Web^{Ten} User's Guide

New Dimensions in Personal Workstation Technology

1123 Chapala Street, Santa Barbara, CA 93101
TEL 805-963-6983 • FAX 805-962-8202
info@tenon.com • http://www.tenon.com

Tenon, Web[Ten] and the Tenon logo are trademarks of Tenon Intersystems.
Apple, Macintosh, Power Macintosh and OpenTransport are registered trademarks of Apple Computer, Inc.
PowerPC is a trademark of International Business Machines Corporation.

Web[Ten] includes software developed by the Apache Group for use in the Apache HTTP Server Project (http://www.apache.org/)
and Squid (http://www.squid-cache.org/).

TABLE OF CONTENTS

1.0 Web[Ten] — A New Standard in Web Service 1

1.1 Web[Ten] Architecture 3
1.1.1 Apache Design 3
1.1.2 Squid Object Caching 4
1.1.3 Enhanced Networking 5
1.2 Enhanced Services 7
1.2.1 Virtual Hosts 7
1.2.2 Header-Based Virtual Hosting 8
1.2.3 Fast Storage 8
1.2.4 Secure Socket Layer 9
1.2.5 FTP Service 9
1.2.6 NFS Service 9
1.2.7 DNS Service 9
1.2.8 CRON Service 9
1.2.9 Proxy Services 10
1.3 Advanced HTTP Features 10
1.3.1 Host Name Identification 10
1.3.2 Content Negotiation 11
1.3.3 Keep-Alive Connections 11
1.3.4 "As-Is" Documents 11
1.3.5 "PUT" and "DELETE" Requests 11
1.3.6 Chunked Transfers 12
1.3.7 Byte Ranges 12
1.4 Scripting 12
1.4.1 CGI-Based Scripts 12
1.5 Server APIs 14
1.5.1 Apache APIs 14
1.5.2 WSAPIs 14
1.5.3 Installing Plug-Ins 14
1.6 Server-Side Includes 14
1.7 Database Interfaces 15
1.8 Directives 15
1.9 Logging 15

1.10 Special Utilities 16
1.10.1 Fast File First Aid 16
1.10.2 NoFinder 16
1.10.3 Unix <-> Text 17
1.11 Transitioning to Mac OS X 17

2.0 Installing WebTen 19

2.1 Before You Begin 19
2.2 Installation Guide 20
2.3 Re-Installing 23
2.4 Uninstalling 24
2.5 WebTen Extensions 25
2.5.1 ht:/ /Dig 25
2.5.2 WEBmail 25
2.5.3 WebEvent 26
2.5.4 WebCrossing 26
2.5.5 HTML/OS 26
2.5.6 PHP 3 26

3.0 Quick Start Guide 27

3.1 Launching 27
3.1.1 WebTen Application Heap 28
3.1.2 Setting the WebTen Administrator's Password 28
3.2 Connecting 29
3.2.1 Connect to the WebTen Home Page 29
3.2.2 Connect to the Administration Server 30
3.2.3 Try the Examples 31
3.2.4 Add Your Web Pages 31
3.3 Quitting 34
3.4 Web Serving Resources 34

4.0 WebTen Menus 37

4.1 File Menu 37
4.2 Preferences 38
4.2.1 Host and Domain Name 39

4.2.2 Time Zone 39
4.2.3 DNS IP Address 39
4.2.4 Launching Web[Ten] On Startup 40
4.2.5 Replacing OpenTransport 40
4.2.6 Networking with OpenTransport 42
4.2.7 Enabling Web[Ten] Services 42
4.2.8 Testing Web[Ten] in Loopback Mode 42
4.3 Edit Menu 45
4.4 Admin Menu 46
4.4.1 Set Admin Password 47
4.4.2 Change License 48
4.4.3 Start / Stop Web Server 48
4.4.4 Start / Stop Admin Server 49
4.4.5 Cache Status 49
4.4.6 System Status 52
4.4.7 Web Server Status 54
4.4.8 Flush Cache 56
4.4.9 Shell Window 56
4.4.10 Save Display 56
4.5 Log Menu 57
4.5.1 Clear Log Data 57
4.5.2 Reset Log Config 57
4.5.3 Display Access Log 58
4.5.4 Display Error Log 58
4.5.5 Display Plug-In Msgs 58

5.0 Web[Ten] Administration 59

5.1 The Administration Server 59
5.1.1 Starting the Administration Server 59
5.2 Web[Ten] Administration Server 61
5.3 Navigating the Administration Pages 63
5.3.1 Types of Information Fields 63
5.3.2 Making Changes 64
5.3.3 Adding Entries 64
5.3.4 Removing Entries 64

5.3.5 Resetting Entries 65
5.3.6 Inheritance 65

6.0 System-Wide Configuration 67

6.1 Server Defaults 68
6.1.1 ServerAdmin 70
6.1.2 DirectoryIndex 70
6.1.3 ErrorLog 71
6.1.4 TransferLog 71
6.1.5 LogFormat 71
6.1.6 ScriptLog 72
6.1.7 HostnameLookups 72
6.1.8 Plug-In / Apple CGI Settings 73
6.1.9 Error File Settings 75
6.1.10 Alias Settings 76
6.1.11 Redirect Settings 78
6.2 Plug-In Administration 79
6.3 Proxy Settings 80
6.3.1 ProxyRequests 80
6.3.2 CacheSize 81
6.3.3 CacheGcInterval 81
6.3.4 CacheMaxExpire 81
6.3.5 CacheLastModifiedFactor 81
6.3.6 CacheDefaultExpire 82
6.3.7 NoCache 82
6.3.8 Remote Proxies 83
6.3.9 Proxy Access 85
6.4 Server Controls 87
6.4.1 Start/Stop Server 88
6.4.2 Server Status 88
6.4.3 Cache Status 88
6.4.4 Server Info 88
6.4.5 Restart Server 88
6.4.6 Flush Cache 89
6.4.7 Messages 89

6.4.8 Startup Log 89
6.4.9 System Errors 89
6.4.10 Config Log 90
6.4.11 Web[Ten] Version Number 90
6.5 Action Handlers 91
6.5.1 Configuring Plug-In Actions 92
6.6 MIME Extensions 94
6.6.1 The MIME Typing System 95
6.7 MIME Languages 97
6.8 MIME Encodings 98
6.9 Users 99
6.10 Groups 101
6.10.1 Users in Group 102
6.10.2 Import and Export 103
6.11 Cache Settings 105
6.11.1 AcceleratorCache 106
6.11.2 supercache_enable 106
6.11.3 cache_mem 106
6.11.4 cache_swap 106
6.11.5 swap_level1_dirs 106
6.11.6 swap_level2_dirs 106
6.11.7 cache_stoplist 107
6.12 Advanced Settings 107
6.12.1 StartServers 108
6.12.2 MaxClients 108
6.12.3 MaxSpareServers 108
6.12.4 MinSpareServers 108
6.12.5 MaxRequestsPerChild 109
6.12.6 Port 109
6.12.7 TimeOut 109
6.12.8 KeepAlive 110
6.12.9 MaxKeepAliveRequests 110
6.12.10 KeepAliveTimeout 110
6.12.11 PITCPOpenTimeout 110
6.12.12 ACGIReplyTimeout 110

6.12.13 ACGIEventExtensions 111
6.12.14 MyopicPlugInMode 111
6.13 Direct Access to Configuration Files 113
6.13.1 Macintosh File Creators and File Types 113

7.0 Virtual Hosts 115

7.1 Virtual Hosts Table 115
7.1.1 Adding Virtual Hosts 116
7.1.2 Deleting Virtual Hosts 116
7.2 Virtual Host Configuration 117
7.2.1 VirtualHost 118
7.2.2 SSLSecurity 119
7.2.3 DocumentRoot 119
7.2.4 ServerAdmin 119
7.2.5 ServerName 120
7.2.6 ServerAlias 120
7.2.7 ServerPath 120
7.2.8 DirectoryIndex 121
7.2.9 ErrorLog 121
7.2.10 TransferLog 121
7.2.11 LogFormat 123
7.2.12 HostnameLookups 124
7.3 Plug-In / Apple CGI Defaults 125
7.3.1 WSAPIRequests 125
7.3.2 ACGIBinOnly 126
7.3.3 RequestFiltering 126
7.3.4 PIAccessControl 126
7.3.5 PreProcessor 126
7.3.6 PIPreProcessing 126
7.3.7 PostProcessor 127
7.3.8 PIPostProcessing 127
7.3.9 WSAPIPostArgSize 127
7.3.10 SSLCertificateFile 128
7.3.11 SSLCertificateKeyFile 128
7.4 Error File, Alias, and Redirect Settings 128

7.5 Folder Contents 129
7.5.1 Files 130
7.5.2 Folders 130
7.6 Access Controls 131
7.6.1 Domain Name-Based Restrictions 132
7.6.2 Realm-Based Requirements 134
7.6.3 MIME Type Overrides 136
7.6.4 Action Handler Overrides 137

8.0 Secure Socket Layer (SSL) 139

8.1 Server Certificates 139
8.1.1 Obtaining a Server Certificate 140
8.2 SSL Settings 141
8.2.1 Common Name 142
8.2.2 Organization Name 142
8.2.3 Organizational Unit 142
8.2.4 Locality 142
8.2.5 State or Province 142
8.2.6 Country Code 142
8.2.7 Email Address 142
8.2.8 Generating a CSR 143
8.3 Enabling SSL 145
8.4 Ciphers 145
8.4.1 SSL Cipher Restrictions 146
8.5 Using WebTen with Multiple Certificates 147
8.6 Self-Signed Certificates 147
8.7 Safeguarding SSL Keys and Certificates 148
8.7.1 Exporting SSL Files 148
8.7.2 Importing SSL Files 150

9.0 FTP Service 151

9.1 File Encodings 151
9.2 Downloading Files via FTP 152
9.3 Uploading Files via FTP 153
9.3.1 Uploading and Executing CGI Scripts 154

9.4 FTP Settings 155
9.4.1 FTP Status 155
9.4.2 Anonymous 156
9.4.3 User-Pass 157
9.4.4 Limit 157
9.4.5 Logging 157
9.5 Virtual Anonymous FTP Service 158
9.5.1 Host Header-Based Anonymous FTP 159

10.0 NFS Service 161
10.1 Configuring the NFS Server 161
10.2 NFS User and Group Numbers 161
10.3 NFS Settings 162
10.3.1 NFS Server 162
10.3.2 Server Path 162
10.3.3 Local Path 163
10.3.4 Read Only 163

11.0 Domain Name System (DNS) 165
11.1 Virtual Hosting Requirements 166
11.2 WebTen Preferences and DNS 167
11.2.1 Running WebTen with an Unconfigured DNS Server or without DNS 168
11.2.2 Running WebTen with DNS 169
11.3 WebTen Domain Name Server Administration 171
11.3.1 DNS Primary Zone 172
11.3.2 DNS New Primary Zone Page 182
11.3.3 DNS New Secondary Zone Page 183
11.3.4 DNS Secondary Zone 184
11.3.5 Deleting DNS Zones 186
11.4 DNS Database Files 187
11.5 DNS Manager CGI 187
11.6 Registering your DNS Zones 188

12.0 Clock Service (Cron) 189
12.1 Starting Cron 190

12.2 Example crontab File 190

13.0 Using CGIs 191

13.1 Shell CGIs 191
13.1.1 Required Shell Script Content 192
13.1.2 Printenv.sh Example 193
13.1.3 Shell Variables 194
13.2 Perl CGIs 194
13.2.1 Required Script Content 195
13.2.2 Printenv.pl Example 196
13.2.3 Environment Variables 197
13.3 C Language CGIs 197
13.3.1 Printenv.c Example 198
13.4 Fast CGI 201

14.0 WEBmail 202

14.1 Using WEBmail as an e-mail Client 202
14.2 Adding a WEBmail mailbox 204
14.3 Customizing WEBmail 207

15.0 ht://Dig 208

15.1 Build the Web[Ten] Search Engine Index File 209
15.2 Test the Web[Ten] Search Engine Database 211
15.3 Multiple Virtual Hosts 211

16.0 Plug-Ins and Apache Modules 212

16.1 Plug-Ins 212
16.1.1 Installing Plug-Ins 212
16.2 Apache Modules 213
16.2.1 Installing Apache Modules 213

Appendix A A-1

Appendix B B-1

Appendix C C-1

Appendix D D-1

Appendix E E-1

Appendix F F-1

Appendix G G-1

INDEX I-1

LIST OF FIGURES

Figure 1:	WebTen Architecture	2
Figure 2:	Installer Icon	20
Figure 3:	Installer Instructions	20
Figure 4:	Installer Options	21
Figure 5:	Installer Destination Folder	22
Figure 6:	Installer Progress	22
Figure 7:	Uninstall	24
Figure 8:	WebTen Folder	27
Figure 9:	Startup Status Window	28
Figure 10:	WebTen Home Page (*default.html*)	30
Figure 11:	WebTen Folder	31
Figure 12:	Preferences Window	38
Figure 13:	Replacing OpenTransport	41
Figure 14:	The Admin Menu	46
Figure 15:	Set Admin Password Window	47
Figure 16:	Input License Number Window	48
Figure 17:	Cache Status Window	49
Figure 18:	System Status Window	52
Figure 19:	Web Server Status Window	54
Figure 20:	WebTen Administration Server	61
Figure 21:	System-Wide Configuration Table	67
Figure 22:	Server Defaults Table	69
Figure 23:	Error Files Table	75
Figure 24:	Alias Settings Table	77
Figure 25:	Redirect Settings Table	78
Figure 26:	WebTen Plug-In Administration Table	79
Figure 27:	Proxy Settings Table	80
Figure 28:	Remote Proxies	83
Figure 29:	Proxy Access	85
Figure 30:	Server Controls Table	87
Figure 31:	Action Handlers Table	91
Figure 32:	Configuring a Plug-In Action	92
Figure 33:	Adding a Plug-In Extension	93
Figure 34:	MIME Extensions Table	94
Figure 35:	MIME Types and File Extensions	96
Figure 36:	MIME Languages Table	97

Figure 37:	MIME Encodings Table	98
Figure 38:	Users Table	100
Figure 39:	Groups Table	101
Figure 40:	Users in Group Table	102
Figure 41:	Import and Export Users and Groups	103
Figure 42:	Cache Settings Table	105
Figure 43:	Advanced Settings Table	107
Figure 44:	Virtual Hosts Table	115
Figure 45:	Virtual Host Configuration Table	118
Figure 46:	Virtual Host Configuration Table	125
Figure 47:	Folder Contents Table	129
Figure 48:	Sub-Folder Contents	130
Figure 49:	Access Controls Table	131
Figure 50:	Domain Name-Based Restrictions	132
Figure 51:	Realm-Based Requirements	134
Figure 52:	MIME Type Overrides	136
Figure 53:	Action Handler Overrides	137
Figure 54:	SSL Cipher Restrictions	141
Figure 55:	Certificate Signing Request Information	143
Figure 56:	Enabling SSL	145
Figure 57:	SSL Cipher Restrictions	146
Figure 58:	FTP Settings Table	155
Figure 59:	NFS Settings Table	162
Figure 60:	DNS Settings Table	171
Figure 61:	Primary Zone Page	172
Figure 62:	New Host Page	173
Figure 63:	Adding Load Balancing Records	174
Figure 64:	Adding an Alias	175
Figure 65:	A Configured Primary Zone	179
Figure 66:	Reverse Lookup Table	180
Figure 67:	Start of Authority	181
Figure 68:	New Primary Zone	182
Figure 69:	New Secondary Zone	183
Figure 70:	Secondary Zone	185
Figure 71:	Deleting a Zone	186
Figure 72:	WEBmail Login	203
Figure 73:	Choose WEBmail account password	207
Figure 74:	Default Indexing Options	209

Figure 75: All Indexing Options 210
Figure 76: Apache Modules and Plug-Ins 212
Figure 77: Included Apache Modules 213

1.0 Web[Ten] — A New Standard in Web Service

Web[Ten] is a powerful Web server for Power Macintosh. Web[Ten] is based on the most popular Web server in use today — the highly acclaimed Apache Web server, developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

Tenon's unique technology enables the Apache Web server to run in an "invisible UNIX wrapper" on the Macintosh Operating System (Mac OS). Tenon's "UNIX virtual machine" creates a preemptive multitasking environment for the industrial-strength, secure Apache Web server. Apache has been extended in a number of ways to give Macintosh users the best of both the Macintosh and UNIX worlds.

Tenon has extended Apache to support WebSTAR Plug-In APIs and AppleScript CGIs. So, with Web[Ten] you can use standard internet plug-ins, as well as Apple-specific plug-ins. You can use shell, Perl, and binary CGIs, as well as Apple CGIs. In fact, Web[Ten] supports the widest selection of plug-ins and CGIs of any known Web server.

Web[Ten] is easy to install, easy to set up, and easy to administer. The Web-accessible interface allows local or remote administration using any Web browser. Web[Ten] can be configured from anywhere in the world. No special administration tools are needed on the remote site. No special administration agent is required on the server.

Tenon's powerful TCP stack lets Web[Ten] support "true" internet-style virtual hosting. Apple's OpenTransport can be used in place of Tenon's TCP stack, or in combination, giving users the most flexibility for their internet and intranet servers.

Other features unique to Web[Ten] include the ability to use SSL to support secure transmission for multiple virtual hosts on a single machine, integrated FTP and NFS services for uploading or offloading Web content, and built-in domain name service.

Tenon has taken advantage of advanced caching techniques from the Harvest ARPA research project. A derivative software package called "Squid" has been incorporated into WebTen. Integration of Squid with Apache provides high-performance object caching that further improves WebTen's top performance. By using Squid object caching, WebTen can process 65,000 hits per minute, or more than 90 million hits per day.

WebTen fuses the world's most popular Web creation platform with the world's most popular Web server. Apple and Apache — a new standard in Web service for Macintosh.

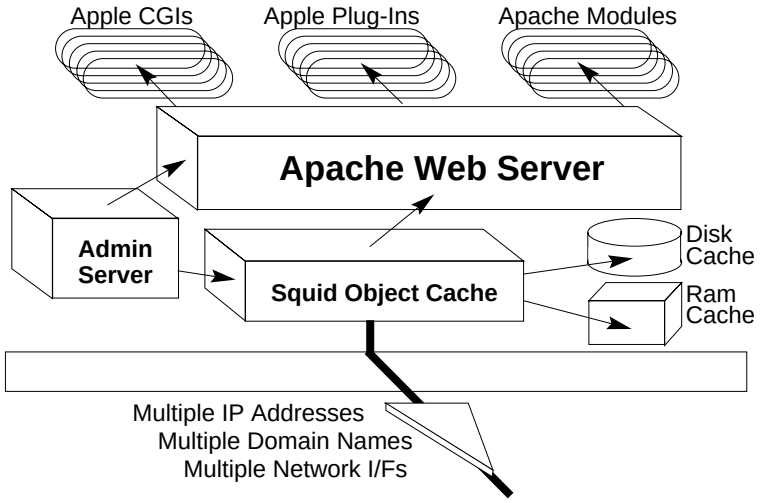


Figure 1: WebTen Architecture

1.1 WebTen Architecture

WebTen is more than simply a port of Apache software to the Macintosh. WebTen extends several Apache and Macintosh system components and, at the same time, brings new and exciting state-of-the-art networking technology to the Macintosh.

Creating a new standard in Macintosh Web service did not simply revolve around the creation of a new and more powerful Web server. It required a series of new or alternative system components, in addition to Apache, to make the system more powerful, flexible, and easy to configure.

The first step in creating WebTen was porting Apache to the Macintosh. This was made possible by Tenon's "UNIX virtual machine" technology, making an Apache port to Macintosh a reality for the first time. Apache was then extended in two dimensions. First, an Administration Server was created to support Web browser-based configuration and maintenance. This gave Apache a point-and-click configuration capability. Next, Apache was extended to support Apple WSAPI-defined CGI and plug-in extensions. As a result, Apple CGIs and plug-ins that work on other Macintosh Web servers will simply "drop in" to WebTen. (See "Figure 1: WebTen Architecture".)

Tenon enhanced Apache's performance by using state-of-the-art caching techniques and fine-tuning WebTen's TCP stack, and extended WebTen's functionality by including a key set of internet and networking services.

1.1.1 Apache Design

A basic understanding of the Apache architecture will enable you to appreciate the power of Apache and fully benefit from the various WebTen displays. Older UNIX Web servers (namely the ones from NCSA and CERN) generated clone servers to handle incoming HTTP requests. The main Web server listened for incoming requests on the *httpd* port and generated a clone server for each request. Under this setup, there was no way to control the number of clones other than by limiting the total number of processes supported by the system.

The Apache designers had a better idea — a large body of persistent "children" running in parallel, coordinated by a parent process. The parent process is able to initiate and terminate children and to control the number of children that are alive. One of Apache's strengths is its tunability. A Webmaster can stipulate the number of persistent children at system startup, and control the number of children that are available at any point in time to respond to requests.

The number of allocated children threads is dynamically set by Web[Ten] as a function of peak loads from system startup. The Web[Ten] Web Server Status Window displays the allocated children threads versus the active children threads at any instance in time. (See section "4.4.7 Web Server Status".)

1.1.2 Squid Object Caching

Web[Ten] incorporates an object cache module that dramatically increases overall performance. It is well known that many Web servers serve a relatively small number of pages many times. Rather than perform full Web service calculations and production for each page, as all Apache Web servers do, Web[Ten] uses an object cache to intercept repeated requests for the same page and to produce a copy of the page directly from a local memory or disk cache. Another advantage provided by internet object caching is a way to store requested internet objects (URLs, FTP requests, gopher requests) on a system closer to the requesting site than the source. Web browsers can use the local cache as a "proxy server", thereby reducing access time and bandwidth consumption. This technique greatly reduces system overhead and increases performance.

The Web[Ten] object cache module is based on the Squid Object Cache software (`http://squid-cache.org`). Squid is derived from the ARPA-funded Harvest project. The Harvest cache has been shown to outperform other popular internet cache implementations by an order of magnitude. In addition, pairing the Harvest cache with HTTP servers has been shown to increase the server throughput by an order of magnitude. Web[Ten]'s object cache is based on this state-of-the-art technology.

The Harvest project spawned a number of commercial and research efforts. For example, Netscape's Catalog Server, a key component of the Netscape Suite Spot, is based on the Harvest design. Squid, a publicly available body of software developed by a community of world-wide internet researchers, incorporates the Harvest cache software. Duane Wessels of the National Laboratory for Applied Network Research (funded by the National Science Foundation) leads the code development.

The Squid Object Cache module sits between the Web[Ten] Apache Web server module and the Macintosh network interfaces, where it is able to monitor inbound network requests for Web service. Each request is examined for the possibility of dynamically caching the Apache response in the local object cache. As time passes, the object cache module contains an increasingly larger variety of previously requested Web data. When a new request is found that matches a previously cached request, the object cache responds directly, without involving the Web[Ten] Apache Web server module. Using this technique, requests for many Web pages can be given a priority response directly from the object cache. This has the effect of greatly increasing the overall operation of the Apache module, freeing it to process more complex or dynamic requests in parallel.

Unlike traditional caching software, Squid handles all requests in a single, non-blocking, I/O-driven process. Squid keeps meta data and especially hot objects cached in RAM. As Web[Ten] starts, Squid caching is enabled by default. Measurement of the Squid cache, like the Harvest cache, suggests an order of magnitude performance improvement over standard Web servers and other caching software. The Squid cache serves as an *httpd* accelerator for Web[Ten].

1.1.3 Enhanced Networking

Web[Ten] includes an alternative TCP/IP networking protocol implementation. Incorporation of the new protocol software provides for enhanced internet performance, increased protection from "hacker" attacks, and support for multiple, simultaneous IP, domain name and network hardware interfaces on a single Macintosh.

The Web[Ten] TCP stack can be used to replace the standard Apple OpenTransport implementation, or it can be run in parallel with OpenTransport on the same Macintosh.

Using the Web[Ten] TCP stack increases performance for two reasons — (1) Tenon's TCP stack includes no built-in limits on the number of simultaneous TCP connections, and (2) Tenon's stack, by including a wider range of packet delay and packet loss profiles than OpenTransport, has been tuned to work extremely well over internet links.

"Hacker" protection has been added to the Tenon TCP stack to prevent a number of "denial-of-service" attacks, such as the "ping-of-death" and SYN attacks, that have recently crippled some Internet Service Providers.

Web[Ten]'s TCP includes support for multiple network hardware interfaces. This allows a single Macintosh to be dual-homed to a single physical network for packet load balancing, or a single Macintosh to be connected to multiple network backbones for robustness and availability.

In addition, Web[Ten]'s TCP allows multiple IP addresses to be assigned to a single physical interface, so that a single Macintosh may be known by different internet addresses. Web[Ten] has support for multiple domain names, all on the same Macintosh. The multiple network interfaces, multiple internet addresses, and multiple domain names can be used in combination with each other, making Web[Ten] the first Macintosh-based Web server that supports true virtual hosting — simultaneous access to different top-level Web server URLs on a single Mac. The flexibility of Apache with respect to Web URLs makes Web[Ten] a great vehicle for supporting Macintosh-based Web service.

1.1.3.1 Dual TCP Stacks

Some Macintosh TCP-based applications require OpenTransport. Since Web[Ten]'s advanced TCP stack is needed to support IP-based virtual hosting, some Webmasters may want to configure their Web[Ten] machines to run dual protocol stacks. It is possible to run Web[Ten]'s TCP and OpenTransport simultaneously on a single Macintosh. The idea is to configure OpenTransport with one IP address and Web[Ten]'s TCP/IP with a second IP address. This allows Macintosh applications to use OpenTransport, and still lets Web[Ten] use its own stack for multihoming and performance advantages. See section "4.2.5 Replacing OpenTransport" for more information.

1.2 Enhanced Services

Web[Ten] includes a native fast file system to provide fast storage. Web[Ten]'s fast storage enables efficient storage and retrieval of thousands of small files on the Macintosh file system. Both the Apache Server and the Squid Object Cache take advantage of this fast file system to give Web[Ten] enhanced performance.

Web[Ten] includes Secure Socket Layer (SSL) support, integrated file transfer, integrated network file service, and integrated domain name service. These services work in concert in Web[Ten]'s preemptive multitasking environment to create a robust, top-performing, full-featured, secure Web server.

1.2.1 Virtual Hosts

Web[Ten] gives you the ability to set up Web service for multiple organizations on a single machine. Using a single machine to "host" Web pages for different organizations will reduce hardware costs and administrative costs. Even within a single organization, you may want to establish multiple Web "sites", each with their own top-level URLs.

For example, Web[Ten] running on host "`www.doodads.com`" could be configured to answer requests for domains "`widgets.com`" and "`trinkets.com`". That way someone using a browser could access the "`doodads`" host in any of the following ways:

```
http://www.doodads.com
http://www.widgets.com
http://www.trinkets.com
```

Note that even though all of these so-called "sites" are hosted on a single machine, each logical host has a first-class URL address.

This setup is much more desirable than the scheme that has been used thus far on Macintosh:

```
http://www.doodads.com
http://www.doodads.com/widgets
http://www.doodads.com/trinkets
```

Having the ability to establish multiple first-class URLs on a single physical

machine is accomplished by creating multiple virtual hosts. Each virtual host can have a unique IP address and a unique domain name, or simply a unique domain name.

The TCP stack that comes with WebTen allows your Macintosh to be configured with multiple network interfaces (i.e., multihomed), each with a unique internet (IP) address. WebTen also allows a Macintosh with one network interface (i.e., single-homed) to appear to be multi-homed through a technique called IP aliasing. By using IP aliasing and domain name aliasing, you can create any number of virtual hosts on a single Macintosh, thus letting a single physical machine behave as if it were several different hosts.

With WebTen, you can set up any number of IP addresses, and each IP address can be assigned any number of domain names. This technique of IP aliasing and domain name aliasing is built into Tenon's TCP stack. Therefore, if you want this capability, you must use the TCP stack that comes with WebTen (see section "4.2.5 Replacing OpenTransport").

To set up virtual hosts, see section "7.1 Virtual Hosts Table" and read section "11.2 WebTen Preferences and DNS" to understand the relationship between virtual hosts and domain name service.

1.2.2 Header-Based Virtual Hosting

The HTTP/1.1 protocol supports a new feature called "header-based virtual hosts". This feature is supported in WebTen. Nowadays, about 95% of all Web browsers support header-based virtual hosts. If you decide to use this, you may use either Tenon's TCP stack or OpenTransport to support virtual hosting.

The setup for this kind of virtual hosting is exactly the same as for IP-based virtual hosting.

1.2.3 Fast Storage

WebTen includes a native fast file system. WebTen's fast storage provides a means to efficiently store and retrieve thousands of small files on the Macintosh File System. Portions of the Apache server and the Squid Object Cache take advantage of this fast file system to give WebTen enhanced performance.

1.2.4 Secure Socket Layer

WebTen incorporates version 3.0 of the Secure Socket Layer (SSL) protocol to encrypt Web server transmissions. Because WebTen is the only Macintosh Web server to support IP-based virtual hosts, WebTen's SSL can support secure transmissions on a per virtual host basis — true multihoming SSL.

1.2.5 FTP Service

WebTen includes FTP service as an integrated component of its Web service, providing high-performance file uploads to Webmasters and Web service providers' customers.

1.2.6 NFS Service

WebTen includes NFS capabilities that allow it to mount NFS volumes from any NFS server. These volumes can then be published within the content hierarchy of the WebTen Web server. NFS servers can contain the Web pages for an entire Web site, a set of specific virtual hosts, or simply a component of a virtual host.

The WebTen NFS client service is compatible with any NFS server implementation. Support for read-only access to the NFS volumes is also provided.

1.2.7 DNS Service

WebTen's domain name service (DNS) is based on the latest internet technology, with improved performance and security. You can use WebTen's built-in DNS as your primary domain name service, as your secondary domain name service in conjunction with any other available DNS service, or simply continue to use your existing domain name service.

1.2.8 CRON Service

WebTen includes an integrated timer-driven execution manager, the popular UNIX *Cron* utility. Using *Cron*, Web managers can specify a schedule for periodic execution of any number of scripts. These scripts can generate Web usage reports or perform various maintenance routines automatically. See section "12.0 Clock Service (Cron)" for more information and a sample *Cron* script.

1.2.9 Proxy Services

A proxy server is one that is able to respond to requests for documents that are on other servers. The simplest use of a proxy is as a document cache. Remote documents can be copied to the hard disk of the proxy server. This not only speeds up access time, but also reduces network congestion. More sophisticated uses of proxies involve security and filtering. A trusted proxy can be positioned inside a firewall. That way employees deal only with the proxy, and the proxy is given special privileges to access documents beyond the firewall. A school can give students internet access via a proxy Web server, with built-in restrictions based on key words or domain names.

Web[Ten] includes two kinds of proxy services — the Apache proxy module and the Squid proxy component. The Apache proxy module can be configured via the Web[Ten] Administration Server (see section "6.3 Proxy Settings"). The Squid proxy software actually has more features, since Squid allows filtering based on partial URLs and key words. See " Appendix C" for instructions on configuring Squid proxy.

1.3 Advanced HTTP Features

Web[Ten]'s Apache is fully compliant with HTTP/1.0 and HTTP/1.1. HTTP/1.1 is the newest version of the HyperText Transfer Protocol. This version allows for greater performance and efficiency when transferring files, and includes a suite of advanced features.

1.3.1 Host Name Identification

Every request sent using HTTP/1.1 must identify the host name of the request. This feature, for the first time, allows non-IP-based virtual hosts. This is the "header-based virtual host" feature discussed in section "1.2.2 Header-Based Virtual Hosting". Based on the host name included in every request, the server can deliver different content, even if the IP address is the same. Therefore, two requests for the same IP address, one coming for "`www.joe.com`" and the other coming for "`www.harry.com`", would each receive different content.

1.3.2 Content Negotiation

This gives the server the ability to choose among several different versions of a single document (e.g., English or French, HTML or PDF) and to serve the one most preferred by the browser.

1.3.3 Keep-Alive Connections

Persistent connections, or "keep-alives", allow the browser to establish a long-lived connection when numerous documents are requested over the same connection. Web[Ten]'s ability to recognize this browser request results in better performance.

1.3.4 "As-Is" Documents

Web[Ten] can be configured to send certain documents "as-is". This allows you to create documents that exhibit special behavior, such as documents that redirect browsers to other sites, or documents that are never cached by the browser, without being concerned that the server will automatically add standard HTTP headers to the data.

Web[Ten] also supports "RAW!" type files, the WebSTAR equivalent of "as-is" documents.

1.3.5 "PUT" and "DELETE" Requests

"PUT" and "DELETE" allow Webmasters to create documents on another system and to upload them to the Web[Ten] system. Conversely, such documents can also be deleted from the Web[Ten] system using the browser on the remote system. In either case, it is necessary for the browser on the remote system to also support the "PUT" and "DELETE" methods.

In order to take advantage of these methods, it is necessary to install a plug-in, Apple CGI, or traditional CGI that specifically handles these transactions.

1.3.6 Chunked Transfers

This feature works in concert with "persistent connections". Chunked encoding lets the server send output a bit (or chunk) at a time. Normally the server would have to know the content length before sending data. If the data is the output of a CGI script, the length may not be known. With this new feature, servers can start sending data before the script is completed. This lets servers send out dynamic content that is either large or produced slowly, without having to disable persistent connections.

1.3.7 Byte Ranges

This feature lets browsers request parts of a document, either to continue after an interrupted transfer or to request a single page of a very large document. PDF documents, for example, are often served in this manner.

1.4 Scripting

In general, when traversing a Web page, clicking on a link causes that client (browser) to send a message to the server (the site maintaining the Web page the client wishes to view) with a given URL. The server gets the file indicated by the URL and sends the contents of the file back to the browser to be displayed to the user. The Common Gateway Interface (CGI) is a mechanism that causes the server to behave differently. The CGI protocol defines communication between the server and an external program. When the URL points to a CGI script file, instead of simply sending the contents of the file to the browser, the server executes the script and then returns the program output to the browser. This allows Webmasters to create dynamic documents and interactive pages. WebTen supports a wide range of executable scripts.

1.4.1 CGI-Based Scripts

CGI scripts are typically written in C, C++ or Perl, or they can be UNIX shell scripts. WebTen supports five different styles of CGIs — shell CGIs, Perl CGIs, AppleScript CGIs, WSAPI CGIs, and C or C++ program CGIs.

See Chapter "13.0 Using CGIs" for more information on CGIs.

1.4.1.1 Shell CGIs

A shell CGI is a text file that contains commands for the Bourne Shell or C Shell command interpreter. Any text editor can be used to create shell CGIs. We recommend BBEdit, but any Macintosh editor will do, even SimpleText. The resultant file will typically have the file extension of ".sh" (e.g., *mycgi.sh*). Place the file in the Web[Ten] *cgi-bin* folder.

1.4.1.2 Perl CGIs

A Perl CGI is a text file that contains commands for the Perl language interpreter. The file name extension is usually ".pl", and the file is placed in the *cgi-bin* folder. A Perl interpreter is included with Web[Ten], so Web[Ten] is able to interpret Perl scripts. We recommend using Tenon's CodeBuilder for developing and debugging Perl CGIs.

1.4.1.3 C Language CGIs

A C language CGI is a computer program. To produce a C language CGI, you need to write the C language source program using any Macintosh text editor. Then, a C language translator called a C compiler is needed to translate the C program into machine language. Tenon's CodeBuilder can be used to construct and compile the C language CGI scripts. The machine language file with the extension ".c" is stored in the *cgi-bin* folder in a file that can be executed by Web[Ten]. F

1.4.1.4 AppleScript CGIs

AppleScript is an OS-specific scripting language. Tenon extended Apache to support AppleScript CGIs (ACGIs). The best reference for writing AppleScript CGIs is Jon Wiederspan's paper "Extending WebSTAR with AppleScript". These techniques can be applied directly to Web[Ten], since running AppleScript CGIs on Web[Ten] is exactly like running ACGIs on WebSTAR. These files, with the extension ".acgi", are placed in the *cgi-bin* folder.

1.5 Server APIs

To maximize server performance, it is possible to add modules directly to the server itself using the server's application programming interface (API). By linking the script directly into the server software, you remove the overhead involved in launching an external program (like a Perl script) and passing the information back and forth between the external scripting program and the Web server.

1.5.1 Apache APIs

Apache modules are the equivalent of WebSTAR plug-ins. WebTen includes many Apache modules and, in most cases, those modules can be configured via the WebTen Administration Server. In some cases, an Apache module provides the full functionality of a common WebSTAR-style plug-in. WebTen's Built-In Plug-Ins and CGIs in Appendix E for a partial list of available plug-ins.

See section "16.0 Plug-Ins and Apache Modules" for more information.

1.5.2 WSAPIs

Tenon included a special Apache module, the "`wsapi_module`", that implements W*API 1.1, providing support for running W*API plug-ins and AppleScript CGIs. In most cases, using WebSTAR-style plug-ins with WebTen will be exactly like using them with WebSTAR. There are some anomalies — for example, plug-ins delivered by StarNINE, ironically, do not conform to W*API 1.1. Please see Guide to Using W*API Plug-Ins and AppleScript CGIs in Appendix D for more details.

1.5.3 Installing Plug-Ins

See Chapter "16.0 Plug-Ins and Apache Modules" for instructions on installing Plug-Ins in WebTen.

1.6 Server-Side Includes

Server-Side Includes (SSIs) are a simple type of script that allows you to create HTML documents with boiler plate information (e.g., time of day, name of the server, built-in page hit counters, etc.). Apache includes a new set of eXtended Server-Side Includes (XSSIs) that lets you set variables and use conditional HTML.

1.7 Database Interfaces

The standard way for Web servers to interact with databases is through CGI scripts. A number of solutions exist on the Macintosh, both in the form of plug-ins and CGIs (e.g., Tango and on Lasso with the FileMaker Pro database). In addition, there are public domain UNIX database applications with CGI script interfaces that could easily be incorporated into Web[Ten].

1.8 Directives

Directives are the key to both Apache and Squid. Directives are ASCII text strings, usually with two or more components (e.g., a tag and a specifier). All server actions are determined by directives. You can use directives to turn Squid logging on, to limit server access, to insert files into an HTML document, or to run a script.

Web[Ten]'s browser-based interface insulates Webmasters from manipulating directives inside configuration files. With Web[Ten]'s interface, mouse clicks are translated into the appropriate directives to yield the required action. Apache-savvy Webmasters, of course, may set directives by editing the appropriate configuration file. For more information on editing directives in the Apache and Squid configuration files. See section Customizing WebTen in Appendix C.

The Web[Ten] W*API module contains some custom directives which may be used in the *httpd.conf* file in the context of the main server or within a <*VirtualHost*> directive. These custom directives control the W*API plug-in / CGI operation. Please see section Guide to Using W*API Plug-Ins and AppleScript CGIs in Appendix D for more details.

The *httpd.conf* file resides in */tenon/apache/conf.httpd.conf* and the *squid.conf* file is in */tenon/squid/etc/squid.conf.*

1.9 Logging

Apache's default log file format is known as the Common Log Format (CLF). This format provides basic information, such as raw hits, pages accessed, client host names, and timestamps. An extension of the Apache *LogFormat* directive lets Webmasters generate WebSTAR-style logging. See Appendix F for more information.

1.10 Special Utilities

Web[Ten] includes some useful utilities in the *tenon:utilities* folder. These utilities are stand-alone Macintosh programs that provide a specific feature or service that aids in the use and maintenance of a Web[Ten] system. In addition, many free, shareware and commercial programs provide other very useful services.

1.10.1 Fast File First Aid

The *Fast File First Aid* program repairs Web[Ten]'s fast storage files. These files may become damaged in the event of a power outage or other unorderly shutdown of the Web[Ten] system. Web[Ten] automatically performs a check and repair (if necessary) on these files each time Web[Ten] is started, so this program should only be used in the atypical event that the automatic repair is failing. To run this program, drag-and-drop one of Web[Ten]'s fast storage files from the *tenon:Storage* folder onto the *Fast File First Aid* program.

1.10.2 NoFinder

The *NoFinder* program provides a means to start and stop the Finder (and other programs) on a Macintosh. The Finder is the program that presents the Desktop interface and supports "point-and-click" and "drag-and-drop" methods for launching programs and managing files.

To use *NoFinder*, simply double-click the icon, select *Finder* from the *Processes* list, and choose *Terminate A Process* from the *Process* menu. To restart the Finder, choose *Launch Process* and select *Finder* from the *System Folder*.

Reasons for running a Macintosh without running the Finder include, but are not limited to:

- Reduced memory requirements. The Finder's memory is released for use by other programs.
- Better performance. The Finder is not competing with other programs for processing cycles.
- Security. The files on the system are protected from unintentional changes.

1.10.3 Unix <-> Text

Web[Ten] can serve text files of any Macintosh text file format, including files to be executed as CGIs. If, however, a Webmaster wishes to access an Apache configuration file directly or modify Apache Log files using Macintosh editors, *Unix*<->*Text* file conversion may be necessary. The *Unix*<->*Text* program converts Macintosh text files back and forth between the different text file formats supported by Web[Ten].

When any Macintosh text file (with the type "TEXT") is dropped on *Unix*<-> *Text*, the file is converted to Web[Ten]'s Macintosh creator (or signature) which is "MUMM" (if it is not already "MUMM"), and the Macintosh type "BINA". "BINA" is Web[Ten]'s native file format. This format provides the fastest possible performance for reading text files in Web[Ten], but most Macintosh text editors do not support this file format. We recommend BBEdit for reading and writing these text files, as it does support this format.

Dragging a "MUMM/BINA" text file onto *Unix*<->*Text* converts the file to the "MUMM/TEXT" format. This format provides excellent performance for reading text files in Web[Ten], and all Macintosh text editors support this format. This format is the suggested format for Web[Ten]'s Perl and shell CGIs.

1.11 Transitioning to Mac OS X

Tenon's iTools provides the same web-based configuration and management interface on Mac OS X as Web[Ten]. So, with Web[Ten], transitioning to Mac OS X is seamless.

2.0 Installing Web[Ten]

2.1 Before You Begin

Check your Macintosh	Web[Ten] is a native Power Macintosh application. It runs only on Power Macintosh computers.
Check Memory and Disk Space	We recommend 32MB physical RAM or more for best performance. Web[Ten] dynamically allocates the memory it needs from the System heap, so it is not necessary to increase the size of the Web[Ten] application heap. We also recommend running with VM turned on, even if you don't need the extra virtual memory. When VM is turned on, portions of the application are dynamically loaded, reducing Web[Ten]'s total memory requirements. Web[Ten] requires about 53 Mbytes of disk space. Additional space will be required, depending upon the amount of Web content you publish.
Check your Networking	Before launching Web[Ten] for the first time, verify that your Macintosh network settings are correct. Before running Web[Ten], ensure that your OpenTransport settings work for other Macintosh applications before running Web[Ten]. Web[Ten] cannot correct erroneous OpenTransport settings.
Check for Name Service	Web[Ten] expects Domain Name Service for normal operation. Verify via the TCP/IP control panel that your Macintosh is properly configured for access to a Domain Name Server. See section "11.0 Domain Name System (DNS)" for more information about DNS.

2.2 Installation Guide

Step 1 — Launch the Installer

To install WebTen, double-click on the *WebTen Installer* icon and launch the installer program.

Figure 2: Installer Icon

Step 2 — Read the Online Instructions

Read the online installation instructions and click *Continue*.

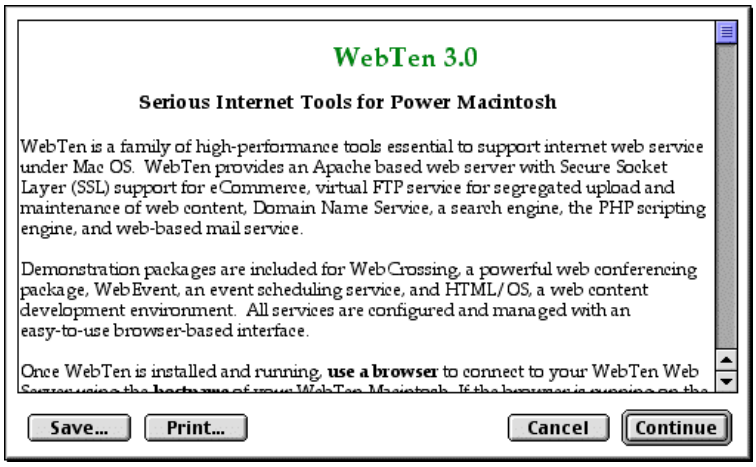


Figure 3: Installer Instructions

Step 3 — Select the Items to be Installed

Select *WebTen Core Package* and all other packages desired and click *Install*. The *WebTen Core Package* must be installed for any other package to be installed. Any of the packages can be installed later individually. See section *"2.5 WebTen Extensions"* for information on the individual packages.

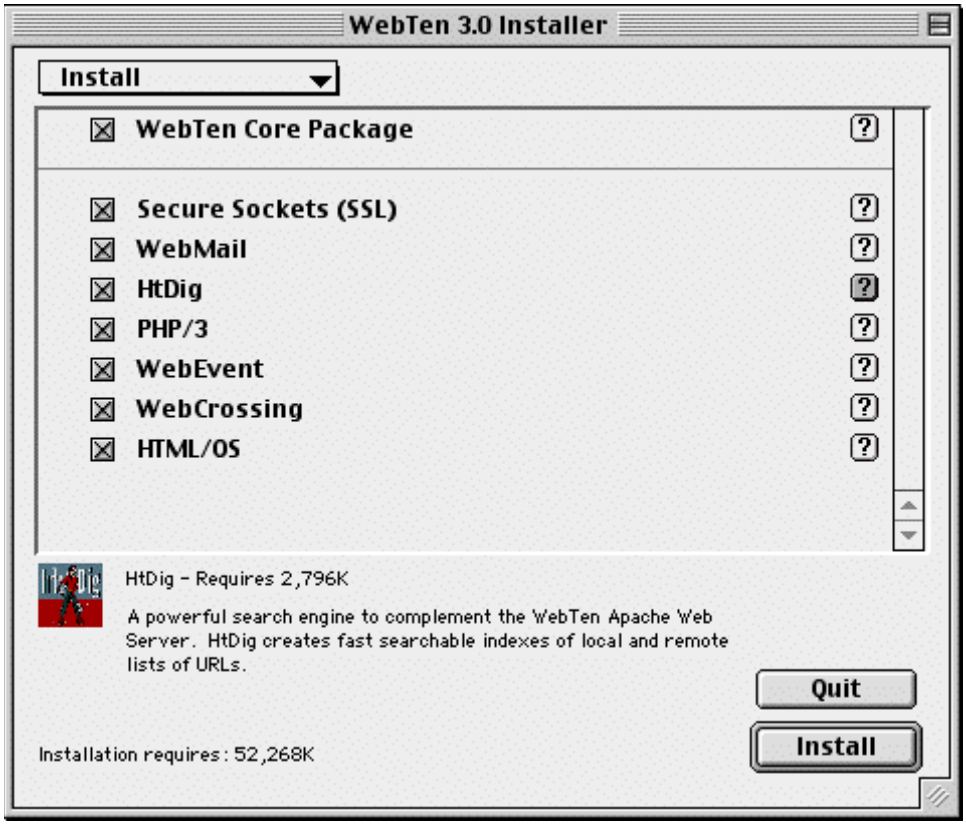


Figure 4: Installer Options

The installer will need about 53 Mbytes of disk space for a complete installation.

Step 4 — Select a Folder

Select a folder in which to install WebTen.

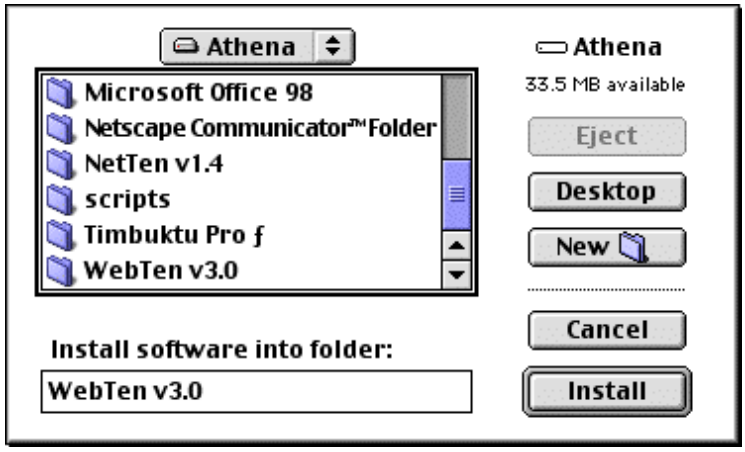


Figure 5: Installer Destination Folder

A window depicting the progress of the installation will appear. Wait a few moments for the installation to complete.

Figure 6: Installer Progress

Step 5 — Close the Installer

When the installer program completes, it displays a status window with a message describing the success of the installation. Click *Quit*. It is not necessary to restart your Macintosh.

2.3 Re-Installing

This step is important even when re-installing the same version of Web[Ten]. These libraries may have been reorganized (by toggling the *Replace OpenTransport* setting in the *Preferences* window). Installing a new version on top of a reorganized version may result in an invalid organization.

Site-specific content files located in the *WebTen* folder or any of its sub-folders should be preserved before de-installing the old *WebTen* folder.

Customizations made via the Web[Ten] Administration Server naturally will need to be re-entered once the new version is installed. Please note all such customizations before de-installing the old *WebTen* folder.

Leave the *WebTen Preferences* file in place to preserve Web[Ten]'s TCP / IP networking settings.

User and Group lists can be preserved by exporting them from the old installation and importing them back into the new installation. See section "6.10.2 Import and Export" for more information.

Specific instructions on upgrading to Web[Ten] 3.0 from previous versions of Web[Ten]can be found on the Web[Ten]support page at http: / / www.tenon.com / support / webten.

2.4 Uninstalling

The WebTen installer includes the option to Uninstall WebTen. The figure below shows this Uninstall option. This Uninstall mode can also be used to Uninstall the separate components of WebTen.

Figure 7: Uninstall

To manually remove WebTen, follow these steps:

Delete the *WebTen Preferences* file from the *Preferences* folder.

Delete the *WebTen* folder.

2.5 Web[Ten] Extensions

Web[Ten] includes separately installable packages that add functionality to the iTools web server. Web[Ten] extensions include ht://dig, a Sherlock-savvy search engine, and WEBmail, a hotmail-style mail server. Third-party packages include WebCatalog, a powerful eCommerce storefront, WebEvent, a calendar and appointment scheduler, WebCrossing, a conferencing application, HTML/OS, a sophisticated tool for creating dynamic web pages, and FrontBase, a powerful SQL92 compliant database . The open source PHP package for creating dynamic web content is also included.

Once these packages are installed, you can browse the URLs listed below to find more information. Note that these URLs only exist if the particular package has been installed.

2.5.1 ht://Dig

For complete information on using ht://Dig, see chapter "13.0 ht://Dig".

This URL will give you a default set of indexing rules to make a searchable database of the local site. Run this index to create a searchable database:
http://hostname.domain/index.cgi

This URL will search the database for key words and phrases:
http://hostname.domain/search.shtml

2.5.2 WEBmail

For WEBmail documentation, see chapter "14.0 WEBmail" .

With the WEBmail client at this URL, you should be able to use any browser to access POP mail from the local system and from remote server:
http://hostname.domainname/webmail

The iTools Administrator uses this URL to add local restricted users to the system database:
http://hostname.domainname/webmail_adduser
These user accounts will be permitted to receive local email but will not be allowed login access to the system. By default, one needs to have an Web[Ten] admin account to add new webmail users.

2.5.3 WebEvent

Matador's (http://www.matadordesign.com) WebEvent is one of the most sophisticated calendar and event scheduling applications under Mac OS. WebEvent supports text, graphic and video calendar operations

This URL accesses a demonstration copy of WebEvent:
http://hostname.domain/webevent. Browse this URL immediately after Web[Ten] installation to set up the WebEvent administrator account.

2.5.4 WebCrossing

Lundeen's (http://www.lundeen.com.) WebCrossing is a powerful meeting and conference application for Mac OS.

This URL accesses the web page for the demo version of WebCrossing:
http://hostname.domain/webx

2.5.5 HTML/OS

Aestiva's (http://www.aestiva.com) HTML/OS is a powerful, high-speed, multi-user development environment for building advanced web sites on Mac OS. Use http://localhost/cgi-bin/htmlos-setup.cgi to complete setup.

During setup, it is important to fill in "Domain Name:" with a fully qualified hostname <hostname>.<domainname> (e.g. goodhost.tenon.com).

This URL accesses the web page for the demo version of HTML/OS:
http://hostname.domain/cgi-bin/htmlos-setup.cgi

2.5.6 PHP 3

PHP is a powerful and complete scripting language, with a rich set of database interfaces, for creating sophisticated dynamic web content. This package is a DSO (Dynamic Shared Object) Apache module.

This URL generates a PHP environment table as an example of PHP's capabilities:
http://hostname.domain/test.php.

3.0 Quick Start Guide

This is a cookbook-style guide to getting started quickly with WebTen. Each step references a later section in this manual which provides more in-depth information. The Quick Start Guide assumes you already know something about internet Web service. If you are not familiar with the Internet and the World Wide Web, please see section "3.4 Web Serving Resources" for more information on building a Web site.

3.1 Launching

Find the *WebTen* folder and double-click on the *WebTen* icon.

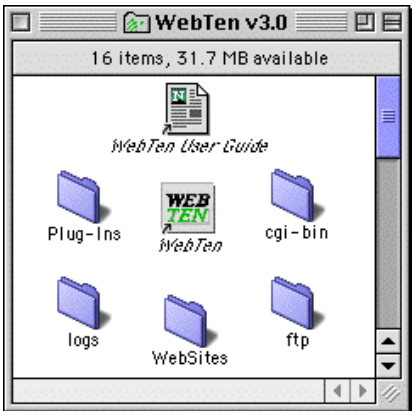


Figure 8: WebTen Folder

The WebTen application requires a minimum amount of information about your system and your network settings before it can start. If WebTen finds a previous preferences file, it will start up. Otherwise, WebTen displays a *Preferences* window and will not start WebTen until the minimum requirements are satisfied. See section "4.2 Preferences" for more information. If, however, the Option key is held down just as WebTen begins starting, the preference dialog box will display itself and the settings can be changed before they are loaded. Once WebTen is satisfied, a *Startup Status* window will appear. This status window reports on WebTen's initialization with a progress bar and text. When WebTen's initialization completes, the window disappears and all of WebTen's menu items are enabled. Once WebTen

is started, changes to the preferences become active only after Web[Ten] is quit and restarted.

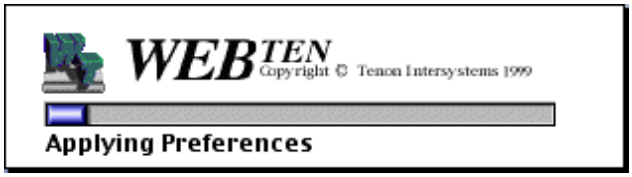


Figure 9: Startup Status Window

3.1.1 Web[Ten] Application Heap

The Web[Ten] application heap is the memory that is allocated specifically to Web[Ten] when it is launched. The size of the heap is specified in the "Get Info" window for the Web[Ten] application. Since Web[Ten] allocates most of its memory from outside of this heap, it is not necessary to increase the size of this heap before launching Web[Ten]. In fact, significantly increasing this setting is a common mistake made by new Web[Ten] users that can actually reduce the memory available to Web[Ten] and affect its behavior and performance.

Do not allocate additional memory to the Web[Ten] application even if there are Plug-ins installed.

3.1.2 Setting the Web[Ten] Administrator's Password

You must set the Web[Ten] administrator's password immediately after launching Web[Ten] the first time. Select *Set Password* under the *Admin* menu. (See section "4.4.1 Set Admin Password".) Type the administrator's user name in the *Admin Name* field, and the administrator's password in the *Password* field. Click *OK*.

Once the administrator's name and password have been entered, additional names and passwords may be added, changed or deleted using the forms and tables presented in the Web[Ten] Administration Server pages under sections "6.9 Users" and "6.10 Groups". *Set Admin Password* can always be used to add new administrators or change the passwords of existing administrators.

You will not be able to connect to the Administration Server if you do not set the Administrator and password set.

3.2 Connecting

3.2.1 Connect to the WebTen Home Page

Use your Web browser to connect to the WebTen Home Page. If your browser is running on the same machine on which WebTen is running (and you are reading this documentation on-line), click on the following local host link:

`http://localhost`

If your browser is running on another machine, enter the URL:

`http://hostname`

where "`hostname`" is replaced with the host name of your WebTen machine.

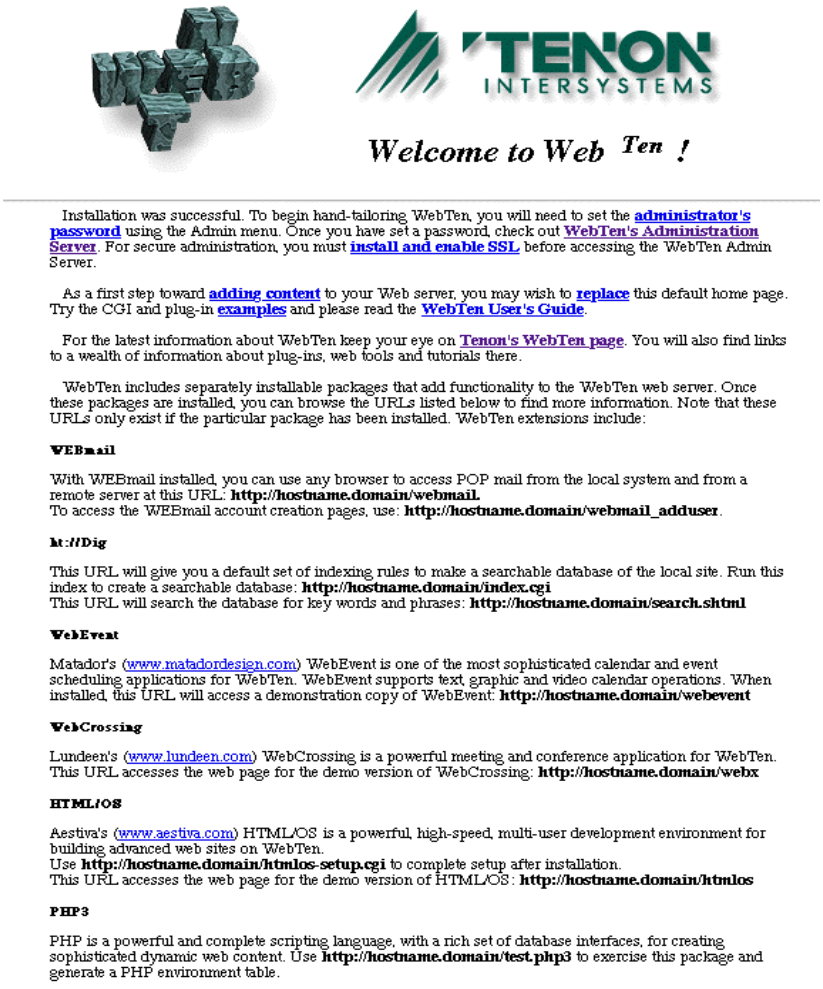
If neither of the above links connects you to the WebTen Home Page, substitute the name of your WebTen machine and your domain name into the following URL, and enter it directly into your browser:

`http://hostname.domainname`

The first time WebTen is accessed via a browser, WebTen uses the file *default.html* (located in the *WebTen* folder) as its home page. The WebTen Home Page contains several useful items, including links to on-line documentation and Web content examples. To access the WebTen Home Page from a browser on the same machine as WebTen, simply point your browser at `http://localhost`.

If your browser is running on another system, you will need to specify the complete URL using the nost name of the WebTen machine and its domain.

A Web server's "home page" is the top-level page of a site — the welcome page. The WebTen Home Page can be used as a template for your own personalized home page. In addition, it serves as a roadmap to help you get started with WebTen. Since WebTen supports virtual hosting, you can have multiple Web sites on a single machine. In that case, each Web site would have its own top-level home page.



Welcome to Web Ten !

Installation was successful. To begin hand-tailoring WebTen, you will need to set the **administrator's password** using the Admin menu. Once you have set a password, check out **WebTen's Administration Server**. For secure administration, you must **install and enable SSL** before accessing the WebTen Admin Server.

As a first step toward **adding content** to your Web server, you may wish to **replace** this default home page. Try the CGI and plug-in **examples** and please read the **WebTen User's Guide**.

For the latest information about WebTen keep your eye on **Tenon's WebTen page**. You will also find links to a wealth of information about plug-ins, web tools and tutorials there.

WebTen includes separately installable packages that add functionality to the WebTen web server. Once these packages are installed, you can browse the URLs listed below to find more information. Note that these URLs only exist if the particular package has been installed. WebTen extensions include:

WEBmail

With WEBmail installed, you can use any browser to access POP mail from the local system and from a remote server at this URL: **http://hostname.domain/webmail.**
To access the WEBmail account creation pages, use: **http://hostname.domain/webmail_adduser**.

ht://Dig

This URL will give you a default set of indexing rules to make a searchable database of the local site. Run this index to create a searchable database: **http://hostname.domain/index.cgi**
This URL will search the database for key words and phrases: **http://hostname.domain/search.shtml**

WebEvent

Matador's (www.matadordesign.com) WebEvent is one of the most sophisticated calendar and event scheduling applications for WebTen. WebEvent supports text, graphic and video calendar operations. When installed, this URL will access a demonstration copy of WebEvent: **http://hostname.domain/webevent**

WebCrossing

Lundeen's (www.lundeen.com) WebCrossing is a powerful meeting and conference application for WebTen. This URL accesses the web page for the demo version of WebCrossing: **http://hostname.domain/webx**

HTML/OS

Aestiva's (www.aestiva.com) HTML/OS is a powerful, high-speed, multi-user development environment for building advanced web sites on WebTen.
Use **http://hostname.domain/htmlos-setup.cgi** to complete setup after installation.
This URL accesses the web page for the demo version of HTML/OS: **http://hostname.domain/htmlos**

PHP3

PHP is a powerful and complete scripting language, with a rich set of database interfaces, for creating sophisticated dynamic web content. Use **http://hostname.domain/test.php3** to exercise this package and generate a PHP environment table.

Figure 10: WebTen Home Page (*default.html*)

3.2.2 Connect to the Administration Server

To connect to the WebTen Administration Server, follow the *WebTen Admin Server* link in the WebTen Home Page. If your browser is running on the WebTen system, can also click on the following link:

http://localhost/webten_admin

If your browser is running on another system, enter a URL of the form:

http://host_name/webten_admin

where "host_name" is replaced with the host name of your Web[Ten] system.

3.2.3 Try the Examples

The Web[Ten] Home Page contains a number of example CGIs including Perl, shell, and binary CGIs. There are also some example Apple CGIs and plug-ins.

3.2.4 Add Your Web Pages

All that remains to be done is to place the pages, folders and sub-folders you wish to publish in the Web[Ten] folder. You may also customize your home page by replacing the file *default.html* with one of your own design.

Web[Ten] lets you publish hypertext and multimedia documents across the Internet. Any Macintosh file (e.g., GIF and JPEG images, QuickTime movies, VRML documents) can be sent by Web[Ten] in response to browser requests. Macintosh text, graphics, video and sound files, and executable scripts can be added to a Web[Ten] system simply by placing the material in the *WebTen* folder.

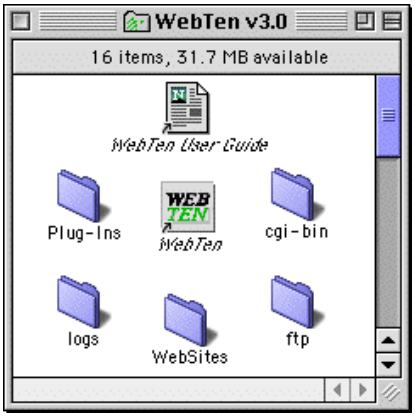


Figure 11: WebTen Folder

The *WebTen* folder includes:

WebSites

All of the content for each virtual host kept in this folder. When you install Web[Ten], a single virtual host folder (the Web[Ten] machine) is automatically created. This folder includes a default home page for Web[Ten] (*default.html*). This sample page can be customized or completely replaced for your site. Use any Macintosh text editor (such as SimpleText) or an HTML editor (such as PageMill) to make changes.

The Web[Ten] application itself is kept in this folder. Do not move the application, since plug-ins and CGI scripts are dependent on the location of the Web[Ten] application.

Web[Ten] User Guide

An alias to an HTML version of the User's Guide. Clicking on (or pointing your browser to) this file will display the Web[Ten] User's Guide. Dragging-and-dropping this alias on your browser enables you to read the Web[Ten] documentation, even when Web[Ten] is not running.

Web[Ten]

This is and alias of the Web[Ten] application. Double-click this icon to start Web[Ten].

Plug-Ins

Plug-ins should be installed in this folder. Some plug-ins are included as examples.

logs

Logs created by either Squid or Apache are stored here. These logs can be read by a text editor (such as BBEdit). By using the *Unix<->Text* utility , the logs can be read by any Macintosh text editor (such as SimpleText).

cgi-bin

This folder contains example CGI scripts for the Web[Ten] server. New scripts should be added to this folder. Scripts in the *cgi-bin* folder are

intended to be shared by all virtual hosts configured under Web[Ten] and referenced by a common */cgi-bin* URL.

Unique *cgi-bin* folders may also be created for each virtual host. See section Customizing WebTen in Appendix C.

By default, scripts running from the *cgi-bin* folder are forbidden to create or modify files within *cgi-bin*. The */cgi-bin/scripts* folder may be used for this purpose.

cgi-bin/scripts

This writable folder may be used for scripts that create temporary files in the directory in which they are executing.

ftp

The contents of this folder are visible to "anonymous" ftp users. All anonymous ftp uploads go into this folder.

Documentation

This folder contains the HTML version of the Web[Ten] User's Guide. Also included is the *Apache* folder, which contains the original Apache documentation (extended by Tenon).

tenon

For Webmasters familiar with Apache, the *tenon* folder is the equivalent of the standard Apache *root* folder. A README in the Apache sub-folder describes the mappings between standard Apache configuration files and the Web[Ten] Administration Server. Other sub-folders include Web[Ten] and Squid configuration files, as well as utilities that may be used with Web[Ten]. Even Webmasters who deal with Web[Ten] via the Web[Ten] Administration Server may occasionally have a need to put special files in this folder.

modules

The modules folder is much like the Plug-ins folder because it holds add-ons to the apache web server that Web[Ten] uses. These modules are loaded when Web[Ten] starts.

3.3 Quitting

To quit Web[Ten], select *Quit* from the *File* menu.

Once you have selected *Quit*, Web[Ten] will begin the process of terminating operations. When the Web[Ten] icon disappears from the monitor, shut down procedures have been completed.

3.4 Web Serving Resources

For the latest information on Web[Ten] and other Tenon products, visit the Tenon Home Page at `http://www.tenon.com`.

A Beginner's Guide to URLs

A complete explanation of the Uniform Resource Locator, from the National Center for Supercomputing Applications. See `http://www.ncsa.uiuc.edu/demoweb/url-primer.html`.

A Beginner's Guide to HTML

An excellent resource concerning HTML markup tags, acronyms and formatting information.
`http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html`.

HTML: The Definitive Guide

A complete guide to creating documents on the World Wide Web. For more information, see `http://www.ora.com/catalog/html2/`.

WebMaster in a Nutshell

One-volume desktop reference covering HTML, CGI, JavaScript, Perl, HTTP, and server configuration. See http://www.ora.com/catalog/webmaster/.

The Common Gateway Interface

An explanation of the standard for interfacing external applications with Web servers, from the National Center for Supercomputing Applications. See http://hoohoo.ncsa.uiuc.edu/cgi/overview.html.

CGI Programming on the World Wide Web

A comprehensive explanation of CGI and related techniques for providing information servers on the Web. For more information, see http://www.ora.com/catalog/cgi/.

Using Perl with MacHTTP

How to create a MacPerl CGI for use with Web[Ten]. See http://www.biap.com/machttp/howto_perl.html.

Designing for the Web

Covers information and techniques useful to anyone who wants to put graphics on-line. See http://www.ora.com/catalog/wdesign/.

Apache: The Definitive Guide

Vital information for Apache programmers and administrators. For more information, see http://www.ora.com/catalog/apache/.

ApacheWeek

A weekly Internet magazine covering all aspects of running a Web server using Apache. For more information, see http://www.apacheweek.com/.

4.0 Web[Ten] Menus

Web[Ten] menus are used to set the system configuration and to display system status. The *Preferences* item of the *File* menu and the *Set Admin Password* item of the *Admin* menu are used early in the configuration of a Web[Ten] installation to set network and user information. Other menu items control Web[Ten] operation and display Web[Ten] status.

4.1 File Menu

The *File* menu contains *Close*, *Preferences* and *Quit* menu items.

The *Close* menu item will close the currently selected window. It has a keyboard equivalent of <Command-w> that will accomplish the same function.

The *Preferences* menu item supports the baseline configuration of the Web[Ten] system and network configurations. See section "4.2 Preferences" for more information.

The *Quit* menu item terminates Web[Ten] operations. It has a keyboard equivalent of <Command-q>. The termination process may take several seconds. During this time, internal file systems are synchronized and each Apple plug-in module is informed that a termination is taking place, giving each module an opportunity to shut down.

4.2 Preferences

Certain minimum requirements must be satisfied before WebTen can be launched. The *Preferences* menu item lets you configure these requirements. By default, WebTen copies the *Host Name* and *Time Zone* settings from the *Sharing Setup* and *Date & Time* control panels, respectively. If OpenTransport has been installed, WebTen gets its default networking settings from the *TCP/IP* control panel. If the previously set values from these control panels satisfy the minimum requirements, WebTen proceeds with full operations. Otherwise, WebTen displays the *Preferences* window with as many of the default fields filled in as possible. WebTen is not launched until the minimum required information has been entered.

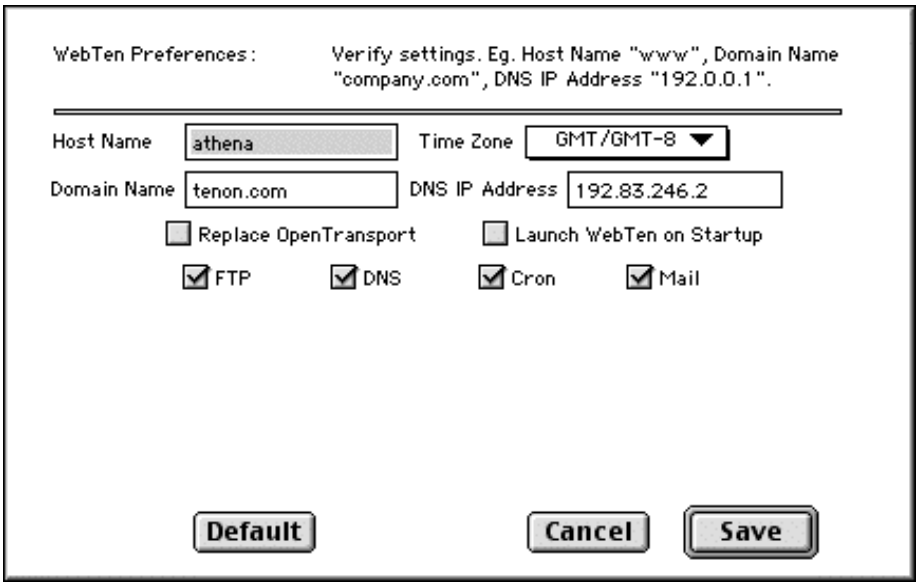


Figure 12: Preferences Window

4.2.1 Host and Domain Name

The *Host Name* is the name assigned to your Macintosh Web server machine. It is the name by which other machines on the network will refer to this host. This information is automatically obtained from the *Macintosh Name* in the *Sharing Setup* control panel. Web[Ten] needs a host name to operate properly. If your machine does not have a host name, you still need to enter something in this field.

The *Domain Name* is the name assigned to your network. If available, this information is obtained from the *TCP/IP* control panel. It is the name by which other machines on external networks will refer to your network and it is the default network name used by machines on your network. If you are using Web[Ten] exclusively within an intranet with its own domain name, or you already have a valid internet domain name, enter the domain name.

In "Figure 12: Preferences Window" for example, the host name is bee and the domain name is tenon.com. This Web[Ten] would then be accessed with the URL http://bee.tenon.com. The URL http://www.tenon.com will request the host 'www' in the domain 'tenon.com.'

Web[Ten] includes a domain name server. If you wish to operate Web[Ten] without the support of a domain name server, please see section "11.0 Domain Name System (DNS)" for information about Web[Ten]'s built-in DNS server or other DNS alternatives. Web[Ten] automatically adds DNS records for the host and domain given to its own DNS server.

4.2.2 Time Zone

The *Time Zone* is the corresponding time zone in which your Macintosh is located. Select the appropriate value from the list. If your time zone does not appear in the list, use the GMT setting with the proper plus or minus value. This information is obtained automatically from the *Date & Time* control panel.

4.2.3 DNS IP Address

The *DNS IP Address* is the IP address, in dot notation, of the Domain Name Server on your network. If available, this information is obtained from the *TCP/IP* control panel. If some machine other than the Web[Ten] machine is to provide the Domain Name Service on you network, enter the IP address of that machine. If you do not

have a Domain Name Server, leave this field blank. See section "11.0 Domain Name System (DNS)" for more details on configuring WebTen without a Domain Name Server. To use WebTen as it's own DNS, put the IP address of the WebTen machine in this field.

4.2.4 Launching WebTen On Startup

WebTen can be configured to start up whenever your Macintosh is restarted. This is accomplished by placing a Finder alias of the WebTen application in the *Startup Items* folder in the active *System Folder*. This check box automatically adds or removes such an alias from the *Startup Items* folder. See "Figure 12: Preferences Window".

4.2.5 Replacing OpenTransport

Some WebTen configurations need the support of multiple IP addresses on a single Macintosh. The WebTen TCP/IP is the only Macintosh TCP/IP that supports these capabilities. In addition, because WebTen's TCP stack is so finely-tuned for top performance, even when OpenTransport supports multihoming, Tenon's stack may be the protocol stack of choice.

At the same time, a number of Macintosh networking applications must have the support of OpenTransport. For example, Timbuktu from Farallon is used to support remote control of a Macintosh. Timbuktu requires OpenTransport. One solution to this problem is to run both the OpenTransport and WebTen TCP/IP protocol implementations at the same time.

The basic strategy is to configure OpenTransport with one IP address and WebTen TCP/IP with a second IP address. This allows Macintosh applications to use OpenTransport, and lets WebTen use its own stack for multi-homing, performance or other reasons.

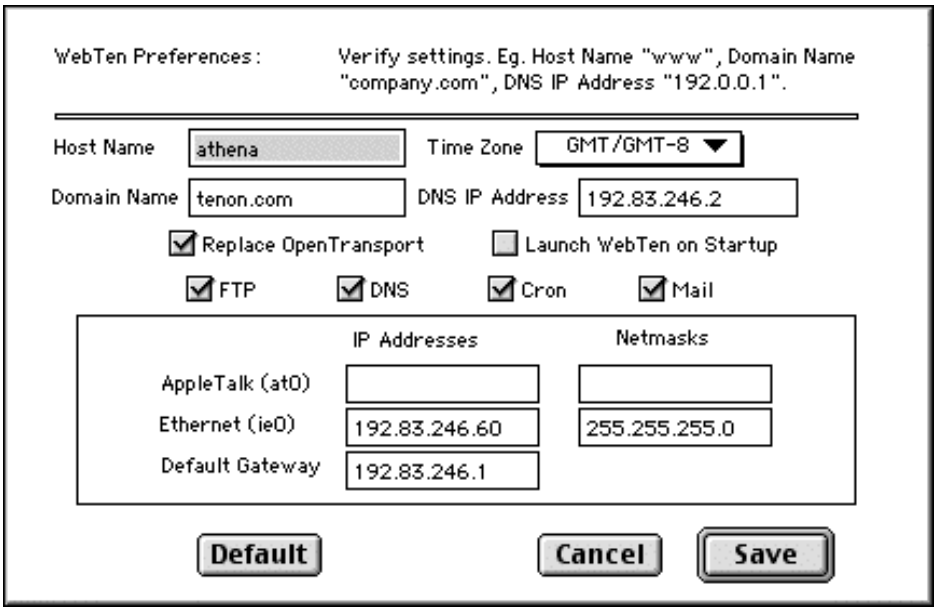


Figure 13: Replacing OpenTransport

A number of new fields appear when *Replace OpenTransport* is checked in the dialog box. The new fields will contain default configuration values taken from the current OpenTransport network configuration. Changing the IP address listed in the Ethernet field to an alternate IP address causes Web[Ten] to respond to that address. As long as that address is different than any other in the network and, importantly, different than the one that is given to the local OpenTransport, both TCP/IP implementations will operate successfully at the time on the same Macintosh.

A final step is to manually move the *MacTCPdLib* and *OTSocketLib* files from your *WebTen/WebTen Libraries* folder to the *Disabled* folder and move the *SocketLib* file to the *WebTen/WebTen Libraries* folder. This will ensure that both classic MacTCP applications and OpenTransport applications that are MacTCP-aware will use the OpenTransport TCP/IP stack.

That's all that is required to run OpenTransport and Web[Ten] TCP/IP on the same machine at the same time. Simply configure OpenTransport with one IP address, and configure Web[Ten] TCP/IP with a another IP address. You should also enter

this new IP address and a corresponding name into your DNS server. If you subsequently use the WebTen Admin Server to add virtual hosts, they should be entered into your DNS database.

More RAM will be used when running both TCP/IP implementations. You should be prepared to dedicate as much as 850K of extra RAM to simultaneous TCP/IP operations. This RAM will not be available to your applications.

4.2.6 Networking with OpenTransport

If OpenTransport is installed, the WebTen *Preferences* default is to use OpenTransport (by not selecting *Replace OpenTransport*). In this mode, the minimum set of preferences that must be set includes the *Host Name* and *Domain Name*.

4.2.7 Enabling WebTen Services

WebTen can be configured to provide FTP, DNS and clock (*Cron*) service in addition to Web (HTTP) service when launched. By default, these services are disabled. When one of these items is checked, it is started when WebTen is started.

For more information on FTP, see chapter "9.0 FTP Service".

For more information on DNS, see chapter "11.0 Domain Name System (DNS)".

For more information on Cron, see chapter "12.0 Clock Service (Cron)".

For more information on Mail, see chapter "14.0 WEBmail".

4.2.8 Testing WebTen in Loopback Mode

Webmasters often want to test, evaluate and experiment with WebTen before deploying it as a live Web server. WebTen's evaluators often choose to perform their tests entirely in a loopback environment (with both the Web server and the Web browser running on the same machine). WebTen's installation is designed to provide the easiest process and fewest steps to set up a real, live Web server. Thus, configuring WebTen for the degenerate, loopback-only case may require additional steps.

Loopback-only mode may be required to support various situations. The Macintosh used to evaluate WebTen may not have a networking interface. The networking interface may be a dial-up PPP connection with a dynamic IP address assignment. The networking interfaces may exist but they may not be properly configured. Whatever the reason, the steps for configuring your system for loopback-only mode are given below.

The simplest environment for testing WebTen is one with a permanently assigned IP address on a properly configured network. In this situation, WebTen will install easily and it can be tested either via a browser running on the same Macintosh as WebTen or via a browser running elsewhere. If this is your current situation and you wish to test WebTen in a loopback-only mode because of some security concerns about running a Web server, don't worry about it. The default WebTen installation contains no sensitive information that would empower anyone else to compromise your system, nor will it serve anything that you don't explicitly put into the WebTen folders for Web service.

Here is how to set up your system for loopback-only testing:

- Open the AppleTalk control panel, and make sure that it is set to "Connect via Printer Port". Verify that AppleTalk is set to "Active" in the Chooser.
- Set the TCP/IP Control Panel "Connect via" menu to "AppleTalk (MacIP)".
- Set the TCP/IP Control Panel "Configure" menu to "Using MacIP Manually".
- Set the TCP/IP Control Panel "IP Address" to "10.0.0.1".
- Set the TCP/IP Control Panel "Router address" to "10.0.0.1".
- Set the TCP/IP Control Panel "Name Server addr" to "10.0.0.1".
- Set the TCP/IP Control Panel "Search Domains" to "bogus.com".
- Delete any previously installed versions of WebTen. If WebTen was previously installed (and failed from some reason likely due to the current networking configuration or lack thereof), delete the WebTen folder, the WebTen Libraries in the Extensions folder and the WebTen Preferences in the Preferences folder.
- Look in the Extensions folder for the WebTen Libraries. In that folder, you should find a file called SocketLib, and in the folderDisabled under the same folder, you should find a file called OTSocketLib. Switch these, so that

SocketLib is under the Disabled folder, and OTSocketLib is in the WebTen Libraries folder. If, for some reason, they are already set up this way, leave them.

- Install WebTen.
- Launch WebTen. In the WebTen Preferences window, check "Replace Open Transport" temporarily to allow you to edit the fields in the box below. Set them up like this:

	IP Addresses	Subnet masks
AppleTalk (at0)	10.0.0.1	255.255.255.255
Ethernet (ie0)	10.0.0.1	
Gateway	10.0.0.1	

- Then, uncheck the "Replace Open Transport" box, and then click on "Save". If you are prompted to confirm the changes, click "Save" again. Quit and Re-Launch WebTen.
- Launch a Web browser.
- Connect to WebTen from the browser using either the loopback IP address "127.0.0.1", or the IP address "10.0.0.1".

4.3 Edit Menu

The *Edit* menu contains *Undo*, *Cut*, *Copy*, *Paste* and *Select All* menu items. These are traditional Macintosh operations to "cut-and-paste" data between the Macintosh clipboard and Web[Ten] windows.

The *Undo* operation reverses the previous *Edit* menu operation.

The *Cut* operation removes the selected text from the Web[Ten] window and copies it onto the Macintosh clipboard. The *Cut* operation has the keyboard equivalent <Command-x>.

The *Copy* peration copies the selected text from the Web[Ten] window onto the Macintosh clipboard. The *Copy* operation has the keyboard equivalent <Command-c>.

The *Paste* operation copies the contents of the Macintosh clipboard into a Web[Ten] window. The text is copied into the window in which the Web[Ten] cursor is located. The *Paste* operation has the keyboard equivalent <Command-v>.

The *Select All* operation selects all the text associated with the current Web[Ten] window. The *Select All* operation has the keyboard equivalent <Command-a>.

4.4 Admin Menu

The *Admin* menu contains *Set Admin Password, Cache Status, Web Server Status, System Status, Flush Cache , Shell Window* and *Save Display* menu items.

Figure 14: The Admin Menu

4.4.1 Set Admin Password

The *Set Admin Password* menu item should be one of the first operations executed after the installation of Web[Ten].

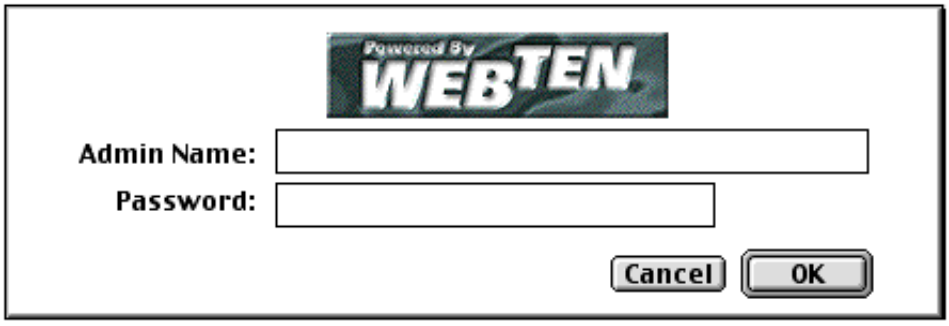


Figure 15: Set Admin Password Window

Use this menu item to set the administrator's user name and password. Both the user name and password are required when using the network-based Web[Ten] Administration Server. ***Do not*** use spaces in the user name. After the *Admin Name* and *Password* fields have been filled in, click the *OK* button to record the user and password, or click *Cancel* to cancel the request. The password will not be echoed to the screen; rather, bullets will be used to represent the characters, preventing your password from being overseen as it is entered. The password is entered only once, so type carefully. If you mistype and accidently click *OK*, simply choose this menu item again and retype the user name and password.

4.4.2 Change License

This menu item is used to modify the Web[Ten] license information while Web[Ten] is running.

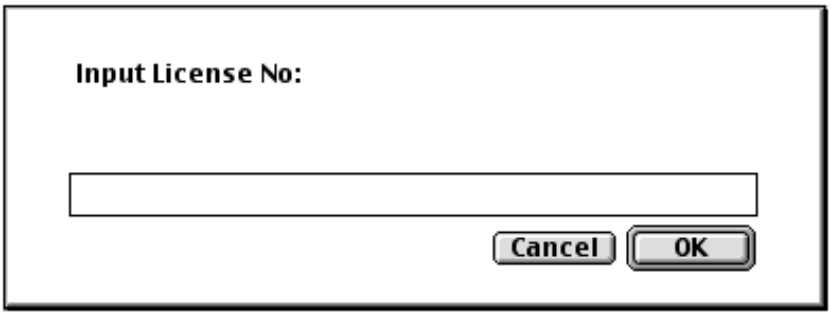


Figure 16: Input License Number Window

Simply input the new license number and double-click on the *OK* button to incorporate the new license number. If you wish to cancel the request, click *Cancel*.

4.4.3 Start/Stop Web Server

The *Start/Stop Web Server* menu item controls the execution of the Apache Web server thread. This menu item is labeled either *Stop Web Server* or *Start Web Server*. When the menu item is labeled *Stop Web Server*, the Apache Web server thread is operating. Selecting *Stop Web Server* terminates the operation of the Apache Web server. This menu item may be used to stop Web operations while system-wide configuration or maintenance is undertaken. When the Web server is stopped, this menu item changes to *Start Web Server*. Selecting *Start Web Server* starts or restarts Web server operations. The Web server operations can be monitored with the *System Status* and *Web Server Status* menu items.

4.4.4 Start/Stop Admin Server

The Administration Server is used to support WebTen configuration and administration from a Web browser. This menu item is labeled either *Stop Admin Server* or *Start Admin Server*. When the menu item is labeled *Stop Admin Server*, the Administration Server is operating and Web browser-based administration is enabled. Selecting this menu item terminates Administration Server operations. If the Administration Server thread is not operating, Web browser-based administration is inhibited. When the Administration Server thread is not operating, this menu item is changed to *Start Admin Server*. Selecting the menu item when it is labeled *Start Admin Server* starts the Administration Server and enables Web browser-based administration.

4.4.5 Cache Status

The *Cache Status* menu item is used to display a *Cache Status* window. The *Cache Status* window contains dynamic information about the operation of the cache portion of WebTen. If the WebTen cache component is not configured, this window will briefly display an error message and then disappear. The *Cache Status* window may be dismissed by clicking in the close box in the upper left-hand corner of the *Cache Status* window, or by using the *Close Window* menu item in the *File* menu. The *Cache Status* window is divided into four sections. Each section is distinguished from the other sections by a horizontal bar.

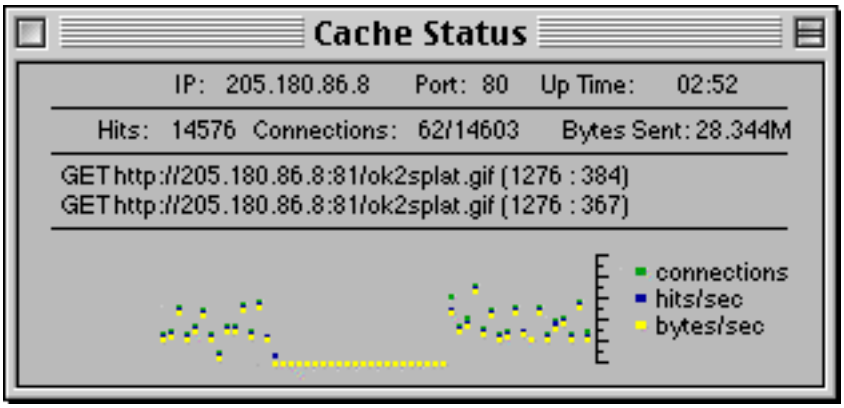
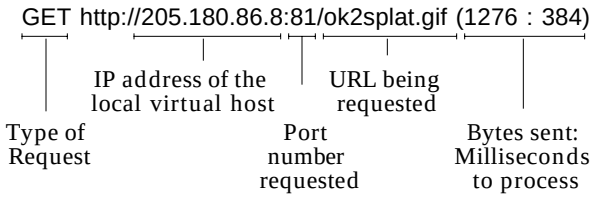


Figure 17: Cache Status Window

The first section of the *Cache Status* window contains *IP*, *Port* and *Up Time* data fields. The *IP* field contains the Internet Protocol address to which the Web[Ten] cache (often referred to as the Cache Manager) is configured to respond. The *Port* field contains the TCP / IP port on which the Cache Manager is listening for HTTP network requests. The *Up Time* field contains the hours, minutes and seconds that the Cache Manager has been running.

The second section of the *Cache Status* window contains *Hits*, *Connections* and *Bytes Sent* data fields. The *Hits* field contains the number of HTTP requests that have taken place since the Cache Manager was started. The *Connections* field has two components. The first component is the number of currently active TCP / IP connections being serviced by the Cache Manager. The second component is the total number of TCP / IP connections that have been serviced since the Cache Manager was started. The *Connections* field components are separated by a slash (" / ") character. The *Bytes Sent* field contains the aggregate number of Mbytes that have been sent by the Cache Manager in response to previous HTTP requests.

The third section of the *Cache Status* window contains a variable number of lines. Each line contains information about a recent request serviced by the Cache Manager. The first element in the service information line is the type of HTTP request currently being processed. "GET" requests are being processed in "Figure 17: Cache Status Window". The information that follows concerns the virtual host, port and URL that have been requested. At the end of the line, in parentheses, is the number of bytes sent in response to the request, and the number of milliseconds used to process the request, divided by a colon (":").



This section displays the first two of the requests that are in progress. More requests may be in progress at any point in time. This section gives an indication of part of the load that has been placed on the Cache Manager. Most of the time two requests will be displayed. Occasionally, if the server is more lightly loaded, this will be reduced to one or no display requests. When the number of requests is reduced to zero, the section is not included in the *Cache Status* display list and is automatically removed from the display.

The last section of the *Cache Status* window contains a graphical representation of some of the data that is contained in the display. This section is updated at one-second intervals. Three different data variables are displayed each second as a colored dot. The green dots refer to the number of connections currently in progress during that second. The blue dots refer to the number of hits that have taken place during that second. The yellow dots refer to the number of bytes transmitted to all clients during an interval.

It is important to understand that this graphical representation is not meant as a detailed plot of the data, but rather as a relative indicator of activity. Each of the plots is "auto-scaled" relative to other values within its group of data. Therefore, comparisons between different data graphs are invalid unless they are on the same "scale". The auto-scaling process takes place each time a new data point is added to the graph. The auto-scaling values are reset to a "0 to 10" scale each time the display is cleared.

4.4.6 System Status

The *System Status* menu item is used to display a Web[Ten] *System Status* window. The *System Status* window contains dynamic information about the operation of Web[Ten]. The *System Status* window may be dismissed by clicking in the close box in the upper left-hand corner of the *System Status* window, or by using the *Close Window* menu item in the *File* menu. The *System Status* window is divided into four sections.

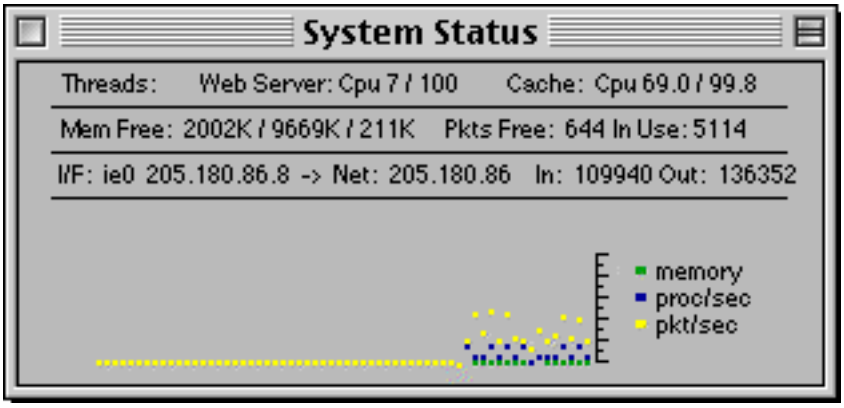


Figure 18: System Status Window

The first section contains *Threads*, *Web Server: CPU*, and *Cache: CPU* fields. Each field contains accumulated user and system CPU information for the Apache Web server thread and for the cache thread. The Apache thread CPU information is accumulated in system ticks, or 1/60th of a second. The information contains an accumulation of all of the system ticks for the main Apache thread and for its constituent children threads. The *Cache: CPU* information is presented in tenths of a second. In both fields, the first number represents the number of ticks, or tenths of a second, used by the Apache or cache software itself. The second number represents the amount of time accumulated performing system-related operations, such as network or file operations.

The second section contains *Mem Free*, *Pkts Free* and *(Pkts) In Use* components. The *Mem Free* component contains three different free memory indications. The first is the amount of free memory in the Web[Ten] Macintosh application heap. The second is the amount of free memory in the Macintosh Process manager heap. This is sometimes referred to as the amount of free temporary memory. The last value is the amount of free memory in the Macintosh system heap. All of the values are specified in Kbytes.

The *Pkts Free* value is the number of Web[Ten] packet buffers that are available for network data. These buffers are essential for carrying network data between the Web[Ten] cache, the Web server software, and the network interface device within the Macintosh Operating System.

The *(Pkts) In Use* value indicates the number of packet buffers that are in use. These data structures are used for a number of different functions and do not necessarily indicate only the presence of network data. Typically, a minimum of 30 or 35 packet buffers are in use. This number will increase to thousands of packets as Web[Ten] is put under load.

The third section contains information specific to each physical network interface attached and configured for use by Web[Ten]. Since Web[Ten] supports multiple physical interfaces, there may be multiple lines in this section. ***This section is active only if Tenon's TCP/IP is configured with Web[Ten]***. This section contains *I/F Net*, *In* and *Out* data values. The *I/F* value contains the name of the network interface and its associated IP address. The *Net* data is the network associated with the interface and its IP address. The *In* value represents the number of packets that have been received by the interface. The *Out* value represents the number of packets that have been transmitted over the interface. Note that the *In* and *Out* counts are per interface counts, not per IP address (many IP addresses may share a single interface).

The last section contains a graphical representation of the memory, CPU and packet data associated with this status window. Like the *Cache Status* and *Web Server Status* windows, this data is plotted as a graphical approximation of the information and resources used by this instantiation of Web[Ten]. Green dots are used to graph the change in memory usage during each one-second interval. Blue dots are used to graph the number of processor ticks that were used during the last second for both the cache and Web server software modules. Yellow dots are used to graph the total number of packets that have been sent and received by the system. The data values are scaled automatically to fit within the graph coordinates.

4.4.7 Web Server Status

The *Web Server Status* menu item is used to display a *Web Server Status* window. The *Web Server Status* window contains dynamic information about the operation of the Apache Web server portion of WebTen. The *Web Server Status* window may be dismissed by clicking in the close box in the upper left-hand corner of the *Web Server Status* window, or by using the *Close Window* menu item in the *File* menu. The *Web Server Status* display is divided into three sections.

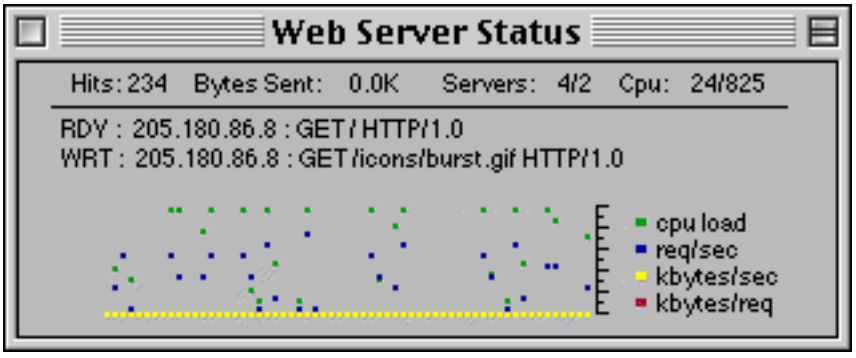


Figure 19: Web Server Status Window

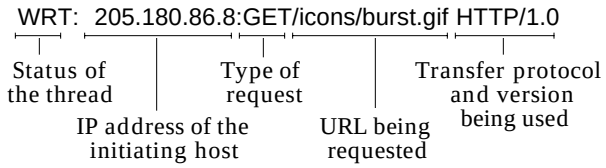
The first section contains *Hits*, *Bytes Sent*, *Servers* and *CPU* fields. The *Hits* field contains the number of HTTP requests that have been processed by the Apache Web server. This field is in contrast to the *Hits* field of the *Cache Status* display, which contains the total number of requests since the Cache Manager was started. If the Cache Manager is running all the time, the *Hits* field of the *Cache Status Display* contains the total number of hits for WebTen as a whole.

The *Bytes Sent* field contains the number of Kbytes sent by the Apache Web server since the server was started.

The *Servers* field contains the number of allocated children threads and the number of active Apache children threads, divided by a slash ("/"). In "Figure 19: Web Server Status Window", there are four currently allocated Apache children and there are two active. The number of allocated Apache children is dynamically set according to the peak load the Apache server has experienced since it was initiated. The number of active servers is a function of the number of HTTP requests that have been passed to the Web server by the Web[Ten] Cache Manager, or directly by contact to the Apache TCP/IP port.

The *CPU* field contains the number of CPU ticks (1/60th of a second) that the Apache server and its children threads have accumulated since Web[Ten] was started. The first number in the field is the number of user level ticks that the Apache server has accumulated. The second number in the field is the number of system level ticks that the Apache server and its constituent threads have accumulated.

The second section of the *Web Server Status* display contains information about two of the Apache children threads. Each thread processes an HTTP request. The first two threads found in the Apache thread list are displayed in this section. Each thread status is displayed on a single line. The first component of the thread status is the state of the thread.[1] The second component of the thread status is the virtual host referenced by the HTTP request. The third component is the IP address of the initiating host. The fourth component is the type of request. In the example below, a "GET" request was last processed by each thread. The fifth component of the thread status is the HTTP URL itself. The last component is the transfer protocol.



1. RDY — Ready; RD — Reading from the network; KA — Keep Alive; and WRT — Writing to the network.

When the Apache server is initialized and no requests have been received, the status lines contain a default indication of "RDY: <-". Note that this does not mean that requests have not been received and processed by Web[Ten], but only that no requests have been received by the Apache component of Web[Ten]. This indication is replaced with a standard state as subsequent Apache HTTP requests are received.

Finally, there is a graphical representation of Apache activity. Like the *Cache Status* window, this information is meant as a rudimentary indication of activity and not as a precise graph. This section contains plots for four data streams. Green dots are used to indicate the number of CPU user and server ticks that were accumulated by the Web server during the previous second. Blue dots are used to indicate the number of HTTP requests that were processed during the previous second. Yellow dots are used to indicate the number of Kbytes of data that were transmitted to the Apache clients during the previous second. Red dots are used to indicate the average number of Kbytes per request that were sent during the previous second. Also like the *Cache Status* window, these graphics are "auto-scaled" for time and space considerations; thus, the same graph's values will be plotted differently depending on the range of values currently displayed for each individual plot.

4.4.8 Flush Cache

This menu item causes the server to delete any data that it has been saved in it's cache. WebSite content is generally what is cached and changes to the content will not be visible to browsers until the cache is emptied. An remotely triggered alternative to this menu item is the flush script. The flush script has the same function as the flush menu item, but is accessed by the URL: http://host.domainname.com/cgi-bin/flush.

4.4.9 Shell Window

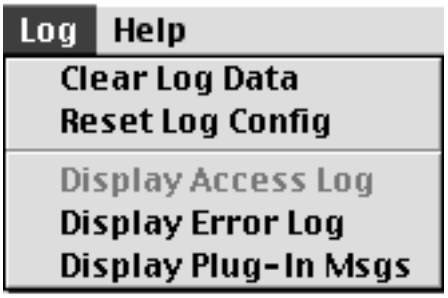
The Shell Window menu option opens a termnal window with a command prompt used for accessing Web[Ten]'s UNIX layer. This interface is meant for andavced manual configuration for users familiar with UNIX.

4.4.10 Save Display

The *Save Display* menu item saves the currently opened windows, as well as their placement. When Web[Ten] is restarted, the display will be re-created.

4.5 Log Menu

The *Log* menu is used manipulate and display various Web[Ten] information logs. The menu items include *Clear Log Data*, *Reset Log Config*, *Display Access Log*, *Display Error Log*, and *Display Plug-In Msgs*.



4.5.1 Clear Log Data

The *Clear Log Data* menu item is used to delete all previously saved log data from disk. This should not be done lightly, as the data files are simply reset to a zero length. The data is not saved or copied.

4.5.2 Reset Log Config

The *Reset Log Config* menu item resets the log configuration to factory default settings. Occasionally, certain logs are started and, due to a system crash or a disk problem, the settings get out of sync with the data files. This menu item attempts to carefully reset of all log information directives to standard "low profile" settings.

All open log windows should be closed before the *Reset Log Config* menu item is selected.

4.5.3 Display Access Log

The *Display Access Log* menu item displays the standard *Access Log* in Common Log Format. Note that log data is also written to a disk file when this window is open. As the system operates, care must be taken that the disk file does not take all available free disk space. The *Clear Log Data* and *Reset Log Config* menu items can be used to clear the disk of *Access Log* data and reset the output of *Access Log* entries.

4.5.4 Display Error Log

The *Display Error Log* menu item displays the *Apache Error Log*. This log contains an entry for each error that occurred during the servicing of any request. Errors that are reported include, but are not limited to, requests for files that do not exist, CGI errors or failures, and requests that are denied due to invalid authorization.

4.5.5 Display Plug-In Msgs

The *Display Plug-In Msgs* menu item displays plug-in output. The log contains entries about the operation of plug-ins. Plug-in initialization, status and error information are also displayed in this window.

5.0 Web[Ten] Administration

Web[Ten] can be configured using any Web browser. The Web browser interface includes easy-to-use tables and forms that eliminate dealing with cryptic Apache directives and the nuisance of updating IP address aliases for each virtual server. Built-in error checking identifies redundant or incomplete entries. Updates are immediately available to the network. And, of course, all documentation is available on-line via the Web.

The browser may be running directly on the Web[Ten] system, or on a remote host connected via a network to the Web[Ten] system. Links within the Web[Ten] Administration Server pages connect you directly to on-line documentation for all aspects of Web[Ten] administration.

5.1 The Administration Server

The Web[Ten] Administration Server is a stand-alone, special purpose Web server that runs within Web[Ten]. This server uses a different port number (the default is port 84), and can be started and stopped independently from Web[Ten]'s main Web server (Apache).

5.1.1 Starting the Administration Server

The Web[Ten] Administration Server is not automatically started when Web[Ten] is started. It is typically started on demand and dispensed with once any administration changes are complete. The Administration Server may be started by using Web[Ten]'s *Admin* menu and selecting the *Start Admin* item. Conversely, the Administration Server may be stopped by selecting the *Stop Admin* item.

The Web[Ten] Administration Server may also be started remotely. Remote startup of the Administration Server uses the Apache server, so it is necessary that Apache is running. The remote startup capability is implemented as a CGI script. Use the path "`/webten_admin`" to invoke this CGI script, thus starting up the Administration Server. For example, if your Web[Ten] system is named "`www.yourdomain.com`", the URL to start up the Administration Server is:

`http://www.yourdomain.com/webten_admin`

The */webten_admin* URL does not start multiple instances of the Administration Server if it is already running. It simply returns a redirection URL to connect the browser to the already-running Administration Server. This URL is included in the default Web[Ten] home page.

If you know that the Administration Server is already running, you may connect directly to it using the host name of your Web[Ten] system and the port number for the Administration Server. For example, if your Web[Ten] system is named `www.yourdomain.com`, the following URL will connect directly to the Administration Server if it is running. It will not start the Administration Server if it is not running.

`http://www.yourdomain.com:84/`

To stop the Administration Server, either pull down the *Stop Admin Server* item under the *Admin* menu, or click on the *Stop Admin Server* button in the Administration Server's home page.

5.2 Web[Ten] Administration Server

Access to the Web[Ten] Administration Server is restricted to users in the *WebTenAdmin* group. There are two ways that users may be added to the *WebTenAdmin* group. Use the *Set Admin Password* item under the *Admin* menu (see section "4.4.1 Set Admin Password"), or use the *Users* and *Groups* tables accessible from within the Web[Ten] Administration Server page (see sections "6.9 Users" and "6.10 Groups").

The latter method requires a valid password to access *Users* and *Groups*. Once you have a valid password for yourself, use this method to add passwords for other users.

Once connected to the Web[Ten] Administration Server, the administration pages appear, as illustrated in ""Figure 20: WebTen Administration Server"".

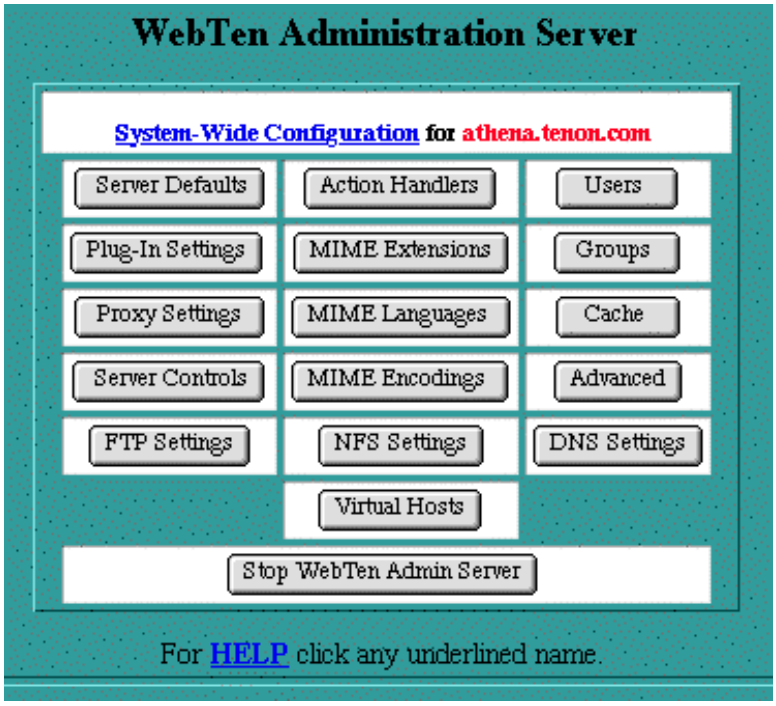


Figure 20: Web[Ten] Administration Server

5.3 Navigating the Administration Pages

The Web[Ten] administration pages use many of the features of HTML forms and CGI scripting to present the Web server's configuration information in tables that are easy to read and easy to modify.

How the information is displayed depends on the type of permissible entries. Related entries are grouped together. Lists are sorted alphabetically. Default or system-wide entries are displayed in the lower portions of the tables, while user-defined changes are displayed in the top portions of the tables. Buttons are provided to save or reset any changes made to these forms, to return to the Web[Ten] Administration Server, or to move on to other tables related to the current table. Many items are displayed as links for quick access to a specific section in the documentation. The following sections explain the conventions used for navigating the configuration settings and making changes to those settings.

5.3.1 Types of Information Fields

Information in the tables may be displayed in the following ways:

- text edit fields
- radio buttons
- check boxes
- pull-down lists

5.3.2 Making Changes

To make changes to an item, either re-type its text, change the radio button or check box settings, or select a different item from a pull-down list. Then click the *Save* button. If an entry in a table is not presented in a text edit field, or as a radio button, check box, or pull-down list, that entry may not be changed. Multiple changes per save are permitted. Once changes are saved, the table is re-displayed with the corresponding changes in place.

Changed items may move to a different row in a table if the rows are sorted and the key used in the sort was one of the changed items.

5.3.3 Adding Entries

New items are usually entered in the first row of a table, which has been left blank by design. When new entries are saved, the table is re-displayed and the new entries appear in their proper place in the table. The first row of the table reverts to blank, awaiting input of another new entry.

5.3.4 Removing Entries

Removing an item from a table can be accomplished by:

- deleting any entry which is displayed in a text edit field.
- unchecking all of the possibilities for a check box.
- selecting the *Inherit* setting from a radio button selection.
- selecting *None* from a pull-down list or radio button selection.

The *Save* button is then clicked to remove the item. Most often, the key field to be deleted is in the first column of the listed item. This may not always be the case.

5.3.5 Resetting Entries

Each of the Web[Ten] administration pages provides a *Reset* button. Clicking on this button will reset all entries in the tables to their current values (i.e., the values they had the last time anything was saved). This button does not make or save any changes in the Web server's configuration; it is essentially an "undo" operation for typos or other incorrect modifications noticed before any such changes are saved.

5.3.6 Inheritance

If certain settings for a particular item are not explicitly set, they are inherited from the parent folder (if the corresponding settings exist). This notion of inheritance is reflected in a table by including the word *Inherited*, in a red font, in the corresponding entry. Modifying an inherited setting and clicking the *Save* button will save an explicit setting for this entry; thus, the *Inherited* flag will not be displayed. Subsequently deleting the explicit entry will cause the setting to be re-inherited from the parent, and the *Inherited* flag will again appear (providing, of course, that the corresponding setting for the parent still exists).

6.0 System-Wide Configuration

The *System-Wide Configuration* table is the starting point for administering Web[Ten]. It contains buttons for each of the major areas of Web[Ten] administration. Clicking on a button will present a table with forms for that specific area. Each of the areas and their tables and forms are discussed below.

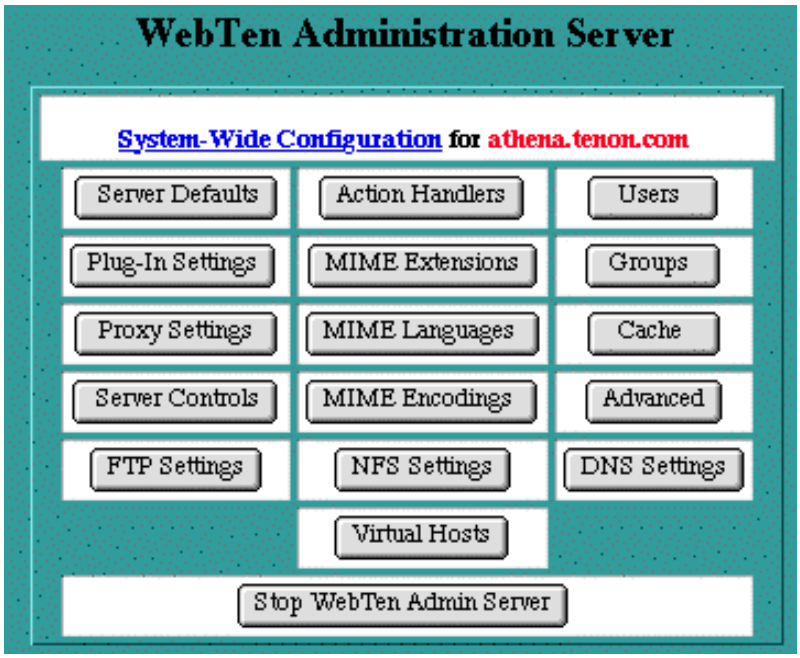


Figure 21: System-Wide Configuration Table

6.1 Server Defaults

There are two key tables in Web[Ten] that control important configuration information for both the default server and any virtual hosts being served — the *Server Defaults* table and the *Virtual Host Configuration* table. Certain items in the *Virtual Host Configuration* table may be inherited (see section "5.3.6 Inheritance") from the initial entries in the *Server Defaults* table.

The *Server Defaults* settings apply to incoming requests that use the default server name. These settings also apply to incoming requests for any virtual host name if the corresponding setting is inherited (i.e., not explicitly set in that *Virtual Host Configuration* table). See section "7.2 Virtual Host Configuration".

To change the *Server Defaults*, modify an existing option or group of options, and click on the *Save Server Defaults* button. If you have not yet saved your changes, use the *Reset* button to restore the information in your browser's page to what it was when the page was first accessed. Note that the *Reset* button does not make any changes to the server's settings; it simply undoes any typing you may have incorrectly entered into this page.

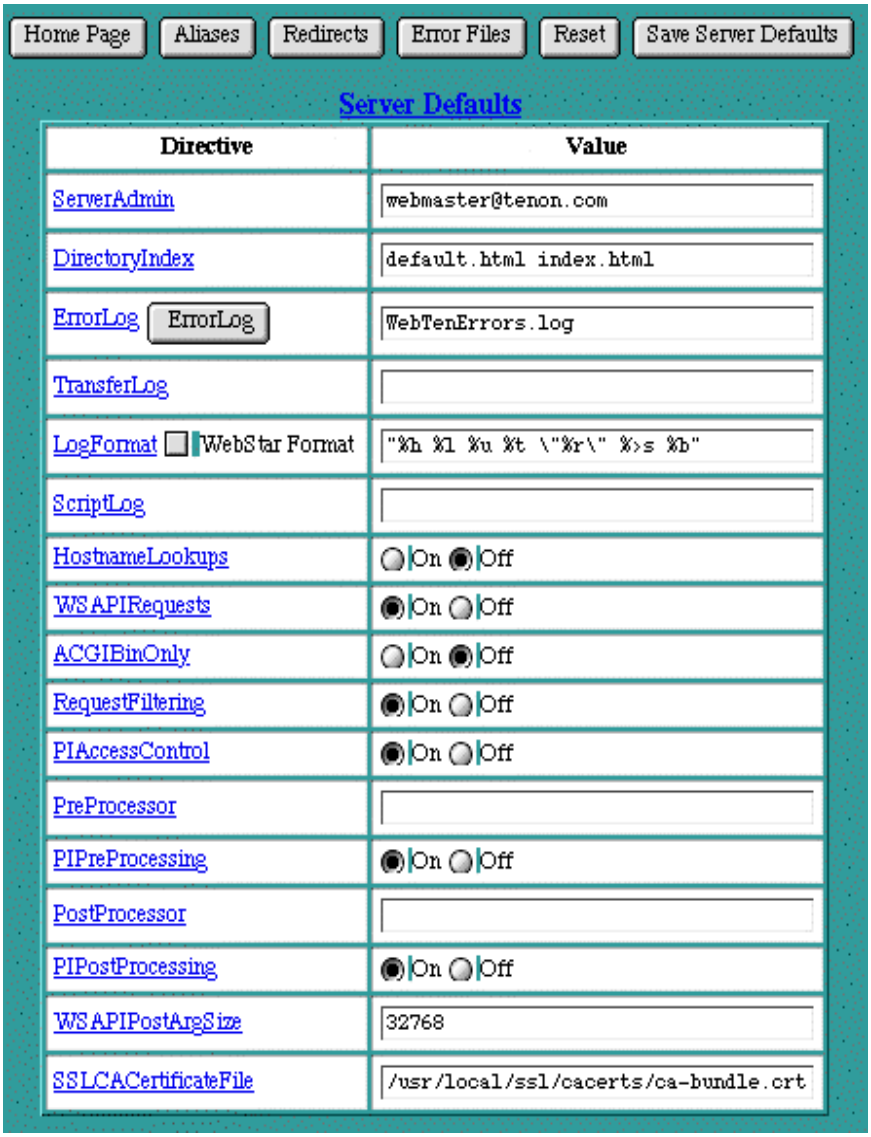


Figure 22: Server Defaults Table

6.1.1 ServerAdmin

The *ServerAdmin* setting is an email address. This address is included in messages sent to a browser whenever a Web server error occurs. Users are encouraged to, and typically do, use this address to notify Webmasters of any problems they are experiencing with a Web server. The established convention is to use the email address "`webmaster@your_domain.com`", but any valid email address is acceptable. The email address must be an existing email address on some other email server. WebTen does not accept incoming email.

In the case of a virtual host, *ServerAdmin* is initially set to the email address "`webmaster@virtualhost`", where "`virtualhost`" is replaced by the virtual host name. Alter this setting to reflect the email address of the Webmaster for this virtual host, or the Webmaster for this WebTen system. Many Web sites follow the convention of using an email address "`webmaster@virtualhost`". To preserve this convention for your WebTen server, add this address to your email server, or make this address an alias to another existing email account on your email server.

If *ServerAdmin* is not set for a particular virtual host, the *"ServerAdmin"* setting is inherited from the *Server Defaults*. In this case, the *ServerAdmin* entry in the *Virtual Host Configuration* table will be flagged with the *Inherited* indicator.

6.1.2 DirectoryIndex

The *DirectoryIndex* setting controls which file is returned when serving a request for a URL that points to a directory (i.e., ending with a "/"). When such a request is made, the *DirectoryIndex* is substituted for the URL, pointing the client request to a default file or CGI. If the *DirectoryIndex* is null, the contents of the directory will be listed on the returned page.

The WebTen default *DirectoryIndex* is "`default.html`", which corresponds to the defaults established by other Macintosh Web servers. The typical Apache setting of *DirectoryIndex* is "`index.html`".

If the *DirectoryIndex* is not set for a virtual host, it will be inherited from *Server Defaults*, and the *Inherited* flag will be displayed.

6.1.3 ErrorLog

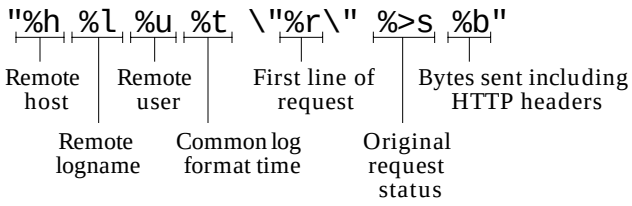
The *ErrorLog* entry in both the *Server Defaults* table and the *Virtual Host Configuration* table is the name of the file Web[Ten] uses to log information about Web server errors. If an *ErrorLog* file is not specifically set for a virtual host, the *ErrorLog* file setting in the *Server Defaults* table will be used.

6.1.4 TransferLog

The *TransferLog* setting is the name of the file Web[Ten] uses to log information about incoming requests. If *TransferLog* is not set for a particular virtual host, it will be inherited from the *Server Defaults*, and flagged accordingly. The *TransferLog* is set to "WebTen.log" by default and will be inherited by all virtual hosts that do not have their own *TransferLog* set.

6.1.5 LogFormat

The *LogFormat* setting is a string that controls the format of the log file. The log file can include literal characters copied from the log format setting and detailed information specific to the actual request that is being logged. Details are encoded using a percent sign ("%") followed by a letter. For example:



Each "%" followed by a letter is a directive to the Web server for a specific piece of information about the request being logged. For example, "%h" logs the name of the remote host placing the request. The order and set of literal characters and details included in the transfer log explicitly follow the order and set of literals and "%" letters in the *LogFormat* setting.

ApacheSSL provides a "c" symbol for custom logging, thus Web[Ten] can be configured with custom SSL log entries using the "c" symbol. For example, a *LogFormat* string to include the SSL version used in an access and the encryption algorithm or cipher used in an access should use:

```
"%{version}c %{cipher}c"
```

If the *TransferLog* is not customized for a particular virtual host, the *LogFormat* setting will be inherited from the *Server Defaults*. This results from the *TransferLog* itself being inherited and utilizing the *Server Defaults' LogFormat*.

Web[Ten] can also create log files in a format compatible with WebSTAR log files. To enable this format, select the *WebSTAR Format* checkbox, and save the virtual host settings.

The cache will keep the log if the accelerator cache is "On". (This is the default.) As such, the *LogFormat* option entered into the Administration Server is not passed to the cache, and the *LogFormat* has no effect.

6.1.6 ScriptLog

The *ScriptLog* setting is the name of the file used to log information about errors in CGI scripts. This feature is meant to be used as an aid in debugging CGI scripts, and should not be used continuously on an active server.

The script log is stored in *CGIErrors.log* file in the top level *logs* folder.
The *ErrorLog*, *TransferLog*, *ScriptLog* and *FTPLog* files are available via the following password-protected URLs:

```
/webten_logs/WebTenErrors.log
/webten_logs/WebTen.log
/webten_logs/CGIErrors.log
/webten_logs/FTP.log
```

6.1.7 HostnameLookups

The *HostnameLookups* setting controls whether reverse DNS lookups are performed for each incoming request using the originator's IP address. Enabling *HostnameLookups* will generally increase the time necessary to satisfy each request, and thus increase the load on your server. However, without *HostnameLookups*, *"Access Controls"* can be based only on IP addresses, not on host names or domain names. If *HostnameLookups* is disabled, IP addresses will be used in the *"ErrorLog"*, *"TransferLog"* and *"FTPLog"*, but these addresses can subsequently be resolved into host names, if necessary.

6.1.8 Plug-In / Apple CGI Settings

After, HostnameLookups, the remaining buttons control various settings for plug-ins and Apple CGIs.

6.1.8.1 WSAPIRequests

The *WSAPIRequests* setting controls whether the Web server will service requests to / from WebSTAR API-style ACGIs and plug-ins. This setting is "On" by default and enables the use of such ACGIs and plug-ins. The *Virtual Host Configuration* table also contains the *WSAPIRequests* entry which, if not specifically set, will be inherited and flagged accordingly.

6.1.8.2 ACGIBinOnly

The *ACGIBinOnly* setting controls whether Apple CGIs are permitted to be executed from within any folder or only from within the *cgi-bin* folder. The default setting is "Off", which enables Apple CGIs to be executed from within any folder. The *Virtual Host Configuration* table also contains the *ACGIBinOnly* entry which, if not specifically set, will be inherited and flagged accordingly.

6.1.8.3 RequestFiltering

The *RequestFiltering* setting controls whether a virtual host will allow "filter" plug-ins to service a request. Filter plug-ins receive the incoming HTTP request before processing has begun. The filter plug-in may modify the request URL before passing it back to Web[Ten] for processing. The default setting is "On", which enables URL filtering within a plug-in. (For more information, see section "6.2 Plug-In Administration".) The *Virtual Host Configuration* table also contains the *RequestFiltering* entry which, if not specifically set, will be inherited and flagged accordingly.

6.1.8.4 PIAccessControl

The *PIAccessControl* flag controls whether security plug-ins may participate in determining whether access should be granted or denied by the server on a per request basis. The default setting is "On", enabling security plug-ins. Security plug-ins may be selectively disabled by each virtual host.

6.1.8.5 PreProcessor

This is a virtual URL to a plug-in acting as a preprocessor (preceded by a slash "/"). *PreProcessors* are run after plug-in filtering is applied and before any access control checking by the server. For example, to have the WebCatalog Plug-in act as a preprocessor, use the entry "/WEBCATALOG_PI" where "WEBCATALOG_PI" is the Action Handler for the WebCatalog Plug-in. See section "6.5 Action Handlers" for more information on Action Handlers.

6.1.8.6 PIPreProcessing

The *PIPreProcessing* flag selectively disables plug-in preprocessors for a virtual host. By default, plug-in preprocessing is enabled and inherited.

6.1.8.7 PostProcessor

This is a virtual URL to a plug-in action, preceded by a slash ("/"). Plug-ins acting as *PostProcessors* receive notification of a completed request. For example, to have the WebCatalog Plug-in act as a postprocessor, use the entry "/WEBCATALOG_PI" where "WEBCATALOG_PI" is the Action Handler for the WebCatalog Plug-in. See section "6.5 Action Handlers" for more information on Action Handlers.

6.1.8.8 PIPostProcessing

The *PIPostProcessing* flag selectively disables plug-in postprocessors for a virtual host. By default, plug-in postprocessing is enabled and inherited.

6.1.8.9 WSAPIPostArgSize

The *WSAPIPostArgSize* setting specifies the argument buffer size for "PUT" and "POST" operators during plug-in and Apple CGI requests. The default size is 32768 bytes.

6.1.8.10 SSLCACertificateFile

This field displays the path to the SSL Certificate Authority file. This path should not be changed under normal circumstances.

6.1.9 Error File Settings

There is a button at the top of each page containing the *Virtual Host Configuration* table and the *Server Defaults* table that allows you access the *Error Files* settings. These settings specify the file to be returned to the client when a Web server error occurs. When such an error occurs, the originally requested page is not returned to the client; instead, the corresponding error file is returned.

To associate an error file to a specific error, select the error code from the pull-down list and type the path to the error file into the text field. Then click the *Save Error Files* button. To change an error code for an existing error file or to change the name of an error file, change the selection in the pull-down list or modify the error file name in an existing text edit field. Then click *Save Error Files* to submit the change.

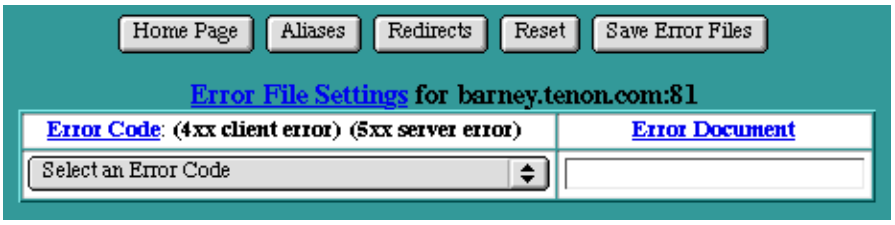


Figure 23: Error Files Table

The two most common errors:

"403: Access to the requested page is denied."

and

"404: The requested page does not exist."

are usually mapped to files with simple messages explaining those errors. However, any of the error cases, from the most common to the most obscure, can be mapped to any URL (including a CGI) for advanced error logging and reporting.

6.1.10 Alias Settings

There is a button at the top of each of the *Virtual Host Configuration* and *Server Defaults* tables that allows you to access the *Alias Settings* for the corresponding virtual host or the default aliases for all virtual hosts.

Alias settings specify components of URLs that are "aliased" or mapped to different folders. When a request is received with a URL that contains one of the aliases, the data returned to the client comes from the specified folder or file.

Aliases may also specify a target folder that contains CGIs (or scripts) rather than normal data. In this case, the alias is referred to as a *ScriptAlias* and is represented in the *Alias Settings* table using a checkbox.

Web[Ten]'s initial *Server Default* settings contain several Aliases used by the Web[Ten] Administration Server, the Web[Ten] documentation, and in the examples. These aliases all begin with the string "`webten_`". The default *cgi-bin* is also specified in this table.

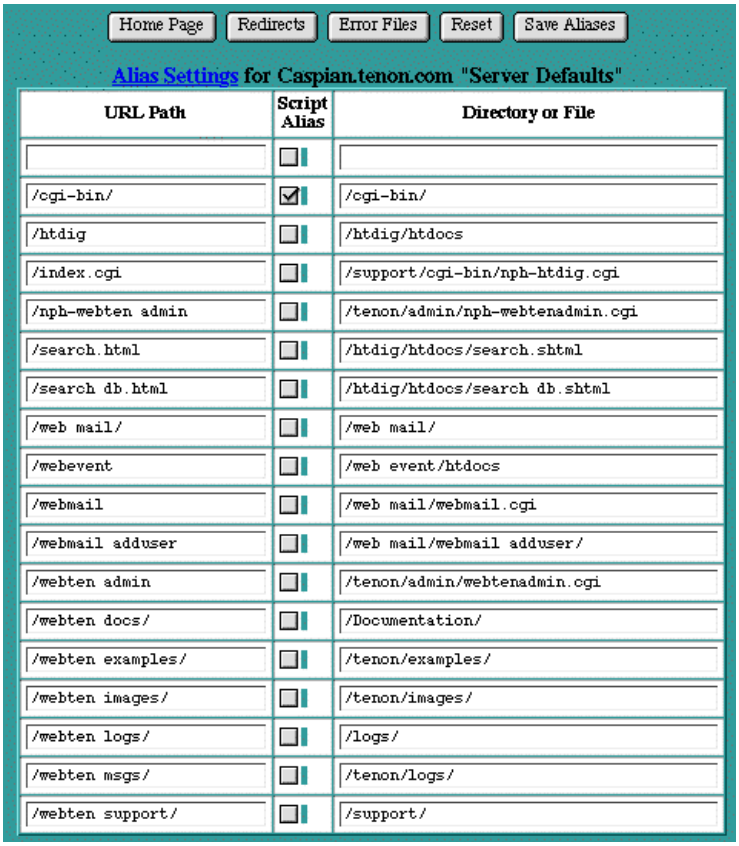


Figure 24: Alias Settings Table

To create a new alias, enter the component of the URL to be aliased into the *URL Path* field of the *Alias Settings* table and enter the path to the folder or file containing the aliased data in the *Directory or File* field. If the *URL Path* or the target represents a folder, it should begin and end with a "/". If it represents a file, it should not end with a "/". If the aliased folder contains CGI scripts, check the *ScriptAlias* checkbox. Click *Save Aliases* to save these settings.

The specified target may reside anywhere within the WebTen hierarchy; it does not necessarily have to reside in the *DocumentRoot* folder for the virtual host servicing the request. The path to the target of the alias always begins in the *WebTen* folder.

6.1.11 Redirect Settings

There is a button at the top of each of the *Virtual Host Configuration* and *Server Defaults* tables that allows you to access the *Redirect Settings* for the corresponding virtual host or the default redirects for all virtual hosts.

Redirect settings specify URLs that are "redirected" or mapped to different servers. When a request is received with a URL that contains one of the redirected entries, the client is instructed (via a return code) to access the data from a different server using the provided URL.Redirect responses contain a reply code and may contain a URL. The reply code can be chosen from a pull-down list.

Web[Ten] does not initially contain any Redirect settings. The following picture shows an example *Redirect Settings* table with a single fictitious entry.

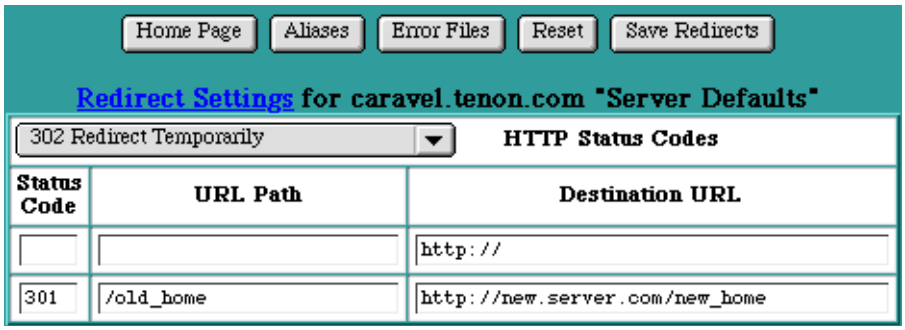


Figure 25: Redirect Settings Table

To create a redirect entry, select the redirect reply code from the pull-down list and enter the URL to be redirected into the *URL Path* field of the Redirect Settings table. If necessary, enter the new URL in the *Destination URL* field. Click *Save Redirects* to save these settings.

Some reply codes require a destination URL and some do not. If you select a reply code that requires a destination URL and do not provide one, an error will be reported. If you select a reply code that does not require a destination URL and one is provided, the destination URL will be discarded when the settings are saved.

6.2 Plug-In Administration

The *WebTen Plug-In Administration* table displays information about each currently installed plug-in. This information includes the name of the plug-in, its version number (if the plug-in provides a version number), its *Action Handler* and its suffix.

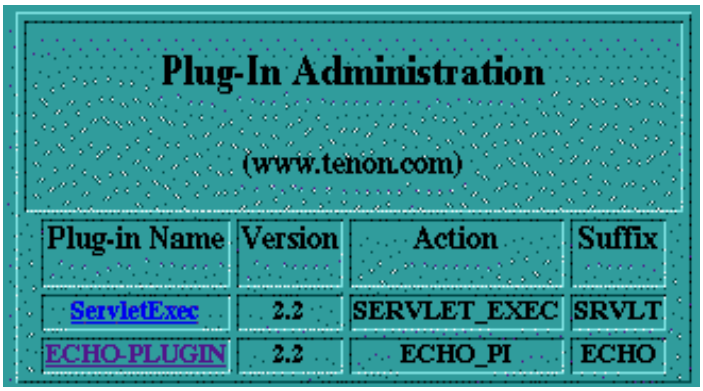


Figure 26: Web[Ten] Plug-In Administration Table

Some plug-ins provide their own Web-based administrative interfaces. In these cases, a link to that plug-in's home page is also provided in the *WebTen Plug-In Administration* table.

If the Web[Ten] cache is enabled, plug-in administration is handled at a different port number than the server's port. This difference may make it necessary to re-enter a Web[Ten] administrator's user name and password when accessing the *WebTen Plug-In Administration* table. Also, if you wish to directly access a plug-in's home page (without first displaying the *WebTen Plug-In Administration* table and following the links it contains), the port number to use will always be one more than the server's port number. For example, if the server is using port 80, the plug-in administration is available on port 81. If the Web[Ten] cache is not enabled, the plug-in administration pages are accessible on the same port number as the Web[Ten] server.

See section "16.0 Plug-Ins and Apache Modules" for instructions on installing Plug-Ins in WebTen.

6.3 Proxy Settings

The *Proxy Settings* table contains some options that control the proxy capabilities of Apache. For more information on Apache and proxy service, see the on-line Apache documentation.

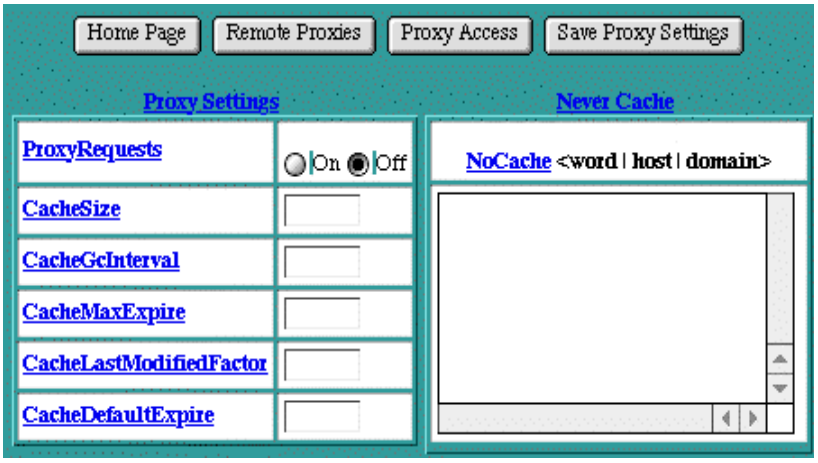


Figure 27: Proxy Settings Table

6.3.1 ProxyRequests

The *ProxyRequests* setting controls whether the proxy service is "On" or "Off". This setting is "Off" by default. If *ProxyRequests* is set to "On", the Squid Accelerator Cache should be turned off.

6.3.2 CacheSize

The *CacheSize* setting controls the disk space, in Kbytes, that the proxy cache files may consume. The proxy service may periodically use disk space beyond this setting, but the proxy "garbage collection" scheme will recover this space after the fact.

6.3.3 CacheGcInterval

The *CacheGcInterval* setting controls the time, in hours, that the proxy "garbage collection" scheme waits between checks to see if the proxy cache size has exceeded its *CacheSize* setting. If it has, files are deleted from the proxy cache until its disk space consumption is less than the *CacheSize* setting.

6.3.4 CacheMaxExpire

The *CacheMaxExpire* setting controls the time, in hours, that a file in the proxy cache may be retained without checking with its origin server. This setting enforces a maximum time that a file may be out of date, even if an expiry date was supplied with the original file.

6.3.5 CacheLastModifiedFactor

If the origin HTTP server did not supply an expiry date for the document, estimate one using the following formula:

```
expiry-period = time-since-last-modification * <factor>
```

For example, if the document was last modified 10 hours ago, and "<`factor`>" is 0.1, then the expiry period will be set to 10*0.1 = 1 hour.

If the expiry period would be longer than that set by *CacheMaxExpire*, the latter takes precedence.

6.3.6 CacheDefaultExpire

If the document is retrieved via a protocol that does not support expiry times, use "<time>" hours as the expiry time. *CacheMaxExpire* does not override this setting.

6.3.7 NoCache

The *NoCache* directive specifies a list of words, hosts and / or domains, separated by spaces. HTTP and anonymous FTP documents matching any words, hosts or domains are not cached by the proxy server. During startup, the proxy module will also attempt to determine IP addresses of any list items which may be host names. These IP addresses will also be cached for use in the match list. In the following example:

```
NoCache some_host.co.uk widgets.doodads.com
```

"widgets.doodads.com" would also be matched if referenced by IP address. Note that "doodads" would also be sufficient to match "doodads.com". Note also that "NoCache *" disables caching completely.

6.3.8 Remote Proxies

Remote proxy servers are other proxy servers that this proxy server may interact with to satisfy a proxy request.

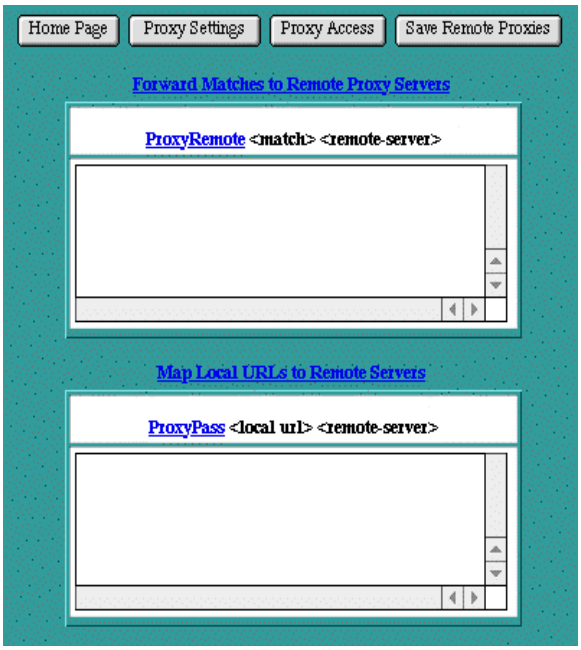


Figure 28: Remote Proxies

6.3.8.1 ProxyRemote

The *ProxyRemote* setting specifies which remote proxy servers are accessible to this proxy server. Each line in the *ProxyRemote* text edit field defines a "`<match>`" string and a "`<remote-server>`" to service URLs that match that string. The match string and the remote server are separated by a space.

The "`<match>`" string is either the name of a URL scheme that the remote server supports, a partial URL for which that remote server should be used, or an asterisk ("*") to indicate that server should be contacted for all requests.

The "<remote-server>" field is the URL for the remote proxy server. Its syntax is "http://<hostname>[:port]". Here are some example entries in the *Remote Proxies* table:

```
http://goodguys.com/     http://mirrorguys.com:8000
*       http://cleversite.com
ftp     http://ftpproxy.mydomain.com:8080
```

In the last example, the proxy will forward FTP requests, encapsulated as yet another HTTP proxy request, to another proxy which will then handle them as FTP requests.

6.3.8.2 ProxyPass

The *ProxyPass* setting allows remote servers to be mapped into the space of the local server. The local server does not act as a proxy in the conventional sense, but appears to be a mirror of the remote server.

Each line in the *ProxyPass* text edit field defines a "<local url>" and a "<remote server>". These fields are separated by a space character.

The "<local url>" is the name of a local virtual path. The "<remote server>" is the URL for the remote server. Suppose the local server has address "http://wibble.org". Typing the following:

```
/mirror/foo     http://foo.com
```

will cause a local request for:

```
http://wibble.org/mirror/foo/bar
```

to be internally converted into a proxy request to:

```
http://foo.com/bar
```

6.3.9 Proxy Access

The *Proxy Access* settings control two things. The *Domain Name Restrictions* control which hosts may use this Web[Ten] server as a proxy server. The *ProxyBlock* acts as a censor list by restricting access to certain URLs, such as pornographic material.

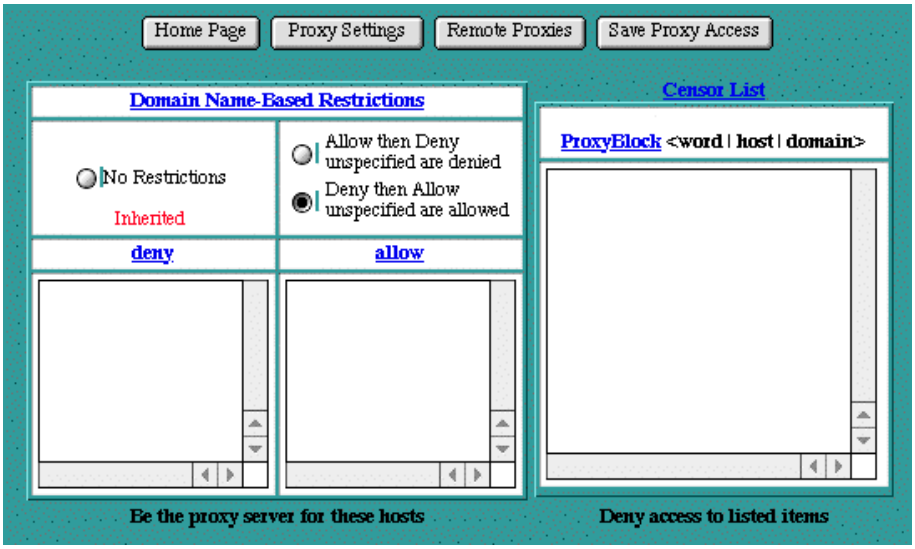


Figure 29: Proxy Access

6.3.9.1 Domain Name Restrictions

The *Domain Name Restrictions* control which hosts may use this Web[Ten] server as a proxy server. These restrictions are applied the same way as Web[Ten] domain name restrictions are applied to any file or folder. See section "7.6.1 Domain Name-Based Restrictions" for more information.

6.3.9.2 ProxyBlock

The *ProxyBlock* directive specifies a list of words, hosts and / or domains, separated by spaces. HTTP, HTTPS and FTP document requests to matched words, hosts or domains are blocked by the proxy server. The proxy module will also attempt to determine IP addresses of list items which may be host names during startup, and cache them for match test as well. For example, if the *ProxyBlock* table contained:

```
nudes
games
some_host.com
```

Access to any URL containing the words "`nudes`" or "`games`" and to "`some_host.com`" would be restricted. "`some_host.com`" would also be matched if referenced by IP address. Note that referencing "`some_host`" would also be sufficient to match "`some_host.com`". Note also that the wild card "`*`" blocks connections to all sites.

6.4 Server Controls

The *Server Controls* table provides some useful information about the current state and version numbers of the Web and cache servers. The buttons on the *Server Controls* page provide a means for the Web[Ten] administrator to examine and control certain aspects of the Web[Ten] server.

The *Server Controls* page first checks on the current state of the Web server. If the server is active, its version number is displayed in the top row of the table; otherwise the word "unavailable" appears. Similarly, if the accelerator cache is active, its version number is displayed in the next row. If the cache has been explicitly turned off in the *Cache Settings*, that row is not displayed in the table.

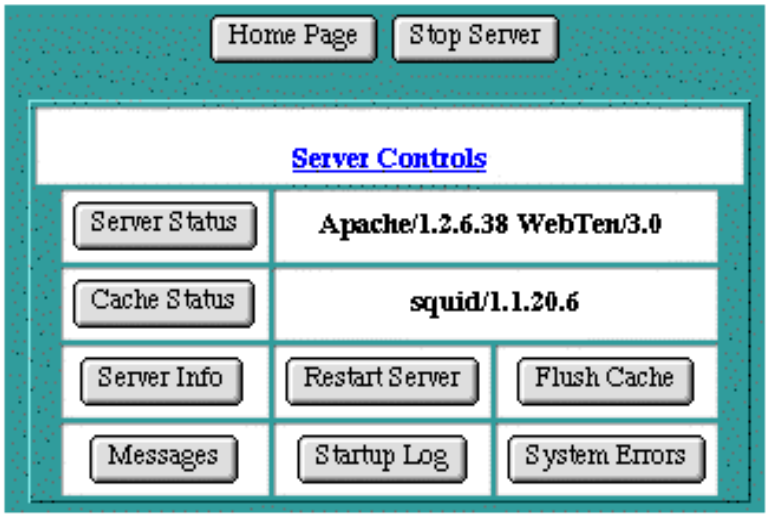


Figure 30: Server Controls Table

6.4.1 Start/Stop Server

If the Web server is active, the *Stop Server* button is provided in the row of buttons above the *Server Controls* table. Clicking on this button will stop the Web server.

If the Web server is not active, the *Start Server* button is provided in the row of buttons above the *Server Controls* table. Clicking on this button will start the Web server.

6.4.2 Server Status

The *Server Status* button provides a connection to Apache's internal status information. This information includes details about the server's version, the current memory available in the system, the time the server was started, CPU usage, and current connections states.

6.4.3 Cache Status

The *Cache Status* button provides a connection to Squid's internal status information. Squid's information is divided into a pull-down list of categories pertaining to all aspects of Squid's operation.

6.4.4 Server Info

The *Server Info* button provides a connection to Apache's current configuration information. This information includes details about which modules are included in this instance of Apache and what its current configuration settings are for each module.

6.4.5 Restart Server

The *Restart Server* button is shown only if the Web server is currently active. Clicking on this button will cause the Web server to completely restart its operation, without shutting down and restarting Web[Ten]. Restarting the server reloads the Apache and Squid configuration files, as well as any plug-ins in the *Plug-Ins* folder. If changes are made directly to these files without using the Web[Ten] Administration Server, or if plug-ins are added to the *Plug-Ins* folder, it is necessary to restart the server in order for these changes to take effect.

6.4.6 Flush Cache

The *Flush Cache* button is shown only if the Web server is currently active and the cache is configured to be "On". Clicking on this button will cause the Web server to completely restart its operation, including a flush of the contents of the accelerator cache.

6.4.7 Messages

During startup, the Web server creates a log file which records the loading of each plug-in it finds. It also records informational Web system messages.

Clicking on the *Messages* button will display the contents of the *WebTen.status* file in the *tenon/logs* folder.

6.4.8 Startup Log

During normal startup, Web[Ten] maintains a log file of the startup activities. Under normal circumstances there is little need to examine this file; however, in the event of startup problems, some useful information may be found here.

Clicking on the *Startup Log* button will display the contents of the *Startup Log* file.

6.4.9 System Errors

This log file is a cumulative record of Web server system errors. Under normal circumstances, there is little need to examine this file; however, errors that may cause the Web server to misbehave or stop serving Web content are recorded here. Clicking on the *System Errors* button will display the contents of the *WebTen.syserrs* file in the *tenon/logs* folder.

6.4.10 Config Log

WebTen's preferences are checked each time WebTen is started. If there have been any changes to the preference settings since the last time WebTen was started, WebTen does some additional startup actions which pass those changes on to the WebTen configuration files. These additional startup actions are recorded in the *WebTen.config* file in the *tenon/logs* folder.

Under normal circumstances there is little need to examine this file; however, some useful information concerning startup and configuration problems may be found here.

6.4.11 WebTen Version Number

WebTen's version number is available from the Mac OS Finder "Get Info" menu command applied to the WebTen application.

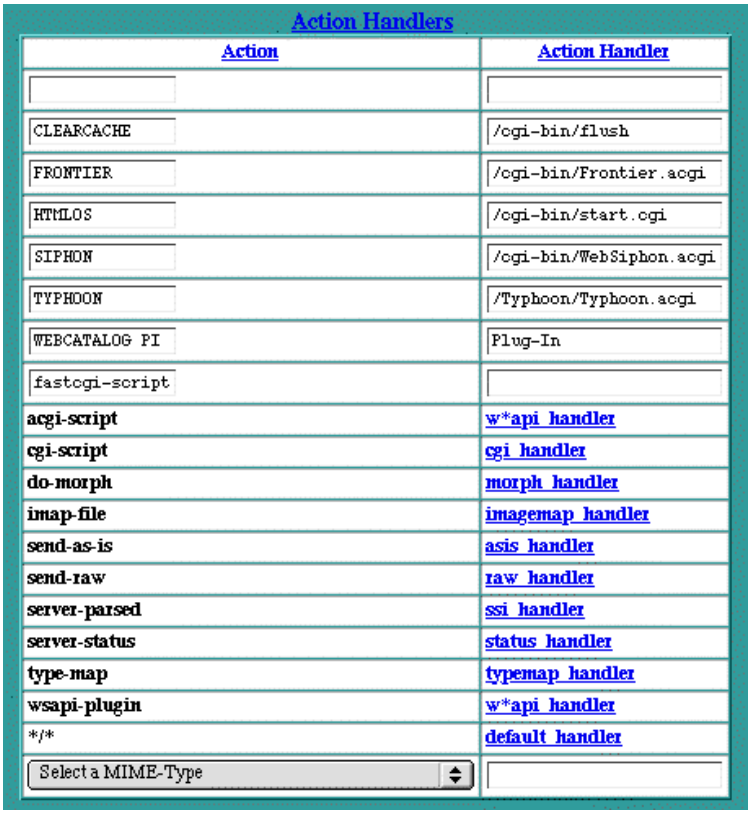
Aall WebTen updates are recorded in the *WebTen.updates* file in the *tenon/logs* folder. This file records the initial version of WebTen that was installed, as well as any updates that are applied.

The *Flush Cache* CGI and the WebTen *Messages*, *Startup Log*, *System Errors*, *Config Log* and *version number* files are available via the following password-protected URLs:

```
/cgi-bin/flush
/webten_msgs/WebTen.status
/webten_msgs/WebTen.startup
/webten_msgs/WebTen.syserrs
/webten_msgs/WebTen.config
/webten_msgs/WebTen.updates
```

6.5 Action Handlers

Action Handlers are an entity internal to Apache. They are used to map files with certain MIME extensions, or files with certain suffixes, to specific actions. These actions may be internal to Apache, or they may be external actions (i.e., CGIs).



Action Handlers

Action	Action Handler
CLEARCACHE	/cgi-bin/flush
FRONTIER	/cgi-bin/Frontier.acgi
HTMLOS	/cgi-bin/start.cgi
SIPHON	/cgi-bin/WebSiphon.acgi
TYPHOON	/Typhoon/Typhoon.acgi
WEBCATALOG PI	Plug-In
fastcgi-script	
acgi-script	w*api_handler
cgi-script	cgi_handler
do-morph	morph_handler
imap-file	imagemap_handler
send-as-is	asis_handler
send-raw	raw_handler
server-parsed	ssi_handler
server-status	status_handler
type-map	typemap_handler
wsapi-plugin	w*api_handler
/	default_handler
Select a MIME-Type	

Figure 31: Action Handlers Table

Before a MIME type or a suffix can be mapped to an action, a handler for that action must be defined. Web[Ten] includes several internal handlers for specific actions. These handlers are displayed in the lower portion of the table and cannot be changed. User-defined handlers can be created for any of the existing MIME types. Use the pull-down list of MIME types or type in a user-defined name, such

as "Frontier" in ""Figure 31: Action Handlers Table"". Enter the path to the external action (CGI), and click on the *Save Handlers* button to submit your changes.

The external actions (CGIs) associated with user-defined handlers must be explicitly added to WebTen. See section "13.0 Using CGIs" for more information.

6.5.1 Configuring Plug-In Actions

Installed plug-ins typically register an action and suffix mapping automatically when WebTen is launched. A plug-in's registered action and suffix are displayed in section "6.2 Plug-In Administration". If a plug-in does not register a suffix or you want to add a suffix to be handled by the plug-in, the *Action Handler* table must be modified.

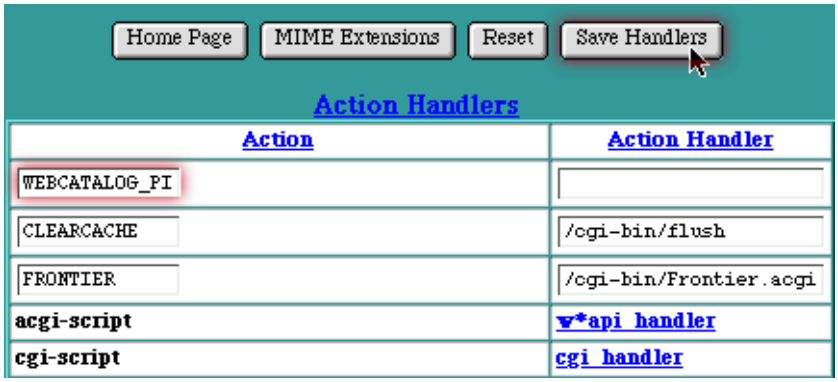


Figure 32: Configuring a Plug-In Action

To configure a suffix for a plug-in, add the registered plug-in action to the empty "*Action*" field. Leave the "*Action Handler*" field blank. Save the new settings. Then, using the *MIME Extensions* table, add the desired suffix extension and map it to the plug-in action.

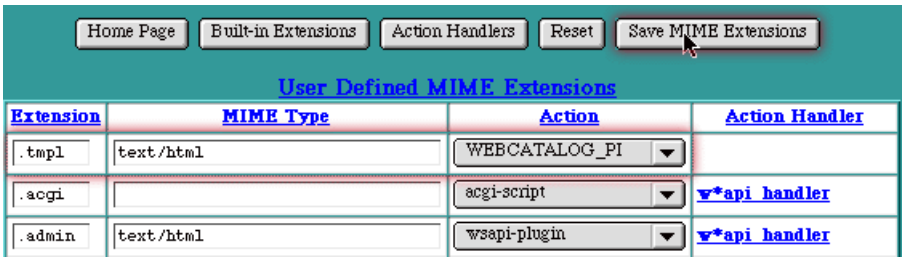


Figure 33: Adding a Plug-In Extension

See "Customizing WebTen" in "Appendix C" for an example of adding an additional suffix to a plug-in.

6.6 MIME Extensions

There are two *MIME Extensions* tables — the *User-Defined MIME Extensions* table and the *Built-In MIME Extensions* table. Both *MIME Extensions* tables map a file name, by its extension, to a MIME type. The extension or MIME type is then mapped to one of the action handlers to control what actions should be taken when any file with this extension is requested. Action handlers can be defined for both MIME types and extensions. If a handler is defined for a specific extension, it overrides any handler specified for that extension's MIME type.

To map a new extension to a MIME type or action handler, enter the new extension into the empty text edit field in the top line of the *User-Defined MIME Extensions* table. Then enter the corresponding MIME type or select a handler from the pull-down list, or do both. Click *Save MIME Extensions* to submit the changes.

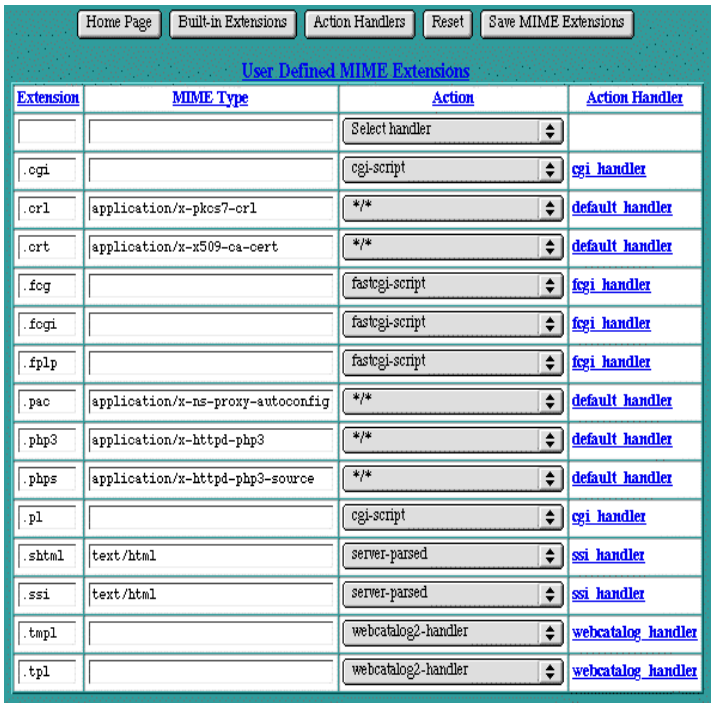


Figure 34: MIME Extensions Table

To change an existing extension, its MIME type, or its handler, modify the extension or MIME type in the text edit field or select a different handler from the pull-down list. Then click on *Save MIME Extensions* to submit the changes.

Web[Ten] includes a long list of well-known extensions and their corresponding MIME types. These extensions are displayed in the *Built-In MIME Extensions* table, accessible via the *Built-In Extensions* button, and cannot be explicitly changed. However, these default extensions can be overridden by entering the extension in the empty text edit field in the *User-Defined MIME Extensions* table, and assigning it a different MIME type. This extension will then appear in that table, and the default setting will no longer appear in the *Built-In MIME Extensions* table. If this extension is subsequently removed, the default setting will remain and will reappear in the *Built-In MIME Extensions* table. Overriding the default extensions in the *Built-In MIME Extensions* table is not recommended, as this setting affects all files with this extension on this server. To explicitly override the default MIME type settings for a specific file or folder, see section "7.6.3 MIME Type Overrides".

6.6.1 The MIME Typing System

The HTTP protocol requires every document served by a Web server to have a "type". By examining the type, a browser can determine how to display the information. For example, an HTML document simply needs to be formatted, a graphics document may require rendering, and an audio file will need to be passed to an application that can deal with the computer's sound system.

MIME (Multipurpose Internet Mail Extensions), originally developed to support multimedia internet mail, is used by the HTTP protocol to describe a document's content. The MIME typing system allows virtually any type of document to be displayed or executed through a browser.

A common way to distinguish one file type from another is to add a distinctive extension to its name. When a browser requests a particular file, the HTTP server determines its MIME type by looking at the file's extension.

Web[Ten] may be configured (on a directory or file basis) to map any file name extension to any MIME type (see section "7.6.4 Action Handler Overrides"). The following table contains some of the basic file extensions and their interpretation:

File Extension	MIME Type	File Type
.html	text/html	Hypertext Markup Language
.raw		Raw Text
.gif	image/gif	GIF Format
.tiff	image/tiff	TIFF Format
.jpeg	image/jpeg	JPEG Format
.mpeg	video/mpeg	MPEG Movie Format
.qt	video/quicktime	QuickTime Movie Format
.snd	audio/basic	Basic Sound Format

Figure 35: MIME Types and File Extensions

Apache includes a lookup table to determine which MIME type to use, based on the file name extension. A complete list of MIME types is displayed in the MIME Extensions table (see section "6.6 MIME Extensions") accessible via the Web[Ten] Administration Server.

6.7 MIME Languages

The *MIME Languages* table provides a means for mapping a file name, by its extension, to a language. The Web server takes no special action based on the language, but the given language is passed back to the client (in the HTTP header) for any specific interpretation in the browser.

To map a new file name extension to a language, enter the extension in the empty text edit field in the first row of the table, and select a language from the pull-down list. Then click *Save MIME Languages* to submit the new setting.

To change an existing setting, either modify the extension in the text edit field or select a new language from the pull-down list. Then click *Save MIME Languages* to submit the changes.

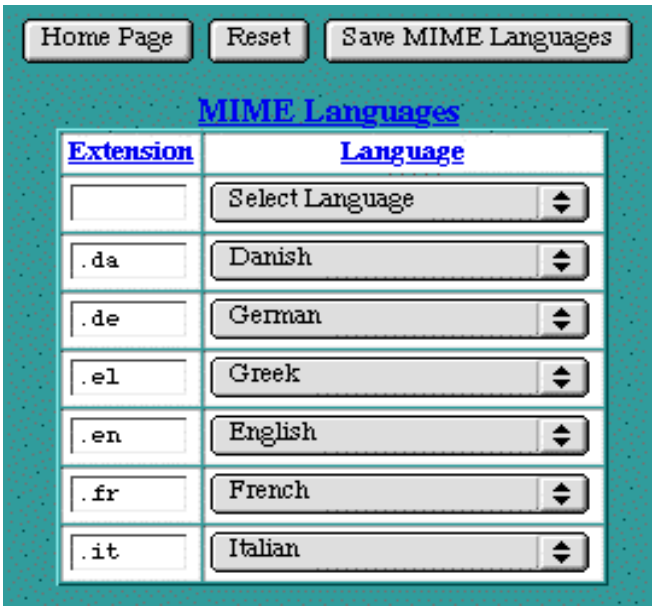


Figure 36: MIME Languages Table

6.8 MIME Encodings

The *MIME Encodings* table provides a means for mapping a file name, by its extension, to a MIME encoding. The Web server takes no special action based on the encoding, but the given encoding is passed back to the client (in the HTTP header) for any specific interpretation in the browser.

To map a new file name extension to an encoding, enter the extension in the empty text edit field in the first row of the table, and enter an encoding in the second text edit field. Then click *Save MIME Encodings* to submit the new setting.

To change an existing setting, modify the extension or the encoding in their respective text edit fields. Then click *Save MIME Encodings* to submit the changes.

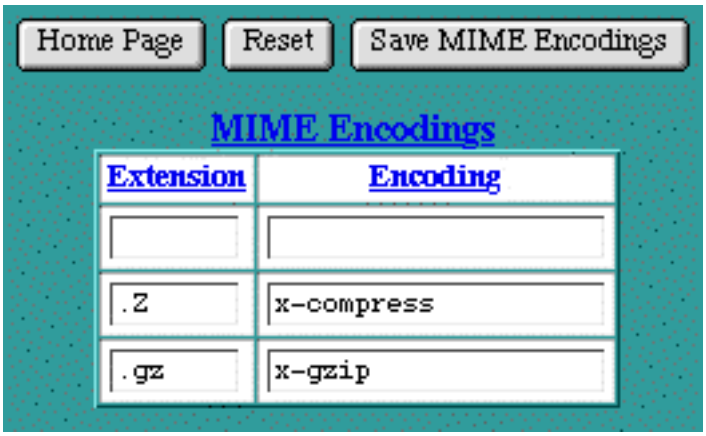


Figure 37: MIME Encodings Table

6.9 Users

Web[Ten] provides a set of realm-based access controls that can restrict access to a particular file or folder based on user names and passwords. (See section "7.6.2 Realm-Based Requirements".) Web[Ten] also provides FTP service based on user names and passwords. User names and passwords for both realm-based access controls and FTP service are entered in the *Users* table.

To enter a new user name and password, type the user name into the empty text edit field in the first row of the table. Type a corresponding password into the second text edit field. The password will not be displayed as it is typed. Instead, bullet characters will be displayed (so type carefully). Click the *Save Users* button to submit the new user name and password.

Click on the FTP checkbox to enable FTP access for this user. If FTP access is enabled, select an *FTP Home* for this user. The *FTP Home* is the folder that this user will be given access to when they FTP into Web[Ten]. Users can be restricted to access only a particular virtual host, only the anonymous FTP hierarchy, or they can be allowed access to all of the virtual hosts, including the anonymous FTP hierarchy. If no *FTP Home* is selected, the default allows access to all of the virtual hosts. Of course, this is enabled only if the FTP checkbox is checked.

Once a user name and password have been entered and the form has been submitted, the new entry will show up in the table. Passwords are always displayed as if they contain eight characters, regardless of how many characters are actually in the password.

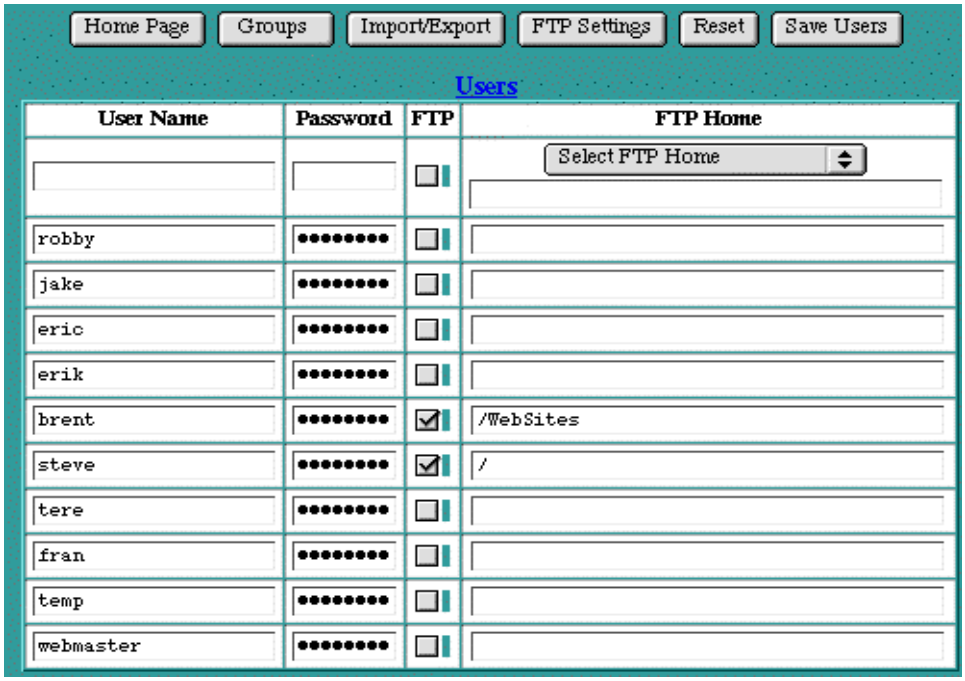


Figure 38: Users Table

To change an existing user name or password, modify the user name, password, FTP checkbox or *FTP Home* settings and click *Save Users* to submit the changes. To delete a user, delete the user name for that user. Click *Save Users* to submit the changes.

6.10 Groups

Web[Ten] provides a set of realm-based access controls that can restrict access to a particular file or folder based on groups of users (each with their own password).

To enter a new group, type the group name into the empty text edit field in the first row of the table. Click the *Save Groups* button to submit the new group. Once a group has been entered, the new entry will show up in alphabetical order in the *Groups* table.

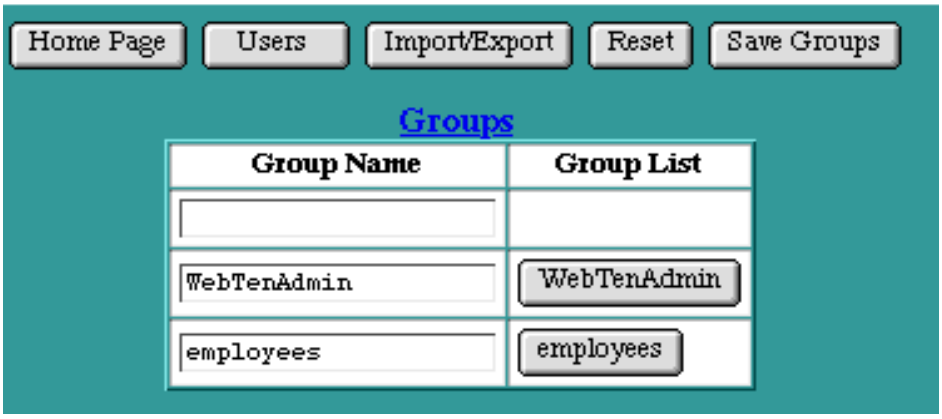


Figure 39: Groups Table

To change an existing group name, modify the name in the text edit field and click *Save Groups* to submit the change.

To select which users are to be members of a group, click on any button in the *Group List* column. The "*Users in Group*" table will be displayed.

The Web[Ten] Administration Server uses a special group named *WebTenAdmin*. Members of this group are permitted access to the Web[Ten] administration pages, and may make changes to the Web[Ten] configuration, including adding and deleting users and groups. If the *WebTenAdmin* group is deleted, or if this group is empty, access to the Web[Ten] Administration Server is completely cut off. In this case, use the *Admin* menu item and follow the instructions in section "4.4.1 Set Admin Password" to add an initial user to this special *WebTenAdmin* group.

6.10.1 Users in Group

The *Users in Group* table controls which users are included in a specific group. To select users for inclusion in a group, click on each privileged user within the scrollable list of all users. Simply clicking on a user's name will select that individual user. Hold the <shift> key and click to select a series of users, or hold the <Apple> key (<control> key on non-Macs) to individually select any combination of users. When a user is selected for inclusion in the group, the user's name will be highlighted. Click on *Save Users in Group* to submit the selected users.

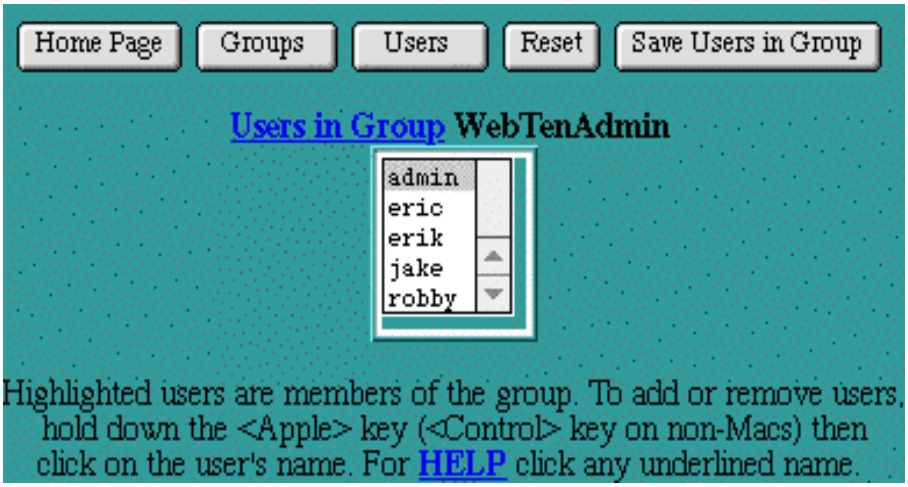


Figure 40: Users in Group Table

6.10.2 Import and Export

Import and *Export* provide a means to manage WebTen's *Users* and *Groups* databases from a text file. The file contains one entry per line, listing a user's name, group, and password. Importing from such a file will read each line of the file, extract valid entries, and append them to the *Users* and *Groups* databases as appropriate. Conversely, exporting the *Users* and *Groups* databases creates a text file (suitable for editing and subsequent importing) containing a line for each entry in the current *Users* and *Groups* databases.

The *Import Users* and *Export Users* buttons are accessible from either the *Users* table or the *Groups* table. Clicking on these buttons presents a simple form for entering the name of the file to be imported from or exported to, and buttons to select either the *Import* or *Export* action.

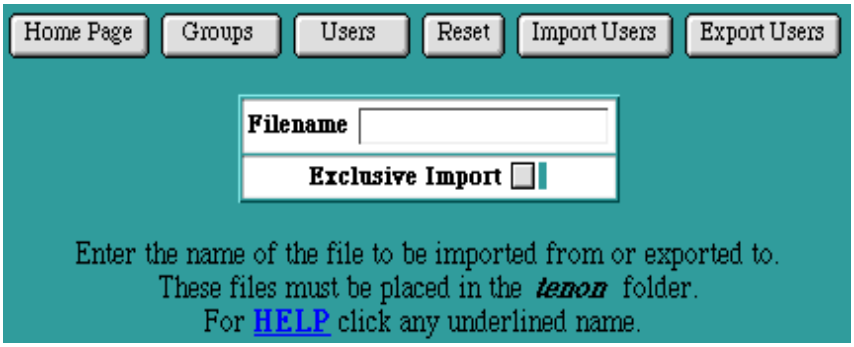


Figure 41: Import and Export Users and Groups

6.10.2.1 Exporting

To export the current *Users* and *Groups* databases, type in a file name and click on the *Export Users* button. The exported file will be placed in the *tenon* folder and will overwrite any existing file of the same name. A table of exported entries is also displayed in your browser.

The exported file is a text file with a Macintosh creator and type of MUMM / BINA. Depending on your choice of Macintosh text editor, it may be necessary to convert this file to type "TEXT" before reading it.

6.10.2.2 Importing

To import a list of user names, groups and passwords, it is first necessary to create a file of the proper format. Typically, the easiest way to get started is to create a file by exporting the current *Users* and *Groups* databases. However, you can create an import file from scratch by following the format below. Web[Ten]'s import format is also compatible with the import file format used by WebSTAR. Import files must be converted to creator and type "MUMM/TEXT" or "MUMM/BINA" before being imported into Web[Ten].

Importing can either append the imported entries into the existing *Users* and *Groups* databases, or it can be an "exclusive" import. "Exclusive" imports completely replace the existing *Users* and *Groups* databases, and thus create only the entries found in the imported file.

Place the file to be imported in the *tenon* folder. If the import is to be an "exclusive" import, select the *Exclusive Import* checkbox. Type the name of the file into the *Import/Export* form and click on the *Import Users* button. A table reporting on the success of each imported entry is displayed in your browser. Imported entries are appended to the previously existing set of *Users* and *Groups*.

6.10.2.3 File Formats

Each line of the *Import* and *Export* files must be formatted as follows. Blank lines and lines beginning with a "#" (comment lines) are ignored.

`username•groupname`	unencrypted-password
`username*groupname`	encrypted-password
`username*groupname`	encrypted-password <*ftp-home*>

Note that the "•" (<option>-8) separating the user name and group name indicates that the password is unencrypted, while "*" (<shift>-8) indicates that the password is encrypted. The user name field must begin in the first column of the file. Every user name in Web[Ten] must be unique. If a user is a member of more than one group, one line must exist (with the same user name and password) for each group to which the user belongs.

The *ftp-home* entry is optional. If it exists, FTP access is enabled for the user in the folder indicated by <*ftp-home*>. If the *ftp-home* entry is omitted, FTP access is disabled.

Web[Ten] will accept either encrypted or unencrypted imported passwords. When Web[Ten] exports passwords, it writes out the passwords in the encrypted form. This is more secure than writing out passwords in unencrypted form, as anyone who reads or accesses the exported file will not have access to the user's passwords, and thus will not be able to access that user's realms. The encrypted passwords are, however, still acceptable for copying or modifying the user name and group entries before importing these changes back into the Web[Ten] system. Also note that in a default Web[Ten] installation, access to the *tenon* folder is restricted. Therefore, exported files are not accessible to anyone browsing this server.

6.11 Cache Settings

Clicking the *Cache* button reveals a *Cache Settings* table. The *Cache Settings* table contains options that control the Web[Ten] memory cache. This cache is object-based and keeps the most recently accessed Web pages in memory, making these pages immediately accessible for subsequent requests. Web[Ten]'s high-performance benchmarks are achieved via extensive use of this memory cache. After changing the *Cache Settings*, click on the *Save Cache Settings* button to preserve your changes.

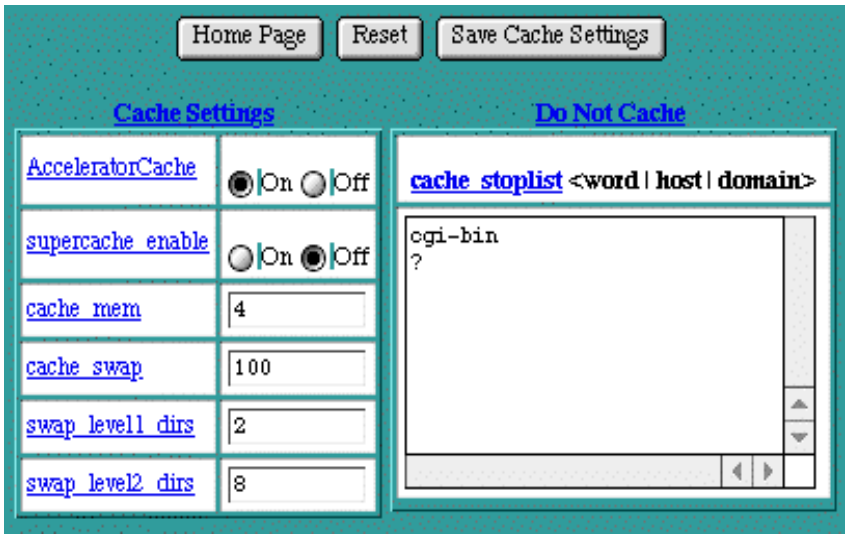


Figure 42: Cache Settings Table

6.11.1 AcceleratorCache

The *AcceleratorCache* setting controls the Squid cache. The default setting is "On". Turning the cache to "Off" will save some memory, so this setting might be useful for servers that are running low on memory. Turning the cache to "Off" will also affect the performance of the server.

6.11.2 supercache_enable

The *supercache_enable* setting controls whether Web[Ten]'s high-performance caching capability is enabled. This feature is disabled by default. If you are logging to disk (which most webmasters do,) then you will not be able to take advantage of supercache because the supercache operates on such a low level that its activity can not be logged.

6.11.3 cache_mem

The *cache_mem* setting controls how much memory, in Mbytes, the cache will use. This setting represents the high-water mark for memory use. The cache will only consume as much memory as it needs, up to this value. The default setting is 4 Mbytes.

6.11.4 cache_swap

The *cache_swap* setting controls how much disk space the cache will use in Mbytes the cache will use. This setting represents the high-water mark for disk usage. The cache will only consume as much disk space as it needs, up to this value. The default setting is 100 Mbytes.

6.11.5 swap_level1_dirs

The *swap_level1_dirs* setting controls how many level 1 (top level) directories the cache will use to organize its cached entries on the disk. The default setting is 2.

6.11.6 swap_level2_dirs

The *swap_level2_dirs* setting controls how many level 2 (second level) directories the cache will use to organize its cached entries on the disk. The default setting is 8.

6.11.7 cache_stoplist

The *cache_stoplist* setting is a list of words or characters. A URL containing any of these values is not cached. The default setting is to not cache URLs containing "`cgi-bin`" or "?". Other words may be added to this list.

6.12 Advanced Settings

The *Advanced Settings* table contains some options that control the inner workings of the Web server. Your choice for these settings may be influenced by certain conditions, such as how much memory the WebTen system has, the expected rate of "hits", the size of the average transfer, the number of simultaneous transfers, and the access bandwidth of the Web server or the clients.

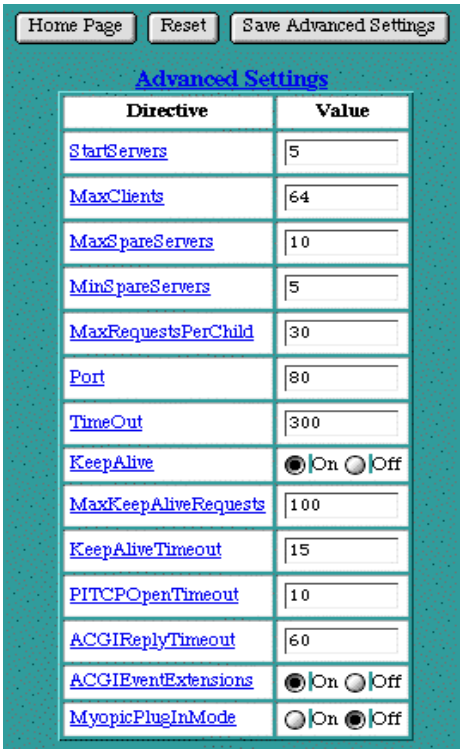


Figure 43: Advanced Settings Table

6.12.1 StartServers

The *StartServers* setting controls how many Web server processes are created when the server is initially started. The number of Web server processes may be dynamically changed (depending on the server's load), so changing this setting has minimal effect once the server is up and has serviced its first few requests.

6.12.2 MaxClients

The *MaxClients* setting controls the number of requests that can be processed simultaneously. If the *MaxClients* are concurrently in progress, subsequent requests are not necessarily lost. Instead, they are queued until an existing request has completed.

6.12.3 MaxSpareServers

The *MaxSpareServers* setting controls the number of idle (i.e., not currently servicing any request) Web server processes. If the number of idle processes exceeds this number, the excess processes are terminated.

6.12.4 MinSpareServers

The *MinSpareServers* setting controls the number of idle (i.e., not currently servicing any request) Web server processes. If the number of idle processes is smaller than this number, extra Web server processes are instantiated at a rate of one per second.

6.12.5 MaxRequestsPerChild

The *MaxRequestsPerChild* setting controls the number of requests each Web server process will service. Web server processes service one request at a time. However, upon completing one request, they may begin servicing another.

Increasing the number of requests each Web server process services reduces the overhead of instantiating and terminating Web server processes. Restricting this number reduces the likelihood of accidental loss of system resources, as these resources are recovered when a process exits. Also, the dynamic control over the number of currently running processes responds to a reduction in load by allowing some Web server processes to exit without instantiating replacements. Therefore, in this case, a smaller number of *MaxRequestsPerChild* leads to a faster reduction in Web server processes.

If the *MaxRequestsPerChild* is set to zero, a Web server process will never expire.

6.12.6 Port

The *Port* setting controls on which port number the Web server accepts incoming connections. The Web server accepts incoming connections on all its IP addresses using the port number specified in the *Port* setting.

6.12.7 TimeOut

The *TimeOut* setting controls the maximum time (in seconds) that the Web server will wait for receipt of a complete incoming request once any initial part of an incoming request is received. The *TimeOut* setting also controls the maximum time the Web server will wait to completely send a response. If the sizes of the files used in the Web transfers are large, and the client's or server's network bandwidth is slow, the *TimeOut* setting must be increased to compensate.

6.12.8 KeepAlive

The *KeepAlive* setting controls whether or not the Web server permits multiple incoming requests (from a single client) in a single connection. Using *KeepAlive* reduces the overhead of connection establishment and termination for each incoming request.

6.12.9 MaxKeepAliveRequests

The *MaxKeepAliveRequests* setting controls the number of incoming requests a client may embed in a single connection. The *MaxKeepAliveRequests* setting is ignored if "*KeepAlive*" is "Off".

6.12.10 KeepAliveTimeout

The *KeepAliveTimeout* setting controls the length of time (in seconds) the Web server will wait for additional incoming requests in a single connection. If the *KeepAliveTimeout* expires, a client can still send additional requests; however, a new connection establishment overhead is incurred. The *KeepAliveTimeout* setting is ignored if "*KeepAlive*" is "Off".

6.12.11 PITCPOpenTimeout

The *PITCPOpenTimeout* setting controls how long (in seconds) Web[Ten] will wait for a connection to be established when a plug-in attempts to open a TCP connection. The default setting is ten seconds.

6.12.12 ACGIReplyTimeout

The *ACGIReplyTimeout* setting controls how long (in seconds) Apple CGIs are given to complete their operation and return their results. The default setting is 60 seconds.

6.12.13 ACGIEventExtensions

The *ACGIEventExtensions* setting controls whether Web[Ten] adds custom virtual host parameters to the "sdoc" Apple Event sent to Apple CGIs during request processing. For backward compatibility with older Macintosh CGIs, *ACGIEventExtensions* may need to be "Off".

6.12.14 MyopicPlugInMode

A number of plug-ins and CGIs designed for Macintosh Web servers that do not support virtual hosting are in wide use today. We refer to these plug-ins as "Virtual Host-Challenged", or myopic plug-ins. Myopic plug-ins and CGIs assume that a Web server supports a single, static, top-level document root and that virtual hosting is accomplished by prepending a unique path or folder name to each request for a virtual host's content. (The standard Apache way of supporting virtual hosts is to allow each virtual host to have a unique, top level document root.)

When *MyopicPlugInMode* is "Off", Web[Ten] supports any numbers of document roots (one for each virtual host) and there is no need to prepend anything to requests for a virtual host's content. With this default "Off" setting, myopic plug-ins and CGIs will not work properly for any of Web[Ten]'s virtual hosts, other than the default virtual host.

When *MyopicPlugInMode* is "On", Web[Ten] checks and filters each incoming request. If the request is for a virtual host and that virtual host's *DocumentRoot* is not explicitly set (i.e., the *DocumentRoot* setting is inherited from the default virtual host) and the virtual host has an explicit *ServerPath* (*ServerPath* specifies the sub-folder where that virtual host's content resides), the *ServerPath* will be prepended to the URL. In this mode, myopic plug-ins or CGIs will work with any of Web[Ten]'s virtual hosts.

Myopic CGIs require some additional configuration under Web[Ten]. A Finder alias of the CGI application residing at the Web[Ten] root level folder must be placed in each virtual hosts's *DocumentRoot* folder for proper operation with a Web[Ten] virtual host. The Action Handler (see section "6.5 Action Handlers") for the myopic CGI should be of the form:

```
/<thecgi>.acgi
```

where `<thecgi>` is the name of the CGI.

When *MyopicPlugInMode* is turned "On" and the *Advanced Settings* are saved, the Web[Ten] Administration Server checks and modifies the configuration of each existing virtual host. If the virtual host's *DocumentRoot* matches its *ServerPath* (this is the default when virtual hosts are created with *MyopicPlugInMode* "Off"), the *DocumentRoot* setting is changed to *inherited*, making this virtual host accessible to work with myopic plug-ins.

When *MyopicPlugInMode* is turned "Off" and the *Advanced Settings* are saved, the Web[Ten] Administration Server checks and modifies the configuration of each existing virtual host. If the virtual host's *DocumentRoot* is inherited (the default when virtual hosts are created with *MyopicPlugInMode* "On"), the *DocumentRoot* setting is changed to match the *ServerPath* setting, restoring the virtual host's configuration for proper operation without concessions for myopic plug-ins.

Any individual virtual host can override *MyopicPlugInMode* (when it is "On") by clearing its *ServerPath* setting or by explicitly adding a *DocumentRoot* setting in that virtual host's configuration.

Contact your plug-in vendors for the latest information about their plug-ins and their compatibility with Web[Ten]'s true virtual hosting.

6.13 Direct Access to Configuration Files

When you configure Web[Ten] by using the browser-based administration tool, the Web[Ten] user experience is exactly like using a Macintosh application. However, because the components that make up Web[Ten] are so rich in features, we have made the native Apache and Squid configuration files accessible. This means that UNIX-savvy Macintosh users can use any Macintosh editor to directly edit the raw Apache and Squid configuration files.

The advantages of this dual approach are many fold. Having the configuration files available makes it easy to later configure other systems in a similar way. Having access to the configuration files means that expert users can take advantage of features that are not yet exported into the point-and-click browser interface. Finally, users can easily integrate new Apache modules into Web[Ten] without having to wait for browser-aware versions to come from Tenon. The best of both worlds — UNIX and Macintosh tightly integrated with no surprises either way.

6.13.1 Macintosh File Creators and File Types

Every Macintosh file has a name, as well as "creator" and "type" fields. These four-letter fields indicate which application created the file and what kind of a file it is. Macintosh files also have two forks, or streams of data — the "data fork" and the "resource fork". When Web[Ten] is serving a file, it serves the contents of the data fork of that file. This is consistent with other Macintosh Web servers. Macintosh Web content creation tools are also designed with this in mind (i.e., they save their Web content in the data forks of their resultant files).

In order to serve both forks of a Macintosh file (e.g., to deliver a Macintosh application or a Microsoft Word document in its complete Word format), it is necessary to convert these files to a format suitable for transferring on the Web. Many conversion and compression tools exist on the Macintosh that encode both forks of the original Macintosh file into the data fork of a new file. This new file may then be transferred on the Web, and re-converted to its original format on the destination system. File name extensions and MIME types are used to tell the destination systems what tool and format the original file was converted with.

CGIs are not "served" in this traditional sense, rather they are launched and run on the Web server. The output they produce is passed back to the browser. In order to run a Web[Ten] binary, Perl or shell CGI, the CGI must be set to the proper

creator or type. Web[Ten] requires the creator "MUMM" and type "BINA" for binary CGIs, and the type "TEXT" for Perl and shell CGIs. CodeBuilder (a companion development tool from Tenon) automatically produces files with the proper creator and type when building binary CGIs for Web[Ten]. Perl and shell CGIs are simply text files, and can be produced with any Macintosh text editor.

When using Web[Ten]'s built-in FTP to upload CGI scripts to the */cgi-bin* directory, the appropriate execution settings for CGI scripts will automatically be turned on. Likewise, when using Web[Ten]'s FTP to upload text, the correct creator and type fields and carriage return / line feed mappings are generated.

7.0 Virtual Hosts

Apache provides the capability to support multiple servers on a single machine, each differentiated by a unique host name. This feature is called virtual hosting. For example, it is often desirable for companies sharing a Web server to have their own domains, with Web servers accessible as "`www.company1.com`" and "`www.company2.com`", without requiring the user to know any extra path information.

7.1 Virtual Hosts Table

The WebTen Administration Server includes a table of virtual host names. This table initially will include a single virtual host, which is the fully qualified domain name of the system on which WebTen is running. In ""Figure 44: Virtual Hosts Table"", that name is "`barney.tenon.com`". It also lists the virtual hosts "`betty.tenon.com`", "`fred.tenon.com`", and "`holly.tenon.com`".

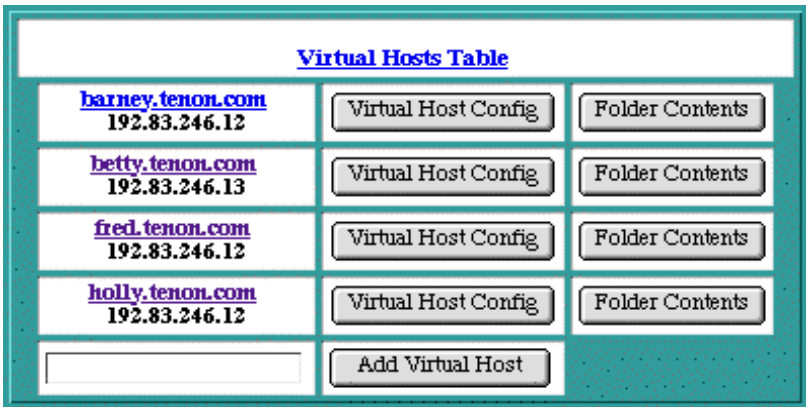


Figure 44: Virtual Hosts Table

7.1.1 Adding Virtual Hosts

Additional virtual host names can be entered directly into the *Virtual Hosts Table*. Simply type the new virtual host name into the empty text edit field. The Domain Name Server is consulted to determine if the host name is an IP-based virtual host or a header-based virtual host. In either case, simply enter the host name, and the proper virtual host type will be created. Click on the *Add Virtual Host* button (or hit the <Return> key) to submit your new virtual host entry.

If you are using a host name, the new host name must be properly configured with your Domain Name Server before WebTen will accept it. If necessary, an IP address can be used in place of a host name. Actual host names can be entered into the table after they have been configured in DNS. Using a host name is preferable to using an IP address, as users will remember a name more readily than a number.

Each virtual host has its own *Virtual Host Configuration*. These settings are accessible via the *Virtual Host Config* button. See section "7.2 Virtual Host Configuration" for more information.

7.1.2 Deleting Virtual Hosts

To delete virtual hosts from the *Virtual Hosts Table*, click on the *Virtual Host Config* button beside the virtual host you wish to delete. Select the *Delete Virtual Host* check box at the bottom of the *Virtual Host Configuration* table. Click on the *Save Virtual Host Config* button to submit the changes. The browser will return to the WebTen Administration Server home page and the *Virtual Hosts Table* should no longer contain the deleted host name.

The default virtual host (the one with the same virtual host name as the fully qualified domain name of the machine running the Web server) does not have the *Delete Virtual Host* check box because it cannot be deleted.

7.2 Virtual Host Configuration

When a virtual host is added to the Web[Ten] configuration, the Web[Ten] Administration Server sets up an initial *Virtual Host Configuration* for the new virtual host. These settings apply to incoming requests that explicitly use this virtual host's name. Each virtual host supported by the Web server corresponds to a unique folder that contains a unique set of Web pages. Client requests using the virtual host name are mapped to the corresponding folder. This prevents the client from unintentionally accessing other virtual hosts' Web pages. The client is unaware that its request is actually supported by a virtual host entry on a Web server with several virtual hosts. To access the *Virtual Host Configuration* table, click on the *Virtual Host Config* button beside the name of the virtual host you wish to configure.

For each folder or file within a virtual host's content, Web[Ten] permits unique *Domain Name-Based Restrictions*, *Realm-Based Requirements*, *MIME Type Overrides*, and *Action Handler Overrides* settings. These settings are accessible via the *Folder Contents* button. See section "7.5 Folder Contents" for more information on finding individual files or folders, and section "7.6 Access Controls" for more information on setting access controls and other overrides for any file or folder served by Web[Ten].

To change the virtual host settings, modify an existing setting or group of settings and click on the *Save Virtual Host Config* button. If you have not yet saved your changes, you can use the *Reset* button to restore the information in your browser's page to what it was when the page was first accessed. Note that the *Reset* button does not make any changes to the virtual host's settings; it simply undoes any typing you may have incorrectly entered into this page.

The following entries in the *Virtual Host Configuration* table are also present in the *Server Defaults* table: *ServerAdmin*, *ServerName*, *DirectoryIndex*, *ErrorLog*, *TransferLog*, *LogFormat* and *HostnameLookups*. For more information on these items, please refer to the corresponding sections in section "6.1 Server Defaults".

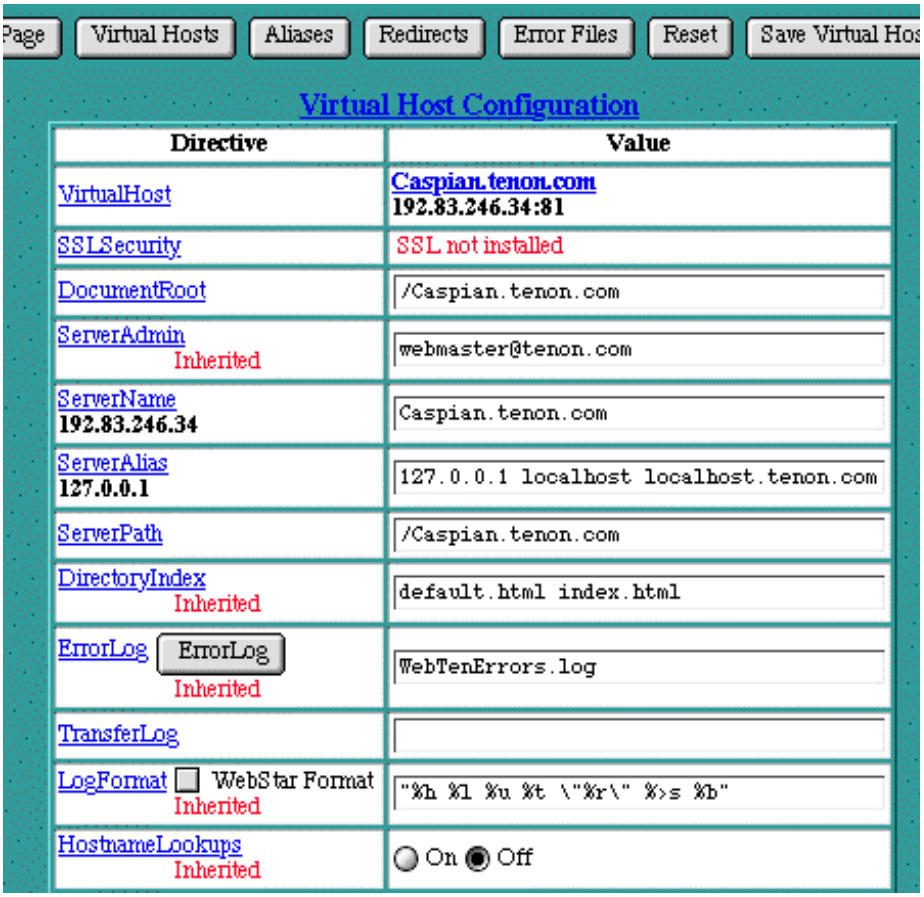


Figure 45: Virtual Host Configuration Table

7.2.1 VirtualHost

The *VirtualHost* entry displays the name of the virtual host to which the following settings apply. It is the same name that was entered in the Web[Ten] Administration Server home page in the *Virtual Hosts Table.*

7.2.2 SSLSecurity

Web[Ten] permits the configuration of secure connections on a per IP virtual host basis via the *SSLSecurity* directive. (For more information on SSL, refer to section "8.0 Secure Socket Layer (SSL)".) The *SSLSecurity* setting defaults to "Off". To enable SSL for this virtual host, set *SSLSecurity* to "On". Before *SSLSecurity* can be enabled, it is necessary to generate a Server Certificate (see section "8.1 Server Certificates") using the SSL Settings page (see section "8.2 SSL Settings").

If WebTen's SSL option is not installed, the SSL Security "On" and "Off" buttons will not be displayed. The message "Not Installed" will be displayed instead.

7.2.3 DocumentRoot

DocumentRoot controls which folder will be used as the base, or top level folder, for this virtual host's content. When a new virtual host is added (see section "7.1.1 Adding Virtual Hosts"), a folder with the same name as the virtual host is automatically created within the Web[Ten] folder. The *DocumentRoot* entry is set to the name of this folder. For example, if your *Virtual Hosts Table* includes two virtual hosts, "`north.test.tenon.com`" and "`south.test.tenon.com`", your *WebTen* folder will contain two sub-folders with corresponding names. Place the content files to be published for this virtual host in this folder. If *DocumentRoot* is not set, the default *DocumentRoot* setting will be used. In this case, the *DocumentRoot* entry will be flagged with the *Inherited* indicator. See section "5.3.6 Inheritance" for more information.

If you change the name of the virtual host's folder or decide to use some other folder, you must make a corresponding change to the *DocumentRoot* setting for this virtual host.

7.2.4 ServerAdmin

The *ServerAdmin* setting is an email address. This address is included in messages sent to a browser whenever a Web server error occurs. Users are encouraged to, and typically do, use this address to notify Webmasters of any problems they are experiencing with a Web server. The established convention is to use the email address "`webmaster@your_domain.com`", but any valid email address is acceptable. The email address must be an existing email address on some other email server. Web[Ten] does not accept incoming email.

In the case of a virtual host, *ServerAdmin* is initially set to the email address "`webmaster@virtualhost`", where "`virtualhost`" is replaced by the virtual host name. Alter this setting to reflect the email address of the Webmaster for this virtual host, or the Webmaster for this Web[Ten] system. Many Web sites follow the convention of using an email address "`webmaster@virtualhost`". To preserve this convention for your Web[Ten] server, add this address to your email server, or make this address an alias to another existing email account on your email server.

If *ServerAdmin* is not set for a particular virtual host, the *ServerAdmin* setting is inherited from the *Server Defaults*. In this case, the *ServerAdmin* entry in the *Virtual Host Configuration* table will be flagged with the *Inherited* indicator.

7.2.5 ServerName

The *ServerName* setting corresponds to the host name of this server. It is only used in redirection URLs. If the *ServerName* setting is not set, a reverse DNS lookup of the server's IP address is used. Note that this reverse DNS lookup may not return the desired host name if, for example, the host's primary name is "`fred.tenon.com`" and the desired *ServerName* is "`www.tenon.com`".

In the case of a virtual host, the *ServerName* is set to the same name as the virtual host. Typically, this setting does not need to be changed. It is only used in redirection URLs.

7.2.6 ServerAlias

The *ServerAlias* denotes which alternate host names should also apply to this virtual host. It is used with host header-based virtual hosts.

The *Server Defaults* do not include a setting for *ServerAlias*, so if the *ServerAlias* is not set, no alternate host names will apply to this virtual host.

7.2.7 ServerPath

The *ServerPath* is set initially to a path beginning with a slash ("/") followed by the virtual host name (e.g., `/north.test.tenon.com`). If this virtual host's home page was previously accessible via a non-virtual host URL, the old, or *legacy* URL, is entered here. Otherwise, this path should be blank

When the Web server receives a request from a browser incapable of supporting host header-based virtual hosts, or from a browser requesting the now outdated legacy URL, this Web server will translate those requests to the proper *DocumentRoot* via the *ServerPath*.

For example, if the new virtual host name is "`company_A.com`" and this data was previously located at "`http://www.tenon.com/company_A/`", the *ServerPath* should be set to "`/company_A`".

7.2.8 DirectoryIndex

The *DirectoryIndex* setting controls which file is returned when serving a request for a URL that points to a directory (i.e., ending with a "/"). When such a request is made, the *DirectoryIndex* is substituted for the URL, pointing the client request to a default file or CGI. If the *DirectoryIndex* is null, the contents of the directory will be listed on the returned page.

The Web[Ten] default *DirectoryIndex* is "`default.html`", which corresponds to the defaults established by other Macintosh Web servers. The typical Apache setting of *DirectoryIndex* is "`index.html`".

If the *DirectoryIndex* is not set for a virtual host, it will be inherited from *Server Defaults*, and the *Inherited* flag will be displayed.

7.2.9 ErrorLog

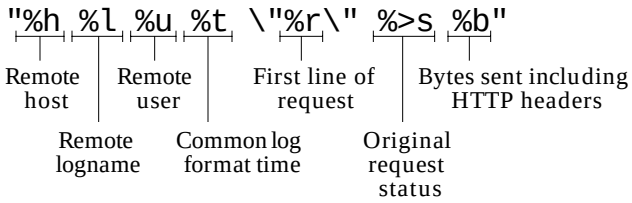
The *ErrorLog* entry in both the *Server Defaults* table and the *Virtual Host Configuration* table is the name of the file Web[Ten] uses to log information about Web server errors. If an *ErrorLog* file is not specifically set for a virtual host, the *ErrorLog* file setting in the *Server Defaults* table will be used.

7.2.10 TransferLog

The *TransferLog* setting is the name of the file Web[Ten] uses to log information about incoming requests. If *TransferLog* is not set for a particular virtual host, it will be inherited from the *Server Defaults*, and flagged accordingly. Apache hits will only be logged in this file if the file is different from the *TransferLog* set in the Server Defaults.

7.2.11 LogFormat

The *LogFormat* setting is a string that controls the format of the log file. The log file can include literal characters copied from the log format setting and detailed information specific to the actual request that is being logged. Details are encoded using a percent sign ("%") followed by a letter. For example:



Each "%" followed by a letter is a directive to the Web server for a specific piece of information about the request being logged. For example, "%h" logs the name of the remote host placing the request. The order and set of literal characters and details included in the transfer log explicitly follow the order and set of literals and "%" letters in the *LogFormat* setting.

ApacheSSL provides a "c" symbol for custom logging, thus Web[Ten] can be configured with custom SSL log entries using the "c" symbol. For example, a *LogFormat* string to include the SSL version used in an access and the encryption algorithm or cipher used in an access should use:

```
"%{version}c %{cipher}c"
```

If the *TransferLog* is not customized for a particular virtual host, the *LogFormat* setting will be inherited from the *Server Defaults*. This results from the *TransferLog* itself being inherited and utilizing the *Server Defaults' LogFormat*.

Web[Ten] can also create log files in a format compatible with WebSTAR log files. To enable this format, select the *WebSTAR Format* check box, and save the virtual host settings.

If the *TransferLog* for the virtual host is inherited from *Server Defaults*, *LogFormat* will not appear as an option in the *Virtual Host Configuration* table. The *LogFormat* option can be edited in the *Server Defaults* table, or a new *TransferLog* for a specific virtual host can be created. Click on the *Save Virtual Host Config* button. The *LogFormat* option should appear in the *Virtual Host Configuration* table.

The cache will keep the log if the accelerator cache is "On". (This is the default.) As such, the *LogFormat* option entered into the Administration Server is not passed to the cache, and the *LogFormat* has no effect.

7.2.12 HostnameLookups

The *HostnameLookups* setting controls whether reverse DNS lookups are performed for each incoming request using the originator's IP address. Enabling *HostnameLookups* will generally increase the time necessary to satisfy each request, and thus increase the load on your server. However, without *HostnameLookups*, *Access Controls* can be based only on IP addresses, not on host names or domain names. If *HostnameLookups* is disabled, IP addresses will be used in the *ErrorLog* and *TransferLog*, but these addresses can subsequently be resolved into host names, if necessary.

If the *HostnameLookups* option is not specifically set, it will be inherited from the *Server Defaults* and flagged accordingly.

7.3 Plug-In / Apple CGI Defaults

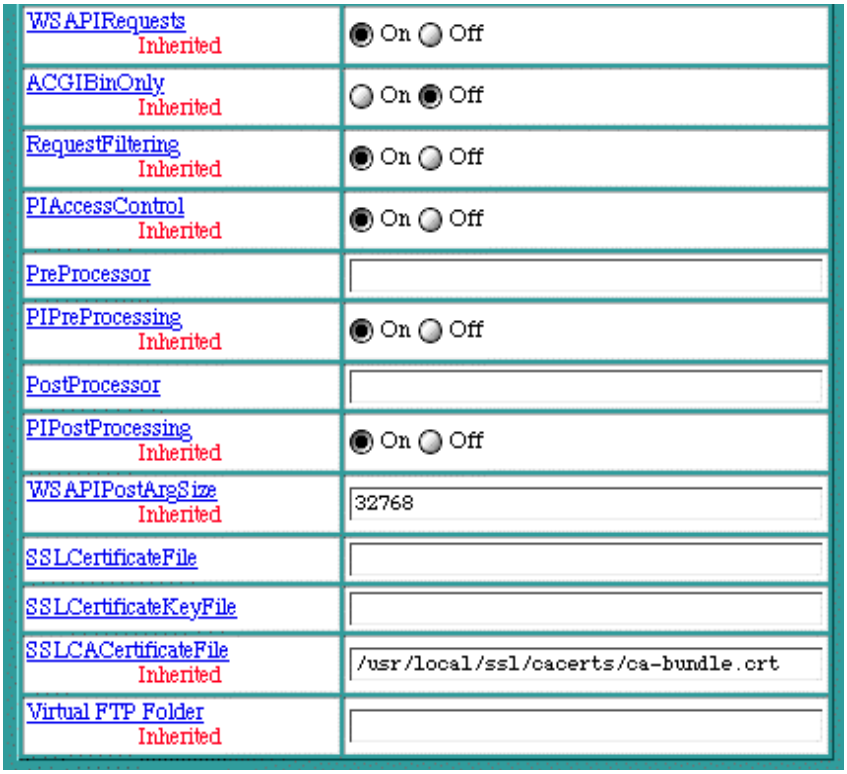


Figure 46: Virtual Host Configuration Table

7.3.1 WSAPIRequests

The *WSAPIRequests* setting controls whether the Web server will service requests to / from WebSTAR API-style ACGIs and plug-ins. This setting is "On" by default and enables the use of such ACGIs and plug-ins. The *Virtual Host Configuration* table also contains the *WSAPIRequests* entry which, if not specifically set, will be inherited and flagged accordingly.

7.3.2 ACGIBinOnly

The *ACGIBinOnly* setting controls whether Apple CGIs are permitted to be executed from within any folder or only from within the *cgi-bin* folder. The default setting is "Off", which enables Apple CGIs to be executed from within any folder. The *Virtual Host Configuration* table also contains the *ACGIBinOnly* entry which, if not specifically set, will be inherited and flagged accordingly.

7.3.3 RequestFiltering

The *RequestFiltering* setting controls whether a virtual host will allow "filter" plug-ins to service a request. Filter plug-ins receive the incoming HTTP request before processing has begun. The filter plug-in may modify the request URL before passing it back to Web[Ten] for processing. The default setting is "On", which enables URL filtering within a plug-in. (For more information, see section "6.2 Plug-In Administration".) The *Virtual Host Configuration* table also contains the *RequestFiltering* entry which, if not specifically set, will be inherited and flagged accordingly.

7.3.4 PIAccessControl

The *PIAccessControl* flag controls whether security plug-ins may participate in determining whether access should be granted or denied by the server on a per request basis. The default setting is "On", enabling security plug-ins. Security plug-ins may be selectively disabled by each virtual host.

7.3.5 PreProcessor

This is a virtual URL to a plug-in acting as a preprocessor (preceded by a slash "/"). *PreProcessors* are run after plug-in filtering is applied and before any access control checking by the server.

7.3.6 PIPreProcessing

The *PIPreProcessing* flag selectively disables plug-in preprocessors for a virtual host. By default, plug-in preprocessing is enabled and inherited.

7.3.7 PostProcessor

This is a virtual URL to a plug-in action, preceded by a slash (" / "). Plug-ins acting as *PostProcessors* receive notification of a completed request.

7.3.8 PIPostProcessing

The *PIPostProcessing* flag selectively disables plug-in postprocessors for a virtual host. By default, plug-in postprocessing is enabled and inherited.

7.3.9 WSAPIPostArgSize

The *WSAPIPostArgSize* setting specifies the argument buffer size for "PUT" and "POST" operators during plug-in and Apple CGI requests. The default size is 32768 bytes.

7.3.10 SSLCertificateFile

The *SSLCertificateFile* is the name of the server certificate for an IP-based virtual server. Host header-based virtual hosts sharing a common IP address must share the same server certificate; however, multiple IP based hosts may also share a single "wildcard" certificate. This setting allows certificate "wildcarding" among several IP hosts.Server certificates are maintained in the *tenon/ssl/certs* folder.

For more information, see section"8.0 Secure Socket Layer (SSL)".

7.3.11 SSLCertificateKeyFile

The *SSLCertificateKey* file is the private key associated with the server certificate. Keys generated by Web[Ten] during certificate signing request generation are normally stored in a secure area of the Web[Ten] internal file system; however, this field may be used for private keys of "wildcard" certificates or when a certificate and key are imported from from another system.

For more information, see section"8.0 Secure Socket Layer (SSL)".

7.4 Error File, Alias, and Redirect Settings

Virtual Host Alias, Error File, and Redirect Settings are exactly the same as their counterparts in the Server Defaults section of the admin server except that the ones applied here only apply to the individual Virtual Host to which they are applied. See section "6.1.9 Error File Settings" for more information on how to set the Error Files, section "6.1.10 Alias Settings" for information on how to set Aliases, and "6.1.11 Redirect Settings" for directions on setting Redirects.

7.5 Folder Contents

The *Virtual Hosts Table* on the Web[Ten] Administration Server home page contains a button for *Folder Contents*.

The *Folder Contents* table contains an entry for each file and sub-folder of a given folder. The given folder may be the *DocumentRoot* for a virtual host, or it may be a sub-folder of the *DocumentRoot*. To display the *Folder Contents* table for the *DocumentRoot* of a virtual host, click on the *Folder Contents* button for that virtual host in the *Virtual Hosts Table*. To display the *Folder Contents* table for a specific sub-folder, simply click on that sub-folder's link in the *Folder Contents* table.

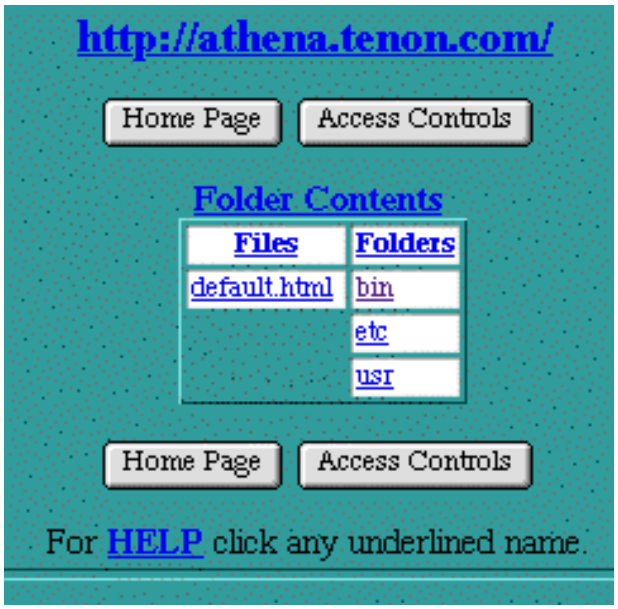


Figure 47: Folder Contents Table

When the *Folder Contents* table is displaying the contents of a folder other than the *DocumentRoot* folder, it displays a *Parent Folder* link as the first entry in the *Folders* column of the table. Clicking on the *Parent Folder* link will display a *Folder Contents* table for the folder in which the current folder resides.

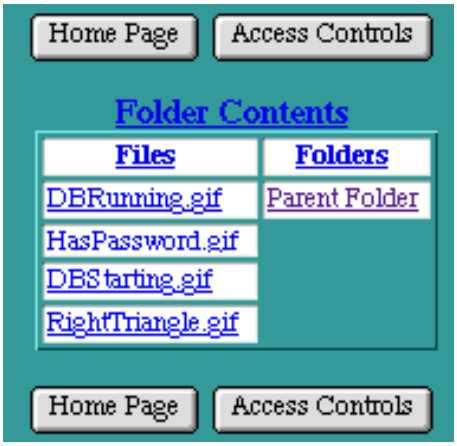


Figure 48: Sub-Folder Contents

Thus, the *Folder Contents* table provides a means for finding any file or sub-folder within a virtual host's hierarchy. To set any *Domain Name-Based Restrictions*, *Realm-Based Requirements*, *MIME Type Overrides*, or *Action Handler Overrides* for a specific folder, find that folder by using the *Folder Contents* table, then click on the *Access Controls* button for that folder. To access settings for a specific file, find that file by using the *Folder Contents* table, and then click on that file's name.

7.5.1 Files

This column displays an alphabetical list of all files contained within the specified folder. To access any settings specific to a particular file, click on that file's name to display the *Access Controls* for that file.

7.5.2 Folders

This column displays an alphabetical list of all sub-folders contained within the specified folder. To make any settings specific to a particular sub-folder, click on that folder's name to display a *Folder Contents* table for that folder, and then click on the *Access Controls* button for that folder.

7.6 Access Controls

The *Access Controls* settings can be set for any file or folder within a virtual host's hierarchy. The name of the file or folder to which these settings apply appears at the top of the table. This is a valid URL to this specific file or folder, complete with the proper virtual host name. Clicking on this URL will request this file or folder from the Web server in the exact same manner as any client browser. Thus, this link provides not only an explicit reference to the file or folder to which these controls apply, but also provides an easy way to test these settings and how the file or folder will be presented to normal incoming requests.

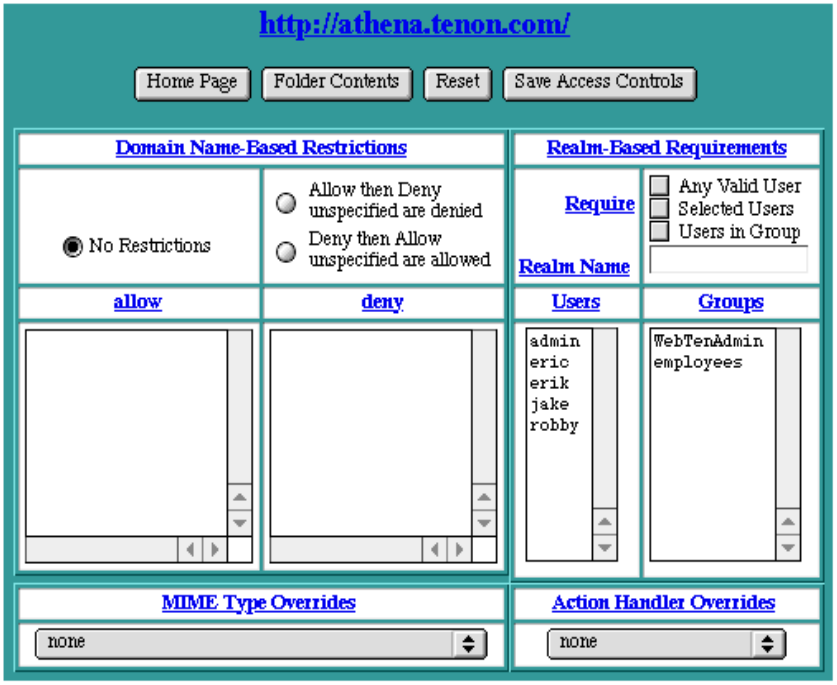


Figure 49: Access Controls Table

7.6.1 Domain Name-Based Restrictions

Domain Name-Based Restrictions specify which client IP addresses are permitted access to this URL, and which are denied. The Web server processes the *Allow* and *Deny* lists in the specified order to determine if a client request will be permitted.

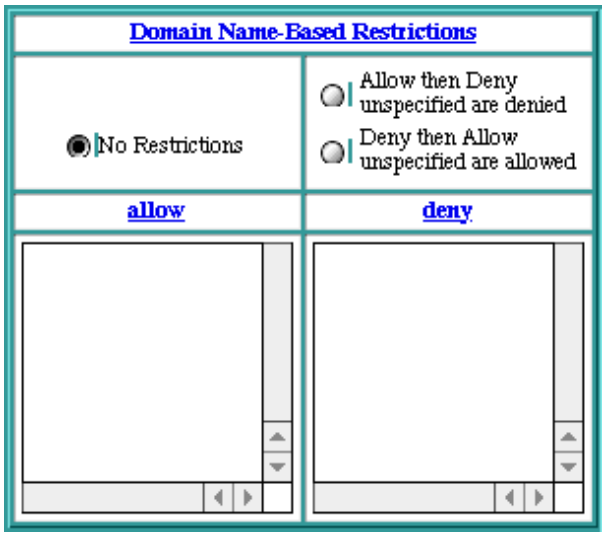


Figure 50: Domain Name-Based Restrictions

Evaluation Selection	Evaluation Order
No Restrictions	All requests are permitted.
Allow then Deny	The *Allow* specifications are evaluated first, followed by the *Deny* specifications. If any *Deny* should contradict any *Allow*, the *Deny* setting takes precedence.
Deny then Allow	The *Deny* specifications are evaluated first, followed by the *Allow* specifications. If any *Allow* should contradict any *Deny*, the *Allow* setting takes precedence.

The *Allow* and *Deny* lists can contain fully qualified domain names or IP addresses. They can also contain partially qualified domain names or IP addresses. If a fully qualified domain name or IP address is used, that specific host is allowed or denied access, as appropriate. If a partially qualified domain name is used, any host whose fully qualified domain name ends with the partially qualified name is allowed or denied, as appropriate. If a partially qualified IP address is used (i.e., the first 1 to 3 bytes of an IP address with a trailing dot ("**.**")), any host whose fully qualified IP addresses begin with the partially qualified IP address is allowed or denied, as appropriate.

If partially qualified domain names or IP addresses are used, any comparison will match whole components of domain names and IP addresses. For example, denying "`non.com`**" denies "**`host1.non.com`**", but not "**`anyone.tenon.com`**". Also, denying "**`192.30.20.`**" denies "**`192.30.20.1`**", but not "**`192.30.201.1`**".**

HostnameLookups must be enabled for this virtual host, in order for domain names to be properly interpreted in the *Allow* and *Deny* lists. See section "7.2.12 HostnameLookups" for more information.

A range of IP addresses may also be specified for a specific subnet by appending a slash ("/") and the number of bits in the subnet mask. For example, specifying "`192.30.20.128/25`" would mean all IP addresses from "`192.30.20.128`" to "`192.30.20.255`", inclusive.

Remember, if *Access Controls* for sub-folders are not explicitly set, the sub-folder will inherit the *Access Controls* from its parent folder. If the *Evaluation Order* for a given file or folder is set to *No Restrictions*, but a parent folder does have explicit restrictions, the parent's restrictions also apply to this file or folder. In this case, the *Access Controls* table will show the inherited *Access Controls* as specified by the parent folder, and the *Inherited* indicator and the *Evaluation Order* setting will be displayed.

7.6.2 Realm-Based Requirements

Realm-based access controls determine who is permitted access to this URL, by means of a user name and a password. If the client fails to provide a correct user name and password, access is denied.

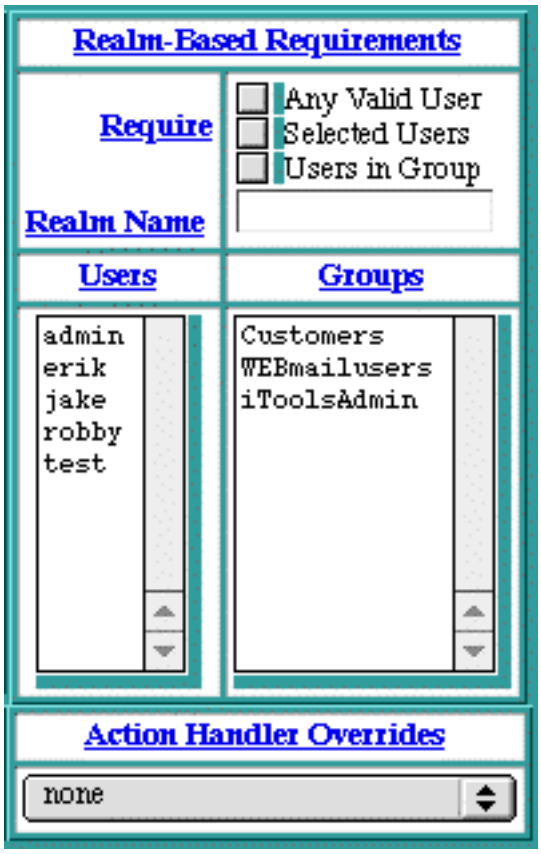


Figure 51: Realm-Based Requirements

The list of privileged users can be any of a combination of settings, as defined below:

Setting	Access
Any Valid User	Any user from the entire list of users is permitted access with the proper password for that user. Selecting *Any Valid User* effectively disables *Selected Users* and *Users in Group*.
Selected Users	Any highlighted user in the *Users* list is permitted access. To select a set of users, see section "6.10.1 Users in Group". Selected users may be checked by itself, or in conjunction with *Users in Group*, in which case any user who is a member of any highlighted group in the *Groups* list, or any highlighted user in the *Users* list, is permitted access.
Users in Group	Any user who is a member of any highlighted group in the *Groups* list is permitted access.

If none of the *Any Valid User*, *Selected Users* or *Users in Group* settings are checked, no *Realm-Based Access Controls* explicitly apply to this file or folder. However, if a parent folder does have explicit *Realm-Based* restrictions, the parent's restrictions also apply to this file or folder. In this case, the *Realm-Based Requirements* table will show the inheritance as specified by the parent folder, and the *Inherited* flag will be displayed, along with the required setting.

If any of the *Any Valid User*, *Selected Users*, or *Users in Group* settings are checked, a *Realm Name* must be entered. This name can be any word or compound word without spaces, such as "`WebTenAdmin`". The *Realm Name* is passed to the browser and displayed to the client as a means of identifying which collection of users is permitted to access this file or folder. Many browsers cache this *Realm Name* and the entered user name and password to relieve the user from re-entering this information when another URL with the same *Realm Name* is requested.

7.6.3 MIME Type Overrides

MIME Type Overrides allow an explicit file, or folder of files, to be served without regard for the file name extensions. For example, if a folder contained only GIF files, but some (or all) of these files did not end with the "`.gif`" extension, the *MIME Type Overrides* could be set to "`image/gif`" to force all of these files to be treated as GIF files, without requiring any renaming of the files.

If the *MIME Type Overrides* setting is not given for a specific file or folder, but a parent folder does have an explicit setting, the parent's setting also applies to this file or folder. In this case, the table will show the inheritance (see section "5.3.6 Inheritance") as specified by the parent folder, and the *Inherited* indicator will be displayed along with the *MIME Type Overrides* setting. See section "6.6 MIME Extensions" for more information.

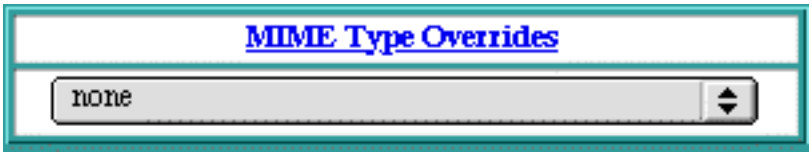


Figure 52: MIME Type Overrides

7.6.4 Action Handler Overrides

Action Handler Overrides allow a specific file, or folder of files, to be passed to the overriding action handlers, without regard for the file name extensions or associated MIME types. See section "6.5 Action Handlers" for more information. For example, if a folder contained only image map files, and some (or all) of these files did not end with the "`.map`" extension, the *Action Handler Overrides* could be set to "`imap-file`" to force all of these files to be passed to the image map handler, without requiring any renaming of the files.

If the *Action Handler Overrides* setting is not given for a specific file or folder, but a parent folder does have an explicit setting, the parent's setting also applies to this file or folder. In this case, the table will show the inherited overrides as specified by the parent folder, and the *Inherited* indicator and the *Action Handler Overrides* setting will be displayed.

Figure 53: Action Handler Overrides

8.0 Secure Socket Layer (SSL)

Web[Ten] incorporates version 3.0 of the Secure Socket Layer (SSL) protocol to encrypt Web server transmissions. The secure socket layer intercepts network calls from the server to encrypt the data before forwarding it to the network layer for transmission to the browser.

The Web server and the browser negotiate an encryption algorithm, or cipher, to be used for the session. A session "key" is securely communicated to the browser using public key cryptography. The session key is then used symmetrically, i.e., to both encode and decode the actual session data.

The first step in setting up SSL is obtaining a Certificate.

8.1 Server Certificates

The server certificate validates the identity of the server. Server certificates are signed by a trusted higher authority (the Certificate Authority, or "CA"), who assures the identity of the server.

In a typical commercial virtual host setup, each IP virtual host will have a unique server certificate.

Named virtual hosts (hosts that share an IP address) must share the certificate of the common IP host. By default, Web[Ten] associates a certificate issued to an IP virtual host with all configured named virtual hosts that share that IP address.

8.1.1 Obtaining a Server Certificate

In order to obtain a server certificate, a Certificate Signing Request (CSR) must be sent to the Certificate Authority, along with other proof of identity documents.

- Fill out the SSL Settings form (see section "8.2 SSL Settings") within the Web[Ten] Administration Server.
- Submit the completed CSR to the Certificate Authority. Verisign Consulting (`www.verisign.com`) has an on-line CSR submission form at:
- Cut and paste the CSR from the SSL Settings form into the CSR submission form.

Other documents validating the identity of the server must be mailed to the CA, along with a nominal service fee. These documents include:

1. Proof of the right to use the organization name, as in a copy of the company articles of incorporation, "doing business as" registration, etc.
2. Proof of domain name registration (except for ".com").
3. A letter, printed on organization letterhead and signed by an authorized representative, requesting certification of the domain name.

Your official certificate will be digitally signed and e-mailed to you by the CA. Rename the certificate to "`xx.xx.xx.xx.crt`" (where <`xx.xx.xx.xx`> is the IP address of the virtual host for which the certificate was generated), and place the official certificate in the *tenon/ssl/private* folder. The official certificate will replace the temporary self-signed certificate generated by Web[Ten] for use prior to receipt of the official certificate.

8.2 SSL Settings

To generate an SSL certificate, click on the *Certificate* button beside the *SSLSecurity* entry in the *Virtual Host Configuration* table (see section "7.2.2 SSLSecurity"). The *SSL Settings* page (shown below in "Figure 54: SSL Cipher Restrictions") is a form for generating a Certificate Signing Request (CSR).

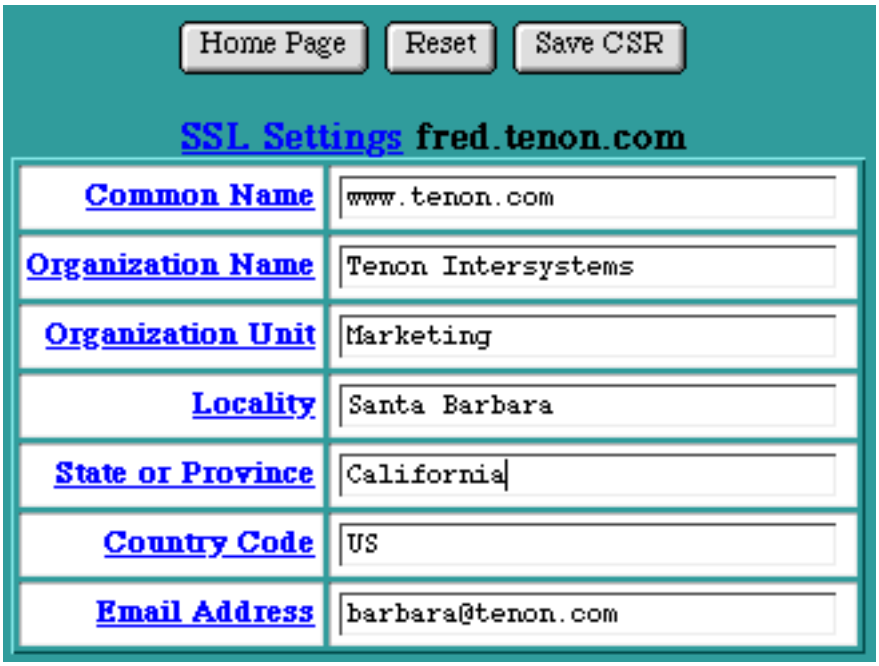


Figure 54: SSL Cipher Restrictions

8.2.1 Common Name

The *Common Name* is the domain name of the Web server or of an IP-based virtual host. This must be a fully qualified domain name, not an IP address or a DNS alias.

8.2.2 Organization Name

The *Organization Name* is the legal organization name.

8.2.3 Organizational Unit

The *Organizational Unit* is the department name or the name of a unit within an organization. This field is optional.

8.2.4 Locality

The *Locality* is the name of the city in which the organization resides. This field is optional.

8.2.5 State or Province

The *State or Province* is the name of the state or province in which the organization resides.

8.2.6 Country Code

The *Country Code* is a two-character country code for the country in which the organization resides. Use "US" for the U.S.A.

8.2.7 Email Address

The *Email Address* is the email address of a contact or representative within this organization.

8.2.8 Generating a CSR

To generate a Certificate Signing Request (CSR) save the *SSL Settings* via the *Save CSR* button. This action has several effects.

If a private key for this virtual host does not exist, such a key is created and saved in a secure area in Web[Ten]'s internal file system.

The actual Certificate Signing Request information is displayed in the Web[Ten] Administration Server (see "Figure 55: Certificate Signing Request Information"). This CSR is a PEM-encoded document which may be emailed to the CA, or it can be copied and pasted into an on-line certificate request form. This CSR is also saved in the *tenon/ssl/certs* folder in a file named `xx.xx.xx.xx.csr` (where <xx.xx.xx.xx> is the IP address of the virtual host for which the CSR was generated).

Figure 55: Certificate Signing Request Information

A temporary, self-signed certificate (for use while your CSR is being processed by the certificate authority) is created and saved in the *tenon/ssl/certs* folder in a file named `xx.xx.xx.xx.crt` (where <xx.xx.xx.xx> is the IP address of the virtual host for which the certificate was generated). This file should be replaced by the real certificate when one is returned from the Certificate Authority.

The self-signed certificate will allow your virtual server to perform secure transactions while your official certificate is being processed.

Browsers will question the validity of any server certificate signed by an authority of which they have no knowledge. The temporary, self-signed certificates should in no way be construed as proof of the virtual host's identity to your browser clients.

8.3 Enabling SSL

Once you have a certificate (even a Tenon-generated temporary one), you will be able to create a secure virtual host by toggling SSL Security "On" in the *Virtual Host Configuration* table. When SSL is activated for a virtual host, a red SSL designation appears to the right of the host name in the *Virtual Hosts Table* (see "Figure 56: Enabling SSL").

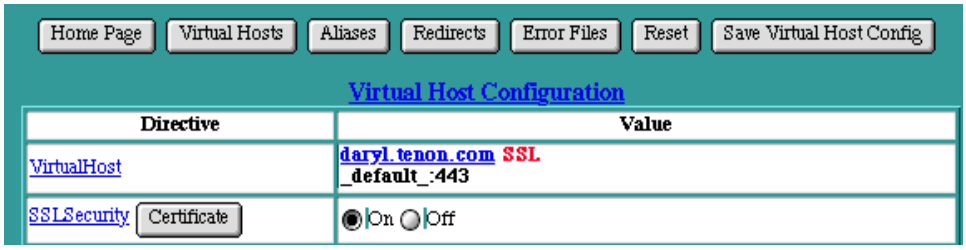


Figure 56: Enabling SSL

8.4 Ciphers

While the SSL 3.0 standard defines how encryption is applied to Web server-browser interactions, the actual encryption itself is performed by the negotiated cipher. Some common ciphers supported by Web[Ten] are shown in the following table:

RC2 and RC4	Block and stream ciphers using 128-bit keys, developed by and licensed from RSA data security, providing a very high level of security.
DES	A well-proven, 168-bit triple-encryption cipher.
Export RC2 and RC4	Identical to the 128-bit versions, except these ciphers use 40-bit keys.

8.4.1 SSL Cipher Restrictions

Clicking on the *Folder Contents* of a secure virtual host in the *Virtual Host Configuration* table will let you stipulate various cipher restrictions for that virtual host.

SSL Cipher Restrictions control whether or not access is allowed or denied to folders or files based on the encryption level negotiated between server and browser when an SSL connection is established (see "Figure 57: SSL Cipher Restrictions"). These controls are only accessible when *SSLSecurity* (see section "7.2.2 SSLSecurity") is enabled for a particular virtual host. The SSL cipher restrictions are not show if *SSLSecurity* is not enabled. Access control checks by SSL cipher are made in addition to any other host or realm-based access controls.

SSL cipher restrictions contain two lists of check boxes for each cipher in the cipher suites. If any checkbox is checked, that cipher is banned or required as indicated by the particular category.

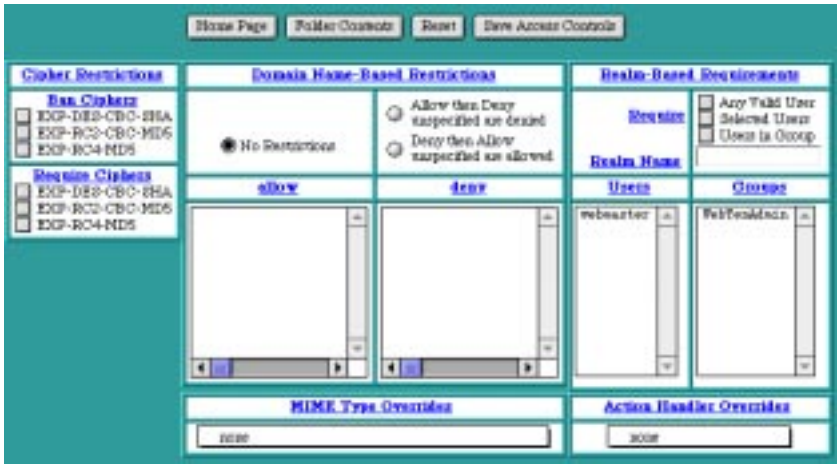


Figure 57: SSL Cipher Restrictions

8.4.1.1 Ban Cipher

If the cipher currently in force on the SSL connection is checked in this list, access to the file or folder is not permitted.

8.4.1.2 Require Cipher

If the cipher currently in force on the SSL connection has not been banned and is checked in this list, access to the file or folder is permitted. Ciphers not checked in this list are automatically banned access. However, if no ciphers are required, access is permitted subject to the *SSLBanCipher* list .

8.5 Using Web[Ten] with Multiple Certificates

Every SSL connection requires a unique IP address. Because Web[Ten] supports IP-based virtual hosting, you can easily set up multiple secure virtual hosts. Each secure virtual host will need its own Certificate. Follow the steps in this chapter to set up subsequent SSL hosts.

8.6 Self-Signed Certificates

If Web[Ten] is on an intranet and is not visible to the Internet at large, it can take advantage of SSL without having their certificate signed by a CA (Certificate Authority such as Verisign).To create your certificate, follow the directions in Section 11 of this document. That will yield a certificate signed by Web[Ten]. While this is not a certificate signed by a CA, it will allow SSL encrypted transactions from your Web[Ten] server. Some browsers will complain that the certificate is not signed by a valid authority (CA), but certificates for only internal or intranet use do not need to be validated by any CA (such as Verisign.)

8.7 Safeguarding SSL Keys and Certificates

Each SSL Certificate works in conjunction with the *SSL Key* file that was produced during the creation of the Certificate Signing Request. SSL Certificates do not stand alone. They require the *SSL Key* file to perform encryption. SSL Certificates will only work with the corresponding *SSL Key* file that was used to produce the actual Certificate Signing Request.

The *SSL Key* file is your private key that ensures that no one can replicate or assume your site's identity on the Web. If the *SSL Key* file is compromised, the inherent security of your SSL Certificate is lost. If the *SSL Key* file is lost, the SSL Certificate is useless and a new certificate will have to be issued.

As you can see, it is important to preserve a copy of your *SSL Ke*y file and to protect it against theft. In Web[Ten], the *SSL Key* file is tightly protected against unauthorized access (for example, rogue Apple or Unix CGIs cannot read the *SSL Key* file). The following steps provide a means to export an *SSL Key* file in order to make a backup copy of it. Once an *SSL Key* file is exported, it should be copied to a floppy disk (or other removable media) and the exported copy should be removed from the Web[Ten] system. The original *SSL Key* file is not deleted when it is exported; it is still available for normal SSL operations, and it is still protected against unauthorized access.

8.7.1 Exporting SSL Files

SSL Key and *SSL Certificate* files may be exported from a Web[Ten] system using a special CGI named *sslcerts.cgi*. For security reasons, this CGI is not installed by default in a Web[Ten] system. It must be installed and executed using the export option on the existing Web[Ten] system prior to upgrading to the new version of Web[Ten]. It then must be installed and executed using the import option on the new Web[Ten] system after that system has been installed. Once the *SSL Key* and *SSL Certificate* files have been imported into the upgraded system, *sslcerts.cgi* should be de-installed from that system

Exporting the *SSL Key* and SSL *Certificate* files does not removethe files it exports, but copies these files to the destination folder.

To export the *SSL Key* and SSL *Certificate* files from an existing Web[Ten] system:

- Copy *sslcerts.cgi* (from the *Utilities* folder on the Web[Ten] CD or from the *support* folder in the Web[Ten] distribution) into the *cgi-bin* folder. If your existing version of Web[Ten] has a *support* folder, copy *sslcerts.cgi* into the *cgi- bin* folder within the *support* folder. Otherwise, copy *sslcerts.cgi* into the active *cgi-bin* folder.

- Execute *sslcerts.cgi*. If you put *sslcerts.cgi* in the */support/cgi-bin* folder, use a URL like the following to execute this CGI. You must substitute your own host and domain names and replace the IP address "`10.0.0.1`" with your own IP address. When executing this CGI in this way, you will be required to provide your Web[Ten] administrator's password. Note that the protocol is *https* as the server is operating with SSL security on.

```
<https://host.domain/webten_support/cgi-
bin/sslcerts.cgi?10.0.0.1+export>
```

- If you put the *sslcerts.cgi* in the */cgi-bin* folder, use a URL like the following to execute this CGI. You must substitute your own host and domain names and replace the IP address "`10.0.0.1`" with your own IP address. When executing this CGI in this way, you will not be required to provide your Web[Ten] administrator's password.

```
<https://host.domain/cgi-bin/sslcerts.cgi?10.0.0.1+export>
```

- The exported SSL Key and SSL Certificate files will be placed in a folder within the /tenon folder. This folder will be named after the IP address that you provided in the URL above. For example, if the IP address was "`10.0.0.1`", the folder will be named `10.0.0.1.ssl`. Save this folder for subsequent importing into the newer version of Web[Ten].

- Remove the *sslcerts.cgi* from the *cgi-bin* folder.

8.7.2 Importing SSL Files

To import the *SSL Key* and *SSL Certificate* files from a previous version of Web[Ten]:

- Copy the folder containing the *SSL Key* and *SSL Certificate* files exported from the previous version of Web[Ten] to the */tenon* folder on the new Web[Ten] installation. Be sure to copy the entire folder (for example, the *10.0.0.1.ssl* folder, not just the contents of this folder).

- Copy *sslcerts.cgi* (from the *Utilities* folder on the Web[Ten] CD or from the *support* folder in the Web[Ten] distribution) into the *cgi-bin* folder within the *support* folder.

- Execute *sslcerts.cgi* using a URL like the following. You must substitute your own host and domain names and replace the IP address "`10.0.0.1`" with your own IP address. When executing this CGI in this way, you will be required to provide your Web[Ten] administrator's password.

  <http://host.domain/webten_support/cgi-bin/sslcerts.cgi?10.0.0.1+import>

- The imported *SSL Key* and *SSL Certificate* files will be placed into their respective places within the Web[Ten] distribution.

- Remove *sslcerts.cgi* from the */support/cgi-bin* folder.

- Remove the folder containing the *SSL Key* and *SSL Certificate* files (for example, the *10.0.0.1.ssl* folder) from the Web[Ten] system. You may choose to save these files in a safe place (preferably not on the Web[Ten] system) for subsequent upgrading or for backups of your *SSL Key* and *SSL Certificates* files.

9.0 FTP Service

Web[Ten] includes a high-performance, full-featured file transfer service that does not rely on AppleShare. The File Transfer Protocol (FTP) is included as an integrated component of Web[Ten]. The FTP service supports content uploading to Web[Ten]. Web[Ten]'s FTP service provides password-protected FTP access to separate access points (i.e., different virtual hosts) for each FTP user. The FTP server can also be configured to permit or deny anonymous FTP access. Anonymous FTP access is always directed to a unique access point not associated with any virtual host.

9.1 File Encodings

The Web[Ten] FTP server supports text and binary file transfers. By default, only the data fork of any file is transferred. It is possible to transfer complete Macintosh files, including their creator and type fields and their resource forks, by encoding these files in either MacBinary or AppleSingle format. The Web[Ten] FTP server uses the file extensions ".bin" and ".as", respectively, to indicate MacBinary or AppleSingle file encodings.

The Web[Ten] FTP server also observes the *MACB* command for MacBinary encoded file transfers. The *MACB* command is widely used by Macintosh FTP clients, including Fetch. These clients typically detect that the FTP server is capable of supporting the *MACB* command, and then automatically handle all file translation encodings. Users of non-Macintosh FTP clients may choose from various encoding mechanisms to transfer files from their systems to the Web[Ten] system.

9.2 Downloading Files via FTP

To download text files from the WebTen FTP server, the client FTP program should specify the FTP ASCII or Text mode for file transfers. In this mode, the WebTen FTP server will always deliver the data fork of the Macintosh file, and the textual data will always be translated to the standard FTP text format. Presumably the client FTP program will translate the incoming data from the standard FTP text format to the native text format for the host system.

It is possible to download text files using the FTP Binary or Image mode of file transfer. In this mode, the WebTen FTP server will deliver the data fork of the Macintosh file, but it will not translate the textual data. The text will be delivered in the exact format as it is stored in the Macintosh text file.

To download binary files from the WebTen FTP server, the client FTP program should specify the FTP Binary or Image mode of file transfer. In this mode, the WebTen FTP server will, by default, transfer the data fork of the Macintosh file in the exact format as it is stored in the file.

Complete text or binary Macintosh files (including the creator and type fields and the resource fork) can be transferred from the WebTen FTP server in a number of ways. The FTP client program can specify the *MACB* command to the FTP server, requesting it to encode files using the MacBinary file format. The popular Macintosh FTP client "Fetch" issues this command immediately upon connecting to the WebTen FTP server, and thus transfers all files to and from the server in MacBinary mode. A client FTP program can also use the extension ".bin" to request that a file be transferred in MacBinary format. For example, to transfer the file "`tenlog1.gif`" in MacBinary format, the FTP client can request to transfer the file "`tenlog1.gif.bin`".

Complete Macintosh files can also be encoded in the AppleSingle file format. To request AppleSingle format, a client FTP program must add the extension ".as" to the file it is requesting to transfer. For example, to transfer the file "`tenlog1.gif`" in AppleSingle format, the FTP client must request to transfer the file "`tenlog1.gif.as`".

9.3 Uploading Files via FTP

To upload text files to the WebTen FTP server, the client FTP program should specify the FTP ASCII or Text mode of file transfer. In this mode, the WebTen FTP server will always write to the data fork of the Macintosh file, and the textual data will always be translated from the standard FTP text format to the Macintosh text format. In addition, the uploaded file will be given the Macintosh type "TEXT" to correspond with its textual data.

To upload binary files to the WebTen FTP server, the client FTP program should specify the FTP Binary or Image mode of file transfer. In this mode, the WebTen FTP server will, by default, write to the data fork of the Macintosh file in the exact format as the data is delivered by the FTP client. If the file existed before the upload took place, the file's type field will not be modified. If the file did not exist before the upload, the file will be given the "BINA" type, corresponding to its binary data. The file's creator will be set to "MUMM".

Uploaded text files using the FTP Binary or Image mode are treated identically to binary files by the WebTen FTP server.

Complete Macintosh files (either text files or binary files) can be uploaded to the WebTen FTP server in a number of ways. If the files to be uploaded are encoded in the MacBinary file format, the FTP client program can specify the *MACB* command to the FTP server. The popular Macintosh FTP client "Fetch" issues this command immediately upon connecting to the WebTen FTP server, and thus transfers all files to and from the server in MacBinary mode. A non-Macintosh client FTP program can use the extension ".bin" to upload a file in MacBinary format. For example, a file named "`tenlog1.gif.bin`" would be interpreted by the WebTen FTP server as a MacBinary file. After the transfer is complete, the FTP client can rename the file and remove the ".bin" extension.

Complete Macintosh files can also be uploaded in the AppleSingle file format. A client FTP program can use the extension ".as" to upload a file in AppleSingle format. For example, a file named "`tenlog1.gif.as`" would be interpreted by the WebTen FTP server as an AppleSingle file. After the transfer is complete, the FTP client can rename the file and remove the ".as" extension.

9.3.1 Uploading and Executing CGI Scripts

The default Web[Ten] configuration has a strict policy on CGI script upload and execution. Web[Ten] will permit the uploading of CGI scripts only into the main */cgi-bin* folder and only for an FTP user configured for access to the content folders of all virtual hosts. Web[Ten] by default will permit the URL-based execution of CGI scripts by network browsers only from the */cgi-bin* folder.

You may override the policy regarding uploading of CGIs by modifying the *ftpaccess* file in the *tenon/etc* folder.

This might be necessary if, for example, you created individual *cgi-bin* folders within each virtual host's content tree on your system and wanted to give unrestricted CGI upload access to the folder to FTP users configured for access to each individual virtual host.

In this case, you would add a directive to the *ftpaccess* file(using a utility such as BBEdit) that reads:

```
upload /usr/local/etc/httpd/WebSites/yourvhost.com /cgi-bin yes webten
webten 0755
```

The first path parameter of the "upload" directive must exactly match the home directory of the configured FTP client. FTP client home directories are configured via the *Users* form in the Web[Ten] Administration Server. The second path parameter is the relative path from the FTP home directory to the folder to contain the CGIs.

Once configured for FTP upload access, CGIs to be run outside of the main */cgi-bin* folder must be given permission to be browsed using the Web[Ten] Administration Server. Use the *cgi-script* Action Handler override described in the "Access Controls" section to permit URL-based execution of the CGI.

The folder name *cgi-bin* has a special function and capability under Web[Ten]. If you create individual *cgi-bin* folders within the document root folders of each virtual host, follow the additional instructions described in the "Customizing WebTen**" section of "Appendix C"**.

9.4 FTP Settings

The *FTP Settings* table contains some options that control the Web[Ten] FTP service. The FTP server is an integrated component of Web[Ten] and is designed to provide separate access points based on virtual hosts for different FTP users. The FTP server can also be configured to permit or deny anonymous FTP access, and FTP transfers can be logged for either anonymous or password-based accesses.

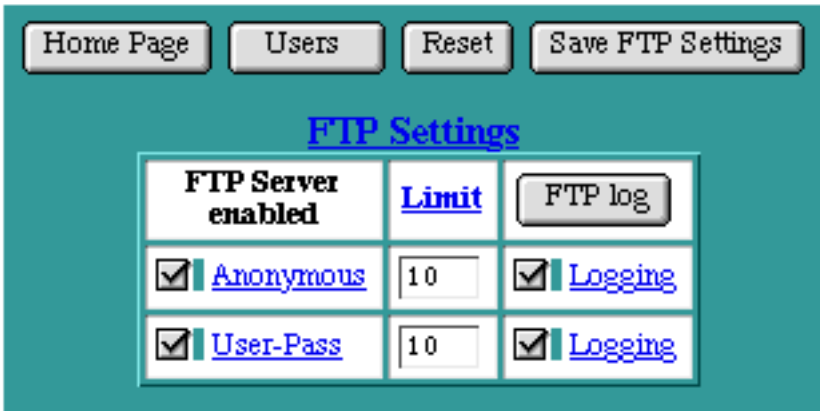


Figure 58: FTP Settings Table

9.4.1 FTP Status

FTP service is controlled by the *Enable FTP* checkbox in the Web[Ten] "Preferences". FTP service should be disabled if another FTP server is being used in conjunction with Web[Ten] on the same machine. When enabled, FTP service for both anonymous access, password access, or both can be explicitly set. The default *Enable FTP* setting is "Off".

9.4.2 Anonymous

The *Anonymous* checkbox enables or disables anonymous FTP access. When a user accesses the Web[Ten] system via anonymous FTP, the Web[Ten] FTP server automatically places that user in the *FTP* folder inside the Web[Ten] folder. Anonymous FTP users are thus restricted from accessing any other folders on the Web[Ten] system. The *FTP* folder contains some default sub-folders which provide different kinds of access to the anonymous FTP clients.

The *pub* folder is the generic placeholder for documents targeted for public consumption. Anonymous FTP users can get files from this folder, but they cannot put files into this folder, or modify any files within this folder. Generally the Web[Ten] administrator controls the organization and contents of this folder. However, password-based FTP users can place files in this folder if their *FTP Home* folder is either *All Virtual Hosts* or *Anonymous FTP*.

The *hidden* folder provides a level of security by obscurity. Anonymous FTP users cannot list or see any of the files within this folder, but if they know the exact name of the file they are looking for, they can get that file from this folder.

The *incoming* folder provides a place for anonymous FTP users to put files on this server. Generally these files are deposited here for consumption by the administrator of the Web[Ten] system. Anonymous FTP users cannot list or see the files in the incoming folder, so other anonymous FTP users cannot get a file deposited by a different FTP user unless they know the exact name of that file.

9.4.3 User-Pass

The *User-Pass* checkbox enables or disables password-based FTP access. When a user accesses the Web[Ten] system via an FTP user name and password, the Web[Ten] server automatically places that user in the folder indicated by the *FTP Home* setting for that user.

Password-based FTP users can read or write files into the folders to which they have access.

9.4.4 Limit

The *Limit* setting controls how many simultaneous sessions the Web[Ten] FTP server will permit for each class of FTP service. Subsequent attempts to FTP into the server will be denied when this limit is reached. A message is provided to the FTP client that the limit has been reached and that they should try again later.

9.4.5 Logging

The *logging* checkbox controls whether or not FTP transfers are logged for each class of FTP service. The Web[Ten] FTP server logs FTP transfers in the *logs/FTP.log* file. The contents of this file can be viewed by clicking on the *FTP Log* button.

9.5 Virtual Anonymous FTP Service

Web[Ten]'s FTP server supports virtual or "multihomed" anonymous FTP service for each IP address Web[Ten] is serving. The virtual FTP service allows an anonymous FTP user to connect to "ftp1.domain.com" and receive one FTP banner message and content location while another anonymous FTP user connecting to "ftp2.domain.com" receives another banner and location, even though they are on the same machine and port.

Note that password-based FTP clients (all users other than the "anonymous" user) have an explicit home folder under Web[Ten] and will be placed in that home folder regardless of which server domain name or IP address they use to connect to the server. Configuration of FTP users and their home folders is discussed in the "Users" section of this manual.

Follow these steps to configure one ore more virtual anonymous FTP server under Web[Ten]:

- Set up any number of IP-based virtual hosts using the Web[Ten] Administration Server (IP-based virtual hosts require Tenon networking, so you must replace OpenTransport in the Web[Ten] preferences).

- Create unique folders to contain the content for each virtual hosts anonymous FTP server:

  Use the Finder to duplicate the *ftp* folder in the *tenon/templates/* folder. Rename the folder, assigning it a unique name (for example, `ftp-virtual1`). Move the folder to the top-level Web[Ten] folder. This "home" folder will contain the content for an anonymous virtual host. Perform this step for one or more of the IP-based virtual hosts you have configured under Web[Ten].

• Edit the *tenon/etc/ftpaccess* file and add lines similar to the following for each virtual host requiring its own virtual anonymous FTP service:

```
virtual 192.1.2.3 /usr/local/etc/httpd/ftp-virtual1 /nobanner
virtual 192.1.2.4 /usr/local/etc/httpd/ftp-virtual2
/usr/local/etc/httpd/ftp-virtual2/banner.msg
```

The first argument is the IP address of the virtual FTP server. The IP address matches one of the IP-based virtual hosts you configured using the Web[Ten] Admininistration Server. The second argument is the path to the "home" folder created in the previous step, and the third argument is a file containing the banner to display to the FTP client upon login. Use */nobanner* to display no login banner.

9.5.1 Host Header-Based Anonymous FTP

Anonymous FTP access to a header-based virtual host will be placed in the anonymous FTP home directory of the IP virtual host whose IP address appears in a "virtual" directive in the *ftpaccess* file.

10.0 NFS Service

Web[Ten] includes NFS capabilities that allow it to mount NFS volumes from any NFS server. This means that enterprise-wide or campus-wide workstations, minis or mainframes can easily be used to store Web[Ten] content. These volumes can then be published within the content hierarchy of the Web[Ten] Web server. The NFS servers can contain the Web pages for an entire Web site, a set of specific virtual hosts, or simply a component of a virtual host.

The Web[Ten] NFS client service is compatible with any NFS server implementation. Support for read-only access to the NFS volumes is also provided.

10.1 Configuring the NFS Server

An NFS server typically requires some configuration to specify which NFS clients are permitted to mount its volumes. For example, an NFS server typically needs to be configured with a host name or IP address for each such client. To permit a Web[Ten] system to mount any of its volumes, the server typically needs to have the host name or IP address of the Web[Ten] systems entered into its NFS configuration database. Each type of NFS server has its own configuration database and instructions. You should review the documentation for that system or consult the system administrator for that system before attempting to mount any NFS system from within Web[Ten].

10.2 NFS User and Group Numbers

When an NFS request is sent from an NFS client to an NFS server, user and group identification numbers are included in the request. These numbers are used by the server to determine what type of access is to be permitted for that specific request. NFS servers can selectively permit or deny access for the reading or writing of any file and the indexing of any directories.

Web[Ten] systems use the user ID 65534 (`nobody -2`) and the group ID 65535 (`nogroup -1`) for all NFS requests. These IDs are configured in the *httpd.conf* configuration file.

Web[Ten] systems use the user ID 65533 and the group ID 65533 for all anonymous FTP requests passed to the NFS server. All other password-based FTP requests use the user ID 1000 and the group ID 100.

10.3 NFS Settings

The *NFS Settings* table contains options that control Web[Ten] NFS capabilities. The NFS capabilities are an integrated component of Web[Ten] and are designed to provide NFS access points (within the Web server's content hierarchy) to NFS server systems. The NFS access points can be based on complete virtual host content trees, or they can be sub-folders within a virtual host's content tree. The NFS access points can also be restricted to read-only access from the NFS server. This protects the NFS server from any modifications attempted via a CGI or a plug-in that is executed on the Web server.

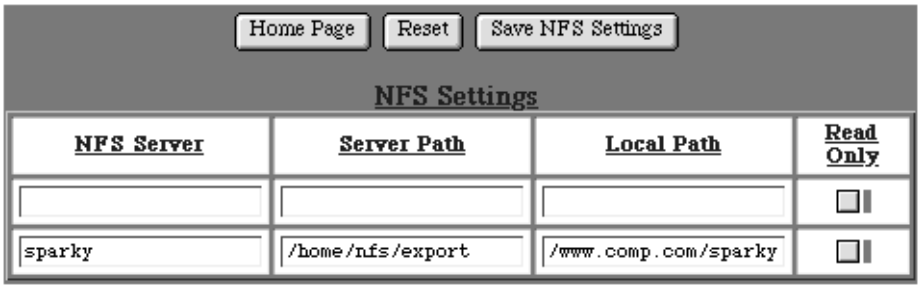


Figure 59: NFS Settings Table

10.3.1 NFS Server

The *NFS Server* is the name of the system on which the NFS volume actually resides. This name may be a fully qualified domain name or a partially qualified domain name within the same domain as the Web[Ten] system. Alternatively, an IP address in dot notation may be used if Web[Ten] cannot resolve (via DNS) the name of the NFS server.

10.3.2 Server Path

The *Server Path* is the path to the desired folder or directory on the NFS server. The NFS server uses this path to identify the top or highest level directory that will be exported to Web[Ten].

10.3.3 Local Path

The *Local Path* is the path to the desired mount point or folder on the Web[Ten] system. This is where Web[Ten] will mount the NFS volume within its own content hierarchy.

10.3.4 Read Only

The *Read Only* checkbox controls whether or not the NFS volume is protected, by restricting the Web[Ten] Web server to read-only access from this NFS volume.

11.0 Domain Name System (DNS)

The Domain Name System (DNS) service acts very much like a telephone company directory assistance service. It provides mapping between Internet "host" computer names and Internet addresses. Given a host name, it will look up and return a host address. Sophisticated DNS features include the mapping of several different names to a single Internet address and the mapping of several different Internet addresses to a single host name.

The Domain Name System itself is a distributed database of domain names and Internet addresses. DNS translates names (for example, panther.wildcats.com) to IP addresses (for example, 205.1.2.3) and vice versa. A client/server scheme, supported by replication and caching, enables these mappings to be available throughout the Internet. The best resource for an in-depth understanding of DNS is "DNS and BIND", published by O'Reilly & Associates, Inc. Domain name servers make up the server half of the client/server mechanism. Name servers contain information about some segment of the DNS database and make that information available to clients, called resolvers. Web[Ten] includes a domain name server.

Having a domain name server properly configured with the name and IP address of your Web[Ten] system greatly facilitates the configuration and use of Web[Ten]. However, if your domain name server has not been configured with the name and address of your Web[Ten] system, you can still configure and use Web[Ten]. When the domain name server is later configured with information about your Web[Ten] system, Web[Ten] will continue to operate properly, and can easily be re-configured to exploit the advantages of DNS.

Web[Ten] includes a complete implementation of the Berkeley Internet Named Domain (BIND) DNS, version 8. BIND, version 8, is the latest incarnation of what is considered the definitive implementation of the DNS protocol. The software is maintained and continually enhanced by the Internet Software Consortium (www.isc.org). This latest version includes significant enhancements, including performance improvements and security-related fixes BIND under Web[Ten] functions independently of Apache, and has been designed to either totally replace or operate in concert with other DNS servers for your domains.

You enable BIND under Web[Ten] by checking the "Enable DNS" box in the Web[Ten] Preferences.

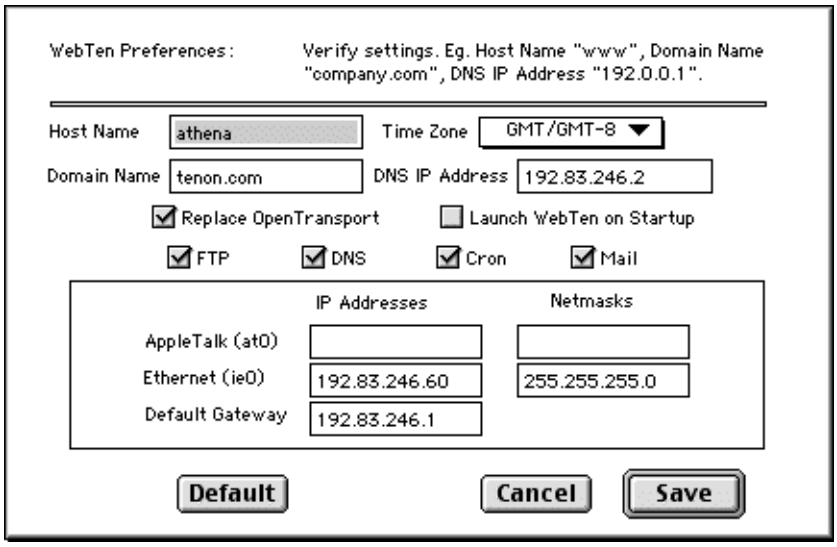
Further configuration of Web[Ten]'s Preferences are affected by your decision to enable BIND under Web[Ten], or by the presence (with proper configuration) or absence of a remote DNS server. Also, how Web[Ten] handles specific features of IP address-based or name-based virtual hosts is affected as well. These related topics are both discussed below.

11.1 Virtual Hosting Requirements

IP-Based Virtual Hosting	Requires checking the option "Replace OpenTransport" in the Web[Ten] Preferences window. Does not require DNS. Explicit IP addresses can be used instead. IP-based virtual hosting can use host names if DNS is available. This method is preferred, as users will identify domain names more readily than numbers.
Host Name-Based Virtual Hosting	Works with or without checking the option "Replace OpenTransport" in the Preferences window. This method of virtual hosting requires access to a properly configured DNS server.

IP-based virtual hosts use the actual IP address of each virtual host. They are also known as IP address-based virtual hosts. Name-based virtual hosts use the DNS served name for each virtual host. They are also known as host header-based virtual hosts.

11.2 Web[Ten] Preferences and DNS



If using you are using OpenTransport, Web[Ten]'s IP address is obtained from the TCP/IP control panel. If you chose to replace OpenTransport, Web[Ten]'s IP address must be entered in the Web[Ten] Preferences "Ethernet (ie0)" field. In either case, Web[Ten] requires an IP address to function.

If you change the "*Replace OpenTransport*" option in the Web[Ten] Preferences field, you MUST quit all applications and restart the system. This allows any other networking applications to recognize the current networking libraries.

11.2.1 Running Web[Ten] with an Unconfigured DNS Server or without DNS

If you have disabled BIND under Web[Ten] and have access to a remote DNS server that is unaware of the Web[Ten] system's host name, and without any previously configured virtual host names; or if you choose to run without a DNS server, the following information must be entered in the Preferences window:

Host Name	This field contains your choice of a name for the system on which Web[Ten] is running. Choose a name which is unique to the DNS server, and which you will not need to alter when DNS becomes available. If the system already has a network designation, be sure to use the well-known designation.
Domain Name	This field contains your choice of a domain name. If you are on a network that has a domain name, use that domain name. Otherwise, enter a fictitious domain name. This can be changed later when a true domain name is officially available.
DNS IP Address	If your remote DNS server is unconfigured for the Web[Ten] system, enter the IP address of the DNS server. If you enabled DNS in the Web[Ten] Preferences, enter the IP address of the local system. If you choose to run without DNS, leave this field blank.

11.2.1.1 Connect to the Web[Ten] Server

If you are connecting to the Web[Ten] server locally (i.e., using a Web browser on the same machine as your Web[Ten] system), use the IP address for "localhost" (`127.0.0.1`), or the host name or explicit IP address assigned in the Preferences window.

If you are connecting via a browser on a remote system, use the explicit IP address of the Web[Ten] system.

11.2.1.2 Adding Virtual Hosts

If you are running Web[Ten] with no DNS server, you will only be able to use IP-based virtual hosts. Name-based virtual hosts *require* DNS configuration for each additional Virtual Host. You *must* check the box "Replace OpenTransport" in the Preferences window. Use the Web[Ten] Administration Server and enter the IP addresses for the additional virtual hosts in the Virtual Hosts Table.

11.2.2 Running Web[Ten] with DNS

If you enabled BIND under Web[Ten] or if you have access to a fully configured DNS server that contains the information about your Web[Ten] system, as well as the virtual host names you wish to create, the following information must be entered in the Preferences window:

Host Name	This field contains the name for the system on which Web[Ten] is running, as it is known by the DNS server.
Domain Name	This field contains the name of the domain being served by the DNS server.
DNS IP Address	This field contains the IP address of the DNS server. If you enabled Web[Ten] DNS, this field contains the IP address of the local system.

11.2.2.1 Connect to the WebTen Server

If you are connecting to the WebTen server locally (i.e., using a Web browser on the same machine as your WebTen system), use the IP address for "localhost" (`127.0.0.1`), or the host name or explicit IP address assigned in the Preferences window.

If you are connecting via a browser on a remote system, and the remote system has access to the DNS server, use the host name of the WebTen system. If the remote system does not have access to DNS, use the explicit IP address of the system on which WebTen is running.

11.2.2.2 Adding Virtual Hosts

If you are running WebTen with access to a DNS server, you will be able to use both IP-based virtual hosts and name-based virtual hosts. If you are using IP-based virtual hosting, you *must* check the box "Replace OpenTransport" in the Preferences window. Name-based virtual hosts can be used with or without OpenTransport. If you enabled BIND under WebTen, use the WebTen DNS Administration Server (see section "11.3 WebTen Domain Name Server Administration") to configure virtual host names and IP addresses into your local DNS. Then use the WebTen Administration Server and enter the IP addresses or the DNS configured virtual host names for each additional virtual host in the Virtual Hosts Table.

11.3 WebTen Domain Name Server Administration

WebTen contains an integrated, browser-based interface for configuring your DNS zones. Changes to the DNS databases are automatically merged into the running DNS. For a description of definitions related to DNS, refer to ""Appendix B"".

The DNS Settings table is a list of Primary and Secondary Zones that are currently being managed by this system. On initial launch, WebTen will create a Primary Zone for the domain entered in the Preferences. The zone will include the Host Name and IP Address from the Preferences (see figure in section "11.2 WebTen Preferences and DNS"), a "localhost" name for loopback connections, and a "www" hostname alias.

The DNS Settings page also presents buttons for creating new Primary Zones, creating new Secondary Zones, and deleting Zones (either Primary or Secondary).

Figure 60: DNS Settings Table

11.3.1 DNS Primary Zone

The "Primary Zone" Page is accessed by selecting an underlined Primary Zone name from the DNS Settings page.

The Primary Zone page displays a table of Host Names and Aliases (sorted alphabetically) that are currently in this Zone. Each row of the table shows the Host Name, its IP Addresses or Alias, its Machine Name and System Name, and its Mail Exchangers (sorted by precedence). To change the information about an entry in the table, select any of the underlined fields to access the specific page for this entry.

This page also presents a row of buttons for managing host names within a Zone:

- "Home Page" returns to the main WebTen Administration Server Settings page
- "Zone List" returns to the DNS Settings page and its table of Zones
- "Address List" displays a table of Zone info sorted by IP Address
- "Start of Authority" displays the Start of Authority values for this Zone
- "New Host" adds a new Host Name to this Zone
- "New Alias" adds a new Alias to this Zone
- "Delete Host" deletes a Host Name or Alias from this Zone

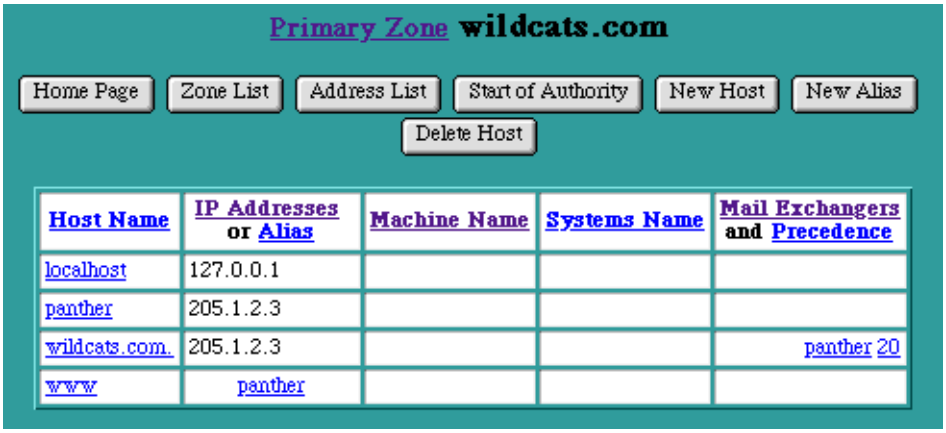


Figure 61: Primary Zone Page

11.3.1.1 Adding DNS Hosts

The "New Host" Page is accessed by selecting the "New Host" button in the Primary Zone Page. This page is used to enter the Host Name of a new Host to be included in this Zone, its IP Addresses, and the optional Machine Name and Systems Name information.

Figure 62: New Host Page

Enter the new Host Name. The new Host Name must be unique within this Zone (i.e., it must be different than any other Host Name or Alias in this Zone). It is not necessary to append the Domain Name at the end of the Host Name (i.e., it is not necessary to enter fully qualified Host Names). If the Domain Name is appended, either with or without a trailing dot ("."), the Domain Name will be stripped off and the abbreviated form will be used in the database and in the presented tables. One exception to the abbreviation rule is the Host Name that exactly matches the Domain Name (either with or without the trailing dot). In this case the Host Name is not truncated (a dot is added if it was omitted). This Host Name is often used to specify the default Mail Exchanger for an entire Domain or to specify a default IP Address for attempts to access this Domain without explicitly specifying a Host Name.

Enter an IP Addresses in the Internet dot (".") notation (e.g., "205.1.2.3") for this Host Name.

Enter the Machine Name and Systems Name. These fields are optional and may be left blank. Typically the Machine Name is used to record the Host's brand of hardware, and the System Name is used to record the name and version of the Operating System in use on this Host. However, these fields may be used to hold any kind of information about the Host. They may contain a space (" "), but must not contain a double quote (" " ").

Select the "Save Host" button to submit the new Host Name information. The new information will be updated in the Primary Zone's records and will be presented in the Host Table for this Zone.

11.3.1.1.1 Adding Load Balancing Hosts

You may use the "New Host" page to assign multiple IP Addresses to a single Host Name. The DNS server will load share resolver requests to this Host equally among the IP Addresses entered. Enter one IP Address per line.

Figure 63: Adding Load Balancing Records

11.3.1.2 Adding Host Aliases

The "New Alias" Page is accessed by selecting the "New Alias" button in the Primary Zone Page. This page is used to enter the Host Name of an Alias or nickname to be included in this Zone, and the name of the Host corresponding to this nickname.

Figure 64: Adding an Alias

Enter the new Alias Name. The new Alias Name must be unique within this Zone (i.e., it must be different than any other Host Name or Alias in this Zone). It is not necessary to append the Domain Name at the end of the Alias Name (i.e., it is not necessary to enter fully qualified Host Names). If the Domain Name is appended, either with or without a trailing dot ("."), the Domain Name will be stripped off and the abbreviated form will be used in the database and in the presented tables.

Select a Host Name from the pop-up list. Typically Aliases correspond to Hosts in the same Zone as the Alias. If this is the case, select the Host Name from the pop-up list of Names currently in this Zone. Otherwise enter the Host Name in the optional Host Name field. If this field is not empty, the entered Host Name will be used rather than the selection from the pop-up list.

If the entered Host Name is not in this Zone, it is necessary to enter a fully qualified Host Name including the dots (".") and a trailing dot.

Select the "Save Alias" button to submit the new Alias Name information. The new information will be updated in the Primary Zone's records and will be presented in the Host Table for this Zone.

11.3.1.2.1 Adding Load Balancing Host Aliases

Host Aliases may be used for DNS load balancing. Select the "New Alias" page once for each load balancing alias to be added to the database. Using the same Alias, select a different corresponding Host Name for each new record added.

11.3.1.3 Deleting a Host

The "Delete Host" Page is accessed by selecting on the "Delete Host" button in the Primary Zone Page. This page is used to select the Name of a Host or Alias to be deleted from this Zone.

Select the Name of the Host or Alias to be deleted from the pop-up list.

Select the "Delete Host" button to delete this Host or Alias. The Deleted name will no longer appear in the Host Table for this Zone.

The "Delete Host" Page is accessed by selecting the Delete Host button in the Primary Zone Page. This page is used to select the Name of a Host or Alias to be deleted from this Zone.

Select the Name of the Host or Alias to be deleted from the pop-up list.

Select the "Delete Host" button to delete this Host or Alias. The Deleted name will no longer appear in the Host Table for this Zone.

11.3.1.4 Changing a Host Name Record

The "Host Name" Page is accessed by selecting an underlined Host Name. This page presents a list of IP Addresses for this host and the optional Machine Name and Systems Name information.

Change the information for this Host by modifing any of the information presented in this page and selecting the "Save Host" button. The new information will be updated in the table of Host Names presented in the Primary Zone Page.

The "Host Name" Page also presents a "Mail Exchangers" button. Select this button to access a page of Mail Exchanger information or to add or delete Mail Exchangers for this host.

11.3.1.5 Changing a Host Alias Record

The "Alias" Page is accessed by selecting an underlined Alias. This page presents a pop-up list of Host Names and Aliases in this Zone with the currently valid Host Name for this Alias selected as the default.

Change the information for this Alias by selecting a Host Name from the pop-up list, following the rules for entering new Host Aliases.

Select the "Save Alias" button to submit the revised Alias Name information. The new information will be updated in the Primary Zone's records and will be presented in the Host Table for this Zone.

11.3.1.6 DNS Mail Exchangers

The "Mail Exchangers" Page is accessed by selecting an underlined Mail Exchanger in the Host Table or by selecting the "Mail Exchangers" button in the Host Page. The Mail Exchanger page presents a list of Mail Exchangers and their Precedence (sorted by precedence) for a specific Host. Mail Exchangers are also Hosts — selecting underlined Mail Exchanger names will display the Host Page for that Mail Exchanger.

To change the Precedence of a Mail Exchanger, select its underlined Precedence to access the specific page for that entry.

The "Mail Exchangers" Page also presents the buttons "Add Mail Exchanger" and "Delete Mail Exchanger" to add and delete Mail Exchangers for this Host. Select these buttons to access the corresponding pages.

11.3.1.6.1 Adding a Mail Exchanger

The "Add Mail Exchangers" Page is accessed by selecting the "Add Mail Exchangers" button in the Mail Exchanger Page. This page is used to enter the Host Name and Precedence of a new Mail Exchanger for a given Host. The Mail Exchanger may be another Host in this Zone, or it may be a Host in another Zone.

Enter the Host Name of the new Mail Exchanger. If the new Mail Exchanger is not in this Zone, enter a fully qualified Host Name including the dots (".") and a trailing dot. Select a Precedence for this Mail Exchanger from the pop-up list.

Select the "Save Mail Exchanger" button to submit the New Mail Exchanger information. The new Mail Exchanger Name will now be included in the Host Table, under the Mail Exchangers column for the given Host Name.

11.3.1.6.2 Deleting a Mail Exchanger

The "Delete Mail Exchangers" Page is accessed by selecting the "Delete Mail Exchangers" button in the Mail Exchanger Page. This page is used to select the Host Name of a Mail Exchanger to be deleted for a given Host.

Select the Host Name of the Mail Exchanger to be deleted from the pop-up list.

Select the "Delete Mail Exchanger" button to delete this Mail Exchanger. The Deleted Mail Exchanger's name will no longer appear in the Host Table under the Mail Exchanger column for the given Host.

11.3.1.6.3 Mail Exchanger Precedence

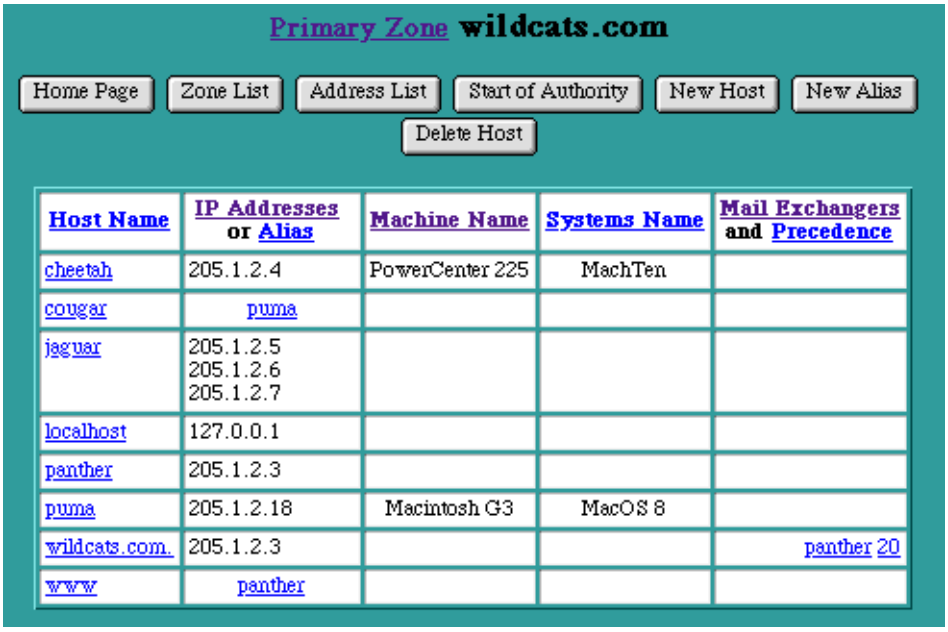
The "Precedence" Page is accessed by selecting on an underlined Precedence in the Mail Exchangers column of the Host Table. The Precedence page presents a pop-up list of Precedences with the current Precedence for the given Mail Exchanger displayed as the default entry.

Select a Precedence for the Mail Exchanger from the pop-up list.

Select the "Save Mail Exchanger" button to submit the new Precedence for this Mail Exchanger. The new Precedence will now be displayed in the Host Table under the Mail Exchangers column for the given Mail Exchanger and Host Name.

11.3.1.7 A Configured DNS Primary Zone

Returning to the Primary Zone page will reflect the Host additions to the database for this zone. To change the information about an entry in the table, select any of the underlined fields to access the specific page for the entry.



Host Name	IP Addresses or Alias	Machine Name	Systems Name	Mail Exchangers and Precedence
cheetah	205.1.2.4	PowerCenter 225	MachTen	
cougar	puma			
jaguar	205.1.2.5 205.1.2.6 205.1.2.7			
localhost	127.0.0.1			
panther	205.1.2.3			
puma	205.1.2.18	Macintosh G3	MacOS 8	
wildcats.com.	205.1.2.3			panther 20
www	panther			

Figure 65: A Configured Primary Zone

11.3.1.8 The DNS Reverse Lookup Zone

Selecting the "Address List" button from the Primary Zone page displays the Host List sorted numerically by IP address. This is the reverse lookup table, allowing the DNS Server to reference a Host Name when queried with an IP address. Changes to the Reverse Zone are made automatically with modifications to the Primary Zone table. A single Reverse Lookup Zone may serve multiple primary zones sharing the same network number.

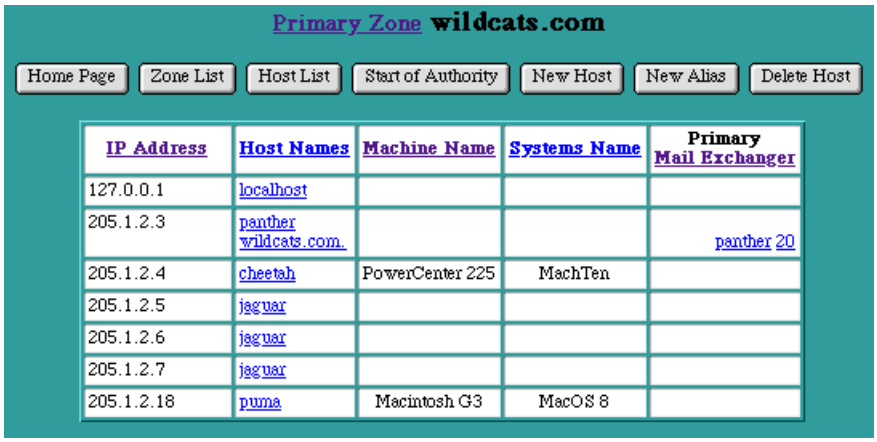


Figure 66: Reverse Lookup Table

11.3.1.9 DNS Start of Authority Record

The "Start of Authority" Page is accessed by selecting the "Start of Authority" button in the Primary Zone Page. This page presents pop-up lists of Start of Authority information with the current entries for the given Primary Zone displayed as the defaults.

Select the Start of Authority values from the pop-up lists. The Start of Authority values govern how often other Domain Name Servers check with this Server to ensure that their information is up to date. The Refresh, Retry, and Expire values are only used by other Domain Name Servers if they are acting as Secondary Servers for this Zone. The Min TTL or Minimum Time-To-Live value is used by any other Domain Name Server that queries any piece of data within this Zone. The time-to-live tells the other DNS Servers how long they may cache the data before checking back with this Server to see if the data has changed.

To change any Start of Authority information for a given Primary Zone, select the new Start of Authority values from the pop-up lists.

Select the "Save Start of Authority" button to submit the Start of Authority information. The new information will be updated in the Primary Zone's records and will be presented in the Start of Authority Page the next time it is accessed. Remote DNS servers that are Secondaries to your zones will pick up the changes no later than when the refresh interval specified in the Secondary's copy of the Zone file expires.

Figure 67: Start of Authority

11.3.2 DNS New Primary Zone Page

The "New Primary Zone" Page is accessed by selecting the "New Primary Zone" button from the main DNS Settings page. This page is used to enter the Domain Name of a Primary Zone to be managed by this system. The Domain Name must be unique — no other Primary or Secondary Zone may have the same Domain Name on this system.

Enter the Domain Name for the Primary Zone. Use the correct spelling, including the dots ("."). The trailing dot is optional. For example, primates.com can be entered as either "`primates.com`" or "`primates.com.`".

Select the Start of Authority values from the pop-up lists. The default Start of Authority values are reasonable for new Primary Zones.

Select the "Save Zone" button to submit the New Primary Zone information. The new Primary Zone Name will now be included (in alphabetical order) in the table of Primary Zones in the DNS Settings Page.

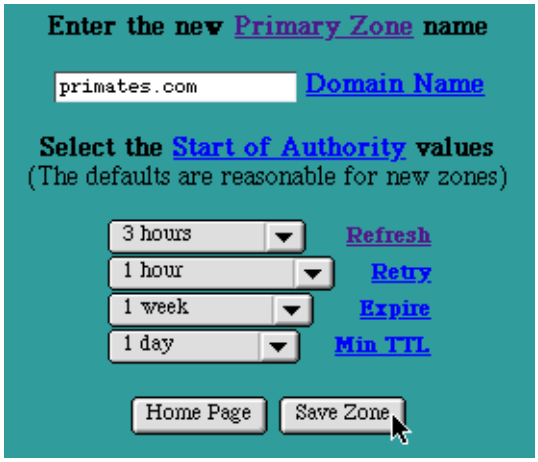


Figure 68: New Primary Zone

11.3.3 DNS New Secondary Zone Page

The "New Secondary Zone" Page is accessed by selecting the "New Secondary Zone" button from the main DNS Settings page. This page is used to enter the Domain Name of a Secondary Zone to be managed by this system. The Domain Name must match the Domain Name for an existing Zone. (Secondary Zones are always redundant copies of existing Zones, on other systems.)

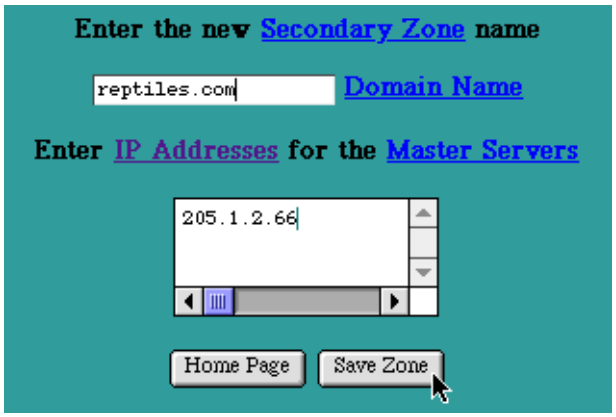


Figure 69: New Secondary Zone

Enter the Domain Name for the Secondary Zone. Use the correct spelling, including the dots (".."). The trailing dot is optional. For example, reptiles.com can be entered as either "`reptiles.com`" or "`reptiles.com.`".

Enter a list of IP Addresses (in the Internet dot "." notation, for example "`205.1.2.66`") for the Master Servers of the existing Zone. The list may include a single IP Address, or multiple IP Addresses (up to ten). Multiple IP Addresses can increase the availability of a Zone's database. In cases where a Master Server has several IP addresses by which it may be contacted, or when multiple Master Servers exist for a given Zone, multiple IP Addresses should be used. The order in which the IP Addresses are entered is the order the Domain Name Server will use when attempting connections. The Domain Name Server will cycle through the list until it successfully contacts a Master Server.

In the case where a Secondary Zone is being created simply to move a Zone from an existing Server, a single IP Address is sufficient. Enter the IP Address of the Master Server for the existing Domain.

Select the "Save Zone" button to submit the New Secondary Zone information. The new Secondary Zone name will now be included (in alphabetical order) in the table of Zones on the DNS Home Page.

11.3.4 DNS Secondary Zone

The "Secondary Zone" Page is accessed by Selecting an underlined Secondary Zone name from the main DNS Settings page. This page presents a list of IP Addresses for the Master Servers for this Secondary Zone.

To change any of the information for the Master Servers for this Secondary Zone, modify any of the IP Addresses in the list.

Select the "Save Zone" button to submit the Secondary Zone information. The new information will be updated in the Secondary Zone's records and will be presented in the Secondary Zone Page the next time it is accessed.

11.3.4.1 Creating a Primary Zone from a Secondary Zone

When Web[Ten] is configured as a Secondary DNS for a Zone, a copy of the Zone database is obtained from the Master Server when Web[Ten] DNS is first launched. The backup copy is kept up to date by periodically querying the Master Server according to the settings in the Start of Authority Record in the backup Zone file or, if the Master Server supports the BIND8 protocol, via an automatic notification mechanism whenever the Primary Zone file is modified.

The "Secondary Zone" Page presents a Transition to Primary Zone button. Selecting this button will transition this Secondary Zone into a Primary Zone on this system. When the Secondary Zone is transitioned into a Primary Zone, the backup Zone file is used as the initial database for the Primary Zone. A Primary reverse lookup zone is automatically created for the transitioned zone.

Part or all of an existing DNS database can be moved to the local system by setting up a Secondary Zone for each Zone you wish to move, then transitioning these Secondary Zones into Primary Zones.

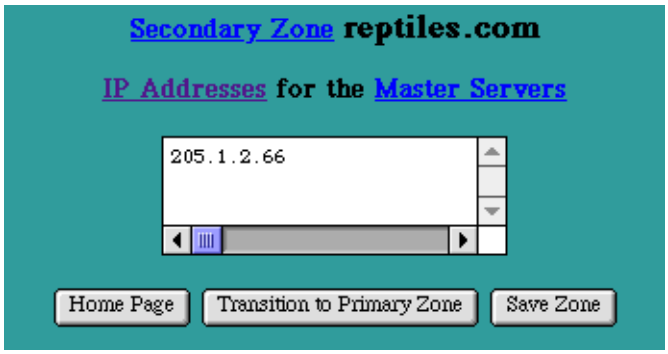


Figure 70: Secondary Zone

11.3.5 Deleting DNS Zones

The "Delete Zone" Page is accessed by selecting the "Delete Zone" button. This page is used to select the Domain Name of a Primary or Secondary Zone to be deleted from this system.

Select the Domain Name of the Zone to be deleted from the pop-up list.

Select the "Delete Zone" button to delete all information about this Zone. The Deleted Zone name will no longer appear in the table of Zones on the DNS Home Page.

Figure 71: Deleting a Zone

11.4 DNS Database Files

The *tenon/etc/named* folder holds the database files for BIND DNS under Web[Ten].

Primary Zone files managed by Web[Ten] DNS are designated "`db.thedomain`", where "`thedomain`" is the domain name.

Secondary Zone files obtained from a Primary Master and managed as back up copies by Web[Ten] DNS are named "`db_s.thedomain`".

Reverse Lookup Zone files are designated "`db.xx.xx.xx`" where "`xx.xx.xx`" is the network number of the reverse domain. The "`db.127.0.0`" file is the reverse lookup file for the loopback "localhost" name.

The *name.root* file contains the names of root domain servers used to initialize the Web[Ten] DNS cache.

The *named.conf* is the start up file for BIND containing the list of zones managed by Web[Ten] DNS, their correspoding zone files, and any DNS options.

11.5 DNS Manager CGI

If necessary, Web[Ten]'s DNS service can be restarted via a browser by using the *dns-mgr* CGI in the */cgi-bin* folder. This CGI is protected via the *WebTenAdmin* realm. To cause the DNS server to re-read its configuration file and reload the database or to query Master servers to update all secondary Zone files:

```
http://yourhost/cgi-bin/dns-mgr?reload
```

To display a list of options for the *dns-mgr* CGI, use:

```
http://yourhost/cgi-bin/dns-mgr?help
```

11.6 Registering your DNS Zones

If data in a newly created Primary Zone is to be made available to the Internet at large, the newly created Zone must be registered with the Internic at http://rs.internic.net, a central registry for Internet Domain Name Servers. If the newly created Zone is a part of an Intranet that is not connected to the Internet, or there is no requirement to make this Zone's data accessible to the Internet, this registration step may be skipped.

12.0 Clock Service (Cron)

Cron executes commands at specified dates and times according to the instructions in the file *tenon/etc/crontab*. Commands may be Perl and Shell scripts, Apple scripts and applications, or any of the UNIX utilities listed in "Appendix G".

The *crontab* file consists of lines of seven fields each. The fields are separated by spaces or tabs. The first five fields are integer patterns to specify:

- minute (0-59)
- hour (0-23)
- day of the month (1-31)
- month of the year (1-12)
- day of the week (1-7 with 1 = Monday)

Each of these patterns may contain:

- A number in the range above.
- Two numbers separated by a minus meaning a range inclusive.
- A list of numbers separated by commas meaning any of the numbers.
- An asterisk meaning all legal values.

The sixth field is a user name — the command will be run with that user's UID and permissions. The seventh field consists of all the text on a line following the sixth field, including spaces and tabs; this text is treated as a command which is executed by the Shell at the specified times. A percent character ("%") in this field is translated to a new-line character.

Lines beginning with a "#" are ignored by *Cron*.

The *crontab* file is checked by *Cron* every minute, on the minute.

12.1 Starting Cron

Check the *Enable Cron* checkbox in the Web[Ten] Preferences and restart Web[Ten] to start *Cron*:

12.2 Example *crontab* File

```
#Roll logs every day at midnight

0 0 * * * root
/usr/local/etc/httpd/tenon/admin/LogRoller

#Run weekly script at 3:45AM on Saturday. Mail result
to Administrator

45 3 * * 6 root
/usr/local/etc/httpd/tenon/admin/weekly.pl 2>&1 | tee
/usr/local/etc/httpd/tenon/logs/weekly.out | mail -s
"weekly output" you@yourserver.com

#Run monthly script on the 1st of the month at 5:30AM

30 5 1 * * root
/usr/local/etc/httpd/tenon/admin/monthly 2>&1 | tee
/usr/local/etc/httpd/tenon/logs/monthly.out | mail -s
"monthly output" you@yourserver.com
```

13.0 Using CGIs

In general, when traversing a Web page, clicking on a link causes that client (browser) to send a message to the server (the site maintaining the Web page the client wishes to view) with a given URL. The server gets the file indicated by the URL and sends the contents of the file back to the browser to be displayed to the user. The Common Gateway Interface (CGI) is a mechanism that causes the server to behave differently.

The CGI protocol defines communication between the server and an external program. When the URL points to a CGI script file, instead of simply sending the contents of the file to the browser, the server executes the script and then returns the program output to the browser. This allows Webmasters to create dynamic documents and interactive pages.

13.1 Shell CGIs

A shell CGI is a text file that contains commands for the Bourne Shell or C Shell command interpreter. Any text editor can be used to create shell CGIs. The resultant file will typically have the file extension of "`.sh`" (e.g., *mycgi.sh*). Place the file in the Web[Ten] *cgi-bin* folder.

The simplest CGI to create and use — the shell CGI — is a text file that contains commands for the Bourne Shell command interpreter. The steps are as follows:

Create a CGI called mycgi.sh. Store the newly created file in the cgi-bin directory. The new CGI can be referenced from a browser with the following URL: /cgi-bin/<cgi-name>. If mycgi.sh is stored in the cgi-bin directory, the URL would be: /cgi-bin/mycgi.sh.

Basic Steps

- Create a text file (see "13.1.1 Required Shell Script Content")
- Place the file in the Web[Ten] cgi-bin directory
- Reference the file from a Web browser

13.1.1 Required Shell Script Content

In addition to creating the text file, there are a few important considerations with respect to the content of the file. First, the top line of the file must contain the following text:

```
#!/bin/sh
```

This tells the system that this is a Bourne Shell script and that the Bourne Shell should be used to interpret the rest of the script.

Second, you can use the echo command to generate text which will be returned to the browser that initiated the URL. The first echo command must contain the following Bourne Shell commands to generate HTTP. This puts WebTen and the browser in the proper mode to accept everything else:

```
echo Content-type: text/plain
echo
```

The first echo indicates that text/plain will follow. The second echo is necessary in order to get the HTTP interpreter to accept the Content-type request. After that, any text sent with an echo command is printed on the originating browser's screen as a response to the URL request.

Shell scripts are text files containing Bourne Shell commands that can generate a stream of characters in response to being executed. There are Bourne Shell commands for assigning integer and string values to shell variables, commands for prescribing conditional flow through the shell script, and commands for running other programs. Relatively sophisticated CGIs can be created by combining different Bourne Shell commands. There are a number of widely available books describing Bourne Shell programming.

Bourne Shell CGIs are used for low-performance, easy-to-develop CGIs. Each Bourne Shell script is text, and is interpreted by a Bourne Shell interpreter controlled by WebTen. Since the interpreter interprets each command, shell scripts operate fairly slowly and use a large number of processing cycles. Therefore, Bourne Shell scripts should be used primarily for rapid CGI development or CGI prototyping. If a CGI will be used in high volume, you may want to consider constructing a more efficient C Language CGI or a Perl CGI.

13.1.2 Printenv.sh Example

A sample shell CGI is included in the printenv.sh file located in the Web[Ten] cgi-bin directory. The first few lines of the file establish the mandatory #!/bin/sh and echo Content-type: text/plain requirements for any shell script. The remaining shell script commands are used to output a few lines of constant text, followed by a dozen or more lines that output the values of a family of shell variables. The following is the content of the printenv.sh CGI:

```
#!/bin/sh
# disable filename globbing
set -f
echo Content-type: text/plain
echo
echo CGI/1.0 test script report:
echo
echo argc is $#. argv is "$*".
echo
echo SERVER_SOFTWARE = $SERVER_SOFTWARE
echo SERVER_NAME = $SERVER_NAME
echo GATEWAY_INTERFACE = $GATEWAY_INTERFACE
echo SERVER_PROTOCOL = $SERVER_PROTOCOL
echo SERVER_PORT = $SERVER_PORT
echo REQUEST_METHOD = $REQUEST_METHOD
echo HTTP_ACCEPT = "$HTTP_ACCEPT"
echo PATH_INFO = "$PATH_INFO"
echo PATH_TRANSLATED = "$PATH_TRANSLATED"

echo QUERY_STRING = $QUERY_STRING

echo SCRIPT_NAME = $SCRIPT_NAME
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_ADDR = $REMOTE_ADDR
echo REMOTE_USER = $REMOTE_USER
echo AUTH_TYPE = $AUTH_TYPE
echo CONTENT_TYPE = $CONTENT_TYPE
echo CONTENT_LENGTH = $CONTENT_LENGTH
```

When the printenv.sh CGI is referenced by a URL, it produces the following output:

```
CGI/1.0 test script report:

argc is 0. argv is .

SERVER_SOFTWARE = Apache/1.2.6.36 WebTen/3.0
SERVER_NAME = www.tenon.com
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
REQUEST_METHOD = GET
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
image/png, */*
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/printenv.sh
QUERY_STRING =
REMOTE_HOST = 192.83.246.60
REMOTE_ADDR = 192.83.246.60
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```

13.1.3 Shell Variables

Shell variables are pre-defined values set by Web[Ten] before the shell CGI is started. Shell variables are referenced by placing a "$" character in front of the name of the shell variable. If the shell interpreter finds a name that matches the string of characters following any "$" character, it substitutes the value of that variable in its processing. In the case of the echo command, the value of the $VAR shell variable is substituted as a parameter to the echo command and is output to the browser as a partial response to the URL request.

13.2 Perl CGIs

A Perl CGI is a text file that contains commands for the Perl language interpreter. The file name extension is usually "`.pl`", and the file is placed in the *cgi-bin* folder.

A Perl interpreter is included with Web[Ten], so Web[Ten] is able to interpret Perl scripts.

This document describes Web[Ten] Perl CGIs. A Perl CGI is a text file that contains commands for the Perl language interpreter.

Create a new CGI called mycgi.pl.

Store the newly created file in the cgi-bin directory, under the Web[Ten] cgi-bin directory. The new CGI can be referenced from a browser with the following URL: /cgi-bin/<cgi-name>.

13.2.1 Required Script Content

In addition to creating the text file, there are a few important considerations with respect to the content of the file. First, the top line of the file must contain the text:

```
#!/usr/bin/perl
```

This tells the Web[Ten] system that this is a Perl script and that Perl should be used to process the remainder of the file.

Second, you can use Perl print statements to generate text which will be returned to the browser that initiated the URL. The first print command must contain an HTTP header. This header indicates what format or kind of data will be output by the remainder of the print commands. The choices are usually plain text or text that is marked up using the HyperText Markup Language (HTML). This first print command puts Web[Ten] and the browser in the proper mode to accept everything else.

For Perl scripts that output plain text, use:

```
print "Content-type: text/plain \n\n";
```

For Perl scripts that output HTML statements, use:

```
print "Content-type: text/html \n\n";
```

The print indicates that text/plain or text/html will follow. After that, any text generated with a print command is sent to the originating browser as a response to the URL request.

Perl scripts are text files containing Perl language statements that generate a stream of text characters in response to being executed. There are Perl statements for assigning integer and string values to variables, statements for prescribing conditional flow through the script, and statements for running other programs. Very sophisticated CGIs can be created by combining different Perl statements. A number of widely available books describing Perl programming are available.

Programming Perl, Second Edition by Larry Wall, Tom Christiansen and Randal L. Schwartz, with Stephen Potter. 1996, O'Reilly & Associates

Perl is used for medium-performance, easy-to-develop CGIs. Each Perl program is text. The scripts are interpreted by a Perl interpreter controlled by WebTen. Since the interpreter interprets each Perl statement, Perl scripts can consume a lot of memory and use a large number of processing cycles.

13.2.2 Printenv.pl Example

A sample Perl CGI is included in the .printenv.pl file located in the WebTen cgi-bin directory. The first few lines of the file establish the mandatory #!/usr/bin/perl and print Content-type: text/plain requirements for any Perl script. The remaining two Perl statements output a dozen or more lines that contain the values of a family of environment variables. The following is the content of the printenv.pl CGI:

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";
while( ($key,$val) = each %ENV ) { print "$key = $val<BR>\n"; }
```

When the printenv.pl CGI is referenced by a URL, it produces the following output:

```
 SERVER_SOFTWARE = Apache/1.2.6.36 WebTen/3.0
GATEWAY_INTERFACE = CGI/1.1
DOCUMENT_ROOT = /usr/local/etc/httpd/WebSites/www.tenon.com
REMOTE_ADDR = 192.83.246.60
APACHE_PORT = 81
SERVER_PROTOCOL = HTTP/1.0
REQUEST_METHOD = GET
```

```
REMOTE_HOST = 192.83.246.60
QUERY_STRING =
HTTP_USER_AGENT = Mozilla/4.61 (Macintosh; I; PPC)
ADMIN_PORT = 84
PATH = /bin:/usr/bin:/usr/ucb:/usr/bsd:/usr/local/bin
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
image/png, */*
REMOTE_PORT = 1138
HTTP_ACCEPT_LANGUAGE = en,pdf
HTTP_CACHE_CONTROL = Max-age=259200
SCRIPT_NAME = /cgi-bin/printenv.pl
SCRIPT_FILENAME = /usr/local/etc/httpd/cgi-bin/printenv.pl
HTTP_ACCEPT_ENCODING = gzip
SERVER_NAME = www.tenon.com
REQUEST_URI = /cgi-bin/printenv.pl
HTTP_ACCEPT_CHARSET = iso-8859-1,*,utf-8
HTTP_X_FORWARDED_FOR = 192.83.246.60
SERVER_PORT = 80
HTTP_HOST = www.tenon.com
SERVER_ADMIN = webmaster@tenon.com
HTTP_VIA = 1.0 www.tenon.com:80 (Squid/1.1.20.6)
```

13.2.3 Environment Variables

Environment variables are pre-defined values set by Web[Ten] before the Perl CGI is started. Environment variables are referenced by the Perl statement $ENV{<env var>}. The Perl statement:

```
$ENV{PATH} = "/bin:/usr/bin";
```

sets the PATH environment variable. The Perl statement:

```
print $ENV{PATH};
```

prints the current value of the PATH environment variable.

13.3 C Language CGIs

A C language CGI is a computer program. To produce a C language CGI, you need to write the C language source program using any text editor. Then, a C language translator called a C compiler is needed to translate the C program into

machine language. The machine language file with the extension ".c" is stored in the *cgi-bin* folder in a file that can be executed by Web[Ten].

A C Language CGI is a computer program. To produce a C Language CGI you must first write the C Language source code using a text editor program. Once the program is written, a C Language translator, called a C compiler, is used to translate the C Language into machine language.

Create a new CGI called mycgi.c. Once the C Language source file is constructed, invoke the C Language compiler using the following format:

```
cc -O -o mycgi mycgi.c
```

This command produces a machine language file named mycgi using the C Language source found in the file mycgi.c. The resulting machine language file or objectfile is directly executable under Web[Ten]. You can use debugging techniques to ensure that the C Language CGI operates correctly. Once the CGI is complete, store the CGI in the Web[Ten] cgi-bin directory. Then reference the CGI with the following URL: /cgi-bin/mycgi

The CGI will be invoked by Web[Ten] and the output will be transported to your browser.

Basic Steps

- Create a C Language source file
- Compile and debug
- Place the file in the Web[Ten] cgi-bin directory
- Reference the file from a Web browser

C Language CGIs are used for high-performance CGIs. Since each C Language CGI is a compiled program, executing a C Language CGI reaps the full benefit of the fastest performance than can be delivered.

13.3.1 Printenv.c Example

The C Language CGI example included with Web[Ten] is in a file named printenv.c, which is located in the Web[Ten] cgi-bin directory.The printenv source code is in tenon/examples/printenv.c.text. Note that this code will not compile and run. It is only listed as an example of how to write C language CGIs. Below is the content

of the printenv.c CGI:

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
  char name[128];
  char val[128];
} entry;
void getword(char *word, char *line, char stop);
char x2c(char *what);
void unescape_url(char *url);
void plustospace(char *str);

entry entries[10000];

main(int argc, char *argv[]) {
  register int x,m=0;
  char *cl;

  printf("Content-type: text/html%c%c",10,10);

  if(strcmp(getenv("REQUEST_METHOD"),"GET")) {
     printf("This script should be referenced with a METHOD of GET.\n");
     printf("If you don't understand this, see this ");
     printf("<A
HREF=\"http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-
forms/overviewhtml\">forms
overview</A>.%c",10);
     exit(1);
  }

  cl = getenv("QUERY_STRING");
  if(cl == NULL) {
    printf("No query information to decode.\n");
    exit(1);
  }
  for(x=0;cl[0] != '\0';x++) {
    m=x;
    getword(entries[x].val,cl,'&');
    plustospace(entries[x].val);
    unescape_url(entries[x].val);
```

```
    getword(entries[x].name,entries[x].val,'=');
  }

  printf("<H1>Query Results</H1>");
  printf("You submitted the following name/value pairs:<p>%c",10);
  printf("<ul>%c",10);

  for(x=0; x <= m; x++)
   printf("<li> <code>%s = %s</code>%c", entries[x].name, entries[x].val,10);
  printf("</ul>%c",10);
}
```

This CGI prints the name/value parameter pairs that are available to any CGI when the CGI is invoked. The general flow of the printenv CGI is that it uses the printf statement to output Content-type: text/html\n\n. This is needed in order for the CGI to inform WebTen and the remote browser of the type of content to follow.

The program then verifies whether or not a GET type of HTTP request was used to initiate the CGI. If a GET request was not used, an error message is returned with several printf statements and the program exits. If a GET HTTP request is found, the environment variable QUERY_STRING is requested. If that string is unavailable, an error message is printed and the program exits. If QUERY_STRING is found, a for loop is entered. The for loop calls the getword subroutine to parse the string into name and value pairs. Once all of the parameters have been parsed, the printf subroutine is called several times to output a constant string "QUERY RESULTS", followed by the string "You submitted the following name/value pairs:", followed by a name and value pair on each line until all of the name/value parameters have been displayed. When the printenv CGI is referenced by the URL:

```
/cgi-bin/printenv?company=Tenon Intersystems&addr=1123 Chapala St.&city=Santa Barbara
```

it produces the following output:

```
Query Results
You submitted the following name/value pairs:
    company = Tenon Intersystems
    addr = 1123 Chapala St.
    city = Santa Barbara
```

13.4 Fast CGI

Web[Ten] includes built-in support for the execution of FastCGI scripts. FastCGI scripts are faster than normal CGI scripts because they are always running, whereas normal CGIs are re-loaded each time they are run. Any CGI can take advantage of FastCGI capabilities if the script's code is modified. Below is an example of the simple printenv.pl script in the form of a FastCGI. The "use CGI::Fast;" line makes the FastCGI capabilities available to the script. The "while" loop must contain the CGI's code. The "$query" variable will change every time the CGI is used by a client and therefore can be used to track which request is being processed.

```
#!/usr/bin/perl
  use CGI::Fast;
  while ($query = new CGI::Fast)
  {
        print "Content-type: text/html<BR>\n";
        while (($key, $val) = each %ENV) {
              print "$key = $val<BR>\n";
        }
  }
```

When a FastCGI such as this is run the first time, mod_fastcgi (an Apache module) spawns a process that keeps the script running while Apache is running. To have the FastCGI run automatically when Apache is first started, put the following lines in Web[Ten]'s httpd.conf file:

```
<IfModule mod_fastcgi.c>
FastCGIServer /usr/local/apache/cgi-bin/printenv.fcgi -processes 1
</IfModule>
```

These lines will create one instance of the printenv.fcgi script whenever Apache is run. The number of processes can be increased if more instances are needed to accommodate the volume of requests. All FastCGI scripts are named with the ".fcgi" extension by convention. Be sure to set the correct path to the FastCGI script in the Apache directives (/usr/local/apache/ is the path to the Web[Ten] folder.)

14.0 WEBmail

WEBmail is both an e-mail client and an e-mail server. Used with Web[Ten], it provides an interface to create and utilize e-mail mailboxes. This dual nature makes WEBmail a one-stop e-mail solution since it is both a self-contained server and client. WEBmail comes pre-configured with a full Web[Ten] installation or it can be installed separately.

With WEBmail installed, all that has to be done to get a working mail server is create new mailboxes. You can access the WEBmail account creation pages by the URL

http://host.yourdomain.com/webmail_adduser.

To immediately use WEBmail as a client, the login page URL is:

http://host.yourdomain.com/webmail

Note that to use WEBmail as a server, "mail" must be enabled in the Web[Ten] Preferences. See section "4.2 Preferences" for more information about the Web[Ten] preferences. Enabling the e-mail server increases memory usage significantly.

14.1 Using WEBmail as an e-mail Client

WEBmail is pre-configured to be a convenient e-mail client application. Using WEBmail as a client is as easy as loading the page

http://host.yourdomain.com/webmail

and entering your full e-mail address and password in the fields provided (see picture below).

The e-mail address entered must be fully qualified (e.g. joe@mail.tenon.com) as opposed to joe@tenon.com.)

Figure 72: WEBmail Login

When logging in, you must be sure to use your full e-mail address. For example, the address "user@domain.com" is not a full e-mail address (though you may receive mail at that address.) WEBmail requires that you include the hostname in your e-mail login so it can properly communicate with your mail server. A "fully qualified" e-mail address, four example, would be "user@mail.domain.com". You e-mail password is the same password you use to log into your e-mail account in any other mail client. Choose "GO" to log into WEBmail once you have entered your e-mail address and password.

Once successfully logged in, you will see a list of your e-mail messages. WEbmail is designed to offer all the features of an e-mail client. For help using the e-mail client portion of WEBmail, refer to the WEBmail documentation at "http://host.domain.com/web_mail/help/".

14.2 Adding a WEBmail mailbox

WEBmail includes an easy to follow web-based interface for creating accounts. The account creation process can be left up to the mailbox user thus minimizing monotonous administration. The WEBmail account setup process is outlined below and can be accessed on your Web[Ten] web server from the URL http://host.yourdomain.com/webmail_adduser (Note that this form is initially restricted to the webmaster and will ask for a user name and password.)

This is the first screen you see when creating a new WEBmail mailbox. Simply enter your name and choose "Submit." This is not the name that will be attached to the account, but a record that you agreed to the usage policy.

The next form will ask you what your mailbox login is to be. This will determine your e-mail address (e.g. newuser@domain.com).

After submitting your mailbox login, you will be asked for your personal information. None of it is required for the form to submit properly, but you must enter a valid e-mail address in the last field so that WEBmail can send confirmation e-mail. The confirmation e-mail must be replied to in order for the account to become active.

Figure 73: Choose WEBmail account password

The final step is entering a password for the new mailbox. Enter it twice and choose "Submit." WEBmail will reply with a message that confirms the account has been successfully created.

14.3 Customizing WEBmail

WEBmail can be customized to allow either user editable WEBmail client and adduser pages or WEBmail pages with no advertisements. These options must be purchased separately from Web[Ten]. For more information, see http://www.tenon.com/products/webmail or send an inquiry to sales@tenon.com.

15.0 ht://Dig

The version of ht://Dig included in WebTen has been extended with a CGI interface that supports the administrative tasks of creating and maintaining searchable databases in a fully integrated, multiple virtual host WebTen package.

ht://Dig is a very customizable utility. The WebTen indexing CGI is designed as an easy to use front-end to htdig. It provides a quick way to get a basic set of htdig's search capabilities working for each virtual host in a WebTen system. To further exploit the power of htdig, refer to the ht://Dig documentation (http://host.domain.com/htdig/doc/index.html). Note that the htdig configuration files created by the indexing CGI are stored in the /htdig/conf/<virtualhostname>.conf file for each virtual host.

You will probably want to customize the HTML search page and the results page from the defaults that are provided. Look in the ht://Dig documentation (http://host.domain.com/htdig/doc/index.html) for a description of the files that it uses for each page. Also look in the WebTen/tenon/apache/conf/httpd.conf file for the extra htdig configuration lines that were added by the WebTen Search Engine Installer. You might want to change these directives if, for example, you wanted to change the URLs for users to access the search engine for a particular virtual host or for your entire Web Server.

Once a searchable database has been built, it may be necessary to periodically rebuild the database to include new or changed pages that have been added to a site. To facilitate periodic updates, the indexing CGI can also be run as a CRON script. For more information, look in the /usr/local/apache/htdig/conf/crontab.tmpl file for some example crontab entries for invoking the indexing CGI.

The indexing process can create large database files. Almost every word that is retrieved from examining a document is stored into a sorted database file for later searching. This means that a lot of disk space may be required to successfully complete an indexing operation. A large site might require as much as 300 Mbytes of available disk space!

15.1 Build the Web[Ten] Search Engine Index File

The Web[Ten] Search Engine Index files are built and maintained using a special indexing CGI. This CGI is intended only for Web[Ten] Administrators and it is protected within the Web[Ten] Admin realm (username and password are required). Use the following URL to open the indexing CGI.

Substitute your Web[Ten] servers name into: http://hostname/index.cgi

The indexing CGI displays a form with a fields for entering the URLs to be indexed, excluded and limited and an optional email address.

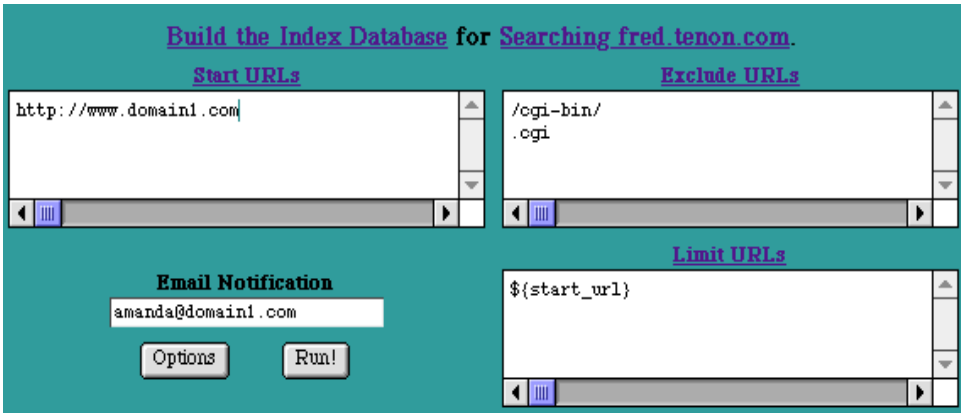


Figure 74: Default Indexing Options

The indexing form contains fields for specifying which URLs should be indexed. The Start URLs are the starting point for the indexing engine. The Exclude URLs are URLs that should not be indexed. The Limit URLs contains sets of patterns that the URLs must match.

The default Start URLs is a single URL matching the virtual host name used in the request. This default instructs the indexing process to visit all of the documents on this virtual host that are reachable (following any numbers of links) from the home page. The default Limit URLs specifies a set that exactly matches the set of Start URLs. In most cases, this is all that is needed to build a complete index of an entire virtual host. Additional URLs can be added to these lists.

The form also provides a field for an email address. It an email address is provided, the results of the indexing process will be emailed to that address.

Additional options may be displayed by clicking on the Options button. In this case, the form is displayed again with the default options shown (below). These defaults can then be modified. (The default options are used if the form is submitted without displaying the options.) The default settings are sufficient to create a search engine index (or database) file for the specified URLs.

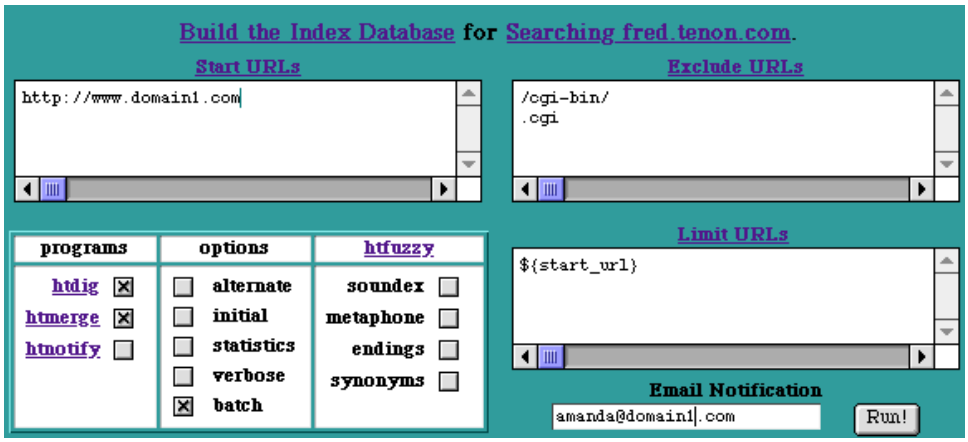


Figure 75: All Indexing Options

To begin the indexing process, click on the Run! button. The CGI will start a batch indexing process (if the batch options is specified) that continues to run after the CGI has completed. A link to a file which will contain the detailed results of the indexing process is provided. Note that it may take some time for the batch indexing process to complete. (For example, a default Web[Ten] installation takes about 10 minutes.) If
the results are referenced before the indexing process is complete, only the completed parts of the indexing process will be shown. Providing an email address is the best way to be notified when the entire indexing operation is complete.

To continually monitor the progress of the indexing process, uncheck the batch option before clicking on the Run! button. In this case, the output from the indexing process is continually displayed in the CGIs output and the CGI does not complete until the indexing process completes.

15.2 Test the Web[Ten] Search Engine Database

The best way to test the searchable database is to perform some actual searches. Use the following URL to search for a particular topic on the indexed site:

Substitute your Web[Ten] servers name into

http://host.domain.com/search.html

15.3 Multiple Virtual Hosts

The Web[Ten] Search Engine supports indexing and searching for multiple virtual hosts. By default, searchable databases are built on a per virtual host basis. For example, to build the index files for virtual hosts www.domain1.com and www.domain2.com, use the following URLs:

http://www.domain1.com/index.cgi
http://www.domain2.com/index.cgi

To search the databases for these virtual hosts, use the following corresponding URLs:

http://www.domain1.com/search.html
http://www.domain2.com/search.html

16.0 Plug-Ins and Apache Modules

Plug-Ins and Apache Modules add extra functionality to the WenTen server package. WebTen is compatible with both dynamically loadable Apache Modules and WebStar-style Plug-Ins.

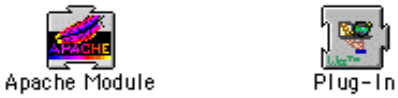


Figure 76: Apache Modules and Plug-Ins

16.1 Plug-Ins

Plug-ins must be installed in the *Plug-Ins* folder. Carefully read and follow the installation instructions provided with your plug-in to install any other files delivered with the plug-in, and to configure the plug-in for your server. WebTen must be restarted to activate (or deactivate) any newly installed (or un-installed) plug-ins. Use the *Restart Server* button in the *Server Controls* page, or quit and restart WebTen whenever any plug-in installations are completed.

16.1.1 Installing Plug-Ins

- Install according to the instructions included with the Plug-In package.
- Re-start WebTen, log into the WebTen admin server and make sure the Plug-In has registered (see section "6.2 Plug-In Administration" for information) and note the "Action" and "Suffix" entries in the Plug-In Administration table.
- Configure an Action Handler for the Plug-In with the Action that the Plug-In reported in the Plug-In Administration table (see "6.5.1 Configuring Plug-In Actions")
- Configure a MIME Extension for the Plug-In with the extension reported in the Plug-In Administration table (see "6.5.1 Configuring Plug-In Actions")
- Test the Plug-In with the content provided with the Plug-In package.

16.2 Apache Modules

Apache modules are the equivalent of WebSTAR plug-ins. WebTen includes many Apache modules and, in most cases, those modules can be configured via the WebTen Administration Server. The Apache Modules included are shown below.

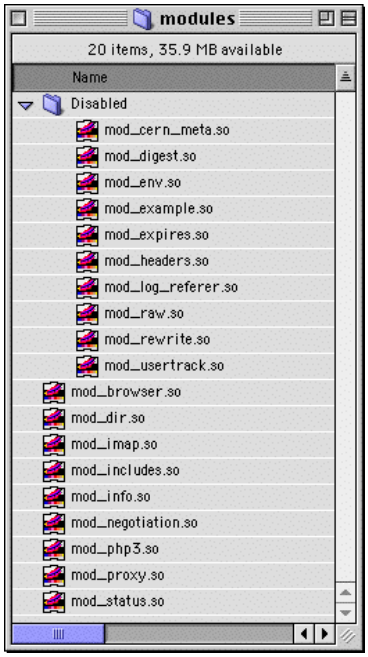


Figure 77: Included Apache Modules

In some cases, an Apache module provides the full functionality of a common WebSTAR-style plug-in.

16.2.1 Installing Apache Modules

To install an Apache Module, put the module file in the *WebTen/Modules* folder and re-start WebTen. Then, if the module requires a MIME Extension and an Action Handler, configure these as they would be configured for a Plug-In (see "16.1.1 Installing Plug-Ins"). Every Apache Module should include documentation that defines the module's requirements.

Appendix A

The Apache Web Server Copyright

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Group and was originally based on public domain software written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. For more information on the Apache Group and the Apache HTTP server project, please see <http://www.apache.org/>.

The Apache SSL Copyright

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and / or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project."
4. The names "Apache-SSL Server" must not be used to endorse or promote products derived from this software without prior written permission.
5. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project."

THIS SOFTWARE IS PROVIDED BY BEN LAURIE ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL

BEN LAURIE OR HIS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of patches to the Apache HTTP server interfacing it to SSLeay.

For more information on Apache-SSL, contact Ben Laurie <ben@algroup.co.uk>.

For more information on Apache see <http://www.apache.org>.

For more information on SSLeay see <http://www.psy.uq.oz.au/~ftp/Crypto/>.

The SSLeay Copyright

Copyright ©1996 Eric Young (eay@mincom.oz.au). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@mincom.oz.au). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@mincom.oz.au).

Please note that MD2, MD5 and IDEA are publicly available standards that contain sample implementations, I have re-coded them in my own way but there is nothing special about those implementations. The DES library is another matter.

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and / or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by Eric Young (eay@mincom.oz.au)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative

of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license (including the GNU Public License).

The reason behind this being stated in this direct manner is past experience in code simply being copied and the attribution removed from it and then being distributed as part of other packages. This implementation was a non-trivial and unpaid effort.

Washington University FTP Server Copyright

Release 2.2 Apr 1, 1994

wu-ftpd is a replacement ftp server for Un*x systems

The following notices apply to this package:

This product includes software developed by Washington University in Saint Louis and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B

Domain Name Service (DNS) Definitions

Alias	Aliases are redundant names or nicknames that are assigned to already named internet Hosts. Aliases permit the association of multiple names to a single host. Aliases may be useful when a host provides several different services or to transition a host to a new name while preserving compatibility via the old name.
Domain	A Domain is a collection of Hosts that are typically related via some logical organization like geographic or political boundaries, government beuracracies, and corporate divisions. However, a Domain does not require a particular organization, any set of Hosts may be collected into a Domain. Note that Domains themselves may be divided into sub-Domains, and so on.
Domain Name	Domain Names are the names assigned to collections of Hosts on the Internet. A Domain may comprise any collection of hosts, but typically a Domain is associated with some logical organization like geographic and political boundaries, government beuracracies, and corporate divisions.

Domain Name Server	A Domain Name Server is a computer that shares its database of DNS Resource Records which are typically Host Names and IP Addresses with other Hosts and Domain Name Servers. A small Domain Name Server may manage a small number of Zones, or a Zone comprising a single Domain, while a large Domain Name Server may manage a large number of Zones comprising several hundred Domains.
Domain Name System	The Domain Name System (DNS) is the address book for the Internet. The Domain Name System divides Hosts into collections called Domains much like Phone Numbers are divided into Area Codes. Note that Domains may themselves be divided into sub-Domains, and so on. The Domain Name System distributes the administration of the Domains between a large number of Domain Name Servers. There are several excellent online and printed Domain Name System References available on the Web.
Expire	The Expire value tells a Secondary Name Server how long to keep using a previously loaded database if the Secondary Name Server continues to fail to connect to a Master Server.
Host	A Host is the word used to refer to a computer. Typically the computer is connected to a network. Most Hosts are general purpose computers like Main Frames or PCs, but a Host may also be some other device like a Printer, an NFS Server, or a Router.
Host Information	Host Information is the name assigned to the collection of information or records that the Domain Name System manages for each Host.

Host Name	Host Names are the names assigned to internet Hosts. Host names are used in place of IP Addresses because they are much more readily remembered by humans.
Host Table	The Host Table is the name of the table displayed in the Primary Zone Page of the DNS Web Pages.
IP Address	IP Addresses (or Internet Protocol Addresses) are the internet address assigned to Hosts. They are usually represented in the Internet dot notation (e.g., `127.0.0.1`), and are typically assigned to a particular host by a network administrator.
Machine Name	The Machine Name was originally intended to to identify a Host hardware type (from the list on page 82, MACHINE NAMES in RFC 1340) but this list is out of date. In practice the Machine Name is used to hold any information a Network Administrator chooses to record. Machine Names are automatically quoted (with double quotes) after being entered into the DNS Web Pages. Therefore, Machine Names may contain a space (" "), but must not contain a double quote (" " ").
Mail Exchanger	Mail Exchangers are the records used by the Internet Mail Service to specify the best destination for mail sent to a particular Host.
Master Server	A Master Server is a Domain Name Server that is used to load zone information into other Domain Name Servers. A Master Server may serve both Primary Zones or Secondary Zones (loaded from some other Master Server).
Minimum Time-To-Live	The Minimum TTL value is included by a Domain Name Server in the response to any query requesting a record from this Server. The Minimum TTL tells the sender of the query how long it is acceptable to cache this record.

Precedence	Each Mail Exchanger is assigned a precendence indicating the order in which the Mail Exchangers should be used when sending mail to a particular Host. Mail Exchangers with the lowest precedence are used first, and if they are unavailable to receive the mail, Mail Exchangers with the next lowest precedence are use next, and so on.
Primary Zone	A Primary Zone is a Zone that a Domain Name Server serves from its local database.
Refresh	The Refresh value tells a Secondary Name Server how long to wait before checking with its Master Server for changes in the Master Server's database.
Resource Records	Resource Records is the name assigned to the collection of information or records that the Domain Name System manages.
Retry	The Retry value tells a Secondary Name Server how long to wait before trying to reestablish a connection with its Master Server if the Secondary failed to contact the Master Server at the end of the Refresh period.
Reverse Lookup Zone	A Reverse Lookup Zone is a Zone that contains the "reverse" mappings of IP Addresses to Host Names. Reverse Lookup Zones are easy to identify because their Domain Names always ends with "`.in-addr.arpa.`". Reverse Lookup Zones may be either Primary Zones or Secondary Zones. Primary Reverse Lookup Zones are handled automatically by the DNS Web pages, and do not appear in the table of Primary Zones. Secondary Reverse Lookup Zones are handled exactly like normal Secondary Zones, and do appear in the table of Secondary Zones.
Secondary Zone	A Secondary Zone is a Zone that a Domain Name Server loads from some other Domain Name Server, called a Master Server.

Serial Number	Each DNS Zone database file is assigned a serial number. When Web[Ten] modifies a Zone, it increments the serial number in the SOA record in the Zone file. Secondary name servers recognize updates in Zone files by comparing the serial number of the local copy of the Zone file against the Master's serial number.
Start of Authority	Start of Authority is the name given to the information with governs how often Name Servers communicate with each other to ensure that the information they are serving is up to date. The Start of Authority values for a Primary Zone on this system control how often other Domain Name Servers check with this system to verify that any cached information about this Zone is current. The Serial Number, Refresh, Retry, and Expire values are used by other Domain Name Servers if they are acting as Secondary Servers for this Zone. These values control how ofter to check with the Master Server, how long to wait if the Master Server is unavaliable, and how long to keep serving any cached information if the Master Server remains unavailable. The Minimum-Time-To-Live or Min TTL value is used by all Domain Name Servers that queries any piece of data within this Zone. The Min TTL tells these other Servers how long they may cache the data before checking back with this Server to see if the data has changed.

System Name	The System Name was originally intended to to identify a Host Operating System (from the list on page 86, SYSTEM NAMES in RFC 1340) but this list is out of date. In practice, the System Name is used to hold any information a Network Administrator chooses to record. System Names are automatically quoted (with double quotes) after being entered into the DNS Web Pages. Therefore, System Names may contain a space (" "), but must not contain a double quote (" " ").
Zone	A Zone is the portion of a Domain that is managed on a single Domain Name Server. A Zone may comprise an entire Domain, or only a part of a Domain, but a Zone is the smallest or atomic unit (within Domains) that Domain Name Servers manage.

Appendix C

Customizing Web[Ten]

The *httpd.conf* and *squid.conf* files are the main configuration database files manipulated by the Web[Ten] Administration Server. Features available to Apache and Squid but not yet configurable in the Administration Server may be added to these files, creating a customized Web site configuration. Some common site customization examples are described below.

Creating Virtual Domain-Specific CGI-BIN Folders

If a unique */cgi-bin* folder is desired for each virtual host, a Finder duplicate copy of the */cgi-bin* folder may be placed in the document root folder for each configured virtual host (see section "7.2 Virtual Host Configuration"). A custom directive is then added to *httpd.conf*:

Place the following directive in the affected <`VirtualHost`> container:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/WebSites/vhost.com/cgi-bin/
```

where "*vhost.com*" is the root folder for the virtual host.

Adding Additional Plug-In Suffix Mappings

This feature permits suffix mappings for plug-ins beyond those registered by the plug-in at server start up. For example, to configure the NetCloak plug-in to handle all *.html* files:

- Type "`CLOAK_PI`" in the *Action* field. The *CLOAK_PI* action is taken from the "Plug-In Administration" in section 6.2.
- Click on *Save Handlers.*
- Click on *MIME Extensions.*
- Type "`.html`" into the *Extension* field and select *CLOAK_PI* from the pull-down *Action* menu. Click on *Save MIME Extensions.*

Overriding a Plug-In Registered Suffix Mapping

This feature can be used to selectively disable an installed plug-in on a per-virtual host basis. Using the NetCloak plug-in as an example, we demonstrate how to prevent NetCloak from serving *.nclk* (the suffix automatically registered by NetCloak) files accessed from VirtualHost 192.0.0.1.

- In *httpd.conf*, place an *AddHandler* directive inside the <VirtualHost> serving the 129.0.0.1 domain.

```
<VirtualHost 192.0.0.1>
AddHandler */* .nclk
</VirtualHost>
```

Running UNIX CGIs Outside of CGI-bin

From the access controls page, set an action handler override to *cgi-script* for each directory where script execution is to be allowed.

Using Squid Proxy Services with WebTen

Under WebTen, the Apache server proxy module is configurable using the WebTen Administration Server. The Squid component of WebTen is used as a caching accelerator front-end to Apache.

But Squid can also provide a very powerful HTTP / 1.1 proxy service. Squid proxy access control mechanisms are more extensive than those in Apache 1.2, particularly in the area of proxy blocking, or the ability to prevent access to certain undesirable sites based on URL, browser, day, hour of the day, even minute of the hour! A simple adjustment will enhance Squid's capabilities under WebTen to provide both acceleration and proxy service.

Because Squid proxy configuration is not yet exported into the WebTen Administration Server interface, direct manipulation of the Squid configuration file is necessary. While this may at first appear daunting to someone unfamiliar with UNIX database files, it is really quite a simple process using a text editor such as BBEdit.

First, using the Administration Server interface, verify these default WebTen settings:

* ProxyRequests Off
* AcceleratorCache On

Next, put on UNIX scuba gear and dive into squid.conf:

The squid.conf configuration file is located in the tenon/squid/etc folder in your WebTen distribution.

* Find and change the line containing 'httpd_accel_with_proxy' to

  httpd_accel_with_proxy on

That's it! Restart the WebTen server using the Administration interface and direct your clients to the proxy on port 80 of your WebTen system.

Setting up Squid Access Control

The access control lists (ACLs) in squid.conf determine what the client browsers can access. The basic format of an acl record is:

```
acl aclname acltype string1 string2 ...
```

aclname is a unique identifying name you give to the acl

acltype is one of: src dst srcdomain dstdomain urlpath_regex port proto method browser user time

For example:

```
acl aclname src ip-address/netmask ...    (clients IP address)
acl aclname src addr1-addr2/netmask ...    (range of addresses)
acl aclname dst ip-address/netmask ...    (URL host's IP address)
acl aclname srcdomain foo.com ...    (taken from reverse DNS lookup)
acl aclname dstdomain foo.com ...    (taken from the URL)
acl aclname url_regex ^http://  ...    (regex matching on whole URL)
acl aclname urlpath_regex  \.gif$ ...    (regex matching on URL path only)
acl aclname port     80 70 21 ...    (port number)
```

```
acl aclname proto    HTTP FTP ...    (protocol)
acl aclname method   GET POST ...    (request method)
acl aclname browser  Mozilla$    (browser regex)
acl aclname user     username ...    (string match on ident output)
acl aclname time [day]  [h1:m1-h2:m2]    (time of day)

   day:
      S - Sunday
      M - Monday
      T - Tuesday
      W - Wednesday
      H - Thursday
      F - Friday
      A - Saturday
```

To activate an acl, use it in an http_accesss statement:

```
http_access deny aclname
http_access allow aclname
http_access deny !aclname
```

Deny will deny access to the aclname, while allow allows access to the aclname. A '!' preceding the aclname denies / allows access to all but the aclname.

There are some sample ACLs in the file which you can enhance using the examples shown below. In the examples, we block access to some less than desirable domains (aclname dirty_domains) and we restrict the downloading of files (aclname download_files) ending with .hqx, .bin, and .gz to all machines that access the proxy except for a trusted group of machines (aclname download_allowed).

Here is an example of ACL-based proxy blocking:

```
#
# These are some basic ACL definitions that come with Squid.
#

acl manager proto cache_object
acl localhost src 127.0.0.1 / 255.255.255.255
acl all src 0.0.0.0 / 0.0.0.0
```

```
acl SSL_ports port 443 563
acl Dangerous_ports port 7 9 19
acl CONNECT method CONNECT

#
# Now some custom ACLs
#

#
#
```

Appendix D

Guide to Using W*API Plug-Ins and AppleScript CGIs

This information is for end users and developers of W*API plug-ins / CGIs and others familiar with WebSTAR. Use it as guide for resolving differences in the serving of your Web content under Web[Ten] vs. WebSTAR and as a starting point for reporting Web[Ten] incompatibilities with your plug-in's / CGI's operation.

- **Web[Ten] Ignores File Creator and Type**

  Web[Ten] ignores the Macintosh file creator and type of the requested document when determining a handler for a URL. The Web[Ten] suffix mapping table has no entries for Macintosh creator / type.

- **W*API Global Parameter Modification**

  Plug-in / ACGI W*API requests to change global server parameters are reflected in the active (in memory) Web[Ten] database, but are not changed in the permanent Web[Ten] database. Permanent changes to Web[Ten] settings are made using the Web[Ten] Administration Server.

- **AppleScript Changes**

  AppleScripts containing the directive

  ```
  tell application "WebSTAR"
  ```

  must be changed to

  ```
  tell application "WebTen"
  ```

  for proper operation under Web[Ten].

- **Plug-In and Server Administration URL**

  Web[Ten] registers the "`.admin`" URL suffix for administration of plug-ins and the main server. Accessing the local URL "`/pi_admin.admin`" will invoke the Web[Ten] plug-in administrator with links to individual plug-in administration URLs and the Web[Ten] Administration Server.

- **WebSTAR Plug-Ins from StarNINE**

  Ironically, the plug-ins delivered with WebSTAR 2.X from StarNINE do not conform to the W*API specification. Functionality of these plug-ins cannot be assured under Web[Ten]; however, copying the *WebSTAR Settings* file to the Web[Ten] server root folder may provide limited capability of WebSTAR plug-ins under Web[Ten].

- **Apple Event Handlers**

  Plug-ins that attempt to install Apple Event handlers will not operate properly under Web[Ten]. Plug-ins should use the W*API Interprocess Communication Callbacks for sending Apple Events and receiving Apple Event replies.

- **Plug-In/CGI Virtual Host Operation**

  Plug-ins/CGIs that run correctly when called from the default Web[Ten] server but fail when called from a Web[Ten] virtual host may be making incorrect assumptions about an underlying WebSTAR run-time environment. Some tips for developers on taking advantage of the W*API in a Web[Ten] environment can be found at `http://www.tenon.com/products/webten/pluginvhost.html`.

- **The *piServerField* Parameter**

  Plug-in developers can determine if their plug-in is running under Web[Ten] by searching for "WebTen" in the text string returned by the *piServerField* parameter.

Appendix E

WebTen's Built-In Plug-Ins and CGIs

The following is a partial list of plug-ins and Apple CGIs that provide features included in WebTen. Note that in many cases WebTen provides functionality above and beyond that which is available from any plug-in. For example, WebTen provides "real" IP-based virtual hosting, UNIX Perl, shell and binary CGIs, and HTTP proxy service, none of which are available as plug-ins.

Name/Functionality	Type	Feature	WebTen Equivalent
ClearWay Simple Cache Manager	PI	Cache Flush Management	WebTen Admin or a protected CGI (see section "6.4 Server Controls")
ClearlyHome	PI	Virtual Domains	WebTen provides real virtual hosting (see section "7.1 Virtual Hosts Table")
MultiHome	PI/CGI	Virtual Domains	WebTen provides real virtual hosting
HomeDoor	PI+	Virtual Domains	WebTen provides real virtual hosting
FireSite	PI	Virtual Domains URL Redirection (see below)	WebTen provides real virtual hosting
WebSTAR Image Map	PI	Image Maps	Built-in Apache module (mod_imap)
WebSTAR Data Cache	PI	Caching	Squid Object Cache

Name/Functionality	Type	Feature	WebTen Equivalent
WebSTAR SSI	PI	Server-Side Includes	Built-in Apache module (mod_include)
WebSTAR Byte Server	PI	Byte Ranging	Core Apache feature
WebSTAR Admin Server	PI	Server Administration	WebTen Administration Server
WebSTAR Dir Indexer	PI	Directory Indexing	Built-in Apache module (mod_dir)
WebSTAR Mail	PI	e-mail serving	WEBmail
WebSTAR Proxy	PI	Proxy Server	Built-in Apache module (mod_proxy)
WebSTAR Virtual Hosts	PI	Virtual Domains	WebTen provides real virtual hosting
WebSTAR FTP	PI	FTP service	Built-in FTP server
WebSTAR Form Mail	PI	and web-based e-mail client	WEBmail
WebSTAR Search	PI	Web Site search and indexing	ht://dig

Appendix F

WebTen Apache Directives

WebTen adds some new directives to the core Apache modules and expands the functionality of some existing directives found in the *httpd.conf* file.

- **LogFormat**

  WebTen uses the configurable logger module, "`mod_log_config`", allowing customization of the *Transfer Log* file format. The default format is a string containing: `<date> <time> <result> <remote hostname> <URL> <bytes sent>`. This corresponds to the directive:

  ```
  LogFormat "%W %d %w %h %>U %b"
  ```

  where the letters following the percent sign mean:

`W`	Enable WebSTAR format: Tab separated record entries. Use of colon (":") instead of slash (" / ") in URLs. Use of "0" instead of "-" to indicate no bytes sent.
`d`	Date time
`w`	WebSTAR result string, one of: `OK`, `ERR!`, or `PRIV`
`h`	Remote host
`>U`	Colon separated URL, after any internal redirection of the original URL
`b`	Bytes sent

- **AcceleratorCache <on/off>**

  This directive controls the operation of the Squid Accelerator Cache.

- **CacheTransferLog <path-to-log-file>**

  When the Squid Accelerator Cache is active and logging, this directive denotes the *ServerRoot* relative path to the accelerator log file as requested by plug-ins / CGIs via the W*API.

- **LogRotation <log file count>**

  This directive controls the number of backup transfer logs maintained by the server during log rolling operations.

- **AuthDBMFormatNCSA <on/off>**

  Web[Ten] uses the "`mod_auth_dbm`" module, providing user authentication using DBM files. Web[Ten] accepts two formats for the DBM group file — the Apache format (user name *key*, list of groups to which the user belongs *value*), and the NCSA format (group *key*, list of users contained in the group *value*) as controlled by the *AuthDBMFormatNCSA* directive. The Web[Ten] Administration Server maintains the default Web[Ten] group file in NCSA format.

- **mod_wsapi Directives**

  The Web[Ten] W*API module adds a number of custom directives used to control plug-in / Apple CGI operation under Web[Ten].

  Please see the Apache online documentation on your Web[Ten] server at `/webten_docs/Apache/manual/mod/mod_wsapi.html` for more information.

Appendix G

Utilities Accessible to Perl and Shell CGI Scripts

Web[Ten]'s UNIX Connection

Web[Ten] runs on MacOS within Tenon's UNIX virtual machine. Web[Ten] has been incorporated into this environment using a very traditional UNIX organization, while preserving the Macintosh experience and the Finder's desktop integration. Most Web publishing tasks do not require any knowledge of the underlying system. However, to fully exploit Web[Ten]'s advanced features (e.g., shell and Perl CGIs) it may be necessary to understand some of the organization of the UNIX virtual machine's file system.

The UNIX virtual machine uses a stand-alone file system that is self contained inside a single Macintosh file called *tenon:Storage:WebTen Storage*. This file is specially formatted for the UNIX virtual machine; its contents are not accessible from the Finder desktop. Web[Ten]'s shell, Perl and binary CGIs do have access to this file system, and the information below is provided as a road map for these CGIs.

Traditional UNIX Paths

Several traditional UNIX file system directories — */bin*, */dev*, */etc*, */sbin*, */tmp*, */usr* and */var* — are preserved within Web[Ten]'s fast storage file. A small subset of UNIX commands have been included with this package to make the broadest possible capabilities available from within Web[Ten]. These commands may be invoked via shell and Perl CGI scripts. The utility programs that may be called from UNIX-style CGIs are shown below. The Bourne shell, for example, is accessible via */bin/sh*, while the file comparison program *diff* is accessible via */usr/bin/diff*.

/bin			
[	df	ls	sh
hostname	domainname	mkdir	sleep
cat	echo	mv	stty
chmod	hostid	pwd	sync
cp	kill	ps	test
date	ln	rm	

/usr/bin			
chcreat	false	perl5	tar
chtype	fgrep	perl5.004	tr
cmp	grep	printenv	true
diff	mactext	printf	unixtext
diff3	mail	sdiff	vi
du	nslookup	sed	yes
egrep	nvi	suidperl	tee
expr	perl	tail	touch

Sendmail

CGIs may send mail using either */usr/bin/mail* or */usr/sbin/sendmail*. This is for outgoing mail only.

The Web[Ten] Folder

The *WebTen* folder (the folder containing the Web[Ten] application and any published Web content) is not contained in the Web[Ten] fast storage file, but it is linked to the path */usr/local/etc/httpd/* within this file system, making it accessible to any CGI. Any files or sub-folders within the Web[Ten] folder are accessible by appending their name to this path. For example, the file *default.html* for a virtual host named `www.host1.com` would be accessible via */usr/local/etc/httpd/www.host1.com/default.html*.

Perl

A complete distribution of Perl 5 is included with Web[Ten]. The Perl executable is linked in several places for maximum compatibility with Perl scripts from other UNIX systems.

```
/usr/bin/perl
/usr/bin/perl5
/usr/bin/perl5.004
/usr/local/bin/perl
/usr/local/bin/perl5
```

The Perl libraries are installed in */usr/lib/perl5*. The *WebTen* folder contains a folder named *tenon*. Within the *tenon* folder, create a folder called *usr*, and within the *usr* folder create a *lib* folder for custom Perl libraries. Custom libraries can then be referred to via */usr/local/etc/httpd/tenon/usr/lib* within any Perl CGI that requires such modules.

INDEX

A

accelerator cache 72, 87, 105, 124
AcceleratorCache 106
Access Controls 72, 124, **131**
  MIME Type Overrides 136
  Realm-Based Requirements 134
  setting 131
  sub-folders inheriting 133
  table 131
  testing 131
access log, display 58
accessing configuration files 113
ACGIBinOnly 73, 126
ACGIReplyTimeout 110
Action Handlers 79, 91
  cgi-script 154
  Overrides 137
adding
  DNS hosts 173
  groups 101
  host aliases 175
  load balancing host aliases 176
  users 99
  virtual hosts 116, 169, 170
Admin
  name 47
  password, setting 28, 47
Admin Menu 28, **46**, 61, 101
  Cache Status 49, 88
  Change License 48
  Flush Cache 56, 89
  Save Display 56
  Set Admin Password 47, 61
  Start / Stop Admin Server 49
  Start / Stop Web Server 48, 88
  System Status 48, 52
  Web Server Status 48, 54
administration
  of DNS 171
  of plug-ins D-1
Administration Server 49, 170, E-2
  connecting to 30
  Start / Stop 49
Advanced Settings
  ACGIReplyTimeout 110
  KeepAlive 110
  KeepAliveTimeout 110
  MaxClients 108
  MaxKeepAliveRequests 110
  MaxRequestsPerChild 109
  MaxSpareServers 108
  MinSpareServers 108
  MyopicPlugInMode 111
  PITCPOpenTimeout 110
  Port 109
  StartServers 108
  Table 107
  TimeOut 109
alias
  settings 76
alias, WebTen 8, 175
"Allow then Deny" lists 132
"Allow" lists 132
anonymous
  FTP 151, 156, 161
    host header-based 159
  virtual hosts 158
Anonymous checkbox, FTP 156
Apache 14, 55, 60, 88, 113
  APIs 14
  architecture 3
  configuration files 33
  copyright A-1
    SSL A-2
  design 3
  directives 15
    AcceleratorCache F-1
    AuthDBMFormatNCSA F-2
    CacheTransferLog F-1
    LogFormat F-1
    LogRotation F-2
    mod_wsapi F-2
  documentation 33
  folder 33
  Group 1
  logs 32
  mod_auth_dbm F-2
  mod_dir E-2
  mod_imap E-1
  mod_include E-2
  mod_log_config F-1
  modules 14, 213
  port to MacOS 3

proxy
settings 80
proxy module 10
root folder 33
The Definitive Guide 35
URL for 1
use of native file system 7
Web Server 1, 54
thread 48
APIs 14
Apache 14
server 14
Apple
Event Handlers D-2
Events 111
AppleScript
changes D-1
AppleScript CGIs (ACGIs) 1, 11, 13, 14, 73, 74, 110, 111, 126, 127, D-1
ACGIReplyTimeout 110
examples 31
WebSTAR API-style 73, 125
AppleShare and FTP 151
AppleSingle
file encodings 151
file format 153
application heap, WebTen 19, 28
Application Programming Interface (API) 14
ARPA-funded Harvest research project 2, 4
.as file extension 152, 153
ASCII
mode in FTP 152, 153
text strings 15
"As-Is" documents 11
attacks, hacker 5
AuthDBMFormatNCSA F-2

B

BBEdit 17, 32
Berkeley Internet Named Domain (see BIND)
.bin file extension 152
BINA 17, 114
binary
CGIs G-1
mode in FTP 152
transfers via FTP 151
BIND
defined 165
enabling 166
Bourne shell CGIs 13, 191, G-1
built-in
CGIS E-1
domain name service 1
MIME extensions 94
table 95
plug-ins E-1
buttons, radio, in tables 63
byte range 12, E-2

C

C Language CGIs 13, 197
cache
accelerator 72, 87, 105, 124
flush 90
manager 50, 55
memory 105
object 4
plug-in E-1
WebTen 79
Cache Settings 87, 105
AcceleratorCache 106
Cache Settings Table 105
cache_mem 106
cache_stoplist 107
cache_swap 106
supercache_enable 106
swap_level1_dirs 106
swap_level2_dirs 106
Cache Status **49**, 88
Bytes Sent 50
Connections 50
data graph 51
display list 51
Hits 50
Ip Addr 50
Port 50
Up 50
Window 49
cache_mem 106
cache_stoplist 107
cache_swap 106
CacheDefaultExpire 82
CacheGcInterval 81
CacheLastModifiedFactor 81
CacheMaxExpire 81
CacheSize 81
CacheTransferLog F-1
CERN 3

certificate
server, SSL 139
Certificate Authority for SSL (CA) 139
Certificate Signing Request (CSR)
defined 140
generating 143
cgi-bin folder 13, 32, 154, 191
cgi-bin/script folder 33
CGIs 11, 14, 17, 91, 113, 191, 213, E-1
AppleScript 13, 14, 126, D-1
binary G-1
Bourne shell 13, 191, G-1
C, C++ 13, 197
defined 12, 191
dns-mgr 187
errors 58
examples 31
Fast 201
Perl 13, 113, 194, G-1
Perl CGIs 194
scripts 12, 32
debugging 72
sending mail G-3
Shell 191
shell 13, 113, G-1
Shell CGIs 191
support for 3
uploading via FTP 154
URL-based execution 154
Virtual Host Operation D-2
WSAPI 12
cgi-script Action Handler 154
Change License, Admin Menu 48
changing
host alias records 177
host name records 176
passwords 100
user names 100
WebTen license information 48
check boxes, in tables 63
child process 4
chunked transfers 12
cipher
ban 147
defined 145
examples 145
requirements 147
restrictions 139, 146
Clear Log Data 57
ClearlyHome E-1
ClearWay Simple Cache Manager E-1
CLF (Common Log Format) 15, 58
clipboard, cutting and pasting 45
clock service (Cron)
Enable Cron checkbox 190
starting 190
"close" command 37
CodeBuilder 13, 114
Common Gateway Interface (CGI) 35
defined 12, 191
Common Log Format (CLF) 15, 58
Config Log 90
configuration
alias settings 76
files
accessing 113
Apache 33
redirect settings 78
system-wide 67
virtual hosts 117
connecting
to the Administration Server 30
to the WebTen server 169, 170
content
negotiation 11
uploading via FTP 151
control panels
MacTCP 19
TCP/IP 19, 38, 167
converting file formats 17
copyright
Apache SSL A-2
Apache Web Server A-1
SSLeay A-3
Washington University FTP Server A-5
creator
fields 113, 151
MUMM 114
Cron 9
Enable Cron checkbox 190
starting 190
crontab file
defined 189
example 190
CSR 140
customizing WebTen 23

D

data
fork 113, 151, 152, 153
graphs 51
database interfaces 15
Date & Time 38, 39
debugging CGI scripts 72
default.html 29, 30, 31, 33, 70, 121
de-installing WebTen 24
"DELETE" and "PUT" requests 11
deleting
DNS Zones 186
hosts 176
Mail Exchangers 178
denial-of-service 6
"Deny then Allow" lists 132
"Deny" lists 132
directives
Apache 15
Squid 15
directory indexing E-2
DirectoryIndex 70
DirectoryIndex setting 121
disk space requirements 19
display
access log 58
errorlog 58
plug-in ssgs 58
DNS 7, 19, 171
adding
aliases 175
host aliases 175
hosts 173
load balancing host aliases 176
virtual hosts 116
administration 171
Alias, defined B-1
BIND
defined 165
enabling 166
changing
host alias records 177
host name records 176
deleting
load balancing host aliases 176
Zones 186
dns-mgr CGI 187
domain
defined B-1
name
defined B-1
server, defined B-2
system, defined B-2
"Enable DNS" checkbox 166
Expire, defined B-2
host
defined B-2
information, defined B-2
name, defined B-3
table, defined B-3
IP Address 39
defined B-3
Machine Name, defined B-3
Mail Exchangers
adding 178
defined 177, B-3
deleting 178
precedence 178
Master Server, defined B-3
Minimum Time-To-Live, defined B-3
New Primary Zone Page 182
New Secondary Zone Page 183
Precedence, defined B-4
Primary Zone
accessing 172
Address List 172, 180
configured 179
creating 171
from Secondary Zone 185
defined 171, B-4
Delete Host 172
deleting 171
Home Page 172
New Alias 172
New Host 172
Start of Authority 172, 181
Zone List 172
Refresh, defined B-4
Resource Records, defined B-4
Retry, defined B-4
reverse DNS lookup 120
table 180
zone 180
defined B-4
running
with DNS 169
without DNS 168
Secondary Zone 184
creating 171
defined 171, B-4
deleting 171
Serial Number, defined B-5
Settings

page 171
table 171
Start of Authority, defined B-5
System Name, defined B-6
Zone
defined B-6
registering 188
document
cache 10
type 95
DocumentRoot 77, 111, 119, 129
documents, "As-Is" 11
Domain Name
fully qualified 133
multiple 6
partially qualified 133
restrictions 132
"Allow then Deny" 132
"Deny then Allow" lists 132
Evaluation Order 132
No Restrictions 132
Domain Name Service (DNS) 7, 8, 19, 171
adding
aliases 175
DNS hosts 173
host aliases 175
load balancing host aliases 176
virtual hosts 116
administration 171
Alias, defined B-1
BIND
defined 165
enabling 166
changing
host alias records 177
host name records 176
deleting
load balancing host aliases 176
Zones 186
dns-mgr CGI 187
domain
defined B-1
name
defined B-1
server, defined B-2
system, defined B-2
"Enable DNS" checkbox 166
Expire, defined B-2
host
defined B-2
information, defined B-2
name, defined B-3
table, defined B-3
IP Address 39
defined B-3
Machine Name, defined B-3
Mail Exchangers
adding 178
defined 177, B-3
deleting 178
precedence 178
Master Server, defined B-3
Minimum Time-To-Live, defined B-3
New Primary Zone Page 182
New Secondary Zone Page 183
Precedence, defined B-4
Primary Zone
accessing 172
Address List 172, 180
configured 179
creating 171
creatingfrom Secondary Zone 185
defined 171, B-4
Delete Host 172
deleting 171
Home Page 172
New Alias 172
New Host 172
Start of Authority 172, 181
Zone List 172
Refresh, defined B-4
Resource Records, defined B-4
Retry, defined B-4
reverse DNS lookup 120
table 180
zone 180
defined B-4
running
with DNS 169
without DNS 168
Secondary Zone 184
creating 171
defined 171, B-4
deleting 171
Serial Number, defined B-5
Settings
page 171
table 171
Start of Authority, defined B-5
System Name, defined B-6
Zone
defined B-6
registering 188
downloading
files via FTP 152
dual

TCP stacks 6
dynamic content serving 12

E

"Edit" menu 45
email address, *ServerAdmin* 70, 119
"Enable DNS" checkbox 166
encrypted passwords 105
encryption algorithm in SSL 139
error
checking 59
codes 75
files 75
in CGI scripts 72
log, display 58
ErrorLog 71, 72, 121, 124
examples
AppleScript CGIs (ACGIs) 31
CGIs 31
expiry date, period 81
exporting
SSL files 148
SSL Keys 148
exporting user and group names 103
eXtended Server-Side Includes (XSSIs) 14

F

fast
file first aid 16
file system 7
storage 7, 16
FastCGI 201
field
creator 113, 151
type 113, 151
"File" menu 37
file
creators 113, D-1
downloading via FTP 152
encodings 151
AppleSingle 151, 152
MacBinary 151
extensions
.as 152, 153
.bin 152
formats 17, 104
Import and Export 104
image map 137
name
extensions 95, 96, 113, 136
system G-1
native 7
transfer 7
Binary mode 152
Image mode 152
types 113, D-1
uploading 153
File Menu
Preferences 38
File Transfer Protocol (FTP) 151
.as file extension 151
.bin file extension 151
anonymous 151
under NFS 161
Anonymous checkbox 156
AppleShare 151
ASCII mode 152, 153
binary
mode 152
transfers 151
cgi-bin folder 154
cgi-script Action Handler 154
client program 152
content uploading 151
downloading files 152
FTP folder 156
FTP Home setting 157
FTP Log button 157
FTP Settings Table 155
ftpaccess file 159
hidden folder 156
host header-based, anonymous 159
Image mode 152
incoming folder 156
Limit Setting 157
logging 157
multihomed anonymous 158
password
based clients 158
protection 151, 157
pub folder 156
simultaneous sessions 157
status 155
tenon/etc/ftpaccess file 159
text
mode 152, 153
transfers 151
uploading
CGI scripts 154
content 151

files 153
URL-based execution of CGI scripts 154
User-Pass checkbox 157
Users form 154
virtual anonymous 158
filtering
URLs 73, 126
via proxy 10
Finder
defined 16
running without 16
FireSite E-1
firewall 10
flush 90
Flush Cache 56, 89
Folder Contents
"Files" column 130
"Folders" column 130
SSL 146
table 129–130
forks
data 113
resource 113
Frontier 92
FTP 151
.as file extension 151
anonymous 151
under NFS 161
Anonymous checkbox 156
AppleShare 151
ASCII mode 152, 153
binary
mode 152
transfers 151
cgi-bin folder 154
cgi-script Action Handler 154
client program 152
content uploading 151
downloading files 152
FTP folder 156
FTP Home setting 157
FTP Log button 157
FTP Settings table 155
ftpaccess file 159
hidden folder 156
host header-based, anonymous 159
image mode 152
incoming folder 156
Limit setting 157
logging 157
multihomed anonymous 158
password
based clients 158
protection 151, 157
pub folder 156
simultaneous sessions 157
status 155
tenon/etc/ftpaccess file 159
text
mode 152, 153
transfers 151
uploading
CGI scripts 154
content 151
files 153
URL-based execution of CGI scripts 154
User-Pass checkbox 157
Users form 154
virtual anonymous 158
FTP folder 156
FTP Home setting 157
FTP Log button 157
FTP Settings table 155
ftpaccess file 159
modifying 154
FTPLog 72
fully qualified domain names 133

G

GIF 31, 96
global server parameters D-1
graphs, data 51
Groups 61, 101
adding 101
Import and Export 103
list 23
names, importing 104
NFS 161
table 101
Users in Group 102
table 102

H

hacker
attacks 5
protection 5
Harvest
ARPA-funded research project 2, 4
cache software 5

Header-Based Virtual Hosting 8
hidden folder, FTP 156
home page
  connecting to 29
  defined 29
  Tenon 1, 34
  WebTen 29
HomeDoor E-1
host
  alias record, changing 177
  alias, adding 175
  header-based
    anonymous FTP 159
    virtual hosting 116, 120, 166
  name 38, **39**
    identification 10
    record changing 176
  name-based virtual hosting 166
HostnameLookups 72, 124, 133
htDig 208
  Database 211
  Index File 209
  Indexing Options 210
  Multiple Virtual Hosts 211
HTML 11, 34, 63, 96
  documents 95
HTML / OS 26
HTTP 8, 10, 55
  protocol 95
httpd
  accelerator 5
  port 3
httpd.conf file 15, 161, F-1
hypertext documents 31
HyperText Markup Language (HTML) 96
HyperText Transfer Protocol (HTTP) 10

I

idle process 108
image maps E-1
  files 137
Import and Export 103
  file formats 104
importing
  SSL Files 150
importing user and group names 104
incoming folder, FTP 156
incoming requests 71
  logging 121
  virtual host settings 117
index.html 70, 121
indexing, directory E-2
inheritance 65
  flag 65
installation
  destination folder 22
  guide 20–22
  of plug-ins 14
  online instructions 20
  options 21
  saving settings 23
  status window 22
Internet 27, 31
Internet Software Consortium 165
Internic 188
IP-based virtual hosting 6, 116, 147, 166, 169, 170, E-1
  SSLCertificateFile 128

J

JPEG 31, 96

K

KeepAlive 110
Keep-Alive Connections 11
KeepAliveTimeout 110

L

Lasso 15
Launching WebTen 27
  on startup 40
  Startup Status window 27
legacy URL 120
libraries
  Perl G-3
license number, WebTen 48
Limit setting, FTP 157
lists
  "Allow then Deny" 132
  "Allow" 132
  "Deny then Allow" 132
  "Deny" 132
  pull-down, in tables 63
load balancing host aliases
  adding 176
  deleting 176
Log Config, Reset 57

Log Menu 57
Clear Log Data 57
Display Access Log 58
Display Error Log 58
Display Plug-In Msgs 58
Reset Log Config 57
LogFormat 71, 123, F-1
logging 15
incoming requests 71, 121
Logging checkbox, FTP 157
LogRotation F-2
logs
Apache 32
Squid 32

M

MacB command 151, 152, 153
MacBinary
file encodings 151
mode 152, 153
Macintosh Operating System(MacOS) 1, 53
MacOS 1, 53
MacPerl CGI 35
MacTCP
control panel 19
MacTCPdLib file, moving 41
Mail Exchangers, DNS 177
adding 178
deleting 178
precedence 178
map
file name to a language 97
file name to a Mime Encoding 98
image E-1
image, files 137
MaxClients 108
MaxKeepAliveRequests 110
MaxRequestsPerChild 109
MaxSpareServers 108
memory
allocating 28
cache 105
requirements 19
virtual 19
Messages 89
MIME
defined 95
encodings 98
extensions 91, 94, 96
built-in 94
MIME type 91, 94
overriding default extensions 95
user-defined 94
languages 97
type
overrides 136
typing system 95
minimum system requirements 38
MinSpareServers 108
mod_auth_dbm F-2
mod_dir E-2
mod_imap E-1
mod_include E-2
mod_log_config F-1
mod_wsapi F-2
modules
adding 14
Apache 14, 213
MPEG Movie Format 96
MultiHome E-1
multihomed
anonymous FTP 158
TCP stack 6, 8, 40
multiple
domain names 6
first-class URLs 7
Multipurpose Internet Mail Extensions (MIME), defined 95
multitasking, preemptive 1, 7
MUMM 17
creator 114
myopic plug-ins 111
MyopicPlugInMode 111

N

name-based virtual hosting 166, 169, 170
native file system 7
navigating the administration pages 63
NCSA 3
Network File Service (NFS) 7
access points 162
anonymous FTP requests 161
configuring 161
defined 161
group IDs 161
httpd.conf file 161
local path 163

read only access 161, 163
server 162
path 162
settings table 162
user IDs 161
networking 5
OpenTransport 42
TCP/IP 23
New Primary Zone Page 182
New Secondary Zone Page 183
NFS
access points 162
anonymous FTP requests 161
configuring 161
defined 161
group IDs 161
httpd.conf file 161
local path 163
read only access 161, 163
server 162
path 162
settings table 162
user IDs 161
NoCache 82
NoFinder 16
non-IP-based virtual hosts 10

O

Object caching 4
OpenTransport 1, 5, 6, 8, 38, 42
networking with 42
Replacing 23, 40, 167, 169, 170
operators
"POST" 74, 127
"PUT" 74, 127

P

packet buffers 53
parent
folder 130
process 4
partially qualified domain names 133
password 99
administrator, setting 28
based FTP clients 158
changing 100
encrypted 105
field 47
protection under FTP 151, 157
unencrypted 105
PDF 11, 12
performance, WebTen 2
Perl G-3
CGIs 13, 113, 194, G-1
libraries G-3
persistent connections 11, 12
PHP 26
PIAccessControl 73, 126
ping-of-death 6
PIPostProcessing 74, 127
PIPreProcessing 74, 126
piServerField D-2
PITCPOpenTimeout 110
plug-ins 11, 31, 73–74, 125–127
administration 79, D-1
Apple CGI Defaults
ACGIBinOnly 73, 126
PIAccessControl 73, 126
PIPostProcessing 74, 127
PIPreProcessing 74, 126
PostProcessor 74, 127
PreProcessor 74, 126
RequestFiltering 73, 126
WSAPIPostArgSize 74, 127
WSAPIRequests 73, 125
built-in E-1
display messages 58
installing 14
interfaces 79
myopic 111
MyopicPlugInMode 111
piServerField D-2
security 73, 126
settings 79
sub-folder 32
support for 1
virtual host operation D-2
W*API D-1
WebSTAR 1
Port 109
"POST" operator 74, 127
PostProcessor 74, 127
Power Macintosh 19
preemptive multitasking 7
"Preferences" menu item 37
Preferences 38, 167
DNS IP Address 168–169
Domain Name 168–169

Host Name 168–169
Preferences folder 24
Preferences window 23, 27
PreProcessor 74, 126
Primary Zone
accessing 172
Address List 172, 180
configured 179
creating 171
creating from Secondary 185
defined 171
Delete Host 172
deletiing 171
Home Page 172
New Alias 172
New Host 172
Start of Authority 172, 181
Zone List 172
processes
child 4
idle 108
parent 4
protection, from hackers 5
proxy
access 85
domain name restrictions 86
ProxyBlock 86
filtering 10
garbage collection 81
security 10
server, defined 4
services 10
settings
CacheDefaultExpire 82
CacheGcInterval 81
CacheLastModifiedFactor 81
CacheMaxExpire 81
CacheSize 81
NoCache 82
Proxy Access 85
ProxyRequests 80
Proxy Settings 80
ProxyBlock 86
ProxyPass 84
ProxyRemote 83
ProxyRequests 80
pub folder, FTP 156
pull-down lists, in tables 63
"PUT" operator 74, 127
"PUT" and "DELETE" requests 11

Q

Quick Start Guide 27–31
QuickTime Movie Format 31, 96
"Quit" menu 37
Quitting WebTen 34, 37

R

radio buttons, in tables 63
Raw Text 96
Realm 134
name 135
Realm-Based
access controls 99, 101, 135
requirements **134**
Any Valid User 135
Realm Name 135
Selected Users 135
Users in Group 135
redirect settings 78
redirection URLs 120, E-1
references, Web server 34
Remote
proxies 83
ProxyPass 84
ProxyRemote 83
startup 60
removing an item from a table 64
Replace OpenTransport setting 23, 166
Replacing OpenTransport 23, 40, 167, 169, 170
RequestFiltering 73, 126
requests
incoming 71
logging 121
requirements
disk space 19
memory 19
Reset button 65
Reset Log Config 57
resource fork 113, 151
Restart Server 88, 212
restarting
WebTen 167
your Macintosh 22
reverse DNS lookup 120
table 180
zone 180
running
with DNS 169
without DNS 168

without the Finder 16

S

safeguarding SSL Keys and certificates 148
"Save" button 64
Save Display 56
"SaveCSR" button 143
scripting 12
for temporary files 33
ScriptLog 72
Secondary Zone 184
creating 171
defined 171
deleting 171
Secure Socket Layer (SSL) 7
Certificate Authority (CA) 139
Certificate Signing Request (CSR)
defined 140
generating 143
cipher 139
ban 147
defined 145
examples 145
requirements 147
restrictions 141, 146
defined 139
enabling 145
encryption algorithm 139
exporting
SSL Certificates 149
SSL Files 148
SSL Keys 148
hosts 147
importing
SSL Certificates 150
SSL Files 150
SSL Keys 150
"SaveCSR" button 143
security 146
server certificate
defined 139
generating 141
multiple 147
obtaining 140
renaming 140
validity of 144
SSL Certificate
safeguarding 148
SSL Keys
exporting 148
safeguarding 148
SSL Settings 141
form 140
page 141
SSLBanCipher list 147
SSLSecurity directive 119
Thawte Consulting 140
secure transmission 1
security
plug-ins 73, 126
SSL 146
via proxy 10
Sendmail G-3
server
administration E-2
APIs 14
certificate
defined 139
for SSL 140
generating 141
multiple 147
obtaining 140
renaming 140
tenon/ssl/certs folder 128
validity of 141, 144
Server Controls 87, 212
Cache Status 88
Config Log 90
Flush Cache 89
Messages 89
Restart Server 88
Server Info 88
Server Status 88
Startup Log 89
Server Defaults 68–74
ACGIBinOnly 73, 126
Changing 68
DirectoryIndex 70, 121
ErrorLog 71, 121
HostnameLookups 72, 124
LogFormat 71, 123
RequestFiltering 73, 126
ScriptLog 72
ServerAdmin 70, 119
ServerAlias 120
ServerName 120
table 68, 69, 75
TransferLog 71
WSAPIRequests 73, 125
Server Info 88
Server Status 88
ServerAdmin 70, 119
ServerAlias 120

ServerName 120
ServerPath 111, 120
Server-Side Includes (SSIs) 14, E-2
sessions, simultaneous under FTP 157
Set Admin Password 28, 47, 61
settings, advanced
ACGIReplyTimeout 110
KeepAlive 110
KeepAliveTimeout 110
MaxClients 108
MaxKeepAliveRequests 110
MaxRequestsPerChild 109
MaxSpareServers 108
MinSpareServers 108
MyopicPlugInMode 111
PITCPOpenTimeout 110
Port 109
StartServers 108
TimeOut 109
Sharing Setup 38
Shell CGIs 191
shell CGIs 13, 113, G-1
simultaneous
sessions under FTP 157
Sound Format 96
Squid 2, 10, 113
Accelerator Cache 80, F-1
Cache Status 88
configuration files 33
defined 5
directives 15
logs 32
Object Cache 4, 5, 7, E-1
use of native file system 7
SSL 7
Certificate Authority (CA) 139
Certificate Signing Request (CSR)
defined 140
generating 143
cipher 139
ban 147
defined 145
examples 145
resquirements 147
restrictions 141, 146
defined 139
enabling 145
encryption algorithm 139
exporting
SSL Certificates 149
SSL Files 148
SSL Keys 148
hosts 147
importing
SSL Certificates 150
SSL Files 150
SSL Keys 150
"SaveCSR" button 143
security 146
Self-Signed Certificates 147
server certificate
defined 139
generating 141
multiple 147
obtaining 140
renaming 140
validity of 144
Settings 141
form 140
page 141
SSL Certificate
safeguarding 148
SSL Keys
exporting 148
safeguarding 148
SSLBanCipher list 147
SSLSecurity directive 119
support for secure transmissions 1
Thawte Consulting 140
SSL Certificate
exporting 148
importing 150
SSL Key file 148
SSL Keys
importing 150
SSLCertificateFile 128
sslcerts.cgi 150
SSLeay copyright A-3
Start / Stop
Admin Server 49
Web Server 48, 88
StartServers 108
Startup
Items Folder 40
Log 89
Startup Status window 27
status, thread 55
Sub-Folder Contents 130
supercache_enable 106
swap_level1_dirs 106
swap_level2_dirs 106

SYN attacks 6
system
heap 19, 53
requirements 19, 27, 38
System Status 48, **52**
(Pkts) In Use 53
"In" value 53
"Out" value 53
Cache: CPU 52
data graph 53
Iface 53
In Use 53
Mem Free 53
Net 53
Pkts Free 53
Threads 52
Web Server: CPU 52
System-Wide Configuration 67
Action Handlers 91
advanced settings 107
Cache Settings 105
Groups 101
MIME
Encodings 98
Extensions 94
Languages 97
Proxy Settings 80
Remote Proxies 83
Server Controls 87
Users 99

T

table, removing an item from 64
Tango 15
TCP / IP 5, 6, 39, 55
control panel 19, 38, 167
networking with 23
Tenon's TCP stack 6, 8
running dual stacks 6
Tenon
home page 1, 34
TCP stack 6, 8, 40
tenon
folder 33, 105, G-1
tenon/etc/ftpaccess file 159
tenon/ssl/certs folder 128
text
edit fields, in tables 63
editors 17
mode in FTP 152, 153
strings, ASCII 15
transfers via FTP 151
TEXT type 114
Thawte Consulting 140
thread status 55
TIFF 96
Time Zone 38, **39**
TimeOut 109
TransferLog 72, 121, 124
TransferLog 71
transfers, chunked 12
transmission, secure 1
type
BINA 114
document 95
field 113, 151
TEXT 114

U

"undo" command 65
unencrypted passwords 105
UNIX
commands G-1
shell scripts 12
virtual machine 1, 3, G-1
Unix<->Text 17, 32
uploading
CGI scripts 154
content via FTP 151
files via FTP 153
URLs 7, 10, 34, 83
filtering 73, 126
legacy 120
redirection 120, E-1
user-defined
action handlers 91
MIME Extensions 95
MIME extensions 94
User-Pass checkbox 157
Users 99
adding 100
form, FTP 154
groups and NFS 161
IDs and NFS 161
in Group 102
table 102
list 23
name 99

changing 100
importing 104
table 61, 100
Using 191
Utilities 16
Fast File First Aid 16
NoFinder 16
Unix<->Text 17

V

version number, WebTen 90
virtual
anonymous FTP service 158
domains E-1
memory 19
virtual hosting 1, 6, 8, 10, 29, 70, 120, 166, E-1
defined 6, 115
IP based 147
requirements 166
Virtual Hosts 7, **115**, 115–121
Adding 116, 169, 170
challenged 111
configuration 116, **117**
ACGIBinOnly 126
DirectoryIndex 121
DocumentRoot 119, 129
ErrorLog 121
HostnameLookups 124
LogFormat 71, 123
RequestFiltering 126
ServerAdmin 119
ServerAlias 120
ServerName 120
ServerPath 120
Table 118, 125
table 68, 75, 117, 141, 146
VirtualHost 118
WSAPIRequests 125
deleting 116
folder, renaming 119
host header-based 116, 166
host name-based 166
IP address-based 116, 166, 169, 170
SSLCertificateFile 128
name-based 166, 169, 170
operation of plug-ins and CGIs D-2
settings 117
SSLSecurity directive 119
Table 115, 145, 169, 170
VirtualHost 118
VRML documents 31

W

W*API
module 15
Plug-Ins D-1
Washington University FTP Server
copyright A-5
Web Server
References 34
Start / Stop 48
Web Server Status 48, 54
Bytes Sent 54
CPU 55
Hits 54
Servers 55
thread status 55
WebCrossing 26
WebEvent 26
WEBmail 202
Adding a mailbox 204
Customizing 207
Using as e-mail Client 202
WebSTAR 104, D-1
Admin Server E-2
Byte Server E-2
Data Cache E-1
Dir Indexer E-2
Format checkbox 72, 123
Image Map E-1
logging 15
plug-ins
Apache equivalents 14, 213
APIs 1
from StarNINE D-2
SSI E-2
WebTen 20
Administration **59**
Adding Entries 64
Making Changes 64
Removing Entries 64
Resetting Entries 65
System-Wide 67
Administration Server 14, 23, 30, 33, 47, **59**, 63, 115, 170, 213, D-1
connecting to 30
alias 8
application heap 19, 28
architecture 3, 167, 171
diagram 2
Built-In Plug-Ins and CGIs E-1
cache 79

configuration files 33
connecting to 169, 170
customizing 23
defined 1
de-installing 24
enabling 42
features 7
folder 27, 31, G-3
Home Page 29
connecting to 29
installation options 21
launching
Startup Status window 27
license number 48
menus 37
overview 1
performance 2
Preferences file 24
quitting 34, 37
restarting 167
running
with DNS 169
without DNS 168
Tenon's TCP stack 5
version number 90
Web Server Status Window 4
webten_admin 60
WebTenAdmin group 61, 101, 135
World Wide Web (WWW) 27
WSAPI 14
CGIs 3, 12
wsapi_module 14
WSAPIPostArgSize 74, 127
WSAPIRequests 73, 125
WWW (World Wide Web) 27

X

XSSIs (eXtended Server-Side Includes) 14

Z

Zone
deleting 186
Primary
accessing 172
Address List 172
Delete Host 172
Home Page 172
New Alias 172
New Host 172
Start of Authority 172
Zone List 172
Secondary 184
creating 171
defined 171
deleting 171