

# ASP Filefunctions 1.5

J. Ritmeijer

[jarit@xs4all.nl](mailto:jarit@xs4all.nl)

[www.xs4all.nl/~jarit/asp/](http://www.xs4all.nl/~jarit/asp/)

## FILEFUNCTIONS LICENSE AGREEMENT

NOTICE: THIS IS A CONTRACT. BY INSTALLING AND USING THIS SOFTWARE, YOU ARE AGREEING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. PLEASE READ CAREFULLY BEFORE PROCEEDING. IF YOU DO NOT AGREE TO THESE TERMS, PROMPTLY DESTROY ALL COPIES OF THE SOFTWARE IN YOUR POSSESSION.

If you find this program usefull and continue to use it, you must register it. Details are provided in the section "Registration and Licenses" in this manual

The Software is owned by its creator, the code is a valuable trade secret protected by copyright law and international treaty. Recipient will not disclose, publish, or distribute the Software except as permitted, and agrees to take reasonable precautions to prevent any unauthorized use, disclosure, publication, or distribution. Proprietary rights notices on the Software shall be preserved on all copies made, and the Software shall not be modified, reverse engineered, or decompiled for any purpose, or translated into another computer language.

THE SOFTWARE IS PROVIDED "AS IS," AND WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, AS TO NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, OR AS TO ITS PERFORMANCE, ACCURACY OR COMPLETENESS. The creator of this program does not guaranty that the Software (or any similar product) will be available in the future. Use of the Software is entirely at Recipient's risk.

THE CREATOR OF THIS PROGRAM SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, WHETHER ARISING IN TORT (INCLUDING NEGLIGENCE), CONTRACT OR OTHERWISE, THAT ARISE FROM ANY USE OF THE SOFTWARE AND/OR OTHER CONFIDENTIAL INFORMATION OR FAILURE TO PERFORM UNDER THIS AGREEMENT.

This license does not entitle you to any hard-copy documentation, support or telephone assistance. You shall only copy the Software for archival purposes and only if such copy contains all of the original Software's proprietary notices. You shall not (and you shall not permit others to): (i) transfer or use the Software except as expressly and unambiguously permitted in this Agreement; (ii) remove any proprietary notices or labels on the Software; or (iii) reverse assemble, reverse compile or reverse engineer the Software.

This License will terminate immediately without notice if the Recipient fails to comply with any provision of this License. Upon termination Recipient must destroy all copies of the Software.

Recipient may terminate this License at any time by destroying all copies of the Software.

Copyright (C) 1997 J. Ritmeijer All rights reserved.

## Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>1</b>
<b>FILEFUNCTIONS OBJECT MODEL.....</b>	<b>2</b>
<b>Installation.....</b>	<b>2</b>
Uninstalling automatically:.....	2
Uninstalling by hand (if the above procedure fails).....	2
<b>Registration and Licenses.....</b>	<b>2</b>
<b>Changes in version 1.5.....</b>	<b>3</b>
<b>FileFunctions.Files.....</b>	<b>4</b>
Properties:.....	4
Path.....	4
Mask.....	4
Attributes.....	5
LastError.....	5
Item.....	5
Count.....	5
Sorted.....	6
Methods.....	7
GetFileList.....	7
ClearFileList.....	7
Exists.....	7
Copy.....	7
KillFile.....	8
Rename.....	8
MakeDir.....	8
RemoveDir.....	8
Add.....	8
Remove.....	9
Sort.....	9
<b>FileFunctions.File.....</b>	<b>10</b>
Properties:.....	10
Name.....	10
Size.....	10
Modified.....	10
Attributes.....	11
Path.....	11
LastError.....	11
Methods.....	12
Copy.....	12
Rename.....	12
KillFile.....	12

## FileFunctions Object Model

The functionality of the filefunctions object can be divided into two parts. Single file/Directory operations and operations that can be done on single files in a filelist. Confused ???? read on....

For a demonstration check out the *filefunc.asp* file included in the distribution.

```
Filefunctions.Files
|
|-- FileFunctions.File
```

### Installation

The easiest way to install the FileFunctions component is to download the easy setup version. This version includes a user friendly installation program, which automatically installs necessary dll's and registers the component for you.

People who know what they are doing and already have the *MSVBVM50.DLL* (VB Service Pack 3) file on their machine can download the smaller version and register the component by hand by placing the filefunctions.dll file in their Windows System directory and running the command:

```
regsvr32 filefunctions.dll
```

It's also possible to drag the DLL on a Microsoft Transaction Server package.

If you have the demo or a previous version installed on your machine it's best to follow the following procedure to uninstall it:

#### Uninstalling automatically:

- Reboot the computer
- From the [control Panel/Add remove programs] choose filefunctions
- The old version is now removed
- Install the new version

#### Uninstalling by hand (if the above procedure fails)

- go to the command prompt
- go to the directory %systemroot%\system32
- type 'regsvr32 /u filefunctions.dll'
- type 'del filefunctions.dll'
- if you get the error 'The process cannot access the file' reboot the computer and then remove the 'filefunctions.dll' file.
- Install the new version

Both versions include a registration program, which allows you to register via Fax, e-mail or postal service.

### Registration and Licenses

If you like this program and continue to use it, you must register it.

The easiest way to register ASP FileFunctions is to pay on-line by using the secure order form that's available at <http://www.xs4all.nl/~jarit/asp/register.html>. People who are behind a firewall or who use an older browser can use an unsecured version of the same form.

You can also use the program register.exe, which is included in the filefunctions zip file to pay via fax, e-mail or regular mail.

You can pay using:

- Cash, in many currencies
- Check or money order
- VISA
- MasterCard
- American Express
- First Virtual
- Invoice

There are 3 types of license available for this component:

- Single Domain license                   \$30
- Server license                            \$100
- Unlimited license + Source code       \$500

The Domain license allows you to use the component on one (virtual) domain, if the server handles multiple domains, which use the component, you need to acquire a license for each single domain.

For servers that handle many virtual domains it's more economic to buy the *Server license* or the *Unlimited license*. New in version 1.5 is the ISP safe version. Read *Changes in version 1.5* for more information.

The unlimited license allows you to use the component on as many servers as you like. This license also includes the source code, which you can modify and use, but not redistribute or resell.

Minor upgrades are free (v1.0 - 1.99), major upgrades (v1.0 - 2.0) may require you to pay an additional fee.

### **Changes in version 1.5**

- **Sorting**  
FileFunctions now supports sorting of all items in the file list. You can sort both ascending and descending on: Path, Filename, File extension, File size, Modified Date and File Attributes. See the 'Sort' method below for more information.
- **Improved Error handling**  
The Filefunctions component now has improved error handling. Before it was necessary to query the 'LastError' property to see what error occurred. The current version has normal error handling just like all other ASP objects. The *LastError* property remains supported.
- **Variables are passed By Value**  
All variables are now passed 'By Value' not 'By Reference'. The following code used to be illegal in previous versions of the component:  

```
Mask = "*" . "*"
FFunc.Mask = Mask
```
- **Bug fix 16bit character set systems**  
On versions of Window NT that have 16bit character sets (e.g. the Chinese version of NT) the file copying routine did not work properly. This has been fixed.
- **ISP Safe version available**  
There is a stripped down version available of the component that has all potential hazardous methods removed. This is ideal for large Windows NT Internet Service Providers who host many domains on the same machine. The price is the same as for the full version. Just mention '*ISP Safe Version*' in the Comments field on the Registration page or in the registration program.  
The following methods have been removed: Copy, KillFile, Rename, MakeDir and RemoveDir.  
Custom versions are available on request.

### FileFunctions.Files

Methods and properties in the .Files object are usually used for single file operations. In other words if you have the need to just *move* or *rename* one file or create a directory anywhere on the system then you'll use this object.

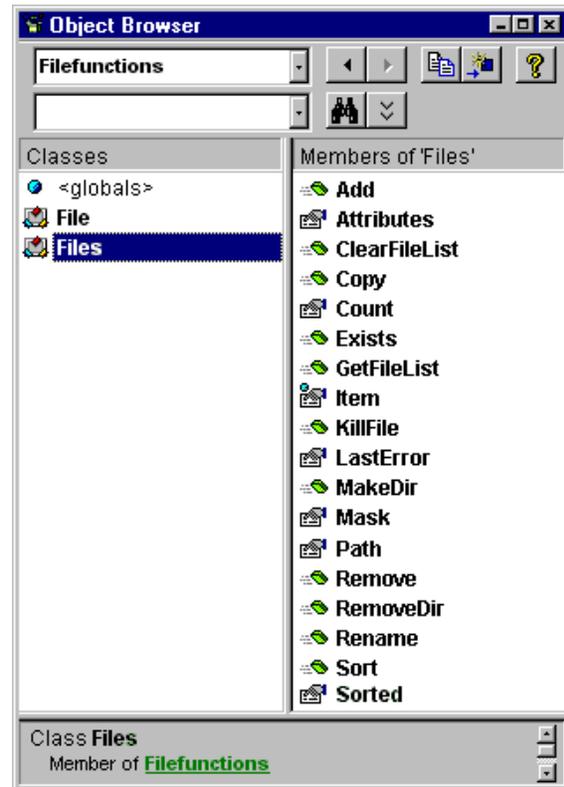
This object also makes it possible to retrieve a list of files and their related information like size, properties, modifieddate etc.

#### Properties:

- Path (r/w)
- Mask (r/w)
- Attributes (r/w)
- LastError (r/w)
- Item (r)
- Count (r)
- Sorted (r/w) (*new in 1.5*)

#### Methods

- GetFileList
- ClearFileList
- Exists
- Copy
- KillFile
- Rename
- MakeDir
- RemoveDir
- Add
- Remove
- Sort (*new in 1.5*)



Before testing any of the examples, don't forget to include the following line in the source code.

```
Set FFunc = Server.CreateObject("FileFunctions.files")
```

#### Properties:

##### Path

**DataType:** String  
**Description:** Path of the files to retrieve into the list.  
**Example:** `FFunc.Path = "c:\pdffiles\"`

##### Mask

**DataType:** String  
**Description:** Mask, indicating which files to retrieve. Wildcards are allowed  
**Example:** `FFunc.Mask = "*.pdf"`

**Attributes****Data Type:** Long**Description:** File attributes, a combination of one or more of the following values.

0 = Normal  
 2 = Hidden  
 4 = System file  
 8 = Volume label  
 16 = Directory or folder

By default this value is -1 (Get all files).

A Hidden System file has a value of 2 + 4

**Example:**  

```
'get directory names
FFunc.Attributes = 16
```

**LastError****Data Type:** String**Description:** Description of the last error occurred in the Files object. Version 1.5 supports normal error handling and reports errors to the VBScript Err object. The *LastError* property remains supported.

**Example:**  

```
If not FFunc.RemoveDir "c:\xx" then
    response.write FFunc.LastError
end if
```

**Item****Data Type:** FileFunctions.File**Description:** A collection of .File objects. This is the default property. When the list is sorted and the *.sorted* property is set to true then the *.item* property returns an item from the sorted list. For more information see the *.sort* method.

**Example:**  

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"
For I = 1 to FFunc.Count
    Response.write(FFunc.Item(I).Name & "<BR>")
Next
```

This is the same as

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"
For each File in Ffunc
    Response.write(File.Name & "<BR>")
Next
```

However the 'for each' method does not work on sorted lists

**Count****Data Type:** Long**Description:** Number of .File objects in the collection.

**Example:**  

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"
Response.write("This Directory contains " & FFunc.Count _
    & " Files")
```

**Sorted****DataType:** Boolean**Description:** This property is automatically set to TRUE when you call the *.sort* method. When you set this property to FALSE the FileList is automatically 'unsorted'. Setting this property to TRUE without calling the *.sort* method will result in errors.

The following methods will also set this property to FALSE:

- *.Add*
- *.ClearFileList*
- *.GetFileList*
- *.Remove*

For more information see the *.sort* method.

**Example:**

```
\ This example displays the files both sorted and unsorted
FFunc.GetFileList "c:\pdffiles\*.pdf"
FFunc.Sort "Name", "ascending"
\ *** sorted ***
For I = 1 to FFunc.count
    Response.write(FFunc(I).Name & "<BR>")
Next
\ *** unsorted ***
Ffunc.Sorted = false
For I = 1 to FFunc.count
    Response.write(FFunc(I).Name & "<BR>")
Next
```

**Methods****GetFileList**

**Syntax:** GetFileList([Path as String])  
**ReturnType:** Boolean (Indicating an Error)  
**Description:** This function retrieves a list of files. You can clear the list with ClearFileList or add another set of files to the list with a second call to GetFileList.  
 You can override the previously set *Path* and *Mask* properties by passing a new Path as argument.  
 This method resets the *.sorted* property to FALSE.  
**Example:**

```
FFunc.GetFileList "c:\pdffiles\*.pdf"
For each File in Ffunc
    Response.write(File.Name & "<BR>")
Next
```

**ClearFileList**

**Syntax:** ClearFileList()  
**ReturnType:** Boolean (Indicating an Error)  
**Description:** Clear a previously retrieved FileList.  
 This method resets the *.sorted* property to FALSE.  
**Example:**

```
FFunc.GetFileList "c:\pdffiles\*.pdf"
For each File in Ffunc
    Response.write(File.Name & "<BR>")
Next
FFunc.ClearFileList
FFunc.GetFileList "c:\pdffiles\extra\*.pdf"
For each File in Ffunc
    Response.write(File.Name & " - " & File.Size & "<BR>")
Next
```

**Exists**

**Syntax:** Exists([FileName])  
**ReturnType:** Boolean Indicating if the file exists  
**Description:** Checks if a file or directory exists.  
 If you don't specify a filename this function assumes the path & mask specified in the *.path* and the *.mask* property.  
**Example:**

```
if FFunc.Exists("c:\autoexec.bat") then
    Response.write("autoexec exists")
Else
    Response.write("autoexec does not exist")
end if
```

**Copy**

**Syntax:** Copy(FromName As String, ToName As String)  
**ReturnType:** Boolean (Indicating an Error)  
**Description:** Copies a file to a different location.  
 If you don't specify a path this function assumes the path specified in the *.path* property.  
**Example:**

```
FFunc.Path = "c:\users\jeroen"
Er = FFunc.Copy("c:\pdffiles\choice.pdf", "item1.pdf")
If not Er then
    Response.write(FFunc.LastError)
End if
```

**KillFile****Syntax:** KillFile(FileName As String)**ReturnType:** Boolean (Indicating an Error)**Description:** This function erases a file from the harddisk.

If you don't specify a path this function assumes the path specified in the .path property.

**Example:**

```
Er = FFunc.KillFile("c:\users\jeroen\item1.pdf")
```

```
If not Er then
```

```
    Response.write(FFunc.LastError)
```

```
End if
```

**Rename****Syntax:** Rename(FromName As String, ToName As String)**ReturnType:** Boolean (Indicating an Error)**Description:** Use this function to rename a file. You can also use this function to move a file to a different directory or harddisk. If you rename or move a directory to a different location the entire content of the directory (including subdirectories) is moved.

If you don't specify a path this function assumes the path specified in the .path property.

**Example:**

```
FFunc.Rename "old.pdf", "new.pdf"
```

```
FFunc.Rename "c:\autoexec.bat", "d:\autoexec.bat"
```

**MakeDir****Syntax:** MakeDir(DirName As String)**ReturnType:** Boolean (Indicating an Error)**Description:** This function creates a new directory.

If you don't specify a path this function assumes the path specified in the .path property.

**Example:**

```
FFunc.MakeDir "c:\pdffiles\new"
```

**RemoveDir****Syntax:** RemoveDir(DirName As String)**ReturnType:** Boolean (Indicating an Error)**Description:** This function removes an *empty* directory.

If you don't specify a path this function assumes the path specified in the .path property.

**Example:**

```
Er = FFunc.RemoveDir("c:\pdffiles\new")
```

```
If not Er then
```

```
    Response.write(FFunc.LastError)
```

```
End if
```

**Add****Syntax:** Add(Path as String, Name As String, [Attributes As Long], [Size As Long], [Modified As String])**ReturnType:** FileFunctions.File**Description:** *Add* is used internally by the component to add individual files to the collection of files. You probably don't need this method, but hey...it's there.This method resets the *.sorted* property to FALSE.**Example:**

```
FFunc.Path = "c:\pdffiles"
```

```
FFunc.Mask = "*.pdf"
```

```
FFunc.GetFilesList
```

```
FFunc.Add "", "This is the end of the list"
```

```
For each File in Ffunc
```

```
    Response.write(File.Name & "<BR>")
```

```
Next
```

**Remove****Syntax:** Remove(Key As Variant)**ReturnType:** n.a.**Description:** This method removes an item from a previously retrieved filelist. This method *does not* remove files from your harddisk.This method resets the *.sorted* property to FALSE.**Example:**

```
FFunc.GetFileList("c:\pdffiles\*.pdf")
FFunc.Remove 13 ' remove the 13nd item
FFunc.Remove "trash.pdf" ' Remove the item with the
filename 'trash.pdf'
```

**Sort****Syntax:** Sort([Field as Variant], [Order as Variant])**ReturnType:** Boolean (Indicating an Error)**Description:** This is a very flexible method to sort the file list. You can sort both ascending and descending on: Path, Filename, File extension, File size, Modified Date and File Attributes.This method automatically sets the *.sorted* property to TRUE.

Field can be:

- *Name* (This is the default order if you don't specify the *Field* variable). If the FileList contains multiple paths then the paths are sorted as well.
- *Extension* (Sort by file extension)
- *Path* (Sort by pathname)
- *Size* (Sort by filesize)
- *Modified* (Sort by file modified date). This does not work 100% since the file date and time are converted to a String before sorting. However the dates are grouped together.
- *Attributes* (Sort by file attributes)

Order can be:

- Ascending (Order A – Z. This is the default order if you don't specify the *Order* variable)
- Descending (Order Z-A)

For more information see the *.sorted* property.**Example:**

```
` This example lets the enduser choose the sortorder
Sort = Request("sort")
Order = Request("order")
if Sort <> "" then
    FFunc.sort Sort,Order
end if
For i = 1 to ffunc.count
    response.write(FFunc(i).Path & "<BR>")
    response.write(FFunc(i).name & "<BR>")
    response.write(FFunc(i).Size & "<BR>")
    response.write(FFunc(i).Modified & "<BR>")
    response.write(FFunc(i).Attributes & "<P>")
Next
```

See 'filefunc.asp' for the complete example

### FileFunctions.File

It doesn't make much sense to create this object by using the CreateObject function. Normally one or more instances of this object are created by the FileFunctions.Files.GetFileList() function.

#### Properties:

- Name (r/w)
- Size (r/w)
- Modified (r/w)
- Attributes (r/w)
- Path (r/w)
- LastError (r/w)

#### Methods

- Copy
- Rename
- KillFile



Before testing any of the examples, don't forget to include the following line in the source code.

```
Set FFunc = Server.CreateObject("FileFunctions.files")
```

#### Properties:

##### Name

**DataType:** String

**Description:** FileName. This is the default property

**Example:**

```
FFunc.GetFileList "c:\pdf\files\*.pdf"
For each File in FFunc
    Response.write(File.Name & "<BR>")
Next
```

##### Size

**DataType:** Long

**Description:** FileSize

**Example:**

```
FFunc.GetFileList "c:\pdf\files\*.pdf"
For each File in FFunc
    Response.write(File.Size & "<BR>")
Next
```

##### Modified

**DataType:** String (not Date)

**Description:** File Modified Date. Note that this property returns the Date/Time in the following format: mm/dd/yyyy hh:mm:ss. This is done for performance reasons, because calculating 'real' dates on directories with 1500 or more files is very CPU intensive.

**Example:**

```
FFunc.GetFileList "c:\pdf\files\*.pdf"
For each File in FFunc
    Response.write(File.Modified & "<BR>")
Next
```

**Attributes****Data Type:** Long**Description:** File attributes, a combination of one or more of the following values.

0 = Normal  
 2 = Hidden  
 4 = System file  
 8 = Volume label  
 16 = Directory or folder

A Hidden System files has a value of 2 + 4.

**Example:**

```
` print directories first, then hidden files
FFunc.Attributes = 16 + 2
FFunc.GetFilesList("c:\*.*)
Response.write("<B>Directories</B><BR>")
For Each file In ffunc
  If (file.Attributes And 16) Then
    Response.write(file.Name & "<BR>")
  End If
Next
Response.write("<P><B>Hidden files</B><BR>")
For Each file In ffunc
  If (file.Attributes And 2) Then
    Response.write(file.Name & "<BR>")
  End If
Next
```

**Path****Data Type:** String**Description:** Returns the full path of the file**Example:**

```
` this example always prints "c:\pdffiles"
FFunc.GetFilesList "c:\pdffiles\*.pdf"
For each File in FFunc
  Response.write(File.Path & "<BR>")
Next
```

**LastError****Data Type:** String**Description:** Description of the last error occurred in the current file object. Version 1.5 supports normal error handling and reports errors to the VBScript Err object. The *LastError* property remains supported.**Example:**

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"
If not FFunc.items(1).copy("c:\xpdffiles\test.pdf")
  Response.write(FFunc.items(1).LastError)
End if
```

## Methods

### Copy

**Syntax:** Copy(ToName As String)

**ReturnType:** Boolean (Indicating an Error)

**Description:** Copies a file to a different location.

If you don't specify a path then this function assumes the path specified in the .path property. This function does not copy directories.

**Example:**  

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"  
FFunc("test.pdf").copy("c:\pdffiles\backup\test.pdf")
```

### Rename

**Syntax:** Rename(ToName As String)

**ReturnType:** Boolean (Indicating an Error)

**Description:** Use this function to rename a file or directory. You can also use this function to move a file to a different directory or harddisk. If you rename or move a directory to a different location the entire content of the directory (including subdirectories) is moved.

If you don't specify a path this function assumes the path specified in the .path property.

**Example:**  

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"  
FFunc("test.pdf").rename("c:\pdffiles\backup\test.pdf")
```

### KillFile

**Syntax:** KillFile()

**ReturnType:** Boolean (Indicating an Error)

**Description:** This function erases a file from the harddisk and also works on empty directories.

**Example:**  

```
FFunc.GetFilesList "c:\pdffiles\*.pdf"
```

For each file in Ffunc

```
    file.killfile
```

```
next
```