

Mediaprotect User manual



1	INTRODUCTION	4
2	INSTALLATION	5
2.1	Post install	5
3	PROTECT MEDIA	7
3.1	Protect media items	7
3.1.1	Single User Protection	8
3.1.2	Role based protection	8
3.1.3	Verify protection	9
4	LOG PROTECTED FILES REQUESTS	11
5	LIBRARY	12
5.1	XSLT Library	12
5.1.1	MediaProtect.Library:AllowedGroups	12
5.1.2	MediaProtect.Library:AllowedGroups	12
5.1.3	MediaProtect.Library:AllowedMembers	12
5.1.4	MediaProtect.Library: AllowedMembers	12
5.1.5	MediaProtect.Library: GetProtectedNodesForRole	13
5.1.6	MediaProtect.Library: GetProtectedNodesForUser	13
5.1.7	MediaProtect.Library:HasAccess	13
5.1.8	MediaProtect.Library:HasAccess	13
5.1.9	MediaProtect.Library:IsProtected	13
5.1.10	MediaProtect.Library:IsProtected	14
5.1.11	MediaProtect.Library: IsProtectedByUserName	14
5.2	XSLT Sample	15
5.3	.Net Library	15
5.3.1	Protectmedia	15
5.3.2	AddMembershipUser	15
5.3.3	AddMembershipRole	15
5.3.4	RemoveMembershipRoleFromMedia	15
5.3.5	RemoveMembershipUserFromMedia	16
5.3.6	RemoveProtection	16
5.3.7	IsProtectedByUserName	16
5.4	Events	16
5.4.1	Requesting event	17
5.4.2	Requested event	18
6	CONFIGURATION	19
6.1	IIS Configuration	19
6.2	Media Protect configuration	19
6.2.1	Show images for logged-in Umbraco users	19
6.3	Add support for additional media pickers	19
6.4	Enable/disable logging	20
6.5	Disable public access dialog	20
6.6	Disable use of returnUrl querystring	20
6.7	Default login and error node	20

7	MANUAL INSTALLATION	21
7.1	Manual installation of files	21
7.2	Additional web.config configuration	21
7.2.1	Add the media protect module.....	21
7.2.2	Run all managed modules for all requests setting	21
7.3	Add the XSLT Extension	22
7.4	Add the license file	22
8	TROUBLESHOOTING	23
8.1	I don't see the MediaProtect package in my media section	23
8.2	I get an Invalid License exception.	23
8.3	When I browse to the file it's still visible on the website	23
8.4	How can I create a login box on my login page.....	23
8.5	I have another question	23

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, or content nodes 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 ASP.NET Membership API, so you can use the existing asp.net login controls for authentication. And you can plug in your own Membership provider.

Mediaprotect is compatible with the following third party upload packages:

- Multi file upload
- Desktop Media Uploader
- Zip Upload
- Import Media
- CMSImport Pro (related media import option)

Besides the default Media picker, Mediaprotect is compatible with the following third party pickers:

- Multi node tree picker (part of UComponents)
- Digibiz Advanced Media Picker

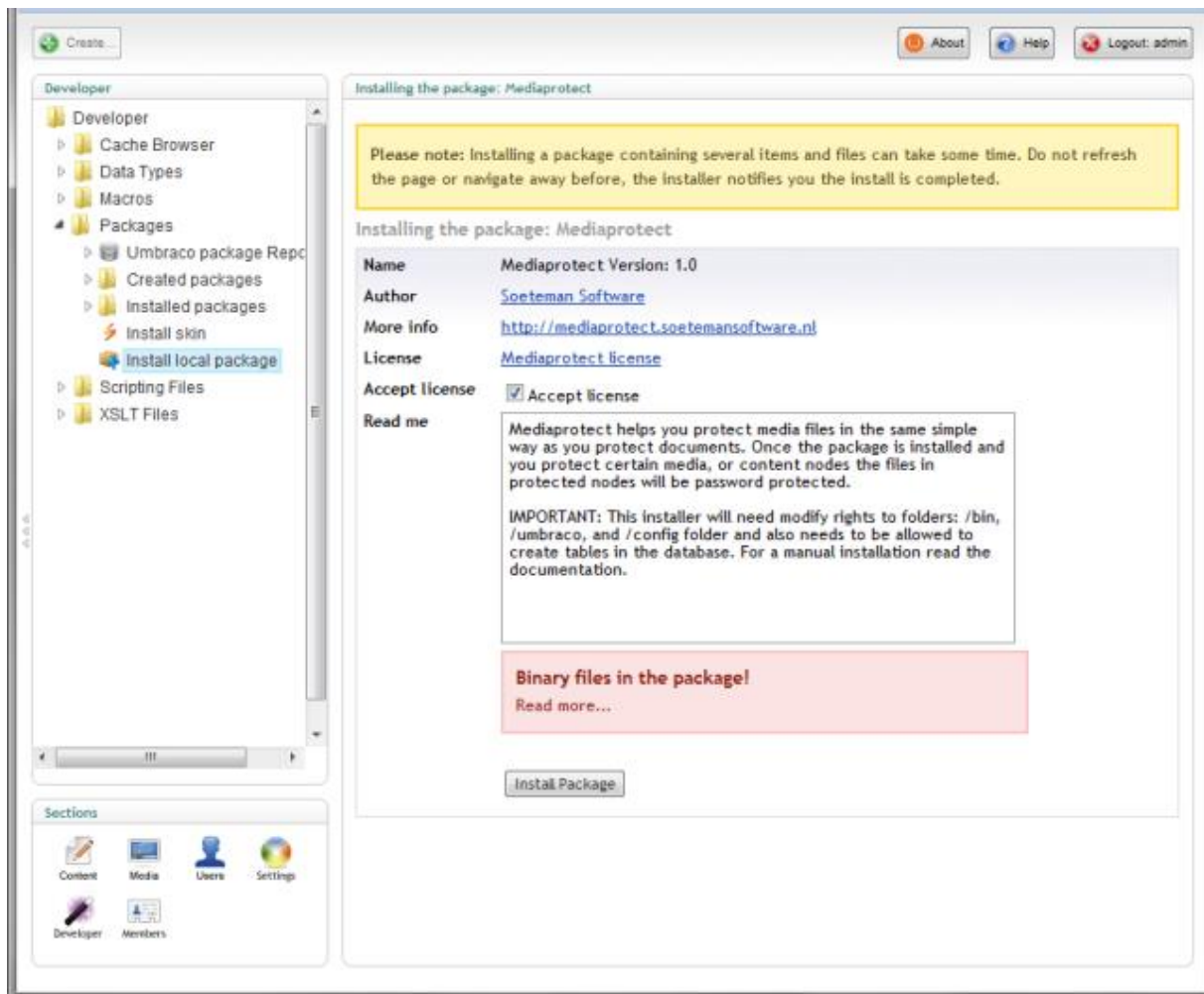
Mediaprotect requires IIS6 or IIS7¹ and is compatible with Umbraco version 4.5 and up.

Mediaprotect is compatible with SQL Server 2005/2008 (Express) or SQL CE 4 (Umbraco v4.6+) only

¹ For IIS6 or IIS7 classic mode you need to map Mime types also check the configuration section for more info.

2 Installation

Install the Mediaprotect package using the Umbraco package installer.



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

- /bin
- App_Data
- /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.

2.1 Post install

Once the package is installed it will validate the settings. It should say that your settings are perfect in any other situation see the chapter Troubleshooting

Install package

Thank you for installing Mediaprotect

With media protect you can protect media as easy as you protect documents. Just right click on the media node and select the protect option.

You might need to logout and login again after install to see this option.

If you have any questions [download the manual](#) or [contact our support team](#).

Validate configuration settings

Your settings are perfect!

You can always revalidate the page using the </umbraco/plugins/mediaprotect/ValidateInstall.aspx> page.

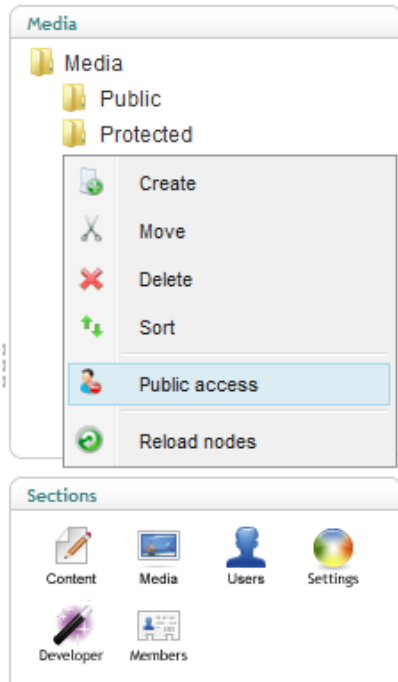
Since context menu's might not be picked up automatically please logout and login again after install.

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.

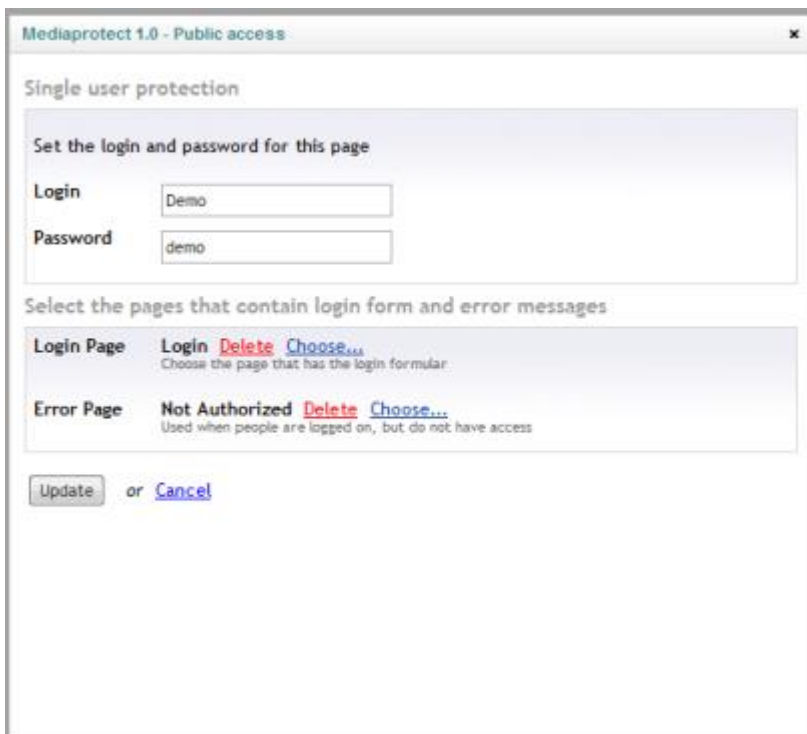


Single user protection will protected a media node based on an username and password, role based protection allows you to select one or more roles that have access to the media node.

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.

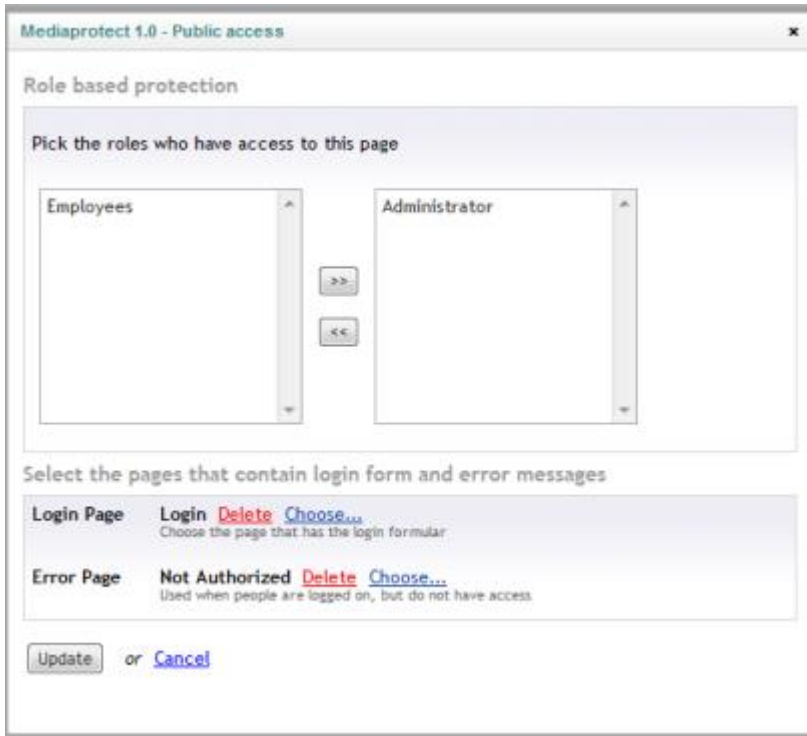


The screenshot shows a web interface titled "Mediaprotect 1.0 - Public access" with a sub-header "Single user protection". The main heading is "Set the login and password for this page". There are two input fields: "Login" with the value "Demo" and "Password" with the value "demo". Below this is a section titled "Select the pages that contain login form and error messages". It contains two rows: "Login Page" with the value "Login" and "Error Page" with the value "Not Authorized". Each row has "Delete" and "Choose..." links. At the bottom, there are "Update" and "Cancel" buttons.

3.1.2 Role based protection

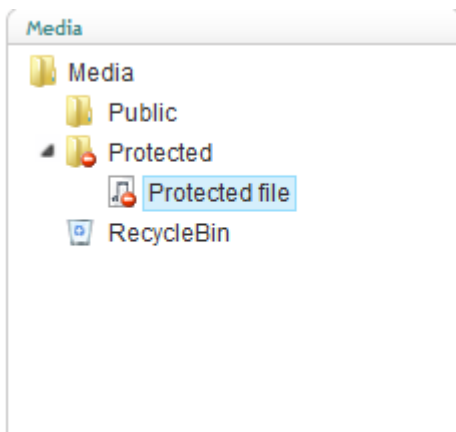
When you select role protection you can specify one or more roles that are allowed to access the media items.

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.



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.

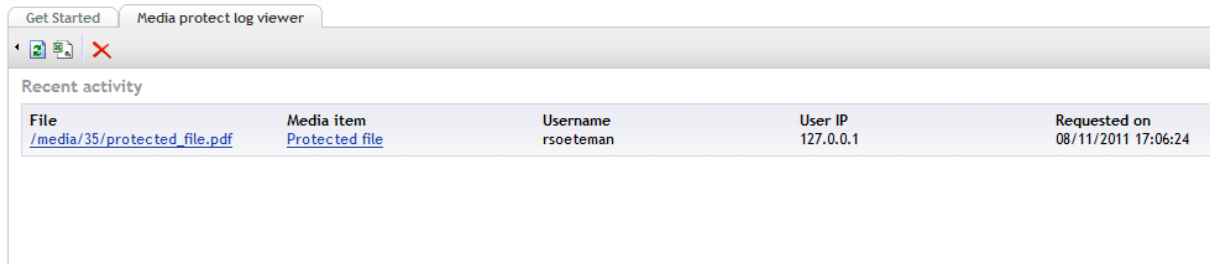


When you uploaded a file on a protected content node the file is protected as well.

4 Log protected files requests

During install you can specify if protected files need to be logged when they are requested. When you enable logging a new dashboard will be installed 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 a web interface titled "Media protect log viewer". It features a "Recent activity" section with a table containing one record. The table has five columns: File, Media item, Username, User IP, and Requested on. The data in the table is as follows:

File	Media item	Username	User IP	Requested on
/media/35/protected_file.pdf	Protected file	rsoeteman	127.0.0.1	08/11/2011 17:06:24

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 XSLT Library

5.1.1 MediaProtect.Library:AllowedGroups

This extension method returns all roles that have access to the certain Media Item, based on media id and media path.

```
<xsl:for-each select="MediaProtect.Library:AllowedGroups('Int32 nodeId',  
'String path')">  
<!--Loop through all roles -->  
</xsl:for-each>
```

5.1.2 MediaProtect.Library:AllowedGroups

This extension method returns all roles that have access to the certain Media Item, based on the filename.

```
<xsl:for-each select="MediaProtect.Library:AllowedGroups(' filename ')">  
<!--Loop through all roles -->  
</xsl:for-each>
```

5.1.3 MediaProtect.Library:AllowedMembers

This extension method returns all members that have access to the certain Media Item, based on media id and media path.

Only for single user protection

```
<xsl:for-each select="MediaProtect.Library:AllowedMembers('Int32 nodeId',  
'String path')">  
<!--Loop through all roles -->  
</xsl:for-each>
```

5.1.4 MediaProtect.Library: AllowedMembers

This extension method returns all membersthat have access to the certain Media Item, based on the filename.

Only for single user protection

```
<xsl:for-each select="MediaProtect.Library: AllowedMembers (' filename ')">  
<!--Loop through all roles -->  
</xsl:for-each>
```

5.1.5 MediaProtect.Library: GetProtectedNodesForRole

This extension method returns all Root media nodes that are protected by the specified role.

```
<xsl:for-each select="MediaProtect.Library:GetProtectedNodesForRole('String
roleName')">
<!--Loop through all protected ROOT nodes -->
</xsl:for-each>
```

5.1.6 MediaProtect.Library: GetProtectedNodesForUser

This extension method returns all Root media nodes that are protected by the specified username.

Only for single user protection

```
<xsl:for-each select="MediaProtect.Library:GetProtectedNodesForUser('String
userName')">
<!--Loop through all protected ROOT nodes -->
</xsl:for-each>
```

5.1.7 MediaProtect.Library:HasAccess

This extension method checks if the current logged in user has access to a media item, based on media id and media path.

```
<xsl:if test="MediaProtect.Library:HasAccess('Int32 nodeId', 'String
path')">
<!-- The user has access-->
</xsl:if>
```

5.1.8 MediaProtect.Library:HasAccess

This extension method checks if the current logged in user has access to a media item, based on the filename.

```
<xsl:if test="MediaProtect.Library:HasAccess('String fileName')">
<!-- The user has access-->
</xsl:if>
```

5.1.9 MediaProtect.Library:IsProtected

This extension method checks if the media item, based on media id and media path is protected for the current logged in user.

```
<xsl:if test="MediaProtect.Library:IsProtected('Int32 nodeId', 'String
path')">
<!-- The media item is protected-->
</xsl:if>
```

5.1.10 MediaProtect.Library:IsProtected

This extension method checks if the media item, based filename is protected for the current logged in user.

```
<xsl:if test="MediaProtect.Library:IsProtected('String fileName')">
<!-- The media item is protected-->
</xsl:if>
```

5.1.11 MediaProtect.Library: IsProtectedByUserName

This extension method returns the username that is allowed to access the media item

Only for single user protection

```
<xsl:value-of select="MediaProtect.Library:IsProtectedByUserName('Int32
nodeId')"/>
```

5.2 XSLT Sample

The following sample only shows the download link if the member is allowed to access the file

```
<xsl:variable name="mediaId" select="$currentPage/mediaPicker"/>
<!--Test if a media item is assigned-->
<xsl:if test="$mediaId !=''">
<xsl:variable name="mediaItem"
    select="umbraco.library:GetMedia($mediaId, '0')"/>

<!--Test if the loggedin member can access the media item-->
<xsl:if test="MediaProtect.Library:HasAccess($mediaItem/@id,
$mediaItem/@path) =1">

<p>Download
  <a href="{ $mediaItem/umbracoFile }"><xsl:value-of
select="$mediaItem/@nodeName"/></a>
</p>
  </xsl:if>
</xsl:if>
```

5.3 .Net Library

When you want to use the Library in .Net all you need to do is add a reference to the **MediaProtect.Library.dll**. This will give you access to the above mentioned Library methods and give you the ability to modify protection settings.

With version 4.7 of Umbraco Razor is supported also. When you want to use Razor, use the .Net Library methods.

5.3.1 Protectmedia

This methods allows you to protect a media node

```
MediaProtect.Library.Protection.ProtectMedia (
    Simpleprotection,
    nodeId,
    loginNode,
    noRightsNode);
```

5.3.2 AddMembershipUser

This method allows a membership user to access the node. Only works with Simple protection

```
MediaProtect.Library.Protection.AddMembershipUserToMedia (nodeId, userName);
```

5.3.3 AddMembershipRole

This method allows a membership role to access the node

```
MediaProtect.Library.Protection.AddMembershipRoleToMedia (nodeId, role);
```

5.3.4 RemoveMembershipRoleFromMedia

This method removes authorization for a certain role and node

```
MediaProtect.Library.Protection.RemoveMembershipRoleFromMedia (nodeId,  
role);
```

5.3.5 RemoveMembershipUserFromMedia

This method removes authorization for a certain member and node

```
MediaProtect.Library.Protection.RemoveMembershipUserFromMedia (nodeId,  
role);
```

5.3.6 RemoveProtection

This method removes protection for a node.

```
MediaProtect.Library.Protection.RemoveProtection (nodeId);
```

5.3.7 IsProtectedByUserName

Determines whether a media node is protected by userName (Simple protection).

```
MediaProtect.Library.Protection.IsProtectedByUserName (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 ApplicationBase. In the constructor wire up the event. The events are available on the MediaProtect.Library.Request object.

The event samples below can be downloaded from the MediaProtect site

<http://mediaprotect.soetemansoftware.nl/documentation.aspx>

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 : ApplicationBase
{
    /// <summary>
    /// Constructor to wire up the requesting event
    /// Make sure your class derives from ApplicationBase otherwise this constructor never gets hit
    /// </summary>
    public ProtectedFolderRequest()
    {
        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 :ApplicationBase
{
    /// <summary>
    /// Constructor to wire up the requesting event
    /// Make sure your class derives from ApplicationBase otherwise this constructor never gets hit
    /// </summary>
    public DownloadCounter()
    {
        Request.FileRequested += new Request.FileRequestedEventHandler(Request_FileRequested);
    }

    /// <summary>
    /// Handles the File requested event
    /// </summary>
    private void Request_FileRequested(object sender, MediaPlayer.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 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!

6.3 Add support for additional media pickers

By default the following media pickers are supported and will show the no-entry sign on protected media nodes when picking a media item:

- Media Picker
- Multi node tree picker (part of UComponents)
- Digibiz Advanced Media Picker

When you want to add an additional media picker you can add the alias of the picker to the media picker collection.²

```
<mediaPickers>  
  <mediapicker alias="myCustomMediaPickerAlias"/>  
  <mediapicker alias="anotherCustomMediaPickerAlias"/>  
</mediaPickers>
```

² If you use a third party media picker which should be added by default send us an email support@soetemansoftware.nl and we will add it for the next release.

6.4 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>
```

6.5 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>
```

6.6 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>
```

6.7 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>
```

7 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, /umbraco you need to install Mediaprotect Manually.

7.1 Manual installation of files

- Open the folder in the zip file.
- Copy all **.dll** files to the /bin folder of your Umbraco installation.
- Open the /umbraco/plugins/ folder.
- Create the folder **MediaProtect**.
- Open the folder MediaProtect and create the following folders:
 - o Config
 - o Dialogs
 - o Scripts
 - o Usercontrols
- Copy all .config from the zip file to the folder /umbraco/plugins/MediaProtect/Config/
- Copy ProtectMedia.aspx to /umbraco/plugins/MediaProtect/Dialogs/
- Copy MediaProtect.js to /umbraco/plugins/MediaProtect/Scripts/
- Copy all .ascx files to the folder /umbraco/plugins/MediaProtect/Usercontrols/
- Copy the following files to the folder /umbraco/plugins/MediaProtect/
 - o mediaprotect.txt
 - o LicenseError.aspx
 - o ConfigurationError.aspx
 - o ValidateInstall.aspx

7.2 Additional web.config configuration

7.2.1 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" />
```

7.2.2 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" />
```

7.3 Add the XSLT Extension

If you want to use the Library methods in Xslt You need to configure the Xslt extension also. Add the following line to the /config/xsltExtensions.config file.

```
<ext assembly="MediaProtect.Library" type="MediaProtect.Library.Library"  
alias="MediaProtect.Library">
```

7.4 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.

8 Troubleshooting

8.1 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

8.2 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.

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

Make sure the file is protected in either the media or content section and make sure the file doesn't exist in the cache of the browser³

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

Mediaprotect is using the standard ASP.NET Membership API so you can use the default asp.net login controls which can be found at <http://msdn.microsoft.com/en-us/library/ms178329.aspx>

8.5 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.