

# Apache HTTP Server Version 2.4

## Apache Module `mod_autoindex`

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

### Summary

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

- A file located in that directory, typically called `index.html`. The `DirectoryIndex` directive sets the name of the file or files to be used. 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 the `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

## Directives

- [AddAlt](#)
- [AddAltByEncoding](#)
- [AddAltByType](#)
- [AddDescription](#)
- [AddIcon](#)
- [AddIconByEncoding](#)
- [AddIconByType](#)
- [DefaultIcon](#)
- [HeaderName](#)
- [IndexHeadInsert](#)
- [IndexIgnore](#)
- [IndexIgnoreReset](#)
- [IndexOptions](#)
- [IndexOrderDefault](#)
- [IndexStyleSheet](#)
- [ReadmeName](#)

## Bugfix checklist

- [httpd changelog](#)
- [Known issues](#)
- [Report a bug](#)

## See also

- [Comments](#)

## Autoindex Request Query Arguments

---

Various query string arguments are available to give the client some control over the ordering of the directory listing, as well as what files are listed. If you do not wish to give the client this control, the `IndexOptions IgnoreClient` option disables that functionality.

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'attern 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

**Syntax:**     AddAlt *string file [file] ...*

<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	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.

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

## AddAltByEncoding Directive

---

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

```
AddAltByEncoding gzip x-gzip
```

## AddAltByType Directive

---

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

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

## AddDescription Directive

---

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

```
AddDescription "The planet Mars" mars.gif
AddDescription "My friend Marshall" friends/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.

Since the *File* argument may be a partial file name, please remember that a too-short partial filename may match unintended files. For example, `le.html` will match the file `le.html` but will also match the file `example.html`. In the event that there may be ambiguity, use as complete a filename as you can, but keep in mind that the first match encountered will be used, and order your list of AddDescription directives accordingly.

See the DescriptionWidth ([↗ #indexoptions.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.

### Arguments with path information

Absolute paths are not currently supported and do not match anything at runtime. Arguments with relative path information, which would normally only be used in htaccess context, are implicitly prefixed with `*/` to avoid matching partial directory names.

## AddIcon Directive

---

<b>Description:</b> Icon to display for a file selected by name
<b>Syntax:</b> <code>AddIcon <i>icon name</i> [<i>name</i>] ...</code>
<b>Context:</b> server config, virtual host, directory, .htaccess
<b>Override:</b> Indexes
<b>Status:</b> Base
<b>Module:</b> 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, a fully qualified remote URL, or of the format (*alttext*, *url*) where *alttext* is the text tag given for an icon for non-graphical browsers.

*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.

`^^BLANKICON^^` is only used for formatting, and so is unnecessary if you're using `IndexOptions HTMLTable`.

### *#Examples*

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

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

## AddIconByEncoding Directive

---

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

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

*MIME-encoding* is a valid content-encoding, such as `x-compress`.

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

## AddIconByType Directive

---

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

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

## DefaultIcon Directive

---

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

```
DefaultIcon /icon/unknown.png
```

## HeaderName Directive

---

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

<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	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.

```
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`.

```
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 negotiation ([↗ ../content-negotiation.html](#)) 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 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` ([↗ #indexoptions.suppresshtmlpreamble](#)), so that these tags are not repeated.

## See also

- [ReadmeName](#)

## IndexHeadInsert Directive

---

<b>Description:</b>	Inserts text in the HEAD section of an index page.
<b>Syntax:</b>	<code>IndexHeadInsert "markup ..."</code>
<b>Context:</b>	server config, virtual host, directory, <code>.htaccess</code>
<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	mod_autoindex

The `IndexHeadInsert` directive specifies a string to insert in the `<head>` section of the HTML generated for the index page.

```
IndexHeadInsert "<link rel=\"sitemap\" href=\"/sitemap.html\">"
```

## IndexIgnore Directive

---

**Description:** Adds to the list of files to hide when listing a directory

**Syntax:** `IndexIgnore file [file] ...`

**Default:** `IndexIgnore "."`

**Context:** server config, virtual host, directory, `.htaccess`

**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 shell-style wildcard expression or full filename. Multiple `IndexIgnore` directives add to the list, rather than replacing the list of ignored files. By default, the list contains `.` (the current directory).

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

### Regular Expressions

This directive does not currently work in configuration sections that have regular expression arguments, such as `<DirectoryMatch>`

## IndexIgnoreReset Directive

---

**Description:** Empties the list of files to hide when listing a directory

**Syntax:** `IndexIgnoreReset ON|OFF`

**Context:** server config, virtual host, directory, `.htaccess`

**Override:** Indexes

**Status:** Base

**Module:** `mod_autoindex`

**Compatibility:** 2.3.10 and later

The `IndexIgnoreReset` directive removes any files ignored by `IndexIgnore` otherwise inherited from other configuration sections.

```
<Directory "/var/www">  
IndexIgnore *.bak .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

```

</Directory>
<Directory "/var/www/backups">
    IndexIgnoreReset ON
    IndexIgnore .??* *# HEADER* README* RCS CVS *,v *,t
</Directory>

```

Review the default configuration for a list of patterns that you might want to explicitly ignore after using this directive.

## IndexOptions Directive

<b>Description:</b>	Various configuration settings for directory indexing
<b>Syntax:</b>	IndexOptions [+ -] <i>option</i> [[+ -] <i>option</i> ] ...
<b>Default:</b>	By default, no options are enabled.
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	mod_autoindex

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

### AddAltClass

Adds an additional CSS class declaration to each row of the directory listing table when `IndexOptions HTMLTable` is in effect and an `IndexStyleSheet` is defined. Rather than the standard `even` and `odd` classes that would otherwise be applied to each row of the table, a class of `even-ALT` or `odd-ALT` where *ALT* is either the standard alt text associated with the file style (eg. *snd*, *txt*, *img*, etc) or the alt text defined by one of the various `AddAlt*` directives.

### Charset=*character-set* (Apache HTTP Server 2.0.61 and later)

The `Charset` keyword allows you to specify the character set of the generated page. The default is UTF-8 on Windows and Mac OS X, and ISO-8859-1 elsewhere. (It depends on whether the underlying file system uses Unicode filenames or not.)

```
IndexOptions Charset=UTF-8
```

### DescriptionWidth=[*n* | \*]

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

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

This option with `FancyIndexing` constructs a simple table for the fancy directory listing. It is necessary for utf-8 enabled platforms or if file names or description text will alternate between left-to-right and right-to-left reading order.

### 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 httpd software. **This option only has an effect if `FancyIndexing` is also enabled.**

### 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 httpd software.

### IgnoreCase

If this option is enabled, names are sorted in a case-insensitive manner. For instance, if the sort order is ascending by name, and `IgnoreCase` is enabled, file `Zeta` will be listed after file `alfa` (Note: file `GAMMA` will always be listed before file `gamma`).

### 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, but only up to 20 bytes wide.

`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.

### ShowForbidden

If specified, Apache httpd will show files normally hidden because the subrequest returned `HTTP_UNAUTHORIZED` or `HTTP_FORBIDDEN`

### SuppressColumnSorting

If specified, Apache httpd 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. However, query string arguments which are appended to the URL will still be honored. That behavior is controlled by `IndexOptions IgnoreClient`.

### 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. **This option only has an effect if FancyIndexing is also enabled.**

### 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. If you also specify a [ReadmeName](#), and if that file exists, The closing `</body></html>` tags are also omitted from the output, under the assumption that you'll likely put those closing tags in that file.

### SuppressIcon

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. **This option only has an effect if FancyIndexing is also enabled.**

### SuppressRules

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.) **This option only has an effect if FancyIndexing is also enabled.**

### SuppressSize

This will suppress the file size in fancy indexing listings. **This option only has an effect if FancyIndexing is also enabled.**

### TrackModified

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.

### Type=*MIME content-type (Apache HTTP Server 2.0.61 and later)*

The `Type` keyword allows you to specify the MIME content-type of the generated page. The default is `text/html`.

```
IndexOptions Type=text/plain
```

### UseOldDateFormat (*Apache HTTP Server 2.4.26 and later*)

The date format used for the `Last Modified` field was inadvertently changed to `"%Y-%m-%d %H:%M"` from `"%d-%b-%Y %H:%M"` in 2.4.0. Setting this option restores the date format from 2.2 and earlier.

### VersionSort (*Apache HTTP Server 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
```

**XHTML (Apache HTTP Server 2.0.49 and later)**

The XHTML keyword forces `mod_autoindex` to emit XHTML 1.0 code instead of HTML 3.2. **This option only has an effect if `FancyIndexing` is also enabled.**

**Incremental IndexOptions**

Be aware of how multiple `IndexOptions` are handled.

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

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

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 unprefix `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

---

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

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, if desired, prevent the client from reordering the list by also adding the `SuppressColumnSorting` index option to remove the sort link from the top of the column, along with the `IgnoreClient` index option to prevent them from manually adding sort options to the query string in order to override your ordering preferences.

## IndexStyleSheet Directive

---

<b>Description:</b>	Adds a CSS stylesheet to the directory index
<b>Syntax:</b>	<code>IndexStyleSheet <i>url-path</i></code>
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	mod_autoindex

The `IndexStyleSheet` directive sets the name of the file that will be used as the CSS for the index listing.

```
IndexStyleSheet "/css/style.css"
```

Using this directive in conjunction with `IndexOptions HTMLTable` adds a number of CSS classes to the resulting HTML. The entire table is given a CSS id of `indexlist` and the following classes are associated with the various parts of the listing:

Class	Definition
tr.indexhead	Header row of listing
th.indexcolicon and td.indexcolicon	Icon column
th.indexcolname and td.indexcolname	File name column
th.indexcollastmod and td.indexcollastmod	Last modified column
th.indexcolsize and td.indexcolsize	File size column
th.indexcoldesc and td.indexcoldesc	Description column
tr.breakrow	Horizontal rule at the bottom of the table
tr.odd and tr.even	Alternating even and odd rows

## ReadmeName Directive

---

<b>Description:</b>	Name of the file that will be inserted at the end of the index listing
<b>Syntax:</b>	<code>ReadmeName filename</code>
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	Indexes
<b>Status:</b>	Base
<b>Module:</b>	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, as in example 2, it will be taken to be relative to the `DocumentRoot`.

```
# Example 1
ReadmeName FOOTER.html
```

```
# Example 2
ReadmeName /include/FOOTER.html
```

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

## Comments

---

<b>Notice:</b>
----------------

This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed again by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Freenode, or sent to our mailing lists.

 Post a comment

 [RSS](#) [Log in / register](#)

**shamoon14** 1223 days ago Rating: 0 (register an account in order to rate comments)

Well, you can use Long Path Tool for such problems, it works good.

 Reply

---

**Copyright 2017 The Apache Software Foundation.  
Licensed under the Apache License, Version 2.0.**