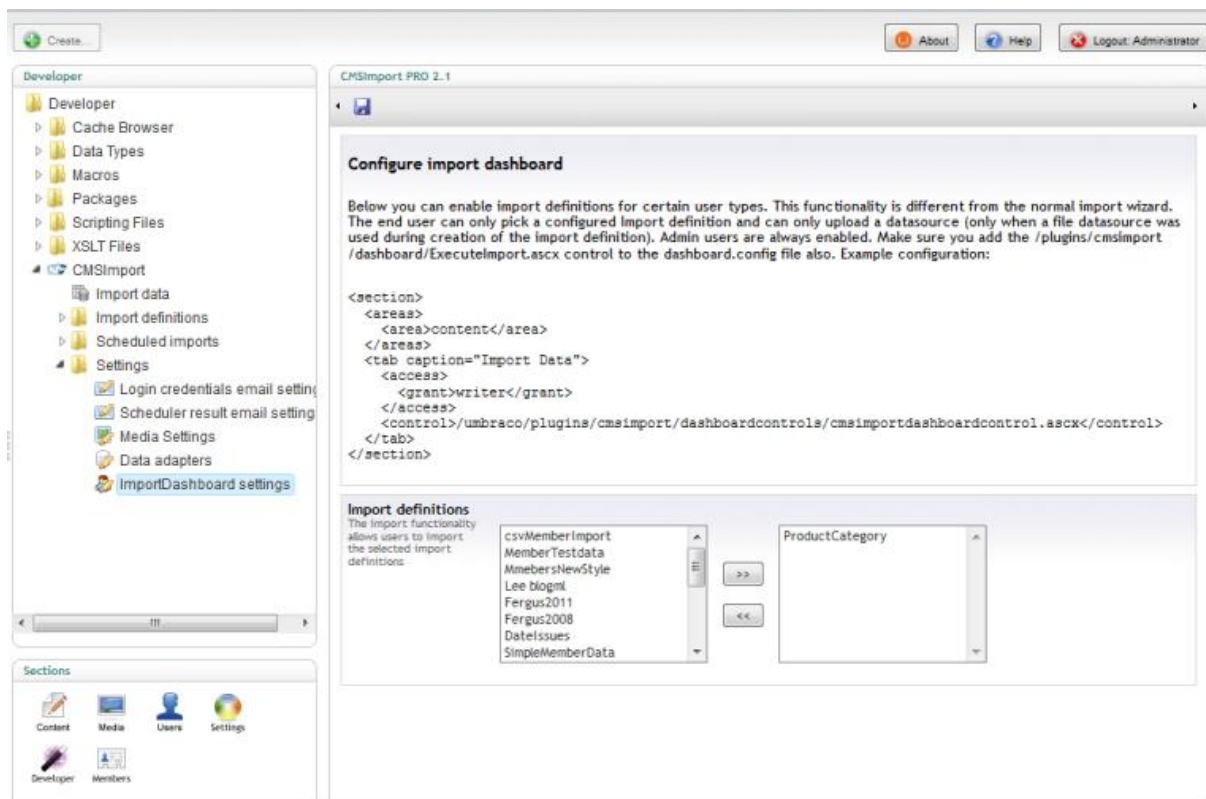
## 8.5 Configure Import Dashboard

In this section you can configure the Import Dashboard. You can specify which import definitions you want to allow to be used on the Dashboard. Only datasource that supports uploading a datasource file (such as Excel, Xml, Csv etc) are supported.

Since it's a dashboard and you can put this dashboard anywhere you like, only a sample xml snippet will be shown that you can copy and paste into the Dashboard.config file.

**To only allow selected usergroups access to the Dashboard use the Access element**

More information how to use the Dashboard.config file can be found at <http://our.umbraco.org/wiki/reference/files-and-folders/dashboardconfig>



## 9 Extend CMSImport

**Although you don't need to**, there are several ways to extend CMSImport. This chapter describes how you can make use of these extension points in your code. All samples can be downloaded from <http://www.cmsimport.com/download/cmsimportsamplepackage.zip> and a sample VS2010 solution can be downloaded from <http://www.cmsimport.com/download/CMSImportVs2010Samples.zip>. To use the samples you need to install the AdventureWorks Lite database which can be downloaded from the CodePlex site <http://msftdbprodsamples.codeplex.com/> and use CMSImport PRO(although some samples might work in the free version)

### 9.1 Setting up Visual Studio

When you want to create an extension for CMSImport you can create a new Class Library and add a reference to the Assembly **CMSImport.Extensions**. When you need to use Umbraco functionality, or want to use CMSImport events also you can create references to the assemblies Umbraco, interfaces, cms and businesslogic.

### 9.2 What's in CMSImport.extensions

#### 9.2.1 DataAdapter

A DataAdapter is used as a generic interface to talk to datasources. You can use a DataAdapter by using ImportDataAdapter as a base class. Checkout the samples for a full implementation of a DataAdapter.

#### 9.2.2 FieldAdapter

A FieldAdapter could be used to convert original data in a datasource to a datatype specific format during import. CMSImport uses this already to make sure values like "true/false" will be converted to a Boolean value which could be mapped against a true/false datatype without causing an error and it uses FieldAdapters to import media.

A FieldAdapter is basically an implementation of the IFieldAdapter interface, which contains the DataAdapterId property and the Parse method. The DataAdapterID must match with the DataAdapterControl. This will ensure that the parse method gets called when data is imported for the Color property.

Edit datatype: ProductColor

Name: ProductColor

Render control: Dropdown list

Data Type GUID: a74ea9c9-8e18-4d2a-8cf6-73c6206c5da6

Settings

Database datatype: Integer

Add prevalue:

This Parse method accepts the following parameters:

- Value. The original value which we can manipulate
- Document property. Gives us information about the document type
- FieldAdapterOptions Bunch of general setting mostly related to Media Import

### 9.2.3 Events

There are several events that you can use for both content and member import. You can hook up events in the same way you hook up events for other Umbraco functionality by deriving from `ApplicationBase` and hook up the event in the constructor.

```
public class Sample :umbraco.BusinessLogic.ApplicationBase
{
    public Sample()
    {
        //Hook up event
        MemberImport.RecordImporting +=
            new MemberImport.RecordImportingEventHandler (MemberImport_RecordImporting);
    }

    void MemberImport_RecordImporting(object sender, RecordImportingEventArgs e)
    {
        //Logic here
    }
}
```

#### 9.2.3.1 RecordImporting

The recordImporting event gets hit before a record got imported, the sender object contains the document or member being imported. `RecordImportingEventArgs` contains the following information:

- DataAdapter
- DataKeyName (Primary key column)
- DataKeyValue (Primary key value)
- Items (collection of items that contains the original data)

**You can cancel the import for this record by setting the cancel property to true.**

#### 9.2.3.2 RecordImported

The RecordImported event gets hit after a record got imported, the sender object contains the imported document or member. `RecordImportedEventArgs` contains the following information:

- DataAdapter
- DataKeyName (Primary key column)
- DataKeyValue (Primary key value)
- ImportAction (ImportAsNew, update or skip record)
- Items (collection of items that contains the original data)

#### 9.2.3.3 Importing

The Importing event gets hit before the import starts, `ImportEventArgs` contains the alias of the DataAdapter.

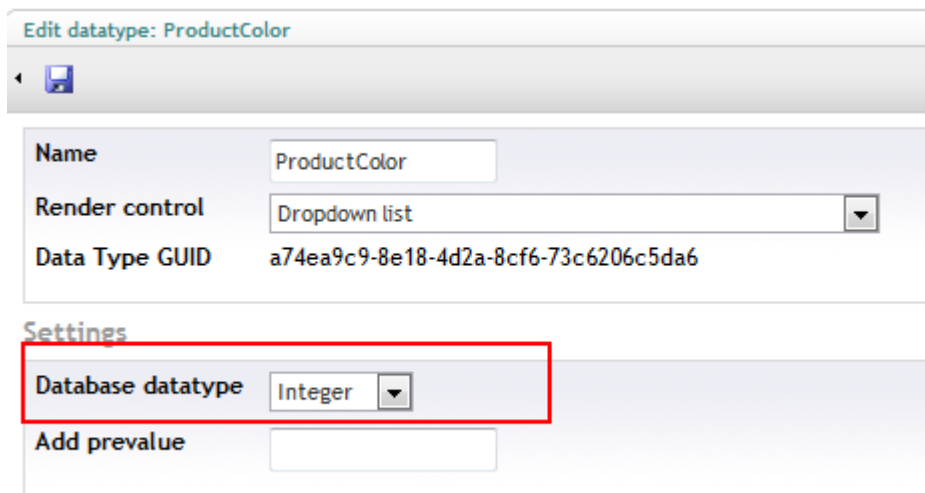
#### 9.2.3.4 Imported

The Imported event gets hit after the import has finished, ImportEventArgs contains the alias of the DataAdapter.

### 9.3 Samples

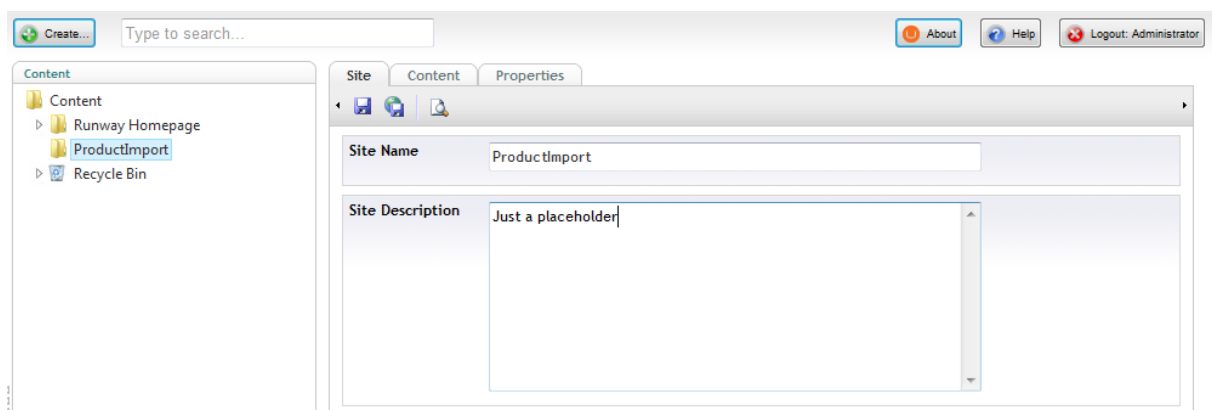
When you've downloaded and installed the samples you can use two extra document types in your Umbraco installation, ProductCategory and Product.

Before you start make sure the ProductColor datatype uses the integer Database datatype.



The screenshot shows the 'Edit datatype: ProductColor' dialog. The 'Name' field is 'ProductColor', the 'Render control' is 'Dropdown list', and the 'Data Type GUID' is 'a74ea9c9-8e18-4d2a-8cf6-73c6206c5da6'. Under the 'Settings' section, the 'Database datatype' is set to 'Integer' (highlighted with a red rectangle), and the 'Add prevalue' field is empty.

All samples will use the above mentioned document types. Only thing you need to do is to create a root folder and use that folder as import location for all the samples.



The screenshot shows the Umbraco CMS interface. On the left is the 'Content' tree with 'Content', 'Runway Homepage', 'ProductImport', and 'Recycle Bin'. The main area shows the 'Site' configuration for 'ProductImport'. The 'Site Name' is 'ProductImport' and the 'Site Description' is 'Just a placeholder'.

### 9.4 Create a DataAdapter for Product Categories

A DataAdapter could be used to communicate with a DataSource. Basically it lets you set some properties and returns data to CMSImport which can then be imported. In the Sample project we are using two DataAdapters

1. ProductCategoryAdapter, responsible for importing all product category data
2. ProductDataAdapter, responsible for importing all product data.

#### 9.4.1 Create the class

To create a custom DataAdapter you need to create a class that derives from **CMSImport.Extensions.DataAdapter.ImportDataAdapter**

```
public class AdventureWorksLTProductCategoriesDataAdapter : ImportDataAdapter
{
    ....
}
```

#### 9.4.2 Specify an alias

The alias you specify by overriding the Alias property will be shown in the pulldownlist where the user can pick a DataAdapter. This alias must be unique.

```
public override string Alias
{
    get { return "AdventureWorks ProductCategories"; }
}
```

#### 9.4.3 Add UI

The ImportAdapter class itself derives from the Webcontrol Class, so we can directly add UI Controls to this class.

```
private Panel _contentPanel = new Panel();
private TextBox _datasourceTextBox = new TextBox();
private Literal _selectDataSourceLiteral = new Literal();

protected override void OnInit(EventArgs e)
{
    base.OnInit(e);
    _contentPanel.ID = "CategorycontentPanel";
    _contentPanel.CssClass = "propertypane";

    //Labels
    _selectDataSourceLiteral.ID = "SelectDataSourceLiteral";
    _selectDataSourceLiteral.Text = "Enter the connection string";

    //TextBox
    _datasourceTextBox.ID = "datasource";
    _datasourceTextBox.CssClass = "umbEditorTextField";
    _datasourceTextBox.Text = DataSource;

    //Create Layout
    _contentPanel.Controls.Add(_selectDataSourceLiteral);
    _contentPanel.Controls.Add(new LiteralControl("&nbsp;&nbsp;&nbsp;"));
    _contentPanel.Controls.Add(_datasourceTextBox);

    Controls.Add(_contentPanel);
}
```

#### 9.4.4 Update DataAdapter values

When the user clicks next you need to update the DataAdapter properties, in this case DataSource and DataCommand.

```
public override void UpdateAdapter()
{
    //Uses the SqlConnection from the TextBox as the Datasource
    DataSource = _datasourceTextBox.Text;
}
```

```
//We are creating a custom DataAdapter for a productCategories, don't bother
//the user with sql
DataCommand = "Select * from SalesLT.ProductCategory order by
ParentProductCategoryID, Name";
}
```

#### 9.4.5 Validate

When the user clicks next the validate method is called **after the DataAdapter values are updated**. Return true in this method when the DataSource is valid, false when not valid

```
public override bool Validate()
{
    bool result = true;
    try
    {
        using (SqlConnection sqlConnection = new SqlConnection(DataSource))
        {
            sqlConnection.Open();
            SqlCommand sqlCommand = new SqlCommand(DataCommand,
            sqlConnection);
            sqlCommand.ExecuteReader();
        }
    }
    catch (Exception ex)
    {
        //Cannot validate against the Datasource;
        result = false;
        //Set the ValidationErrorMessage with the correct error.
        ValidationErrorMessage = string.Format("Error validating the Data
source: {0}", ex.Message);
    }
    return result;
}
```

#### 9.4.6 GetData

The GetData method gets called by CMSImport during the import process and it will return the data from DataSource as an IDataReader. In the example below you'll see the GetData Method that we are using in the AdventureWorks ProductCategory DataAdapter

```
public override IDataReader GetData()
{
    SqlConnection sqlConnection = new SqlConnection(DataSource);
    sqlConnection.Open();
    SqlCommand sqlCommand = new SqlCommand(DataCommand, sqlConnection);
    return sqlCommand.ExecuteReader();
}
```

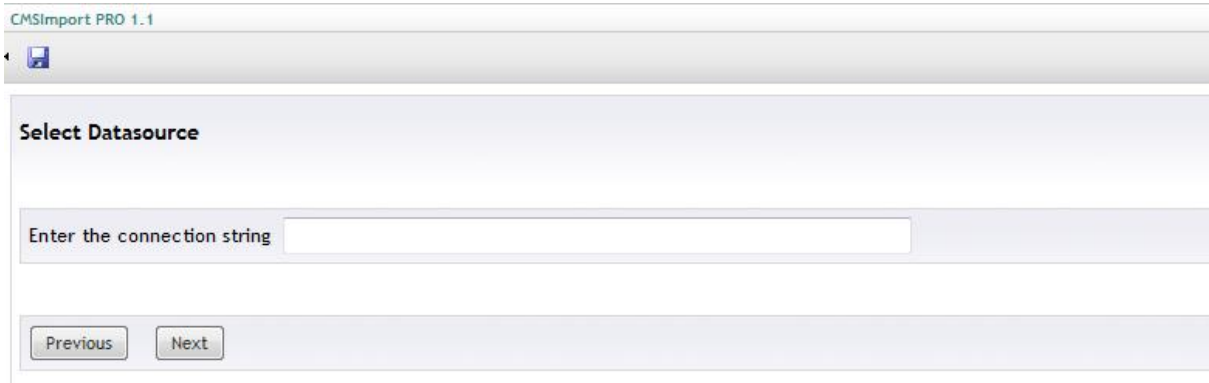
CMSImport will dispose the Reader once it's finished with the import.

If you need to convert XML to IDataReader you can use the helper method XmlToDataReader which takes an xml file/url and xpath Expression. Currently this only works on elements attributes will be ignored.



#### 9.4.7 End result

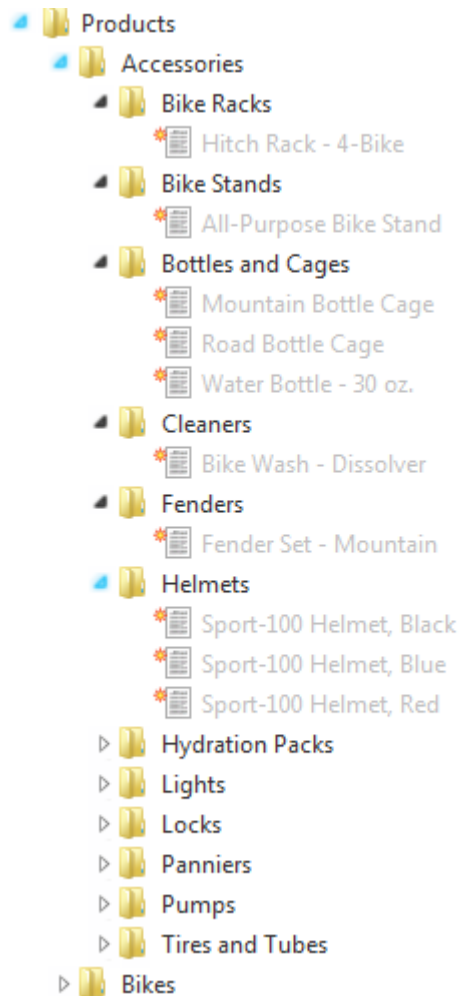
When we will build the dll and drop it into the bin folder of the Umbraco install we will see the “AdventureWorks ProductCategory” in the pulldown list of possible datasources and when we select the AdventureWorks ProductCategory DataAdapter we will see the following screen:



We can provide a Connection string and click next to continue the import process the normal way.

#### 9.5 Create a DataAdapter for Products.

Since we will be importing both Categories and Products we want to have a Custom DataAdapter for products also. Do this by copying the ProductCategories DataAdapter and name it AdventureWorksProducts, rename the alias to “AdventureWorks Products” and replace the query in the UpdateAdapter method to “SELECT \* FROM SalesLT.Product order by Name”



## 9.6 Using a FieldAdapter to map text values against a Dropdownlist DataType

As you might have seen our Product DocumentType contains a property ProductColor which is based on the ProductColor Dropdownlist. When we map the color column from the AdventureWorks Products table directly against the ProductColor document property the import will fail because it can't convert text to integer values.

**Import Finished**

Import is finished

Import duration (h:m:s)	00:00:39
Records retrieved	295
Records imported	50
Errors	245

The following errors occurred

Conversion failed when converting the nvarchar value 'Multi' to data type int.  
 Conversion failed when converting the nvarchar value 'Silver' to data type int.  
 Conversion failed when converting the nvarchar value 'Blue' to data type int.  
 Conversion failed when converting the nvarchar value 'Blue' to data type int.  
 Conversion failed when converting the nvarchar value 'Silver' to data type int.  
 Conversion failed when converting the nvarchar value 'Silver' to data type int.  
 Conversion failed when converting the nvarchar value 'Black' to data type int.  
 Conversion failed when converting the nvarchar value 'Black' to data type int.  
 Conversion failed when converting the nvarchar value 'Black' to data type int.  
 Conversion failed when converting the nvarchar value 'Black' to data type int.  
 Conversion failed when converting the nvarchar value 'Black' to data type int.

This can be solved using a FieldAdapter

For our implementation we want to retrieve the Id of the prevalue and return that id instead of the text value. Additional it would also be nice if we can create the prevalue when it doesn't exists so we don't need to create the prevalues first.

Below you find the FieldAdapter. The DataTypeId returns the Guid of the Dropdown Datatype. The parse method first checks if the datatype is the ProductColor datatype and when it is the ProductColor datatype it will get (or create) the integer id of the prevalue.

```

/// </summary>
public class AutoAddProductColorToList : IFieldAdapter
{
    /// <summary>
    /// Gets the data type id.
    /// </summary>
    /// <value>The data type id.</value>
    public Guid DataTypeId
    {
        get { return new Guid("a74ea9c9-8e18-4d2a-8cf6-73c6206c5da6 "); }
    }

    public object Parse(object value, umbraco.cms.businesslogic.property.Property
property, FieldAdapterOptions fieldAdapterOptions)
    {
        if (property.PropertyType.DataTypeDefinition.Text == "ProductColor")
        {
            DataTypeDefinition dt = property.PropertyType.DataTypeDefinition;
            value = GetOrCreatePrevalue(dt.Id, value.ToString());
        }
        return value;
    }
}

```

Once the import has finished using this FieldAdapter you'll see that all options are added to the ProductColor datatype.

Edit datatype: ProductColor

Name: ProductColor

Render control: Dropdown list

Data Type GUID: a74ea9c9-8e18-4d2a-8cf6-73c6206c5da6

Settings

Database datatype: Integer

Text	Value	
Multi	286	<a href="#">Delete</a>
Silver	287	<a href="#">Delete</a>
Blue	288	<a href="#">Delete</a>
Black	289	<a href="#">Delete</a>
Silver/Black	290	<a href="#">Delete</a>
Red	291	<a href="#">Delete</a>
Yellow	292	<a href="#">Delete</a>
White	293	<a href="#">Delete</a>
Grey	294	<a href="#">Delete</a>

Add prevalue:

And you'll see that these Prevalues are mapped correctly against the document

Product Properties

ProductName: Road-150 Red, 52

Product Number: BK-R93R-52

ProductPrice: 3578.2700

Product color: Red

## 10 Manual Installation/Configuration

If you renamed the Umbraco folder or for some reason can't give the installer sufficient rights to create tables in the database, or the sufficient rights to modify the following folders /bin, /config, /umbraco you need to install CMSImport PRO Manually.

### 10.1 Manual installation of files

- Open the folder in the zip file.
- Copy all (Except for HtmlAgilityPack.dll) .dll files to the /bin folder of your Umbraco installation.
- Copy HtmlAgilityPack.dll to the bin/legacy folder of your Umbraco installation.
- Copy your license file to the /bin folder of your Umbraco installation.
- Open the /umbraco/plugins/ folder.
- Create the folder **CMSImport**.
- Create the following folders in the /umbraco/plugins/CMSImport folder.
  - o Config
  - o Handlers
  - o Pages
  - o Usercontrols
- In the /umbraco/plugins/CMSImport /Usercontrols folder create the folder **ImportSteps**
- In the /umbraco/plugins/CMSImport /Usercontrols/ImportSteps folder create the folder **ContentImport**
- In the /umbraco/plugins/CMSImport /Usercontrols/ImportSteps folder create the folder **MemberImport**
- Copy all .config files from the zip file to the folder /umbraco/plugins/CMSImport/Config/
- Copy all .ashx files from the zip file to the folder /umbraco/plugins/CMSImport/Handlers/
- Copy all .aspx files from the zip file to the folder /umbraco/plugins/CMSImport/Pages/
- Copy the files **CMSImport.ascx** and **CMSImportInstaller.ascx** from the zip file to the folder /umbraco/plugins/CMSImport/Usercontrols/
- Copy the files **ConfirmSelectedOptions.ascx**, **Importing.ascx**, **Intro.ascx**, **MapProperties.ascx**, **SelectDataSource.ascx**, **SelectDataSourceType.ascx** from the zip file to the folder /umbraco/plugins/CMSImport/Usercontrols/ImportSteps/
- Copy **SelectUmbracoTypeAndLocation.ascx** from the zip file to the folder /umbraco/plugins/CMSImport/Usercontrols/ImportSteps/ContentImport
- Copy **SelectMemberType.ascx** from the zip file to the folder /umbraco/plugins/CMSImport/Usercontrols/ImportSteps/MemberImport
- Copy all the png files from the zip file to the folder /umbraco/Images/

### 10.2 Manual configuration of Database

Run the following script to install the database tables

```
CREATE TABLE [CMSImportState] (
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UniqueIdentifier] [uniqueidentifier] NOT NULL,
    [Name] [nvarchar] (250) NOT NULL,
    [ImportType] [nvarchar] (250) NOT NULL,
    [ImportState] [nvarchar] (max) NOT NULL,
    CONSTRAINT [PK_CMSImportState] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [CMSImportScheduledTask] (
```

```

        [ScheduleId] [int] IDENTITY(1,1) NOT NULL,
        [ScheduleGUID] [uniqueidentifier] NOT NULL,
        [ImportStateGUID] [uniqueidentifier] NOT NULL,
        [ScheduledTaskName] [nvarchar](50) NOT NULL,
        [NotifyEmailAddress] [nvarchar](250) NOT NULL,
        [ExecuteEvery] [nvarchar](50) NOT NULL,
        [ExecuteDays] [nvarchar](50) NOT NULL,
        [ExecuteHour] [int] NOT NULL,
        [ExecuteMinute] [int] NOT NULL,
        CONSTRAINT [PK_CMSImportScheduledTask] PRIMARY KEY CLUSTERED
    (
        [ScheduleId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

CREATE TABLE [CMSImportRelation](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UmbracoID] [int] NOT NULL,
    [UmbracoParentId] [int] NOT NULL,
    [DataSourceKey] [nvarchar](250) NOT NULL,
    CONSTRAINT [PK_CMSImportRelation] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

CREATE TABLE [CMSImportMediaRelation](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UmbracoMediaId] [int] NOT NULL,
    [SourceUrl] [nvarchar](500) NOT NULL,
    CONSTRAINT [PK_CMSImportMediaRelation] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

CREATE TABLE [dbo].[CMSImportScheduledItems](
    [ScheduledItemId] [int] IDENTITY(1,1) NOT NULL,
    [ScheduleIdTaskId] [int] NOT NULL,
    [ScheduledOn] [smalldatetime] NOT NULL,
    [ExecutedOn] [smalldatetime] NULL,
    CONSTRAINT [PK_CMSImportScheduledItems] PRIMARY KEY CLUSTERED
    (
        [ScheduledItemId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

CREATE TABLE CMSImportCustomRelation(
    CustomRelationId int IDENTITY(1,1) NOT NULL,
    UmbracoId int NOT NULL,
    Alias nvarchar(250) NOT NULL,
    DataSourceKey nvarchar(250) NOT NULL,
    RelatedId nvarchar(250) NOT NULL,
    CONSTRAINT PK_CMSImportCustomRelation PRIMARY KEY CLUSTERED
    (
        CustomRelationId ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

ALTER TABLE dbo.CMSImportState ADD
    Recursive bit NULL,
    Parent uniqueidentifier NULL

ALTER TABLE dbo.CMSImportScheduledItems ADD
    InProgress bit NULL

ALTER TABLE dbo.CMSImportScheduledTask ADD
    ImportAsUser int NULL

```

```

INSERT [umbracoAppTree] ([treeSilent], [treeInitialize], [treeSortOrder], [appAlias],
[treeAlias], [treeTitle], [treeIconClosed], [treeIconOpen], [treeHandlerAssembly],
[treeHandlerType], [action]) VALUES (0, 0, 16, N'developer', N'CMSImportSettings',
N'Settings', N'folder.gif', N'folder_o.gif', N'CMSImportLibrary',
N'ApplicationTree.ImportSettingsTree', N'')
INSERT [umbracoAppTree] ([treeSilent], [treeInitialize], [treeSortOrder], [appAlias],
[treeAlias], [treeTitle], [treeIconClosed], [treeIconOpen], [treeHandlerAssembly],
[treeHandlerType], [action]) VALUES (0, 1, 10, N'developer', N'CMSImportWizard', N'CMSImport',
N'folder.gif', N'folder_o.gif', N'CMSImportLibrary', N'ApplicationTree.CMSImportTree', N'')
INSERT [umbracoAppTree] ([treeSilent], [treeInitialize], [treeSortOrder], [appAlias],
[treeAlias], [treeTitle], [treeIconClosed], [treeIconOpen], [treeHandlerAssembly],
[treeHandlerType], [action]) VALUES (0, 0, 11, N'developer', N'CMSImportWizardSaveImportTree',
N'Saved Imports', N'folder.gif', N'folder_o.gif', N'CMSImportLibrary',
N'ApplicationTree.SavedImportTree', N'')
INSERT [umbracoAppTree] ([treeSilent], [treeInitialize], [treeSortOrder], [appAlias],
[treeAlias], [treeTitle], [treeIconClosed], [treeIconOpen], [treeHandlerAssembly],
[treeHandlerType], [action]) VALUES (0, 0, 15, N'developer',
N'CMSImportWizardScheduleImportTree', N'Scheduled Imports', N'folder.gif', N'folder_o.gif',
N'CMSImportLibrary', N'ApplicationTree.ScheduledTaskTree', N'')

```

### 10.3 Manual configuration of the language files

When the install failed due insufficient rights of the installer. It's better to assign sufficient rights to the /umbraco/config/lang/ folder (and all the xml files in that folder). Then start CMSImport PRO again. CMSImport Pro will determine that the language file is not updated and will automatically update the language files again.

If for some reason this isn't the case you can modify the files manually. To do this open the necessary language files. For example if you use the English language in Umbraco open en.xml, if you use the Dutch language in Umbraco open nl.xml etc.

Replace

```
<area alias="actions">
```

With

```

<area alias="actions">

<key alias="DeleteCMSImportAction">Delete</key>
<key alias="ExecuteCMSImportWizardAction">Execute</key>
<key alias="ScheduleCMSImportWizardAction">Schedule</key>
<key alias="StartCMSImportWizardAction">Import Data</key>
<key alias="StartCMSImportChildDefinitionAction">Create child definition</key>
<key alias="LastRunInfoAction">Scheduled task log</key>

```

### 10.4 Manual configuration of the scheduled task handler

Open the /config/umbracoSettings.config file.

Replace

```
</scheduledTasks>
```

With

```

<task log="true" alias="CMSImportScheduler" interval="60" url="[url to
mysite/umbraco/] /plugins/CMSImport/Handlers/ScheduleTaskHandler.ashx" />
</scheduledTasks>

```

And replace [url to mysite/umbraco/] with the url of your site including the umbraco path. For example: <http://cmsimport.com/umbraco//plugins/CMSImport/Handlers/ScheduleTaskHandler.ashx>

When the scheduled task handler is configured correctly and you browse to that url using your favorite browser it should report "CMSImport scheduler"



## 11 Troubleshooting

### 11.1 I don't see the CMSImport package in my developer section

Make sure you have sufficient rights to install the package. See chapter 2, otherwise perform a manual installation see chapter9.

### 11.2 I don't see my column names when importing from a CSV file.

Make sure that your csv file contains column names

### 11.3 I get weird column names when importing from a CSV file.

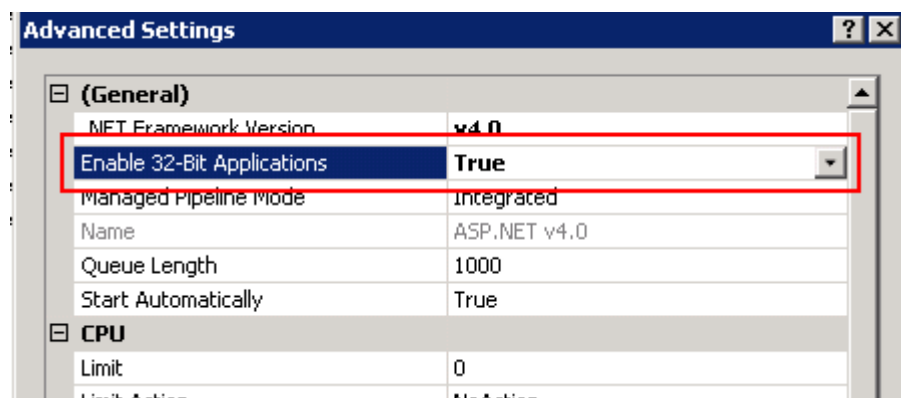
Make sure that you set the correct csv options to display the CSV file. For example, choose ; as the delimiter and " as a string indicator.

### 11.4 I get an error not supported error when using access or excel files

Make sure you've installed the **Microsoft Data Connectivity Components** installed. You can download them via

<http://www.microsoft.com/downloads/en/confirmation.aspx?familyId=7554f536-8c28-4598-9b72-ef94e038c891&displayLang=en%22>

Make sure to enable 32 bits applications on the application pool running the website .



### 11.5 Email is not send when importing a member

Make sure you have configured your smtp server in your web.config file. Also check the UmbracoLog table for SMTP errors.

### 11.6 I get an Invalid License exception.

Make sure your license file exists in the bin folder and you've bought the correct license. Contact [support@soetemansoftware.nl](mailto:support@soetemansoftware.nl) for help.