

Apache Module `mod_autoindex`

Description:	Generates directory indexes, automatically, similar to the Unix <code>ls</code> command or the Win32 <code>dir</code> shell command
Status:	Base
Module Identifier:	<code>autoindex_module</code>
Source File:	<code>mod_autoindex.c</code>

Summary

The index of a directory can come from one of two sources:

- A file written by the user, typically called `index.html`. The `DirectoryIndex` directive sets the name of this file. This is controlled by `mod_dir`.
- Otherwise, a listing generated by the server. The other directives control the format of this listing. The `AddIcon`, `AddIconByEncoding` and `AddIconByType` are used to set a list of icons to display for various file types; for each file listed, the first icon listed that matches the file is displayed. These are controlled by `mod_autoindex`.

The two functions are separated so that you can completely remove (or replace) automatic index generation should you want to.

Automatic index generation is enabled with using `Options +Indexes`. See the `Options` directive for more details.

If the `FancyIndexing` option is given with the `IndexOptions` directive, the column headers are links that control the order of the display. If you select a header link, the listing will be regenerated, sorted by the values in that column. Selecting the same header repeatedly toggles between ascending and descending order. These column header links are suppressed with `IndexOptions` directive's `SuppressColumnSorting` option.

Note that when the display is sorted by "Size", it's the *actual* size of the files that's used, not the displayed value - so a 1010-byte file will always be displayed before a 1011-byte file (if in ascending order) even though they both are shown as "1K".

Topics

Autoindex Request Query Arguments	1
URI References	11

Directives

AddAlt	2	DefaultIcon	5
AddAltByEncoding	3	HeaderName	6
AddAltByType	3	IndexIgnore	7
AddDescription	4	IndexOptions	7
AddIcon	4	IndexOrderDefault	10
AddIconByEncoding	5	ReadmeName	10
AddIconByType	5		

Autoindex Request Query Arguments

Apache 2.0.23 reorganized the Query Arguments for Column Sorting, and introduced an entire group of new query options. To effectively eliminate all client control over the output, the `IndexOptions IgnoreClient` option was introduced.

 Apache Module mod_autoindex

The column sorting headers themselves are self-referencing hyperlinks that add the sort query options shown below. Any option below may be added to any request for the directory resource.

- C=N sorts the directory by file name
- C=M sorts the directory by last-modified date, then file name
- C=S sorts the directory by size, then file name
- C=D sorts the directory by description, then file name
- O=A sorts the listing in Ascending Order
- O=D sorts the listing in Descending Order
- F=0 formats the listing as a simple list (not FancyIndexed)
- F=1 formats the listing as a FancyIndexed list
- F=2 formats the listing as an HTMLTable FancyIndexed list
- V=0 disables version sorting
- V=1 enables version sorting
- P=*pattern* lists only files matching the given *pattern*

Note that the 'P' pattern query argument is tested *after* the usual `IndexIgnore` directives are processed, and all file names are still subjected to the same criteria as any other autoindex listing. The Query Arguments parser in `mod_autoindex` will stop abruptly when an unrecognized option is encountered. The Query Arguments must be well formed, according to the table above.

The simple example below, which can be clipped and saved in a header.html file, illustrates these query options. Note that the unknown "X" argument, for the submit button, is listed last to assure the arguments are all parsed before `mod_autoindex` encounters the X=Go input.

```
<form action="" method="get">
  Show me a <select name="F">
    <option value="0"> Plain list</option>
    <option value="1" selected="selected"> Fancy list</option>
    <option value="2"> Table list</option>
  </select>
  Sorted by <select name="C">
    <option value="N" selected="selected"> Name</option>
    <option value="M"> Date Modified</option>
    <option value="S"> Size</option>
    <option value="D"> Description</option>
  </select>
  <select name="O">
    <option value="A" selected="selected"> Ascending</option>
    <option value="D"> Descending</option>
  </select>
  <select name="V">
    <option value="0" selected="selected"> in Normal order</option>
    <option value="1"> in Version order</option>
  </select>
  Matching <input type="text" name="P" value="" />
  <input type="submit" name="X" value="Go" />
</form>
```

AddAlt Directive

Description: Alternate text to display for a file, instead of an icon selected by filename

Apache Module mod_autoindex

Syntax:	AddAlt <i>string file</i> [<i>file</i>] ...
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

AddAlt provides the alternate text to display for a file, instead of an icon, for FancyIndexing. *File* is a file extension, partial filename, wild-card expression or full filename for files to describe. If *String* contains any whitespace, you have to enclose it in quotes (" or '). This alternate text is displayed if the client is image-incapable, has image loading disabled, or fails to retrieve the icon.

Examples

```
AddAlt "PDF file" *.pdf
AddAlt Compressed *.gz *.zip *.Z
```

AddAltByEncoding Directive

Description:	Alternate text to display for a file instead of an icon selected by MIME-encoding
Syntax:	AddAltByEncoding <i>string MIME-encoding</i> [<i>MIME-encoding</i>] ...
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

AddAltByEncoding provides the alternate text to display for a file, instead of an icon, for FancyIndexing. *MIME-encoding* is a valid content-encoding, such as x-compress. If *String* contains any whitespace, you have to enclose it in quotes (" or '). This alternate text is displayed if the client is image-incapable, has image loading disabled, or fails to retrieve the icon.

Example

```
AddAltByEncoding gzip x-gzip
```

AddAltByType Directive

Description:	Alternate text to display for a file, instead of an icon selected by MIME content-type
Syntax:	AddAltByType <i>string MIME-type</i> [<i>MIME-type</i>] ...
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

AddAltByType sets the alternate text to display for a file, instead of an icon, for FancyIndexing. *MIME-type* is a valid content-type, such as text/html. If *String* contains any whitespace, you have to enclose it in quotes (" or '). This alternate text is displayed if the client is image-incapable, has image loading disabled, or fails to retrieve the icon.

Example

```
AddAltByType 'plain text' text/plain
```

AddDescription Directive

Description:	Description to display for a file
Syntax:	AddDescription <i>string file</i> [<i>file</i>] ...
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

This sets the description to display for a file, for FancyIndexing. *File* is a file extension, partial filename, wild-card expression or full filename for files to describe. *String* is enclosed in double quotes (").

Example

```
AddDescription "The planet Mars" /web/pics/mars.gif
```

The typical, default description field is 23 bytes wide. 6 more bytes are added by the `IndexOptions SuppressIcon` option, 7 bytes are added by the `IndexOptions SuppressSize` option, and 19 bytes are added by the `IndexOptions SuppressLastModified` option. Therefore, the widest default the description column is ever assigned is 55 bytes.

See the `DescriptionWidth` `IndexOptions` keyword for details on overriding the size of this column, or allowing descriptions of unlimited length.

Caution

Descriptive text defined with `AddDescription` may contain HTML markup, such as tags and character entities. If the width of the description column should happen to truncate a tagged element (such as cutting off the end of a bolded phrase), the results may affect the rest of the directory listing.

AddIcon Directive

Description:	Icon to display for a file selected by name
Syntax:	AddIcon <i>icon name</i> [<i>name</i>] ...
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

This sets the icon to display next to a file ending in *name* for FancyIndexing. *Icon* is either a (%-escaped) relative URL to the icon, or of the format (*alttext, url*) where *alttext* is the text tag given for an icon for non-graphical browsers.

Apache Module mod_autoindex

Name is either `^^DIRECTORY^^` for directories, `^^BLANKICON^^` for blank lines (to format the list correctly), a file extension, a wildcard expression, a partial filename or a complete filename.

Examples

```
AddIcon (IMG,/icons/image.xbm) .gif .jpg .xbm
AddIcon /icons/dir.xbm ^^DIRECTORY^^
AddIcon /icons/backup.xbm *~
```

`AddIconByType` should be used in preference to `AddIcon`, when possible.

AddIconByEncoding Directive

Description:	Icon to display next to files selected by MIME content-encoding
Syntax:	<code>AddIconByEncoding icon MIME-encoding [MIME-encoding] ...</code>
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

This sets the icon to display next to files with `FancyIndexing`. *Icon* is either a (%-escaped) relative URL to the icon, or of the format `(alttext,url)` where *alttext* is the text tag given for an icon for non-graphical browsers.

MIME-encoding is a wildcard expression matching required the content-encoding.

Example

```
AddIconByEncoding /icons/compress.xbm x-compress
```

AddIconByType Directive

Description:	Icon to display next to files selected by MIME content-type
Syntax:	<code>AddIconByType icon MIME-type [MIME-type] ...</code>
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

This sets the icon to display next to files of type *MIME-type* for `FancyIndexing`. *Icon* is either a (%-escaped) relative URL to the icon, or of the format `(alttext,url)` where *alttext* is the text tag given for an icon for non-graphical browsers.

MIME-type is a wildcard expression matching required the mime types.

Example

```
AddIconByType (IMG,/icons/image.xbm) image/*
```

DefaultIcon Directive

Description:	Icon to display for files when no specific icon is configured
Syntax:	DefaultIcon <i>url-path</i>
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

The `DefaultIcon` directive sets the icon to display for files when no specific icon is known, for FancyIndexing. *Url-path* is a (%-escaped) relative URL to the icon.

Example

```
DefaultIcon /icon/unknown.xbm
```

HeaderName Directive

Description:	Name of the file that will be inserted at the top of the index listing
Syntax:	HeaderName <i>filename</i>
Context:	server config, virtual host, directory, .htaccess
Override:	Indexes
Status:	Base
Module:	mod_autoindex

The `HeaderName` directive sets the name of the file that will be inserted at the top of the index listing. *Filename* is the name of the file to include.

Example

```
HeaderName HEADER.html
```

Both `HeaderName` and `ReadmeName` now treat *Filename* as a URI path relative to the one used to access the directory being indexed. If *Filename* begins with a slash, it will be taken to be relative to the `DocumentRoot`.

Example

```
HeaderName /include/HEADER.html
```

Filename must resolve to a document with a major content type of `text/*` (e.g., `text/html`, `text/plain`, etc.). This means that *filename* may refer to a CGI script if the script's actual file type (as opposed to its output) is marked as `text/html` such as with a directive like:

```
AddType text/html .cgi
```

Content negotiation¹ will be performed if `Options MultiViews` is in effect. If *filename* resolves to a static `text/html` document (not a CGI script) and either one of the `options`

 Apache Module mod_autoindex

Includes or IncludesNOEXEC is enabled, the file will be processed for server-side includes (see the `mod_include` documentation).

If the file specified by `HeaderName` contains the beginnings of an HTML document (`<html>`, `<head>`, etc.) then you will probably want to set `IndexOptions +SuppressHTMLPreamble`, so that these tags are not repeated.

IndexIgnore Directive

Description:	Adds to the list of files to hide when listing a directory
Syntax:	<code>IndexIgnore file [file] ...</code>
Context:	server config, virtual host, directory, <code>.htaccess</code>
Override:	Indexes
Status:	Base
Module:	mod_autoindex

The `IndexIgnore` directive adds to the list of files to hide when listing a directory. *File* is a file extension, partial filename, wildcard expression or full filename for files to ignore. Multiple `IndexIgnore` directives add to the list, rather than the replacing the list of ignored files. By default, the list contains `.` (the current directory).

```
IndexIgnore README .htaccess *~
```

IndexOptions Directive

Description:	Various configuration settings for directory indexing
Syntax:	<code>IndexOptions [+ -]option [[+ -]option] ...</code>
Context:	server config, virtual host, directory, <code>.htaccess</code>
Override:	Indexes
Status:	Base
Module:	mod_autoindex

The `IndexOptions` directive specifies the behavior of the directory indexing. *Option* can be one of

DescriptionWidth=[*n* | *] (Apache 2.0.23 and later)

The `DescriptionWidth` keyword allows you to specify the width of the description column in characters.

`-DescriptionWidth` (or unset) allows `mod_autoindex` to calculate the best width.

`DescriptionWidth=n` fixes the column width to *n* bytes wide.

`DescriptionWidth=*` grows the column to the width necessary to accommodate the longest description string.

See the section on `AddDescription` for dangers inherent in truncating descriptions.

FancyIndexing

This turns on fancy indexing of directories.

FoldersFirst (Apache 2.0.23 and later)

If this option is enabled, subdirectory listings will *always* appear first, followed by normal files in the directory. The listing is basically broken into two components, the files and the subdirectories, and each is sorted separately and then displayed subdirectories-first. For instance,

if the sort order is descending by name, and `FoldersFirst` is enabled, subdirectory `Zed` will be listed before subdirectory `Beta`, which will be listed before normal files `Gamma` and `Alpha`. **This option only has an effect if `FancyIndexing` is also enabled.**

HTMLTable (*Experimental, Apache 2.0.23 and later*)

This experimental option with `FancyIndexing` constructs a simple table for the fancy directory listing. Note this will confuse older browsers. It is particularly necessary if file names or description text will alternate between left-to-right and right-to-left reading order, as can happen on WinNT or other utf-8 enabled platforms.

IconsAreLinks

This makes the icons part of the anchor for the filename, for fancy indexing.

IconHeight[=*pixels*]

Presence of this option, when used with `IconWidth`, will cause the server to include `height` and `width` attributes in the `img` tag for the file icon. This allows browser to precalculate the page layout without having to wait until all the images have been loaded. If no value is given for the option, it defaults to the standard height of the icons supplied with the Apache software.

IconWidth[=*pixels*]

Presence of this option, when used with `IconHeight`, will cause the server to include `height` and `width` attributes in the `img` tag for the file icon. This allows browser to precalculate the page layout without having to wait until all the images have been loaded. If no value is given for the option, it defaults to the standard width of the icons supplied with the Apache software.

IgnoreClient

This option causes `mod_autoindex` to ignore all query variables from the client, including sort order (implies `SuppressColumnSorting`.)

NameWidth=[*n* | *]

The `NameWidth` keyword allows you to specify the width of the filename column in bytes.

-`NameWidth` (or unset) allows `mod_autoindex` to calculate the best width.

`NameWidth=n` fixes the column width to *n* bytes wide.

`NameWidth=*` grows the column to the necessary width.

ScanHTMLTitles

This enables the extraction of the title from HTML documents for fancy indexing. If the file does not have a description given by `AddDescription` then httpd will read the document for the value of the `title` element. This is CPU and disk intensive.

SuppressColumnSorting

If specified, Apache will not make the column headings in a FancyIndexed directory listing into links for sorting. The default behavior is for them to be links; selecting the column heading will sort the directory listing by the values in that column. **Prior to Apache 2.0.23, this also disabled parsing the Query Arguments for the sort string.** That behavior is now controlled by `IndexOptions IgnoreClient` in Apache 2.0.23.

SuppressDescription

This will suppress the file description in fancy indexing listings. By default, no file descriptions are defined, and so the use of this option will regain 23 characters of screen space to use for something else. See `AddDescription` for information about setting the file description. See also the `DescriptionWidth` index option to limit the size of the description column.

SuppressHTMLPreamble

If the directory actually contains a file specified by the `HeaderName` directive, the module usually includes the contents of the file after a standard HTML preamble (`<html>`, `<head>`, *et cetera*). The `SuppressHTMLPreamble` option disables this behaviour, causing the module to start the display with the header file contents. The header file must contain appropriate HTML instructions in this case. If there is no header file, the preamble is generated as usual.

SuppressIcon (*Apache 2.0.23 and later*)

This will suppress the icon in fancy indexing listings. Combining both `SuppressIcon` and `SuppressRules` yields proper HTML 3.2 output, which by the final specification prohibits `img` and `hr` elements from the `pre` block (used to format `FancyIndexed` listings.)

SuppressLastModified

This will suppress the display of the last modification date, in fancy indexing listings.

SuppressRules (*Apache 2.0.23 and later*)

This will suppress the horizontal rule lines (`hr` elements) in directory listings. Combining both `SuppressIcon` and `SuppressRules` yields proper HTML 3.2 output, which by the final specification prohibits `img` and `hr` elements from the `pre` block (used to format `FancyIndexed` listings.)

SuppressSize

This will suppress the file size in fancy indexing listings.

TrackModified (*Apache 2.0.23 and later*)

This returns the Last-Modified and ETag values for the listed directory in the HTTP header. It is only valid if the operating system and file system return appropriate `stat()` results. Some Unix systems do so, as do OS2's JFS and Win32's NTFS volumes. OS2 and Win32 FAT volumes, for example, do not. Once this feature is enabled, the client or proxy can track changes to the list of files when they perform a `HEAD` request. Note some operating systems correctly track new and removed files, but do not track changes for sizes or dates of the files within the directory.

Changes to the size or date stamp of an existing file will not update the Last-Modified header on all Unix platforms. If this is a concern, leave this option disabled.

VersionSort (*Apache 2.0a3 and later*)

The `VersionSort` keyword causes files containing version numbers to sort in a natural way. Strings are sorted as usual, except that substrings of digits in the name and description are compared according to their numeric value.

Example:

```
foo-1.7
foo-1.7.2
foo-1.7.12
foo-1.8.2
foo-1.8.2a
foo-1.12
```

If the number starts with a zero, then it is considered to be a fraction:

```
foo-1.001
foo-1.002
foo-1.030
foo-1.04
```

Incremental IndexOptions

Apache 1.3.3 introduced some significant changes in the handling of `IndexOptions` directives. In particular:

- Multiple `IndexOptions` directives for a single directory are now merged together. The result of:

```
<Directory /foo>
  IndexOptions HTMLTable
  IndexOptions SuppressColumnsorting
</Directory>
```

 Apache Module mod_autoindex

will be the equivalent of

```
IndexOptions HTMLTable SuppressColumnSorting
```

- The addition of the incremental syntax (*i.e.*, prefixing keywords with + or -).

Whenever a '+' or '-' prefixed keyword is encountered, it is applied to the current `IndexOptions` settings (which may have been inherited from an upper-level directory). However, whenever an unprefixed keyword is processed, it clears all inherited options and any incremental settings encountered so far. Consider the following example:

```
IndexOptions +ScanHTMLTitles -IconsAreLinks FancyIndexing
IndexOptions +SuppressSize
```

The net effect is equivalent to `IndexOptions FancyIndexing +SuppressSize`, because the unprefixed `FancyIndexing` discarded the incremental keywords before it, but allowed them to start accumulating again afterward.

To unconditionally set the `IndexOptions` for a particular directory, clearing the inherited settings, specify keywords without any + or - prefixes.

IndexOrderDefault Directive

Description:	Sets the default ordering of the directory index
Syntax:	<code>IndexOrderDefault Ascending Descending Name Date Size Description</code>
Default:	<code>IndexOrderDefault Ascending Name</code>
Context:	server config, virtual host, directory, <code>.htaccess</code>
Override:	Indexes
Status:	Base
Module:	<code>mod_autoindex</code>

The `IndexOrderDefault` directive is used in combination with the `FancyIndexing` index option. By default, fancyindexed directory listings are displayed in ascending order by filename; the `IndexOrderDefault` allows you to change this initial display order.

`IndexOrderDefault` takes two arguments. The first must be either `Ascending` or `Descending`, indicating the direction of the sort. The second argument must be one of the keywords `Name`, `Date`, `Size`, or `Description`, and identifies the primary key. The secondary key is *always* the ascending filename.

You can force a directory listing to only be displayed in a particular order by combining this directive with the `SuppressColumnSorting` index option; this will prevent the client from requesting the directory listing in a different order.

ReadmeName Directive

Description:	Name of the file that will be inserted at the end of the index listing
Syntax:	<code>ReadmeName filename</code>
Context:	server config, virtual host, directory, <code>.htaccess</code>

Apache Module mod_autoindex

Override:	Indexes
Status:	Base
Module:	mod_autoindex

The `ReadmeName` directive sets the name of the file that will be appended to the end of the index listing. *Filename* is the name of the file to include, and is taken to be relative to the location being indexed. If *Filename* begins with a slash, it will be taken to be relative to the `DocumentRoot`.

Example

```
ReadmeName FOOTER.html
```

Example 2

```
ReadmeName /include/FOOTER.html
```

See also `HeaderName`, where this behavior is described in greater detail.

URI References

[1] <http://httpd.apache.org/docs-2.1/content-negotiation.html>