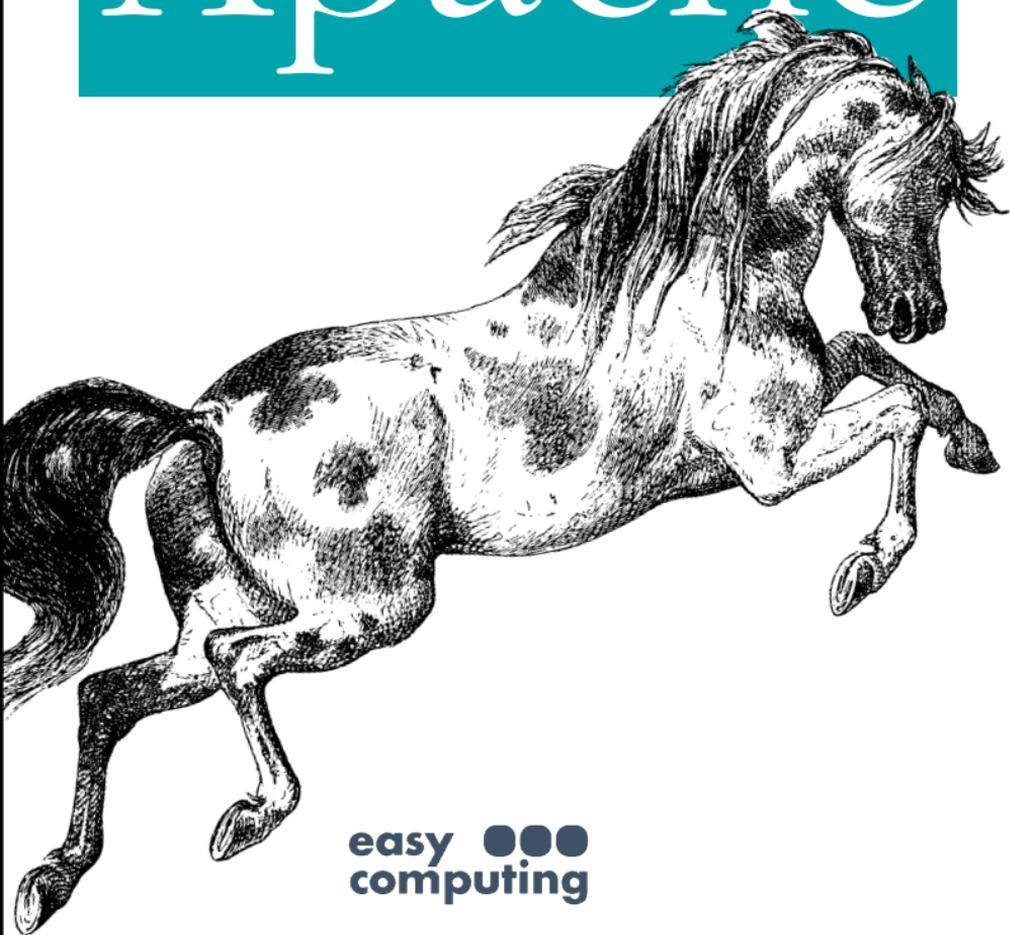


*For Apache Programmers  
& Administrators*

**Covers  
Apache 1.3.12**

# Apache



easy ●●●  
computing

**O'REILLY®**

*Andrew Ford*

---

# Apache

## *Pocket Reference*

Andrew Ford

easy ●●●  
computing

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo*

## **Apache Pocket Reference**

by Andrew Ford

Copyright © 2000 Ford & Mason Ltd. All rights reserved. Printed in the United States of America.

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

**Editor:** Gigi Estabrook

**Production Editor:** Madeleine Newell

**Cover Designer:** Ellie Volckhausen

### **Printing History:**

June 2000: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly & Associates, Inc. The use of the image of an Appaloosa horse in association with Apache is a trademark of O'Reilly & Associates, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly & Associates, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

### **Library of Congress Cataloging-in-Publication Data**

Ford, Andrew.

Apache pocket reference / Andrew Ford.--p. cm.

ISBN 1-56592-706-0

1. Apache (Computer file : Apache Group)--Handbooks, manuals, etc. I. Web sites. II. Computer programs--Handbooks, manuals, etc. I. Title.

TK5105.8885.A63 F67 2000

005.7'13769--dc21

00-033975

1-56592-706-0

[8/00]

[C]

---

# Table of Contents

Acknowledgments .....	1
Conventions .....	2
Starting and Stopping Apache .....	3
Apache Modules .....	5
Directory Layout .....	7
Support Utilities .....	8
General Configuration .....	9
Request Processing .....	30
Post-Read Request Phase .....	31
URL Translation Phase .....	33
Access Restrictions .....	41
MIME Type Checking Phase .....	47
Content Generation Phase .....	53
Request Logging Phase .....	73
strftime() Time Formats .....	76
Secure Versions of Apache .....	77
CGI Environment Variables .....	96

# Apache Pocket Reference

Apache is far and away the most widely used web server in the world, running on Windows NT as well as Unix and other platforms. Probably one of the most popular pieces of open source software in existence, it powers over half of the world's web sites and is still increasing its market share. Apache forms the basis of a number of commercial web servers, such as C2Net's Stronghold, Covalent's Raven, IBM HTTP Server powered by Apache, and Red Hat Secure Server.

Apache is at the leading edge of web server development and is often where new technologies are first implemented. It has a flexible architecture that allows independent developers to add their own functionality by way of modules, written either in C or, with the advent of *mod\_perl*, in Perl.

This pocket reference summarizes Apache's command-line options and configuration directives. It covers Apache version 1.3.12, but is applicable to other versions as well as to web servers derived from Apache.

For more information on Apache, visit the Apache Software Foundation web site at <http://www.apache.org>.

## Acknowledgments

I would like to thank Stas Bekman, Gigi Estabrook, Tony Finch, Martin Kraemer, Doug MacEachern, Catherine Mason, Lenny Muellner, Madeleine Newell, and all my reviewers.



# Conventions

Apache directive syntax uses a number of conventions:

constant width text

Denotes literal text

*constant width italic text*

Denotes dummy parameters

{ A | B }

Denotes alternatives

[text]

Denotes optional text

... Indicates that the previous element may be repeated

Apache configuration directives are described in a standard format, as shown here.

---

<b>DirectiveName</b>	module_name
DirectiveName arg1 arg2 arg3 ...	
<i>Contexts:</i> GVSF* (Override)	<i>Compatibility:</i> 1.x +
<i>Default:</i> default	

The top line gives the name of the directive, along with the name of the Apache module that implements the directive (the module name is omitted if the directive is implemented by the core module). The next line gives the directive syntax: the name of the directive is given in bold type followed by its arguments. Directives are case-insensitive, as are most arguments except those that refer to case-sensitive objects such as filenames. Subsequent lines give the list of contexts in which the directive may be used, version compatibility notes, and the default value for the directive (if applicable).

The list of valid contexts can contain one or more of the following abbreviations:



- G* Valid in global context; i.e., in the server configuration files outside of any virtual host or directory-type container
- V* Valid in a virtual host section
- S* Valid in a directory-type section (<Directory>, <Files>, and <Location>)
- F* Valid in a per-directory configuration file (named *.htaccess* by default)
- \** Indicates that the directive may be given more than once in a context

An override keyword is given in parentheses after the context abbreviations, if the directive can be used in a per-directory configuration file and is controlled by an AllowOverride directive category.

## Starting and Stopping Apache

Apache is usually set up to be started automatically when the system is booted, and stopped when the system is halted. On Unix systems this is normally handled by an *rc-file*. A shell script, `apachectl`, is provided with Apache to automate the process.

Should you need to start Apache manually, it takes the following command-line options:

- cdirective*  
Process *directive* after reading configuration files.
- Cdirective*  
Process *directive* before reading configuration files.
- ddirectory*  
Initial value for `ServerRoot`.
- Dparameter*  
Define parameter for <IfDefine> sections.

easy computing

-*ffile*

Configuration file (default is *conf/httpd.conf*).

-k {shutdown | restart}

Windows only; shutdown or restart Apache (1.3.3 +).

-l List compiled-in modules and exit.

-L List available configuration directives (provided by compiled-in modules) and exit.

-*nname*

Windows only; service name for Apache (1.3.7 +).

-S Show virtual host settings and exit.

-t Test syntax of configuration files, checking for the existence of document root directories, and exit.

-T Test the syntax of configuration files, but without checking for the existence of document root directories, and exit.

-v Print version and build date, and exit.

-V Show compilation settings and exit.

-X Run in single-process debug mode.

On Unix systems, Apache responds to the following signals sent to the parent process (the process ID of which is stored in the *pid* file):

TERM

Stops the server by causing the parent process to attempt to kill each of the child processes and then terminate.

HUP Restarts the server by causing the parent process to kill off each of the child processes and then reread the configuration files and spawn new child processes. Server statistics are reset to zero on a restart.

USR1

Initiates a graceful shutdown. Child processes are advised to terminate after processing the current request, or immediately if not currently servicing a request. The parent

**easy** ●●●  
**computing**

process rereads the configuration files and starts to spawn new child processes to maintain the appropriate number of server processes. Server statistics are not reset on a graceful restart.

On Windows NT, use the `-k` command-line option, or if Apache is installed as a service, use the `NET START` and `NET STOP` commands.

## Apache Modules

Apache modules may have differing statuses (which may change between releases):

### *BASE*

Base modules are included in the Apache distribution and are compiled in by default.

### *OPTIONAL*

Optional modules are included in the Apache distribution but are not compiled in by default.

### *EXPERIMENTAL*

Experimental modules are included in the Apache distribution but are not compiled in by default.

Roughly three dozen modules are included in the Apache distribution, as follows.

Module Name	Status	Description
<i>mod_access</i>	BASE	Access control based on client hostname or IP address
<i>mod_actions</i>	BASE	Executes scripts based on MIME type or request method
<i>mod_alias</i>	BASE	URL mapping
<i>mod_asis</i>	BASE	Canned responses from files that include HTTP headers
<i>mod_auth</i>	BASE	User authentication using text-based configuration files

Module Name	Status	Description
<i>mod_auth_anon</i>	OPTIONAL	Anonymous authentication
<i>mod_auth_db</i>	OPTIONAL	User authentication using Berkeley DB files
<i>mod_auth_dbm</i>	OPTIONAL	User authentication using DBM files
<i>mod_auth_digest</i>	EXPERIMENTAL	MD5 authentication
<i>mod_autoindex</i>	BASE	Automatic directory listings
<i>mod_cern_meta</i>	OPTIONAL	Support for HTTP header metafiles
<i>mod_cgi</i>	BASE	Executes CGI scripts
<i>mod_digest</i>	OPTIONAL	MD5 authentication
<i>mod_dir</i>	BASE	Basic directory handling
<i>mod_env</i>	BASE	Passing of environments to CGI scripts
<i>mod_expires</i>	OPTIONAL	Applies Expires headers to resources
<i>mod_headers</i>	OPTIONAL	Adds arbitrary HTTP headers to resources
<i>mod_imap</i>	BASE	Imagemap file handler
<i>mod_include</i>	BASE	Server-side include documents
<i>mod_info</i>	OPTIONAL	Server configuration information
<i>mod_isapi</i>	OPTIONAL	Windows ISAPI Extension support
<i>mod_log_agent</i>	OPTIONAL	Logging of the client user agent (deprecated—use <i>mod_log_config</i> instead)
<i>mod_log_config</i>	BASE	User-configurable logging
<i>mod_log_referer</i>	OPTIONAL	Logging of the HTTP referer field (deprecated—use <i>mod_log_config</i> instead)
<i>mod_mime</i>	BASE	Determines document types using file extensions

Module Name	Status	Description
<i>mod_mime_magic</i>	OPTIONAL	Determines document types in the manner of the Unix <code>file</code> command
<i>mod_mmap_static</i>	EXPERIMENTAL	Maps files into memory for faster delivery
<i>mod_negotiation</i>	BASE	Content negotiation
<i>mod_proxy</i>	OPTIONAL	Caching proxy
<i>mod_rewrite</i>	OPTIONAL	Powerful URI-to-filename mapping using regular expressions
<i>mod_setenvif</i>	BASE	Sets environment variables based on client information
<i>mod_so</i>	OPTIONAL	Support for loading modules at runtime
<i>mod_speling</i>	OPTIONAL	Automatically corrects minor mistakes in URLs
<i>mod_status</i>	OPTIONAL	Server status display
<i>mod_unique_id</i>	OPTIONAL	Generates a unique identifier for every request
<i>mod_userdir</i>	BASE	User home directories
<i>mod_usertrack</i>	OPTIONAL	User tracking using cookies
<i>mod_vhost_alias</i>	OPTIONAL	Dynamically configured mass virtual hosting

Further modules can be found on the Apache Module Registry (<http://modules.apache.org>).

## Directory Layout

Conventions for Apache directory structuring vary between releases and distributions. The source distribution will install in the following subdirectories under `/usr/local/apache`, unless configured differently.



## Client certificate information

Client certificate environment variables are set only if client authentication is enabled.

SSL\_CLIENT\_CERT

PEM-encoded certificate (*mod\_ssl* if the `ExportCertData` option is set)

SSL\_CLIENT\_CERT\_CHAIN*n*

PEM-encoded certificate *n* in certificate chain (Apache-SSL if the `SSLExportClientCertificates` directive is specified)

SSL\_CLIENT\_M\_VERSION

Certificate version (*mod\_ssl*)

SSL\_CLIENT\_M\_SERIAL

Certificate serial number (*mod\_ssl*)

SSL\_CLIENT\_SERIALNUM

Certificate serial number (IHS)

SSL\_CLIENT\_A\_SIG

Algorithm used for signature of certificate (*mod\_ssl*)

SSL\_CLIENT\_SIGNATURE\_ALGORITHM

Algorithm used for signature of certificate (*mod\_ssl*♦, Stronghold)

SSL\_CLIENT\_A\_KEY

Algorithm used for public key of certificate (*mod\_ssl*)

SSL\_CLIENT\_V\_START

Start time of certificate validity (*mod\_ssl*)

SSL\_CLIENT\_CERT\_START

Start time of certificate validity (*mod\_ssl*♦, Stronghold)

SSL\_CLIENT\_V\_END

End time of certificate validity (*mod\_ssl*)

SSL\_CLIENT\_CERT\_END

End time of certificate validity (*mod\_ssl*♦, Stronghold)

easy computing

SSL\_CLIENT\_VERIFY  
Validity of certificate (*mod\_ssl*)

SSL\_CLIENT\_S\_DN  
Subject DN (*mod\_ssl*)

SSL\_CLIENT\_DN  
Subject DN (*mod\_ssl*♦, Apache-SSL, IHS, Stronghold)

SSL\_CLIENT\_S\_component  
Component of subject DN (*mod\_ssl*)

SSL\_CLIENT\_component  
Component of subject DN  
(*mod\_ssl*♦, Apache-SSL, IHS, Stronghold)

SSL\_CLIENT\_I\_DN  
Issuer DN (*mod\_ssl*)

SSL\_CLIENT\_IDN  
Issuer DN (*mod\_ssl*♦, Apache-SSL, IHS)

SSL\_CLIENT\_I\_component  
Component of issuer DN (*mod\_ssl*)

SSL\_CLIENT\_Icomponent  
Component of issuer DN (*mod\_ssl*♦, Apache-SSL, IHS)

SSL\_CLIENT\_CERT\_BODY  
Client certificate as a string (IHS)

SSL\_CLIENT\_CERT\_BODY\_LEN  
Length of the client certificate string (IHS)

SSL\_CLIENT\_SESSSIONID  
Session ID (IHS)

SSL\_CLIENT\_NEW\_SESSSIONID  
Set to "TRUE" if the session ID is new (IHS)

SSL\_CLIENT\_SERIALNUM  
Client certificate serial number (IHS)

easy   
computing