

fatgeorge FileUpload Control v3.5 for XMod

The File Upload control enables your form's users with an easy means of uploading a file to your web server.

It's really two controls in one. It's both a file picker control and a file upload control. By setting the *displaymode* attribute, you can use the control as a file picker, a file upload control, or both. Plus, it works with DNN's list of allowable file extensions so only extensions allowed (via DNN's HostSettings page) are available to the control for potential upload. The control can further limit the types of files uploaded by setting the *extensions* attribute.

The control also provides a list control which lists available files in the target upload directory and dynamically refreshed after each upload. This allows your users to upload a file and then select it.

Installation

Use the XMod Administration screen to install the fatgeorge_fileupload_x_x.zip file

Or

Copy the fatgeorge.fileupload.dll to your dotnetnuke/bin folder then use in your XMod form as specified below.

The File Upload control is customizable. Using its properties, you can filter which files can be uploaded according to their file extensions (this filter is applied after the DNN extension filter), dynamically generate unique filenames for uploaded files using a pre-defined pattern, overwrite or preserve existing files, specify an upload directory, and much more.

To use the File Upload control, you must `<register>` it in your form and create a `<custom>` tag where you want to use the control. Below are attributes of the FileUpload control. These include those attributes required by the `<custom>` control tag (ref, name, tagprefix, etc.)

NOTE: To use the File Upload control, you must enable custom controls in XMod. For security purposes, this is turned off by default. For information on enabling custom controls, see the "Usage Notes" section below.

ATTRIBUTES :

Property	Description	New in Version	Updated in Version
ref	The ID of the control, used by XMod to set the name of the field for the form. This must be unique within the form. (Required) Names must begin with a letter and may only include letters, numbers and the underscore character.		
extensions	Determines which file types can be uploaded. This serves as an additional filter beyond what is allowed by DNN. Thus, if a file type is allowed here, but is not allowed by DNN, the file will not be allowed. Extension should be specified as a comma-separated list. So, to only allow images, you		

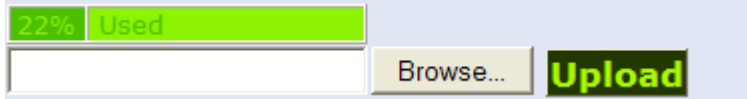
	would specify "jpg,gif,png" as the value for this attribute. The default value is the extension list allowed by DNN. (Optional)		
tagprefix	This is the prefix XMod uses to match the control with the assembly that contains the custom control (specified in the <register> tag). (Required) [String value]		
name	The name of the control. This should always be "FileUpload" (Required)		
cssclass	Assigns a CSS class name to the internal HTML file control. (Optional)		
uploadlinktext	The text of the link the user clicks to upload the file they've selected. The default value is 'Upload'. (Optional)		
uploaddir	The directory into which the file will be placed and the directory where the file picker portion of the control will load its list of files from. See usage notes below for important additional information. The default value is the portal's upload directory. (Optional)		
overwrite	If <i>true</i> , a file with the same filename as the file being uploaded will be overwritten. If <i>false</i> , the control will generate a unique filename for the uploaded file in the case of a naming conflict by appending a number to the end of the filename. The default value is <i>false</i> (Optional)		
namingpattern	If blank, the uploaded file's name is used. Otherwise, use this attribute to dynamically create a name for the new file. The control provides a number of placeholders you can use, along with your own text, to generate filenames. Use of the placeholders is optional. <ul style="list-style-type: none"> • {FileUpload_FileName} Replaced at runtime by the uploaded file's name, without the extension. • {FileUpload_FileExt} Replaced at runtime by the uploaded file's extension. • {FileUpload_Date} Replaced at runtime by the current date in the 		

	<p>format YYYYMMDD (i.e. 20050131 for Jan 31 2005)</p> <ul style="list-style-type: none"> • {FileUpload_Time} Replaced at runtime by the current time in the format HHMMSS (i.e. 132405 for 1:24:05 pm - hours are 0-23) 		
noneselectd	If true (the default value), the control will insert "<None Selected>" as the first item in the file list. (Optional)		
displaymode	<p>(Optional) Valid values are:</p> <ul style="list-style-type: none"> • filelistonly - the control acts as a file picker control and does not give the user the option of uploading a file. • uploadonly - the file list picker isn't shown. Only the upload control is shown. • uploadandselect - this will just show the file upload control, but will store the most recently uploaded file information (as if the file were uploaded then selected in the file picker). This is helpful for allowing users to upload files without seeing other files in the directory. • If not specified, the control will display the file list picker control with a link to add a new file which then displays the upload control. 		
uploadonce	If <i>false</i> (the default), users may upload more than one file. If <i>true</i> , users are only allowed one successful upload per editing session (i.e. when adding or editing a record). If they edit the record again, they will be allowed to upload another file. (Optional)		
style	This pass-through HTML property allows you to set the CSS style properties of the control. (Optional)		
addnewlinktext	This is the text displayed in the link the user clicks to display the file upload control. The default value is 'Add New File' (Optional)		

listwidth	Allows you to control the width of the file picker list control in standard .NET units (pixels, points, etc.) (Optional)		
-----------	---	--	--

Property	Description	New in Version	Updated in Version
Resize	Specifies the new width and height to resize an image to. Aspect ratio is maintained. N.B. only applies to image files. e.g. <code>resize="800,600"</code> (see example below)		
ShrinkOnly	Boolean flag – defaults to false Used in conjunction with the Resize function. If you don't want small images to be enlarged by the resize function then set this to True.		
Aspect	Specifies if Aspect Ratio is maintained during image resize. (defaults to TRUE) If you wish to switch it off for image resize use <code>aspect="false"</code>	2.2	
FixedSize	Defaults to FALSE If set to TRUE then the image will be resized to the exact size specified in the resize option. The image will be "padded" so that it fits the exact size specified using the background colour specified in ImageBackColour. e.g. <code>fixedsize="true"</code>	2.2	
ImageBackColour	Defaults to Transparent. Can be set to any of the known .net colours. Is used by both the FixedSize and ThumbFixedSize options. (see Appendix 2 for a list of valid colours) Please note the spelling of this option if you are not from the United Kingdom!	2.2	
Thumbnail	Specifies size of thumbnail to generate. Creates a new file with same filename as original but prefixed with "thm_", thumbnail is stored in same folder as original file. Aspect ratio is maintained. N.B. only applies to image files. e.g. <code>thumbnail="150,150"</code> (see example below)		
ThumbShrinkOnly	Boolean flag – defaults to false Used in conjunction with the Thumbnail function. If you don't want small images to be enlarged by the Thumbnail function then set this to True. If you wish all your thumbnail images to be set to the same size then leave this as false.		
ThumbAspect	Specifies if Aspect Ratio is maintained during image thumbnail generation. (defaults to TRUE) If you wish to switch it off for thumbnails use <code>aspect="false"</code>	2.2	
ThumbFixedSize	Defaults to FALSE If set to TRUE then the thumbnail will be resized to the exact size specified in the Thumbnail option. The image will be "padded" so that it fits the exact size specified using the background colour specified in ImageBackColour. e.g. <code>thumbfixedsize="true"</code>	2.2	
ThumbnailFolder	Specifies a sub folder for storing thumbnails. Will create and use a subfolder under your specified <code>uploaddirectory</code> , thumbnail images are created with the same filename as the original file. (No prefix!)		

Watermark	<p>Specified text is used to produce an image with watermark. Watermark is semi-transparent and positioned in the centre at the bottom of the image. The specified text is automatically prefixed with the © Copyright symbol. (see example below)</p> <p>The original image is stored with a prefix of "orig_" should you wish to use it.</p>											
watermarkposition	<p>You can specify any 1 of 9 possible positions for your watermark:</p> <table border="1" data-bbox="355 539 608 689"> <tr> <td>tl</td> <td>tc</td> <td>tr</td> </tr> <tr> <td>ml</td> <td>mc</td> <td>mr</td> </tr> <tr> <td>bl</td> <td>bc</td> <td>br</td> </tr> </table> <p>e.g. watermarkposition="mc" Default = bc</p>	tl	tc	tr	ml	mc	mr	bl	bc	br		
tl	tc	tr										
ml	mc	mr										
bl	bc	br										
wmCompression	<p>Watermark Compression default=50% When using the watermark feature the resulting watermarked file increased in size dramatically. The compression property defaults to 50% if you do not specify it, you can set it to whatever value you like 0 = no compression 100 = maximum compression (not pretty but interesting results)</p>											
WMFontSize	<p>A numeric value which allows you to control the size/ratio of the watermark text. e.g. WMFontSize="1" is full width of the image, "0.5" would take up half the image width etc.</p>											
Maxfilesize	<p>Allows you to specify a maximum allowable file size (in kb) If file is over maxfilesize the following message is displayed: "File exceeds maximum allowed size (1024kb). File was not uploaded."</p>											
DiskQuota	<p>Allows you to limit the amount of disk space used. Specified in MB. E.g. diskquota ="5" = 5MB</p> <p>During file upload the current size of the files in the specified upload directory plus the size of the current file is checked against the diskquota. If there is not enough space remaining to upload this file the following message is displayed:</p> <p>"Unable to upload file as this will exceed your allocated file space."</p> <p>N.B. Use this setting in conjunction with the UploadDirectory parameter to control individual user's allocation.</p> <p>e.g. uploaddirectory="{XMOD_UserUsername}"</p> <p>You can control the colour of the display using dqdisplay="forecolour,backcolour"</p> <p>e.g. dqdisplay="#4A6C00,#8CF302"</p>											



You can specify a helpful tooltip e.g. dqTooltip="My Space"


quotasearch

You can also limit what gets counted by the quota check.
 e.g. if you do not think it is fair to include the generated thumbnails and watermarked images in the count and only want to include the original file that was uploaded by the user then you could specify:
 quotasearch="orig_*.*"

If you do not specify a quotasearch then all files in the folder get included.

	<pre> <form> <controls> <register tagprefix="fgfu" namespace="fatgeorge.FileUpload" assembly="fatgeorge.FileUpload" /> <custom tagprefix="fgfu" name="FileUpload" ref="ShowQuota" cssclass="NormalTextBox" cssclasslink="Normal" style="background-color: white; color: green;" extensions="txt,jpg,gif,png" uploaddirectory="{XMOD_UserUsername}" diskquota="8" dqdisplay="#4A6C00,#8CF302" dqTooltip="My Space" dqonly="true" > </custom> <addbutton display="linkbutton"/> <cancelbutton display="linkbutton"/> </controls> </form> </pre> <p>NB: ensure that the uploaddirectory and quotasearch matches that of your actual upload form!</p>		
Purge	<p>When you delete or expire a record in XMod it will not remove the uploaded file associated with that record, this results in many orphaned files and a waste of vital disk space.</p> <p>This can also be very annoying for users if you are utilising the diskquota option, as they can run out of quota even though they have deleted all their records.</p> <p>The purge option specifies a number of days; any files in your specified uploaddirectory older than (today – purge days) will be deleted.</p> <p>NB: The timestamp is taken into consideration during this comparison but beware of timedifferences if your DNN installation is hosted in a different timezone.</p> <p>NB: PLEASE USE WITH CAUTION!</p> <p>An example use case: For a project that I am working on the uploaded images and records don't need to stay on the system for a long time (I have specified 14 days) so when a new record is added I set the XMod expiry date on the form.</p> <p>When configuring the XMod module I ensure that the default (enabled) for content scheduling is still true. So the built in content scheduling takes care of the XMod records and setting the purge option to the same value takes care of the then orphaned images. E.g. Purge="14"</p> <p>In this project there are 3 images for every record:</p> <ul style="list-style-type: none"> • Original image • Watermarked image • Thumbnail <p>Because we are using the datestamp in our comparison and all 3 files were created at the same time we can remove all 3.</p> <p>Note: I plan to create a scheduled task which will also perform this purge</p>		

	<p>functionality; I need to wait for the next release of dnn which will provide the ability to pass "parameters" to scheduled tasks (see Joel Brinkman's blog for details).</p> <p>This will allow me to pass the directory and purge period to the scheduled task which can then run overnight purging files.</p>		
ImageConfirm	<p>ImageConfirm - A Boolean flag which defaults to False.</p> <p>When an image is uploaded and you have specified thumbnail generation then setting this to true will result in the thumbnail image being displayed as confirmation of the upload success. If you have not specified Thumbnail generation then the normal success message is displayed.</p> <p>Your uploadsuccessmsg is used as the AlternateText for the thumbnail image.</p> <p>Also includes ImageConfirmUpdate, if you edit a record the current image's thumbnail will be displayed (if it exists.).</p> <p>Note: The ImageConfirmUpdate feature only works if you are running on DNN 3.3.1 or 4.3.1 or higher.</p> <p>In v3.4 + ImageConfirm will now display an icon image for any non-image file uploaded. e.g. if you have uploaded a pdf file you can have a pdf icon displayed. The icon filename is constructed from the uploaded file's extension plus the specified IconExt. e.g. UserGuide.pdf would use an icon of pdf.gif</p> <p>If the iconimage cannot be found then the uploaded filename is displayed.</p>		3.4
IconPath	<p>The location of icon images for non-image file uploads, relative to the portal home/upload directory. Defaults to images/filemanager.</p>	3.4	
IconExt	<p>Specifies the extension of the icon image file to use. Defaults to .gif</p>	3.4	
ShowOpenLink	<p>You can make the IconImage a link to the file. Default = false</p>	3.4	
AltText	<p>When ShowOpenLink = True the normal iconimage alt text of the filename is prefixed with this text. Default = "click here to open "</p>	3.4	
HideUploadOnEdit	<p>When set to true the upload part of the control no longer shows when the record is being edited and ImageConfirm="true". Instead the ImageConfirm graphic/image is displayed and the link to deleted the uploaded file. If the file is deleted then the upload control becomes visible.</p> <p>Default = False</p>	3.5	
Sync	<p>DNN Synchronisation - set sync="true" to enable DNN file synchronisation for the current targetdir when upload form is saved.</p>		
Delete Image Button	<p>If you are using UploadandSelect method once you have uploaded an image the file browse and upload control should disappear and the image will be displayed (DNN 3.3.1+/4.3.1+) along with a delete button. Clicking the delete button will delete the uploaded image along with any generated files such as thumbnails. The file browse and upload controls will become visible again.</p>		
Deletelinktext	<p>The text of the link the user clicks to delete the file they've uploaded. The default value is 'Delete'. (Optional)</p>	2.2.2	

IsRequired	<p>Default = false Provides semi Required Field Validation. Whilst using UploadandSelect if the browse control has not been used and therefore there is no filepath in the browse textbox then an error message will be displayed as specified by invalidmessage.</p> <p>Whilst this does not stop a user from submitting a form without uploading a file it does prompt the user to browse for a file before submitting.</p> <p>It would be a good idea to include some on screen instructions on your form about uploading files.</p>	2.2	
invalidmessage	<p>Allows you to customize the error message for the IsRequired setting. If this is not specified the default message is used: "Please upload a file."</p>	2.2	
Compression	<p>Compression default=0 When uploading an image you can optionally choose to compress it in order to save storage space. The compression property defaults to 0 if you do not specify it, you can set it to whatever value you like 0 = no compression 100 = maximum compression (not pretty but interesting results)</p>	2.2.5	
maxDimension	<p>A boolean value which defaults to false. Switches on the maximum dimension upload restriction checking if set to True.</p>	3.2	
maxWidth	<p>Maxium width allowed for uploaded image.</p>	3.2	
maxHeight	<p>Maxium height allowed for uploaded image.</p>	3.2	
maxDimensionMsg	<p>Error message to display if the uploaded image dimensions are greater than the maxWidth or maxHeight. Default: "
File exceeds maximum allowed dimensions (Maximum Width:{0} Height:{1}).
 File was not uploaded" You can customise the message and use the following tokens to insert the dimension values:</p> <p>{0} = MaxWidth value {1} = MaxHeight value {2} = Image Width value {3} = Image Height value</p> <p>e.g. maxDimensionMsg="It's too big! it was {2}w by {3}h
&it has to be no more than {0}w by {1}h."</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p style="text-align: center;"> <input type="text"/> <input type="button" value="Browse..."/> <input type="button" value="Upload"/> </p> <p style="text-align: center; color: red;">It's too big! it was 2048w by 1536h it has to be no more than 2048w by 1530h.</p> <p style="text-align: center;">  <input type="button" value="Remove"/> </p> </div> <p>results in:</p>	3.2	

USAGE NOTES

Custom Controls Must Be Enabled by the Host in order for the FileUpload control to be used. To do so, login as Host, select "Administration" from the Actions menu, and check the

box next to "Enable Custom Form Controls" (in the Global Settings section). For security, custom controls are disabled by default.

Upload Directory: This directory is used by both portions of the control. Uploaded files are placed in the directory and the file list control looks at the directory to get its list of files. If this directory doesn't exist, the control will attempt to create it. This directory is relative to the portal's upload directory. So, if you specify "images", the file will be placed in the "/Portals/*nn*/images" directory (where *nn* is the portal's ID). The control will also create child directories. So, if you specify "2005/images", the control will attempt to create a "2005" directory with a child "images" directory.

Required Fields: You may encounter some problems when using the "required=true" attribute of the <input> control tag. If no value has been entered in the field, the required field validator will fire when clicking the upload link. While this is normal behavior, it may be a bit confusing for your users. There are two ways to alleviate this problem - 1) set a default value for those controls; 2) use the server-side required field validation offered as a new feature in XMod version 3 (see <validate>).

EXAMPLES

The following example registers the file upload control, then uses it via the <custom> tag. The control will display the file picker control with an "Add New File" link which, when clicked, will display the file upload control. Any uploaded file will be saved into the 2005/images subdirectory of the portal's upload directory

```
<form>
  <controls>
    <register tagprefix="fgfu" namespace="fatgeorge.FileUpload"
assembly="fatgeorge.FileUpload" />
    <input ref="Name">
      <label>Image Name</label>
    </input>
    <input ref="Caption">
      <label>Email</label>
    </input>
    <custom tagprefix="fgfu" name="FileUpload" ref="FileUpload"
cssclass="NormalTextBox" cssclasslink="Normal" style="background-
color: blue; color: yellow;" extensions="jpg,gif,png"
uploaddirectory="{XMOD_UserUsername}" displaymode="uploadandselect"
uploadonce="true" uploadsuccesmsg="Your image has been uploaded"
watermark="{XMOD_PortalName}" thumbnail="150,150" maxfilesize="2048"
resize="800,600" diskquota="5"
namingpattern="{XMOD_UserUsername}_{FileUpload_Date}_{FileUpload_Time
}_{FileUpload_FileExt}">
      <label>File Upload</label>
    </custom>
  </controls>
</form>
```

Appendix 1

Examples of images

Uploaded Image resized to 800 x 600 with watermark



Thumbnail 150 x 150 with watermark



