MemberExport PRO
User manual

1 Introduction

MemberExport Pro helps you export members from your Umbraco installation to an Excel or Csv file. It's also possible to save the export options steps for later use .

MemberExport PRO can export thousands of members in a few seconds.

MemberExport uses the database directly and will only work with the default Umbraco Membership provider!

2 Installation

Install the MemberExport PRO package using the Nuget package manager.

Install-Package MemberExport

During startup of the site the install will be completed. Make sure that the installer has modify rights on the following folders:

- **/bin**
- **/app_plugins**

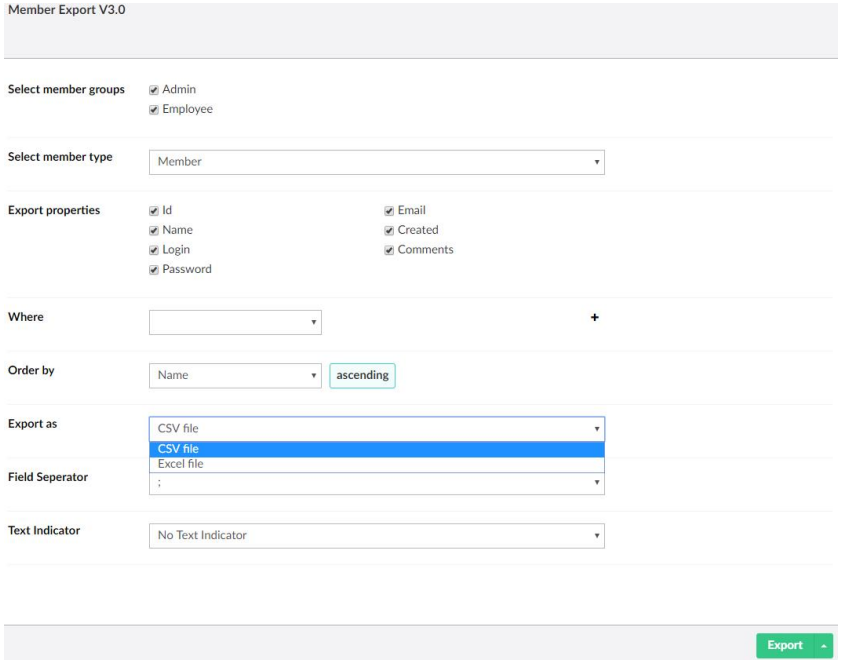
The installer also needs rights to create tables in the database.

Once the package is installed you have an extra folder in your member section called MemberExport. You might need a page refresh to see this folder.

3 Export Members

To export members you browse to the Member section of your Umbraco site. Open the “Export members” folder and click on the “Export members” menu option. The export member screen will appear.

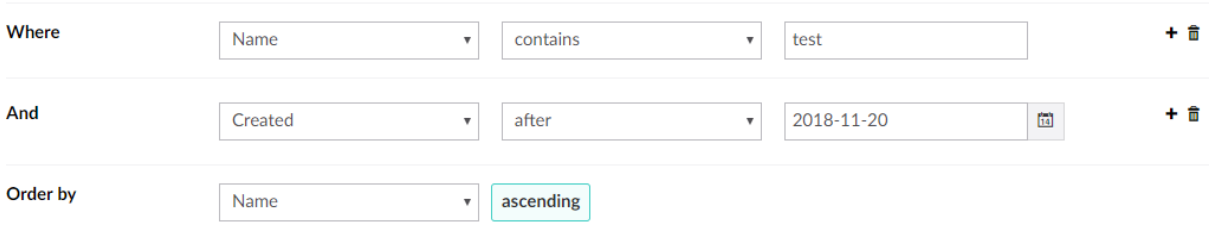
Here you can specify what to export, specify the export format (Excel, or csv) and you can specify the csv export options in case of csv export.



When you click the export button you can download or open the csv file.

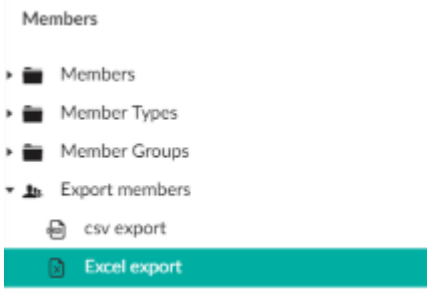
3.1 Filter and sort data

Since Version 3 and higher you can filter and sort data before exporting. In the Where clause select a field you want to filter on and add the criteria. You can query on multiple criteria. Use + to add criteria and the thrash can icon to remove a criteria. Order by field allows you to sort data in the exported file ascending or descending.



3.2 Save exports

You can save the export options for later use, by hitting the save button. The saved exports will be stored in the same “Export Members folder” The save option will only save the export definition. The actual export file will be generated on the fly .



4 Extend MemberExport

MemberExport is an out of the box product, still in some cases you might want to modify the package to your needs. MemberExport can be extended by writing your own value parsers and export providers.

4.1 Value parsers

By default MemberExport exports the stored value from the database. When this value is an Id value you might want to export the text value instead. This can be achieved using a value parser. By default MemberExport comes with value parsers for the following datatypes :

- Checkbox
- Dropdownlist
- MNTP
- MuiltipleDropdownlist
- Radiobox
- Ultimatepicker

In the example below we will write a value parser for the Checkbox datatype. By default values will be exported as 1/0. We want the values to be exported as yes/no for this example.

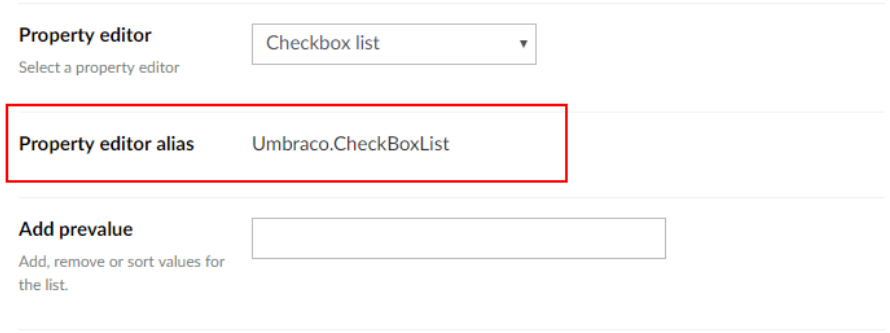
First we need to add a reference to the MemberExport.Library dll. Then we can create a class that implements the IValueParser interface and decorate the class with the ValueParser attribute

```
 [ValueParser(PropertyEditorAlias = "b4471851-82b6-4c75-afa4-39fa9c6a75e9")]
   [ValueParser(PropertyEditorAlias = "Umbraco.CheckBoxList")]
   public  class CheckboxValueParser : IValueParser
   {

       public virtual object Parse(object value)
       {
           return null;
       }
       public virtual object Parse(MemberField memberField, object value, FieldParserOptions
fieldParserOptions)
       {
           if (value != null)
           {
               var sb = new StringBuilder();

               foreach (string item in value.ToString().Split(','))
               {
                   var prevalue = memberField.GetPrevalues().FindLast(p => p.Id.ToString() == item);
                   if (prevalue != null)
                   {
                       if (sb.Length > 0)
                       {
                           sb.Append(',');
                       }
                       sb.Append(prevalue.Value);
                   }
               }
               value = sb.ToString();
           }
           return value;
       }
   }
```

The propertyeditor alias of the valueparser attribute needs to return the GUID which is displayed on the datatype edit screen.



The Parse method will be called when a true/false property value is exported. In the above example we transform the value to a csv list of selected items.

The following information about the exported field is available in the memberFieldInfo variable:

- **PropertyId**. The id of the property
- **PropertyEditorAlias**. The alias of the property ediutor
- **DatatypeNodeId**. The node id of the Datatype.
- **Alias**. The Property alias.
- **Text**. The Property text.

To use the Value Parser all we need to compile the project and add the dll to the bin folder of the Umbraco install. Then it will be picked up automatically and values will be exported as csv.

5 Manual Configuration

5.1 Manual configuration of Database

Run the following script to install the database tables

```
CREATE TABLE [dbo].[MemberExportState](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[UniqueIdentifier] [uniqueidentifier] NOT NULL,
	[Name] [nvarchar](255) NOT NULL,
	[ExportState] [ntext] NOT NULL,
	[Icon] [nvarchar](255) NOT NULL,
 CONSTRAINT [PK_MemberExportState] 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] TEXTIMAGE_ON [PRIMARY]
```

6 Configuration

By default Memberexport runs out of the box with a default configuration. If you need to include/exclude certain properties you can add the following snippet to the appsettings.json file

```
,
"MemberExportConfiguration":
       {
       "ExcludePropertiesConfiguration":
       [
              "umbracoMemberLastLockoutDate",
              "umbracoMemberLastPasswordChangeDate"
       ],
       "LogDebugInfo": false
      }
```

6.1 Exclude properties

When you want to exclude a property from the export options screen. You can add it to the ExcludePropertiesConfiguration array

By default the new technical properties of the Umbraco membership provide such as LockOutDate etc are ignored.

6.1.1 Fixed properties

Besides the user defined properties it's also possible to exclude fixed properties such as Name of the member. Excluding these properties work exactly the same as excluding normal properties, they are prefixed with __member . Below all the fixed properties

Alias	Description
__memberName	The name of the member
__memberCreatedBy	The user created the member (disabled by default)
__memberCreated	The date the member is created
__memberId	The node id of the member
__memberLogin	The login of the member
__memberPassword	The password of the member
__memberEmail	The email address of the member
__memberType	The member type
__memberGroups	A comma separated list of groups assigned to the member

6.2 Debug SQL

When you want to see what SQL Statement gets fired to the database you can set LogDebugInfo to true.

```
"LogDebugInfo": true
```

7 Troubleshooting

7.1 I don't see the MemberExport package in my members section

Make sure you have sufficient rights to install the package. See chapter 2, otherwise perform a manual installation see chapter 4.

7.2 My CSV file is messed up when opening in Excel.

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

7.3 I get an Invalid License exception.

Make sure you've bought the correct license for the (sub)domain , or an enterprise license and added the license file to the bin folder. Contact support@soetemansoftware.nl for help.