

Mediaprotect User manual



1	INTRODUCTION	3
2	INSTALLATION	4
2.1	Package installer	4
2.2	NuGet	4
3	PROTECT MEDIA	5
3.1	Protect media items	5
3.1.1	Single User Protection	6
3.1.2	Role based protection	6
3.1.3	Verify protection	7
4	DASHBOARDS	8
4.1	Log protected files requests	8
4.2	Index Dashboard	8
5	LIBRARY	9
5.1	Umbraco helper extensions	9
5.2	Library methods	9
5.3	Protection API	10
5.4	Events	11
6	CONFIGURATION	14
6.1	IIS Configuration	14
6.2	Media Protect configuration	14
7	AZURE BLOB STORAGE	16
7.1	Set protection level of Blob container to private	16
8	MANUAL INSTALLATION	17
8.1	Manual installation of files	17
8.2	Additional web.config configuration	17
8.3	Dashboards	18
8.4	Add the license file	18
9	TROUBLESHOOTING	19
9.1	I don't see the MediaProtect package in my media section	19
9.2	I get an Invalid License exception.	19
9.3	When I browse to the file it's still visible on the website	19
9.4	How can I create a login box on my login page.	19
9.5	I have another question	19

1 Introduction

Mediaprotect helps you protect media in the same simple way as you protect documents. Once the package is installed and you protect certain media, the files will be password protected. Only authenticated members can open the files, otherwise members will be redirected to the login screen.

Mediaprotect is using the standard Umbraco MemberService or the ASP.NET Membership API, so you can use the existing login views for authentication. And you can plug in your own authentication mechanism.

Mediaprotect requires IIS7⁺ and is compatible with Umbraco version 7.4¹ and up. Mediaprotect is compatible with SQL Server (Express) or SQL CE

¹ Check the Umbraco archive page versions that support older versions of Umbraco

2 Installation

Install Mediaprotect via the Umbraco Package installer or via NuGet.

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

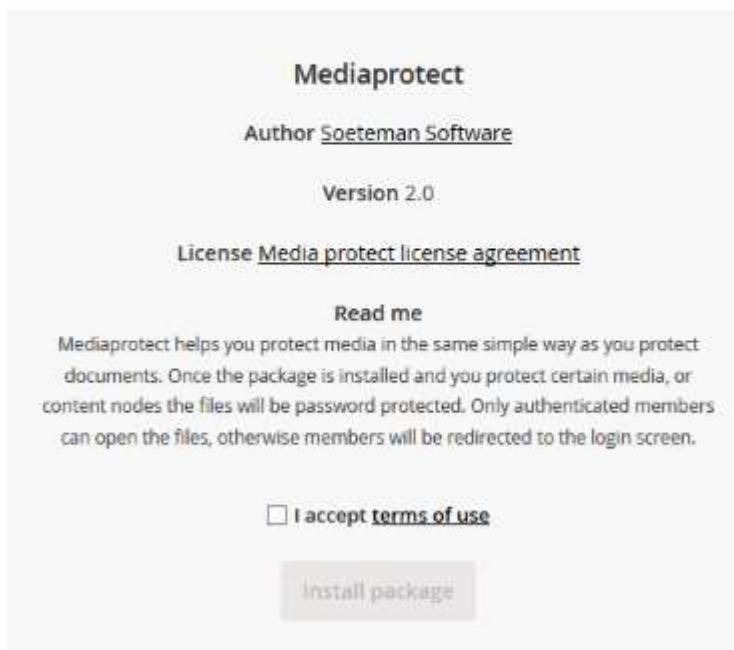
- /bin
- /App_Data
- /App_plugins

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.

2.1 Package installer

Install the Mediaprotect package using the Umbraco package installer.



2.2 NuGet

Use the following command in the Package manager console.

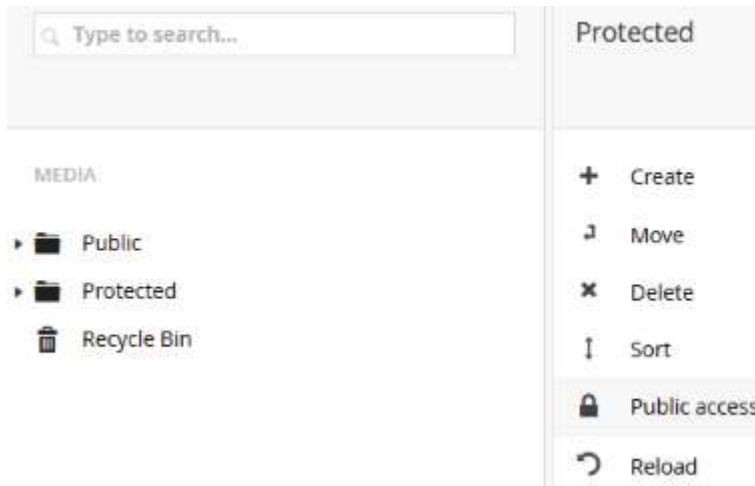
```
PM> Install-Package MediaProtect
```

3 Protect media

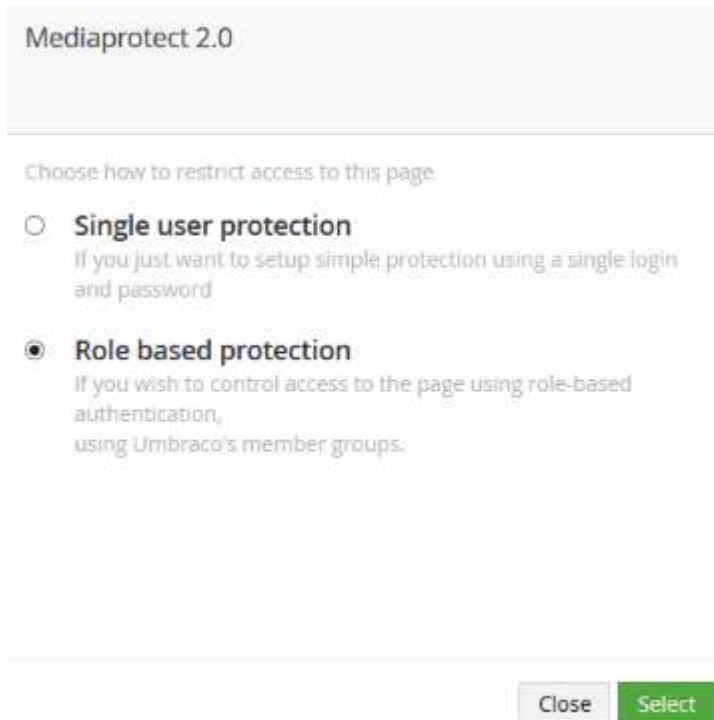
Once the package is installed you can protect your media items.

3.1 Protect media items

You can protect your media items by navigating to the media section, right click on the media folder you want to protect and select Public Access



This will open the same dialog you get when protecting content items.



Single user protection will protected a media node based on an username and password, role based protection allows you to select one or more roles that have access to the media node.

3.1.1 Single User Protection

When you select single user protection you can specify the username and password. The user will be created with the credentials when you select update. You can also specify the login page where the website visitor will be redirected to when not logged in. And you can specify an error page where the member will be redirected to when the member is not authorized to access the requested file.

Mediaprotect 2.0

Set the login and password for this page

Login

Enter your password

Select the pages that contain login form and error messages

Login node **Login** [Delete...](#) [Choose](#)

No rights node **No rights** [Delete...](#) [Choose](#)

[Close](#) [Save](#)

3.1.2 Role based protection

When you select role protection you can specify one or more roles that are allowed to access the media items. You can also specify the login page where the website visitor will be redirected to when not logged in. And you can specify an error page where the member will be redirected to when the member is not authorized to access the requested file.

Mediaprotect 2.0

Pick the roles who have access to this item

<p>Not a member of group(s)</p> <div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div>	<p>>></p> <p><<</p>	<p>Member of group(s)</p> <div style="border: 1px solid #ccc; padding: 5px;"> <p>Admin</p> <p>Employee</p> </div>
--------------------------------------------------------------------------------------------------------	---------------------------------	-------------------------------------------------------------------------------------------------------------------

Select the pages that contain login form and error messages

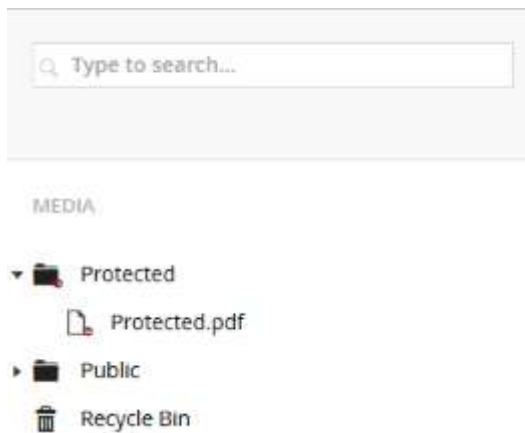
Login node **Login** [Delete...](#) [Choose](#)

No rights node **No rights** [Delete...](#) [Choose](#)

[Close](#) [Save](#)

3.1.3 Verify protection

You can recognize a protected media item in the tree or any supported media picker by the no entry sign.



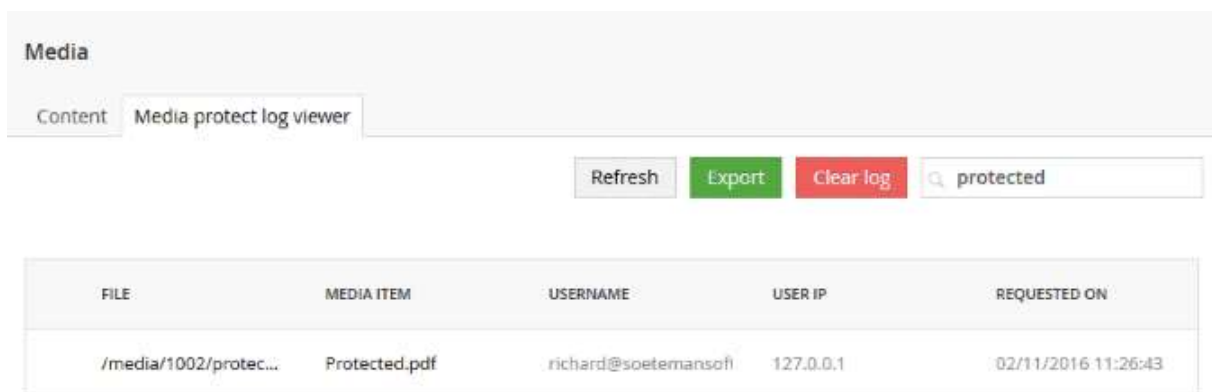
When you open the media item and click on the link you should be redirected to the page you have specified as the login page. If you are already logged in as a website member you should be able to either open the protected media item or being redirected to the not authorized page.

4 Dashboards

4.1 Log protected files requests

During install a dashboard will be installed that enables logging in the media section.

This Dashboard will give you information about which file got requested, the Media item, User info and when the file was requested. You can refresh the overview, export to csv or delete all records.

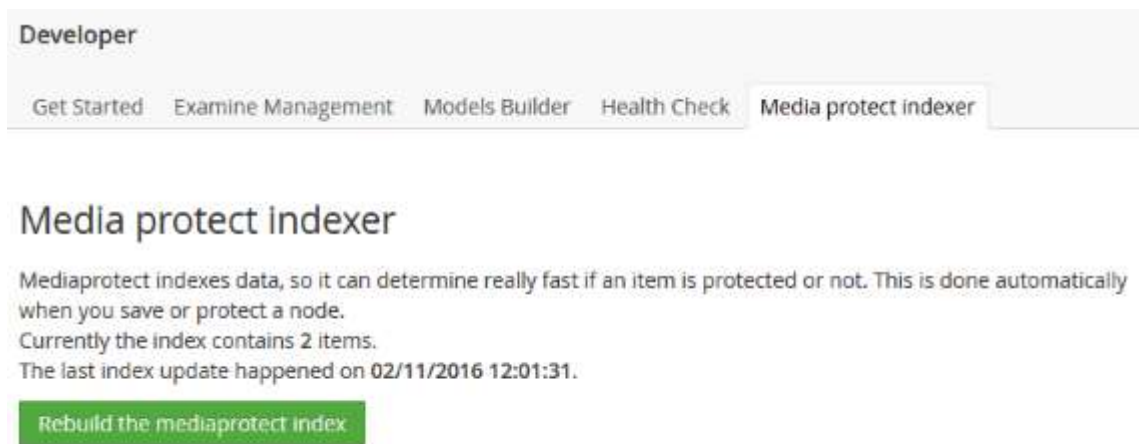


The screenshot shows the 'Media' dashboard with a sub-tab 'Media protect log viewer'. It features a search bar with the text 'protected', and buttons for 'Refresh', 'Export', and 'Clear log'. Below is a table with the following data:

FILE	MEDIA ITEM	USERNAME	USER IP	REQUESTED ON
/media/1002/protec...	Protected.pdf	richard@soetemansoft	127.0.0.1	02/11/2016 11:26:43

4.2 Index Dashboard

Media protect uses an index for fast query of media files. This index gets updated when you set protection or save a media item. In case you ever want to check the last index date, or want to completely rebuild the index you can use the Media Index Dashboard on the developer section.



The screenshot shows the 'Developer' section with a sub-tab 'Media protect indexer'. It contains the following text:

Mediaprotect indexes data, so it can determine really fast if an item is protected or not. This is done automatically when you save or protect a node.
Currently the index contains 2 items.
The last index update happened on 02/11/2016 12:01:31.

Below the text is a green button labeled 'Rebuild the mediaprotect index'.

5 Library

Media protect comes with a small library that you can use in you xslt and/or .net libraries to determine if certain items are protected , or who has access to those items.

5.1 Umbraco helper extensions

The most common used methods of the library are the ones that checks if an item is protected or if the user has access. You can access those methods using the standard Umbraco helper in your view.

5.1.1 MedialsProtected

Use this extension method to check if a media item is protected

```
@Umbraco.MediaIsProtected(1068)
```

5.1.2 MedialsProtected

Use this extension method to check if the current logged in user has access to the media item.

```
@Umbraco.MediaHasAccess(1068)
```

5.2 Library methods

Besides the most used methods that can be retrieved via the Umbraco helper extensions, media protect comes with a library that gives all kind of information about (protected) media items.

5.2.1 IsProtected

Checks if a media item is protected

```
bool MediaProtect.Library.Library.IsProtected(int nodeId, string path)
```

5.2.2 IsProtected

Checks if a filename is protected

```
bool MediaProtect.Library.Library.IsProtected(string fileName)
```

5.2.3 HasAccess

Checks if the current logged in user has access to the media item.

```
bool MediaProtect.Library.Library.HasAccess(int nodeId, string path)
```

5.2.4 HasAccess

Checks if the current logged in user has access to the requested file.

```
bool MediaProtect.Library.Library.HasAccess(string fileName)
```

5.2.5 AllowedGroups

Returns a list of groups that have access to the media item.

```
IEnumerable<string> MediaProtect.Library.Library.AllowedGroups(int nodeId, string path)
```

5.2.6 AllowedGroups

Returns a list of groups that have access to the media file.


```
void MediaProtect.Library.Protection.AddMembershipRoleToMedia(int nodeId,
string roleName)
```

5.3.4 RemoveMembershipRoleFromMedia

This method removes a role from a protected media item

```
void MediaProtect.Library.Protection.RemoveMembershipRoleFromMedia(int
nodeId, string roleName)
```

5.3.5 RemoveMembershipUserFromMedia

This method removes a user from a protected media item

```
void MediaProtect.Library.Protection.RemoveMembershipUserFromMedia(int
nodeId, string userName)
```

5.3.6 RemoveProtection

Removes protection from a media item.

```
void MediaProtect.Library.Protection.RemoveProtection(int nodeId)
```

5.4 Events

MediaProtect comes with two events, Requesting and Requested. The requesting event is triggered before validation occurred and can be used to add custom validation rules. The requested event is triggered after a file is validated. Below you find two examples how these events can be used.

Events are wired up exactly the same as Umbraco events are used. Create a class that derives from ApplicationEventHandler. In the constructor wire up the event. The events are available on the MediaProtect.Library.Request object.

5.4.1 Requesting event

In the example below we protect a custom folder (on disk). Only members in the role Subscriber can download something from that folder. When the member doesn't belong to that role cancel will be set to true and MediaProtect will block the request and redirect to the specified RedirectPage.

```
/// <summary>
/// Example code for the Requesting event
/// </summary>
public class ProtectedFolderRequest : ApplicationEventHandler
{
    /// <summary>
    /// Constructor to wire up the requesting event
    /// Make sure your class derives from ApplicationBase otherwise this constructor never gets hit
    /// </summary>
    protected override void ApplicationStarted(UmbracoApplicationBase umbracoApplication,
        ApplicationContext applicationContext)    {
        Request.FileRequesting += new Request.FileRequestingEventHandler(Request_FileRequesting);
    }

    /// <summary>
    /// Handles the FileRequesting event to check if the user has access
    /// </summary>
    private void Request_FileRequesting(object sender, MediaProtect.Library.EventArgs.FileRequestin
gEventArgs e)
    {
        //Check if the user request a protected folder and the role of the user
        if (e.Path.StartsWith("/protected/", StringComparison.CurrentCultureIgnoreCase) && !Roles.I
sUserInRole("Subscriber"))
        {
            //Cancel the event when a user requests a file from the /protected folder and is not in
the subscriber role
            e.Cancel = true;

            //Specify the login page. MediaProtect will redirect to this page
            e.RedirectPage = "/login/";
        }
    }
}
```

5.4.2 Requested event

In the example below we increase a counter on the protected media item. So you can see how many times a file is requested. **This event is only triggered on items from the ~/media folder.** Once a file is requested the following code will look for a property alias downloadCounter. When it finds the property it will increase its value with 1

```
/// <summary>
/// Sample code to implement a download counter on your media items
/// </summary>
public class DownloadCounter : ApplicationEventHandler
{
    /// <summary>
    /// Constructor to wire up the requesting event
    /// Make sure your class derives from ApplicationBase otherwise this constructor never gets hit
    /// </summary>
    protected override void ApplicationStarted(UmbracoApplicationBase umbracoApplication,
    ApplicationContext applicationContext)    {
        { Request.FileRequested += new Request.FileRequestedEventHandler(Request_FileRequested);
    }

    /// <summary>
    /// Handles the File requested event
    /// </summary>
    private void Request_FileRequested(object sender, MediaProtect.Library.EventArgs.FileRequestedE
ventArgs e)
    {
        //Check if we have a valid media id
        if (e.MediaId != 0)
        {
            //Get the media item
            Media mediaItem = new Media(e.MediaId);

            //Media item exists
            if (mediaItem != null)
            {
                //Get the downloadCounter from the mediaItem
                Property downloadCounterProperty = mediaItem.GetProperty("downloadCounter");
                if (downloadCounterProperty != null)
                {
                    //Property exist, get the current value
                    int current = 0;
                    int.TryParse(downloadCounterProperty.Value.ToString(), out current);

                    //Increase current value with 1
                    current++;

                    //Store the new value
                    downloadCounterProperty.Value = current;

                    //Save the media item
                    mediaItem.Save();
                }
            }
        }
    }
}
```

6 Configuration

6.1 IIS Configuration

This package is only tested on IIS7+. We recommend the use of the integrated pipeline in that case you will be absolutely sure that every request will go through our media protect handler.

When you want to use this package in classic mode make sure you map all requests via the asp.net dll. This following article on LearnIIS.net shows you how to do this.

<http://learn.iis.net/page.aspx/508/wildcard-script-mapping-and-iis-7-integrated-pipeline/>

6.2 Media Protect configuration

You can change a few configuration settings in the Mediaprotect.config file which can be found in the folder /Umbraco/plugins/Mediaprotect/Config

6.2.1 Show images for logged-in Umbraco users

By default protected images are not shown to Umbraco users. When you want to insert a protected media item in the Rich Text Editor, or want to edit a cropped image you'll see a dead image. By setting the showimagesForUmbracoUsers element to true. Images will be shown for logged-in Umbraco Users

```
<showimagesForUmbracoUsers>true</showimagesForUmbracoUsers>
```

Always validate afterwards if the image is really protected by logging out and browsing directly to the image!

6.2.2 Enable/disable logging

To enable or disable logging set the enableLogging to true/false. It's also possible to log public files, set the logPublicMedia to true to enable logging of public files. Override the default export settings by modifying the csvDelimiter and csvStringIndicator attributes

```
<enableLogging logPublicMedia="false" csvDelimiter=";"  
csvStringIndicator="">False</enableLogging>
```

6.2.3 Disable public access dialog

By default the public access dialog on media is enabled, if you only use the API to modify members/roles you might want to disable the Dialog. Set the value of disableMediaProtectDialog element to true when you want to disable the public access dialog on media

```
<disableMediaProtectDialog>>false</disableMediaProtectDialog>
```

6.2.4 Disable use of returnUrl querystring

By default the returnUrl is appended to the url, example
?returnurl=%2fmedia%2f37%2fprotected_file.pdf

If you want to disable this, set the disableReturnUrl config setting to true.

```
<disableReturnUrl>>false</disableReturnUrl>
```

6.2.5 Default login and error node

When you specify default login and error pages these values will be used in the dialog when you protect the node so you don't have to specify them every time.

```
< defaultLoginNode>1048</ defaultLoginNode>  
< defaultErrorNode>1048</ defaultErrorNode>
```

6.2.6 Enable asp.net membership provider

By default Mediaprotect uses the Memberservcie API to query member data. It's possible to switch this to the asp.net membership provider.

```
<authenticationProvider>MemberShipAuthentication</authenticationProvider>
```

In case you want to authenticate to something else. Get in touch with support support@soetemansoftware.nl .

7 Azure Blob Storage

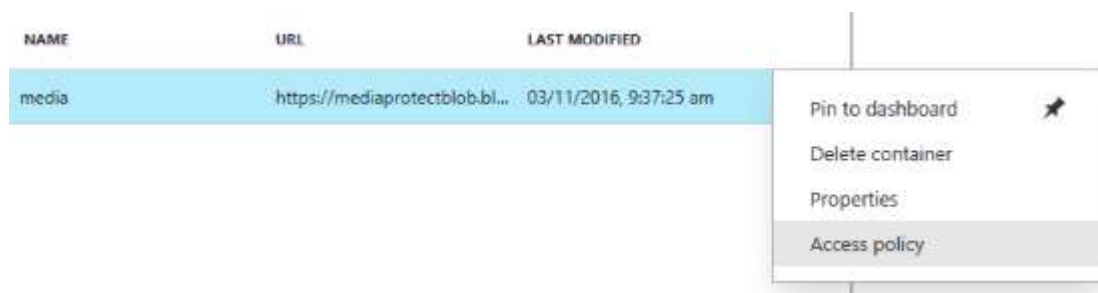
Media protect also works when Umbraco is using BLOB storage in combination with the UmbracoFileSystemProviders.Azure package which can be found at <https://our.umbraco.org/projects/collaboration/umbracofilesystemprovidersazure/>.

Media protect will protect the request from the website `/media/1001/protected.df` and will redirect to the login pages when the person is not logged in or doesn't have enough rights. The Azure Blob container is set to public by default² This means the protected file can still be downloaded via the Azure Url <https://test.blob.core.windows.net/media/1001/protected.pdf> . So besides setting protection on the Umbraco site we need to set the access level to private on the Azure container(s) as well.

7.1 Set protection level of Blob container to private

To make sure the file can't be viewed directly from the Azure Blob container, the Access policy needs to be set to private.

In the Azure portal select the Blob container and select Access policy



Then set Access level to private. Then try to open the url again and you will get an error message instead of the file.

```
<Error>
  <Code>ResourceNotFound</Code>
  <Message>
    The specified resource does not exist. RequestId:206e3516-0001-005f-7cca-35e7f9000000 Time:2016-11-03T12:01:38.0185776Z
  </Message>
</Error>
```

² This will change in the near future so the Azure provider allows you to set protection level to private by default

8 Manual Installation

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, /app_plugins you need to install Mediaprotect Manually.

8.1 Manual installation of files

- Open the folder in the zip file.
- Copy the **MediaProtect.*.dll** files to the /bin folder of your Umbraco installation .
- Open the /app_Plugins / folder.
- Create the folder **mediaProtect**.
- Open the folder mediaProtect and create the following folders:
 - o backoffice
 - o config
 - o css
 - o js
- Open the folder/app_Plugins /mediaprotect/backoffice/ and create the following folders:
 - o dashboard
 - o dialogs
- Copy indexdashboard.html to /app_Plugins /mediaprotect/backoffice/dashboard/
- Copy logviewer.html to /app_Plugins /mediaprotect/backoffice/dashboard/
- Protect Protect.html to /app_Plugins /mediaprotect/backoffice/dialogs/
- Copy mediaprotect.config to the folder **/app_plugins /mediaprotect/config/**
- Copy mediaprotect.css to **/app_plugins /mediaprotect /css/**
- Copy *.js to **/app_plugins /mediaprotect /js/**
- Copy mediaprotect.txt to the folder **/app_plugins /mediaprotect /**
- Copy Package.manifest to the folder **/app_plugins /mediaprotect /**

8.2 Additional web.config configuration

8.2.1 Add the media protect module

To protect media you need to add the following snippet to the system.webServer/modules section in your web.config file.

```
<remove name="MediaProtectModule" />
<add name="MediaProtectModule" type="MediaProtect.HttpModules.MediaProtectModule,
MediaProtect" />
```

8.2.2 Run all managed modules for all requests setting

To make sure the HttpModule gets called during each request set the attribute runAllManagedModulesForAllRequests to true.

```
<system.webServer>
  <validation validateIntegratedModeConfiguration="false" />
  <modules runAllManagedModulesForAllRequests="true">
    <remove name="ScriptModule" />
    <remove name="UrlRewriteModule" />
    <remove name="umbracoRequestModule" />
    <remove name="viewstateMoverModule" />
    <remove name="umbracoBaseRequestModule" />
```

8.3 Dashboards

To enable the dashboards add the following xml to the /config/dashboards.config file.

```
<section alias="MediaProtectIndexer">
  <areas>
    <area>developer</area>
  </areas>
  <tab caption="Media protect indexer">
<control>/app_plugins/mediaprotect/BackOffice/Dashboard/indexdashboard.html</control>
  </tab>
</section>
<section alias="MediaProtectLogViewer">
  <areas>
    <area>media</area>
  </areas>
  <tab caption="Media protect log viewer">
    <control>/app_plugins/mediaprotect/BackOffice/Dashboard/LogViewer.html</control>
  </tab>
</section>
```

8.4 Add the license file

When you bought a license for MediaProtect you can store the .lic file into the Bin folder of your Umbraco install and the trial message should disappear.

9 Troubleshooting

9.1 I don't see the MediaProtect package in my media section

Try logging out and an IIS reset to force rebuilding of the Umbraco tree

9.2 I get an Invalid License exception.

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

9.3 When I browse to the file it's still visible on the website

Make sure Mediaprotect httpmodule is running by browsing to `/app_plugins/mediaprotect/mediaprotect.txt`. This should return true. In any other case the HTTPModule is not configured (See configuration how to install manually). Also make sure the file is protected and make sure the file doesn't exist in the cache of the browser³

9.4 How can I create a login box on my login page.

Umbraco comes with some partial templates for login and loginstatus. Use those.

9.5 I have another question

If you have any question please email us at support@soetemansoftware.nl we love to help you.

³ Even though we force not to cache pages, some browsers ignore these rules.