Server Side Includes Quick Reference Card 1.01

http://www.explainth.at

Color key overleaf

Purpose

Server Side Includes, **SSI**, are directives placed in an HTML document. These directives are processed by the server before the document is served to the viewer.

The document extension **SHTML** is generally used to inform the server that it should parse the document for **SSI** directives prior to pushing it out. To use a different extension, or to get existing HTML files subjected to **SSI** parsing see the section entitled **Server Configuration**.

Generic Form

The generic form of an **SSI** directive is

<!--#tag argument="value" -->

The degree of support for **SSI** is highly server-dependent. This document provides a summary of **SSI** directives that are fully supported by Apache.

There should be no space before or after the **#**. Apache does not permit the use of single quotes. **SSI** directives are not case sensitive. However, it is best to follow commonly used conventions.

The Config Directive

Used to format information returned by other **SSI** directives.

- **errmsg:** Defines message on **SSI** error. e.g. **<!--#config errmsg="SSI Error" -->** would output **SSI Error** in response to the non-existent directive **<!--#site explainthat="www.explainth.at" -->**
- **sizefmt:** Determines format for reporting file sizes. **<!--#config sizefmt="abbrev" -->** This is the default. Reports file size in KB or MB. **<!--#config sizefmt="bytes" -->** Reports file size in bytes.
- **timefmt:** Determines the format for reporting date and time. **<!--#config timefmt="value" -->** where **value** is one of the following

Value	Meaning
%a	Abbreviated day name
%A	Full day name
%b	Abbreviated month name
%B	Full month name
%c	Locale date-time format
%d	Day of month, 01 to 31
%D	Shortcut for %m/%d/%y
%e	Day of month, 1 to 31. A single space character used to pad dates from 1 to 9
%H	Hour, 00 to 23
%I	Hour, 01 to 12
%j	Day of year, 001 to 366
%m	Month of year, 01 to 12
%M	Minutes, 00 to 59
%n	Newline character
%p	AM/PM string
%r	Shortcut for %I:%M:%S %p
%R	Shortcut for %H:%M
%s	Seconds since 00:00:00 on 1 January 1970
%S	Seconds, 00 to 59
%t	Tab character
%T	Shortcut for %H:%M:%S
%w	Day of week, 00 to 06
%W	Week of year, 00 to 53
%y	Year of century, 00 to 99
%Y	Year with century, e.g. 2008
%Z	Timezone name

Examples

- **<!--#config timefmt="%A, %eth of %B %Y" -->**gives **Wednesday, 3th of January 2008**.
- **<!--#config timefmt="%wth day of the %Wth week of %Y" -->**gives **4th day of the 1th week of 2008**.

3th and **1th** are both wrong. **SSI Conditionals**, discussed below, offer a better solution.

The Echo Directive

Used to echo information – i.e. output it to the HTML document being subjected to **SSI** parsing. The generic form of this directive is

<!--#echo var="value" --> where **value** is one of the following

Value	Meaning
CONTENT_LENGTH	Size of client input
CONTENT_TYPE	MIME content type
DATE_GMT	Current date/time, GMT
DATE_LOCAL	Current local date/time
DOCUMENT_NAME	Name of current document
DOCUMENT_URL	Name & Location of current document
DOCUMENT_ROOT	The root directory
LAST_MODIFIED	Document was modified on...
HTTP_COOKIE	Document cookie, if any
HTTP_REFERER	Viewer came from?
REMOTE_ADDR	Viewer's IP address
REMOTE_HOST	Host name of remote client
REQUEST_METHOD	GET or POST?
SCRIPT_NAME	Name of current script
SERVER_NAME	Host name, e.g. explainth.at
SERVER_PORT	Typically, Port 80
SERVER_PROTOCOL	Typically, HTTP/1.1
SERVER_SOFTWARE	e.g. Apache/2.0.52 (CentOS)

value is not case sensitive but is usually specified in uppercase.

CONTENT_LENGTH & CONTENT_TYPE are only relevant when the method is POST or PUT in which case they report the length and type of the information attached to the request.

Dates are reported using the format set by the last **<!--#config timefmt="value" -->**statement. Information provided by echoing values such as DOCUMENT_ROOT or SERVER_SOFTWARE could well be used maliciously.

Example

```
<script type="text/javascript">
var uip='<!--#echo var="REMOTE_ADDR" -->';
</script>
```

This code is a simple way to make the IP address of the user available to scripts run on the browser after the document has been loaded.

The Exec Directive

Used to instruct the server to run an external application or a system command. Takes one of two forms

- **<!--#exec cmd="value" -->** – to run a system command
- **<!--#exec cgi="value" -->** – to run a CGI script

Allowing the use of the **#exec** directive has serious security implications. Many site hosts block the use of this directive or at least the execution of system commands. In some cases, it may be possible to configure your host to run these directives as explained in the section entitled **Server Configuration**.

Examples

- **<!--#exec cmd="ping explainth.at" -->** – Pings explainth.at
- **<!--#exec cmd="ls" -->** – returns a directory listing. If your server is running Windows use **dir** instead.
- **<!--#exec cgi="/scripts/myip.php" -->** – runs the script *myip.php*.

The FSize & FLastMod Directives

Used to return information regarding file size and last modification date.

- **<!--#fsize file="ssiqr.pdf" -->**
- **<!--#fsize virtual="/en/ssi/sfile.shtml" -->**
- **<!--#flastmod virtual="/en/menu/ssi.shtml" -->**

The nature of the output from the **#fsize** directive is determined by the last **<!--#config sizefmt="value" -->** statement. See **Notes** for the meaning of **file** and **virtual**.

The Include Directive

The main rationale for Server Side Includes is to provide a mechanism to deliver information that is repeated across a range of HTML documents from one source in order to minimize duplication and the risk of error. Enter the **#include** directive. The directives

<!--#include virtual="/includes/logo.html" -->

<!--#include file="logo.html" -->

will incorporate the contents of the file *logo.shtml* into the current document. If the document being included must in turn be processed to parse **SSI** directives just use the extension **SHTML** for that document.

<!--#include virtual="/scripts/myip.php" --> – will run the script *myip.php* and incorporate its results into the current document.

See **Notes** for the meaning of **file** and **virtual**.

The Set Directive

The **#set** directive provides a means of setting variables for later use.

Examples

- **<!--#set var="site" value="www.explainth.at" -->** assigns the string www.explainth.at to a variable called **site**.
- **<!--#set var="day" value="$DATE_LOCAL" -->**assigns the intrinsic variable DATE_LOCAL to a variable called **day**. The **$** sign is a way to tell the server that we are referring to a variable rather than a string bearing the name DATE_LOCAL.
- **<!--#set var="price" value="\$9.99" -->** assigns a the string $9.99 to the variable **price**. The reverse solidus, \ is used to escape the **$** character which would otherwise be treated as an attempt to refer to the "variable" **$9.99**.

Apache may at times fail to recognize a variable reference. If this happens, just wrap the reference in braces. e.g. **${DATE_LOCAL}.**

SSI Conditionals

SSI Conditionals provide a mechanism for the

Server Side Includes Quick Reference Card 1.01

conditional inclusion of information in the document being parsed. The generic form of an SSI Conditional is shown below

```
<!--#if expr="valueA" -->
statementA
<!--#elif expr="valueB" -->
statementB
...
<!--#else -->
statementDefault
<!--#endif -->
```

where **value** is the expression to be tested and **statement** is the information to be output to the document being processed if the condition is satisfied. **statement** can be another **SSI** directive such as **#echo** or plain text or **HTML** code. Only the **#if** and **#endif** parts are obligatory. In fact, Apache will not complain if **#endif** is left out. However, the parsed document is liable to be truncated before the **#if** if it resolves to **false**.

value must evaluate to **true** or **false**. It is possible to use the comparison operators **=, <. <=, >=, >** and **!=** as well as the logical operators **||** (OR) and **&&** (and). A comprehensive example is given below

```
<!--#set var="h" value="1" -->
<!--#set var="t" value="1" -->
<!--#set var="s" value="1" -->
<!--#if expr=""$h=1 && !($t=1 || $s=1)"" -->
Cook
<!--#elif expr="$t=1 && !($h=1 || $s=1)" -->
Take A Nap
<!--#elif expr="($h=1 && $t=1) && $s!=1" -->
Order A Pizza
<!--#elif expr="s=1 && !($t=1 || $h=1)" -->
Go To Bed
<!--#elif expr="$h=1 && $s=1 && $t=0" -->
Sleep but not tired? Make up your mind!
<!--#elif expr="($t=1 && $s=1) && $h=0" -->
Off To Bed!
<!--#else -->
Get Bored
<!--#endif -->
```

where **h=hungry**, **t=tired** & **s=sleepy**.

Server Configuration

Depending on the features offered by the version of Apache running on your server and the capabilities permitted by your host you may be able to edit the **.htaccess** file and change the way your server processes **SSI** directives.

.htaccess is merely a text file that controls how resources within a directory – and its sub directories – are accessed by the server. **.htaccess** entries relevant to controlling **SSI** are given below

●Enable/Disable Server Side Includes

This is done using the syntax

Options [+|-]Value1 [[+|-]Value2 [+|-]Value3...]

+ causes the value to be added to the current list of options while **–** causes it to be removed from the current list of options. For instance

Options +Includes would enable **SSI** processing

Options -IncludesNOEXEC would block the **#exec** directive while allowing other directives to be processed.

●Subjecting .EXT files to SSI Parsing

By default most servers will parse all files bearing the extension **SHTML** for **SSI d**irectives**.** This can prove to be an impediment if you need to use **SSI** in a **HTML** file that is currently on your servers and widely indexed by search engines such as Google. To force the server to parse **.HTML** files for **SSI d**irectives just add the following lines to your **.htaccess** file.

AddType text/html .html
AddHandler server-parsed .html

Notes

file indicates a file in the current path. **virtual** indicates a file path relative to the document root.

Color Key

red - **SSI** Directive output.

statement – **SSI** directive, plain text or HTML code.

[text] – text is optional

If you find this reference card useful please help us by creating links to our site http://www.explainth.at where you will find other quick reference cards and many other free programming resources.