nörd

MultipleFileUpload v1.0 Package for Umbraco Documentation v1.0

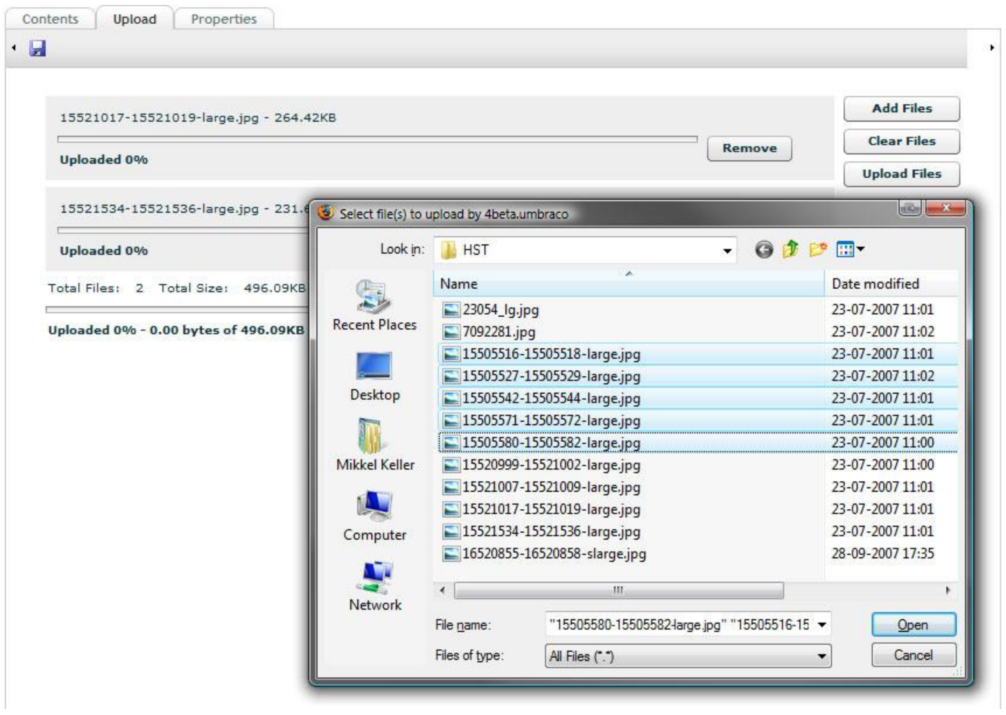
2008-07-30

Table of content

1 Overview

The package enables you to upload multiple files in one go.

A new tab called "Upload" will be added to the native Folder media type. The tab contains a Flash based control which enables the user to select multiple files from multiple sources before uploading. The files will be placed as child nodes under the selected folder.


Contents
Upload
Properties
15521017-15521019-large.jpg - 264.42KB
Remove
Uploaded 0%
Add Files
Clear Files
Upload Files
15521534-15521536-large.jpg - 231.
Uploaded 0%
Total Files: 2 Total Size: 496.09KB
Uploaded 0% - 0.00 bytes of 496.09KB
Select file(s) to upload by 4beta.umbraco
Look in: HST
Recent Places
Desktop
Mikkel Keller
Computer
Network
File name: "15505580-15505582-large.jpg" "15505516-15
Files of type: All Files (*.*)
Open
Cancel

Name	Date modified
23054_lg.jpg	23-07-2007 11:01
7092281.jpg	23-07-2007 11:02
15505516-15505518-large.jpg	23-07-2007 11:01
15505527-15505529-large.jpg	23-07-2007 11:02
15505542-15505544-large.jpg	23-07-2007 11:01
15505571-15505572-large.jpg	23-07-2007 11:01
15505580-15505582-large.jpg	23-07-2007 11:00
15520999-15521002-large.jpg	23-07-2007 11:00
15521007-15521009-large.jpg	23-07-2007 11:01
15521017-15521019-large.jpg	23-07-2007 11:01
15521534-15521536-large.jpg	23-07-2007 11:01
16520855-16520858-slarge.jpg	28-09-2007 17:35

Each file will be handled according to its extension. By default the MultipleFileUpload package has two different behaviours. All image files (`jpeg`, `jpg`, `gif`, `bmp`, `png`, `tiff`, `tif`) will be created as image media types, whereas all other files will be created as file media types.

It's possible to customize how individual files will be treated and the user control that is displayed inside Umbraco. Please refer to "Extending the package" section.

1 Installing the package

Umbraco 3.x

Right-click on the Macros item in the Developer section of umbraco. Select Import Package. Navigate to the noerd.Umb.DataTypes.multipleFileUpload.umb file and press *Load package*. Accept the license and click *Install Package*. Accept the license and click Load Package.

Umbraco 4 local file

Goto the *Developer section* and expand the *Packages* node click the *Install local package* node. Navigate to the noerd.Umb.DataTypes.multipleFileUpload.umb file and press *Load package*. Accept the license and click *Install Package*.

Umbraco 4 repository install

Goto the *Developer section* and expand the *Packages* node expand the *Umbraco package Repository* node. Locate the Multiple File Upload package and click on the link. Click *Download and install package* and click *ok* to verify the download. Accept the license and click *Install Package*.

Final steps: Custom installer

The installer performs the following steps

1) Copies the following files to the umbraco instance

   ~/bin/noerd.Umb.DataTypes.multipleFileUpload.dll
   ~/scripts/swfobject.js
   ~/config/MultipleFileUpload.config
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.swf
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.ascx
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.ascx.cs
   ~/usercontrols/MultipleFileUpload/Installer.ascx
   ~/usercontrols/MultipleFileUpload/Installer.ascx.cs
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.pdf

2) Creates a Multiple File Upload datatype in the developer section.
3) Add an "Upload" tab to the Folder media type and adds a property with the Multiple File Upload datatype.
4) Registers the MultipleFileUploadHandler (IHttpHandler) in the web.config file.

It's possible to skip step 2-4 if you prefer to perform these steps manually.

Final steps: Manual install

To perform a manual install you'll have to perform the following steps

Create a Multiple File Upload datatype in the developer section.

1) Go to the *developer section*, rightclick the *Data Types* node and select *Create*
2) Give the Data Type a name like "Multiple File Upload" and press *Create* button
3) Click on the new Data type. In the *Rendercontrol* dropdown select *Multiple File Upload* and press the *save* button.

   Advanced: The Prevalue field, can be used to specify the path to a custom usercontrol as frontend for the Multiple File Upload package.

Add an "Upload" tab to the Folder media type and add a property with the Multiple File Upload datatype.

1) Go to the *settings* section and expand the *Media Types* folder.
2) Select the *Folder* node and press the *Tabs* tab.
3) In the *New tab* field write a name like "Upload" and pres the *New tab* button.
4) Select the *Generic properties* tab and create a new property. Give it whatever name you like and write MultipleFileUpload in the *Alias* field, select the data type created in step 1 in the *Type* field and select the tab you just created in the *Tab* field.
5) Press the *save* button.

Register the MultipleFileUploadHandler (IHttpHandler) in the web.config file.

1) Add the following line to the system.web/httpHandlers section in the web.config file.

```
<add path="MultipleFileUploadHandler.axd" verb="POST"
type="noerd.Umb.DataTypes.multipleFileUpload.MultipleFileUploa
dHandler, noerd.Umb.DataTypes.multipleFileUpload"
validate="false" />
```

2 Uninstalling the package

To uninstall the package you'll need to perform some manual steps:

Warning: *It's very important that you follow the instructions below in the in the correct order, otherwise you could end up invalidating all folders in the media section.*

1) Goto the settings section and expand the *Media Type* node.

2) Select the *Folder* node and click the *Generic properties* tab and delete the *Upload multiple files* property.

3) Goto the *Tabs* tab and delete the *Upload* tab.

4) Go to the developer section and delete the *Multiple File Upload* data type.

5) Open the web.config file, goto the system.web/httpHandlers section and delete this line:

```
<add path="MultipleFileUploadHandler.axd" verb="POST"
type="noerd.Umb.DataTypes.multipleFileUpload.MultipleFileUploa
dHandler, noerd.Umb.DataTypes.multipleFileUpload"
validate="false" />
```

6) **Umbraco v4:** Goto the developer section and expand the *Packages* node and the *Installed packages* node. Click on the the MultipleFileUpload node and press the *Uninstall package* button.

   Umbraco v3.x: Remove the following files from the Umbraco instance.

   ~/bin/noerd.Umb.DataTypes.multipleFileUpload.dll
   ~/scripts/swfobject.js
   ~/config/MultipleFileUpload.config
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.swf
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.ascx
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.ascx.cs
   ~/usercontrols/MultipleFileUpload/Installer.ascx
   ~/usercontrols/MultipleFileUpload/Installer.ascx.cs
   ~/usercontrols/MultipleFileUpload/MultipleFileUpload.pdf

3 Extending the package

The MultipleFileUpload package is highly extendable, while still adhering to the open/closed principle. This means that you can extend the package without changing the MultipleFileUpload codebase.

Custom Media handlers

The heart of Multiple File Upload package is the class MultipleFileUploadHandler, which is an implementation of the IHttpHandler interface. It's responsible of handling the post requests from the flex element in the Umbraco gui.

The MultipleFileUploadHandler delegates the creation of media nodes to concrete instances of the abstract class MediaFactory based on the files extension. In the config folder you'll find the configuration file MultipleFileUpload.config. This configuration file maps extensions to types, namespaces and assemblies.

This is the default MultipleFileUpload.config file.

```
<multipleFileUpload xmlns='urn:MultipleFileUpload-schema'>
   <mediaFacory
         assembly="noerd.Umb.DataTypes.multipleFileUpload"
         namespace="noerd.Umb.DataTypes.multipleFileUpload"
         type="DefaultFileMediaFactory">
      <extensions>
         <!-- <ext>*</ext> below is a wildcard and specifies
         that this MediaFactory will be used if no other
         MediaFactory has a matching extension.

         If multible MediaFactories is configured with a
         <ext>*</ext> the first MediaFactory in document order
         will be use-->
         <ext>*</ext>
      </extensions>
   </mediaFacory>
   <mediaFacory
         assembly="noerd.Umb.DataTypes.multipleFileUpload"
         namespace="noerd.Umb.DataTypes.multipleFileUpload"
         type="DefaultImageMediaFactory">
      <extensions>
         <ext>jpeg</ext>
         <ext>jpg</ext>
         <ext>gif</ext>
         <ext>bmp</ext>
         <ext>png</ext>
         <ext>tiff</ext>
         <ext>tif</ext>
      </extensions>
   </mediaFacory>
</multipleFileUpload>
```

As can be seen, the package has two default implementations of the MediaFactory class. The later handles images while the first is configured with the wildcard and thereby handles everything else. You can easily add your own implementations.

To create a concrete instance of MediaFactory all you have to do is inherit the abstract class MediaFactory and implement one method and add the extension, assembly, namespace and type to the config document.

As an inspiration the following is the source for the build in concrete MediaFactory `DefaultFileMediaFactory`

```
using System.Web;
using umbraco.BasePages;
using umbraco.BusinessLogic.console;
using umbraco.cms.businesslogic.media;

namespace noerd.Umb.DataTypes.multipleFileUpload
{
    /// <summary>
    /// /// The default Media factory for files.
    /// </summary>
    public class DefaultFileMediaFactory : MediaFactory
    {
        // IMediaFactory Members
        // -------------------------------------------------------------------

        #region IMediaFactory Members

        public override Media CreateMedia(IconI parent, HttpPostedFile uploadFile)
        {
            string filename = uploadFile.FileName;

            // Create new media object
            Media media = Media.MakeNew(filename, MediaType.GetByAlias("File"),
                                    UmbracoEnsuredPage.CurrentUser, parent.Id);

            // Get umbracoFile property
            int propertyId = media.getProperty("umbracoFile").Id;

            // Set media properties
            media.getProperty("umbracoFile").Value =
               VirtualPathUtility.Combine(
                               ConstructRelativeDestPath(propertyId), filename);
            media.getProperty("umbracoBytes").Value = uploadFile.ContentLength;
            media.getProperty("umbracoExtension").Value =
               VirtualPathUtility.GetExtension(filename).Substring(1);

            return media;
        }

        #endregion
    }
}
```

Here you can get an idea of what can be done.

If you plan to develop your own IMediaFactory implementations you will get better error feedback by uncommenting the FileUpload WebControl in the ~/usercontrols/MultipleFileUpload/MultipleFileUpload.ascx file and using this instead of the flex based interface.

Custom usercontrol

You can also create your own interface in umbraco by creating a usercontrol that inherits `MultipleFileUploadControl` and still have the HttpHandler and MediaFactories handle the rest for you.

Extension ideas (go for it)

A couple of extension ideas

A MediaFactory, that handles zip files and expands them to folders and files in the media section.

An Ajax based interface to umbraco

4 Flex GUI

The source for the Flex GUI is located in the flex folder.

The project consists of an mxml file located in the mxml folder, and 4 ActionScript files located in the classes folder.

To build the MultipleFileUpload.swf it is necessary to have the Adobe Flex 3 SDK installed.
The SDK can be downloaded here:
http://www.adobe.com/products/flex/flexdownloads/index.html

Modify the first line of the file "build.properties" to define the path to the SDK on your system:

```
# The location of the Flex 3 SDK on your system.
flex3sdk.dir = C:/TheFlashPit/flex_sdk_3
```

(note the use of forward slash in the path)

To build the project, just double-click the build.cmd file, and the resulting MultipleFileUpload.swf is placed in the build folder.

To test the GUI on your Umbraco installation, simply copy the swf to:
~/usercontrols/MultipleFileUpload/MultipleFileUpload.swf

Please note that it may be necessary to empty your browser cache to force your browser to use the new file.

5 Props goes out to

The MultipleFileUpload package is partly based on these open source components, which we'd like to recognize here.

<swfobject>
http://code.google.com/p/swfobject/

The Flex GUI is based on the FlashUpload project on "The Code Project":
http://www.codeproject.com/KB/aspnet/FlashUpload.aspx

To everybody who attended Codegarden 08 in Copenhagen and helped us win the Package coding contest and the XBox360 by voting.

And of cause Umbraco, the Umbraco community, Niels & Per.