hp-ux web server suite

january 2004

migration guide

Migration Guide

iPlanet Web Server to HP-UX Web Server Suite

January 6, 2004

Glossary

Apache	Apache HTTP Server Project developed and maintained by ASF
ASF	Apache Software Foundation
HP-UX Apache	HP-UX Apache-based Web Server
IPF	Itanium Processor Family
IPv6	Internet Protocol Version 6
iWS	iPlanet Web Server, Enterprise Edition 4.1 (iWS EE 4.1)
JDK	Java Development Kit
LDAP	Lightweight Directory Access Protocol
MPM	Multi-Processing Module
NES	Netscape Web Server 3.6 (NES 3.6)
PA-RISC	Precision Architecture, Reduced Instruction Set
PHP	Hypertext Preprocessor
WebDAV	Web-based Distributed Authoring and Versioning

Publication History

Version 6	January 6, 2004	Updating for product restructure and naming to HP-UX Web Server Suite
Version 5	September 3, 2002	Updating mod_perl DSO, auth_ldap 1.6 for HP Apache-based Web Server 1.3.26.06
Version 4	August 12, 2002	Updating OpenSSL v.0.9.6g, mod_ssl v.2.8.10 and MM v.1.2.1 for HP Apache-based Web Server 1.3.26.05 and 2.0.39.05
Version 3	July 24, 2002	Updating PHP to 4.2.2
Version 2	June 26, 2002	Update for HP Apache-based Web Server 2.0.39 and 1.3.26
Version 1	May 16, 2002	First Release

Legal Notices

The information in this document is subject to change without notice.

Warranty Disclaimer. HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS INFORMATION, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance or use of this material.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies. Rights for non-DOD U.S. Government Department and Agencies are as set forth in FAR 52.227-19 (c)(1,2).

Acknowledgements. This product includes software developed by the Apache Software Foundation.
This documentation is based on information from the Apache Software Foundation (http://www.apache.org).

This product includes software developed by the OpenLDAP Project (http://www.openldap.org).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes PHP, freely available from (http://www.php.net).

More information on the HP-UX Web Server Suite can be found at http://www.hp.com/go/webserver

Conventions in this Guide

The following typographical conventions are used in this guide:

$**cat**	**Boldface** type with a "$" represents command or keywords that the user must enter.
ports.sh	**Boldface** type is also used for program names.
<*italic*>	*Italic* text within angle brackets indicates variable values, placeholders and function argument names.
`Fixed Width`	`Fixed Width` typeface indicates information that the computer displays. Examples include source code, file content, and directory paths.
→	A right pointing arrow represents a separator between mouse clicks. For example, go to software.hp.com then click on Featured Products then click on HP-UX Apache-based Web Server: http://software.hp.com→Featured Products→HP-UX Apache-based Web Server
http://	http:// refers to external documents
Appendix	Underlined text refers to a section of the migration guide

1 Using this Guide

This guide covers the migration of Netscape Web Server (NES 3.6) and iPlanet Web Server, Enterprise Edition (iWS EE 4.1) to the HP-UX Web Server Suite. The suite currently runs on these versions and later of HP-UX:

Release	HP-UX 11.0 PA-RISC	HP-UX 11i (11.11) PA-RISC	HP-UX 11i Version 1.6 (11.22) IPF	HP-UX 11i Version 2 (11.23) IPF
32-bit	yes			yes
64-bit			yes	yes
IPv6 (Extended IP addressing)		yes		default

Included in this guide is information on how to:

- Find more resources on the HP-UX Web Server Suite
- Understand how and what functionality is migrated
- Prepare for a migration
- Perform a migration
- Verify the migration

The focus of this document is on setting up HP-UX Web Server Suite with functionality similar to Netscape and iPlanet web server functionality. Appendix A Summary of Web Server Functionality Differences provides a high-level view of basic functionality differences between iPlanet and HP-UX Web Server Suite. A broad range of technical topics are discussed here, some of which may not apply to your own web server setup. Select the subset of requirements and migration steps that are applicable to your installation.

This guide is not intended as a general reference for setting up and running the HP-UX Web Server Suite Because of this focus, not all of the suite's features are described here. Additional features are listed in the Overview of HP-UX Web Server Suite. Please consult the overview and Getting More Information sections for general information about the HP-UX Web Server Suite.

Steps are only shown for migrating to the HP-UX Web Server Suite. Previous releases of Apache by HP on HP-UX (HP Apache-based Web Server) will not be described. For complete information on the differences between HP Apache-based Web Server 1.3.x and HP-UX Web Server Suite, refer to the Migration Guide HP Apache-based Web Server Version 1.3.x to HP-UX Web Server Suite, http://www.hp.com/go/webservers→hp-ux web server suite→Technical Tips. For a PDF version of this document, click on the link, http://www.hp.com/products1/unix/webservers/apache/techtips/index.html

Guide Layout

Overview of HP-UX Web Server Suite describes the suite and its components including the HP-UX Apache-based Web Server.

Getting More Information addresses where to go for in-depth content about HP-UX Web Server Suitereleases and their features.

HP-UX Web Server Suite Requirements contains detailed information about the pre-requisites for product installation including revision numbers and patch information.

Tools and Utilities introduces a set of tools that can be used during certain migration steps.

Performing the Migration consists of a series of detailed steps to migrate specific features. These steps can be performed in any order. Select which features will be migrated from iPlanetthen execute the corresponding implementation steps. Each step includes,

1. A label to identify the functionality category (Core 1, Core 2, Security 1, Security 2, etc.)
2. A brief description of the feature
3. Directions on how to find the feature's setting(s) in your iPlanet web server
4. Commands for setting the equivalent value(s) in the HP-UX Web Server Suite
5. Verification of migration for the feature

Appendix A Summary of Web Server Functionality Differences gives a big-picture view of how features between iPlanet and the HP-UX Web Server Suite compare.

Appendix B Important HP-UX Web Server Suite Files gives the location of important HP-UX Web Server Suite files such as sample files, server bundled documentation and startup/shutdown scripts.

Appendix C Web Servers Component Reference is a table that displays iPlanet and HP-UX Web Server Suite features side-by-side for a quick reference of what each web server contains.

Appendix D Open Source Apache versus HP-UX Web Server Suite is a single table that displays ASF Apache and HP-UX Web Server Suite components as a quick reference for their content and component version numbers.

2 Overview of HP-UX Web Server Suite

The HP-UX Web Server Suite is a free product for the HP-UX platform. It includes key software components necessary to deploy, manage, and implement mission critical web servers. These components are integrated, bundled, tested, distributed, and supported by HP as the HP-UX Web Server Suite product. Support is provided by the Hewlett-Packard Response Center as part of the HP-UX operating system.

The HP-UX Web Server Suite is comprised of:

- HP-UX Apache-based Web Server
- HP-UX Tomcat-based Servlet Engine
- HP-UX Webmin-based Admin
- HP-UX XML Web Server Tools

These components are based on software developed by the Apache Software Foundation (http://www.apache.org) except for the HP-UX Webmin, which is based on the open source Webmin (http://www.webmin.com/).

- HP-UX Apache-based Web Server is based on software developed by the Apache Software Foundation (ASF) and combines numerous popular modules from other Open Source projects for scripting, content management, and security. It also contains HP value-added features for the HP-UX platform
- HP-UX Webmin-based Admin is a Configuration and Administration GUI with extensive enhancements for the HP-UX Apache-based Web Server.
- HP-UX Tomcat-based Servlet Engine provides customers Java-based extensions for dynamic content generation via Servlets and JavaServer Pages (JSPs).
- HP-UX XML Web Server Tools is a collection of a Java-based XML tools used for XML parsing, stylesheet and XSL processing, web-publishing and image translating from the open source projects: Xerces, Xalan, Cocoon, FOP and Batik.

NOTE: The HP Apache-based Web Server 1.3.x has been obsoleted. For more information, please read the support message on the HP-UX Web Server site, http://www.hp.com/products1/unix/webservers/apache/support/index.html

2.1 Summary of Features in HP-UX Web Server Suite

HP-UX Web Server Suite Features	Description
LDAP authentication	Authenticates users in iPlanet (Netscape) Directory Servers or OpenLDAP Directory Servers
Stunnel control utility	Enables secure (SSL) transactions between HP-UX Apache-based Web Server and an LDAP Directory Server
Secure Socket Layer (SSL) mod_ssl, OpenSSL	Secure transactions with 128-bit/168-bit encryption. Optimized for HP platforms.
mod_perl	Enables Perl shared libraries and faster Perl CGI scripts
Tomcat	Java servlets and JSP run by Tomcat and connected to HP-UX Apache-based Web Server through mod_jk or mod_jk2
mod_jk / mod_jk2	Tomcat connector to HP-UX Apache-based Web Server that supports ajpv13 protocol
PHP	Server-side scripting language embedded in HTML with a Java / C++ syntax. Supports many databases.
PHP extension to Oracle	PHP scripts can access and modify Oracle databases

PHP Extensions	Extensions add support for Oracle, PostgreSQL, and LDAP connectivity, XML parsing, Image manipulation and more.
Webmin	Web-based GUI Administrator enhanced and customized for HP-UX Apache
Apache extensibility	APIs for writing customer Apache modules, dynamic loading of shared objects (DSOs)
C++ shared library (DSO) support	HP-UX Apache shared libraries (DSOs) can be written in C++
HP and third-party plug-in support or certification	BEA Application Server certification, Ready-to-Run FrontPage Server Extensions support, HP MC ServiceGuard support, PTC certification, Siebel certification
Automatic restart of Apache/Tomcat/Webmin	Configurable automatic startup at reboot
Chroot	Named, alternate directory becomes the root directory
chroot copy utility	Script that copies commonly used HP-UX Apache files and system resources (i.e. system libraries) into the chroot directory
MM	Library which simplifies shared memory usage
certmig utility	Migration tool for iPlanet (iWS 4.x) certificates (not available for IPF releases)
mkcert utility	Generates private keys, certificate signing requests and certificates
CGI daemon	Daemon that forks child processes to run CGI scripts for improved CGI performance.
suEXEC	Ability to run CGI and SSI programs under user IDs different from the user ID of the web server
port utility	Reads and displays all configured ports
altroot utility	Move the components of HP-UX Web Server Suite to different directories
cache utility	Helps improve caching performance.
test certmig utility	Tests importing and exporting of Netscape certificates
Multi-threaded processes	Multiple threads per process for better scalability
Apache API/APR	APIs to built modules that are loaded by Apache at run-time to provide additional capabilities and the Apache Portable Runtime (APR), a library forming a system portability layer to many operating systems.
Filters	Output of a module can be processed by another module
WebDAV Web Publishing	Web-based Distributed Authoring and Versioning (WebDAV). An IETF standard extension to HTTP to add, modify, move and delete server files locally.

For more details of these features, check Appendix A Summary of Web Server Functionality Differences

2.2 Getting More Information

The following tables list resources for the HP-UX Web Server Suite. For the latest information, check http://www.hp.com/go/webserver.

Table 1: Resources for HP-UX Web Server Suite

Resources	Location
HP-UX Web Server Suite	
Information and technical tips (latest information)	http://www.hp.com/go/webserver
Electronic Download	http://software.hp.com →Featured Products→HP-UX Apache-based Web Server or http://software.hp.com and search for "HP-UX Web Server Suite"
Bundled documents	/opt/hpws/hp_docs
FAQs	http://www.hp.com/products1/unix/webservers/apache/faqs/index.html
HP-UX Apache-based Web Server Version 1.3.x to HP-UX Web Server Suite Migration Guide	http://www.hp.com/products1/unix/webservers/apache/techtips/index.html
iPlanet to HP-UX Web Server Suite Migration Guide	http://www.hp.com/products1/unix/webservers/apache/techtips/index.html
HP Developer and Solution Partner Portal	http://www.hp.com/dspp

Table 2: Applications Integrated with HP-UX Web Server Suite

Integrated Application	Resource
HP Workload Manager (WLM)	http://www.hp.com/go/wlm
BEA WebLogic Server	http://www.bea.com Supported versions: http://e-docs.bea.com/wls/certifications/index.html
PTC Windchill	http://www.ptc.com Supported versions: http://www.ptc.com/partners/hardware/2000i2/windchill.htm
HP-UX Network Server Accelerator for HTTP (NSAHTTP)	http://software.hp.com Search for 'Network Server Accelerator' or 'NSAHTTP'

Table 3: Open Source Resources

Resources	Location
General	
Apache HTTP Web Server Project information	http://httpd.apache.org/
Apache Software Foundation (ASF)	http://www.apache.org/
Apache 2.0 User's Guide	http://httpd.apache.org/docs-2.0/
Apache developer resources	http://dev.apache.org/
Apache Module Registry	http://modules.apache.org/
Servlets	
Tomcat User's Guide	http://jakarta.apache.org/tomcat

Perl	
Perl Interpreter	http://software.hp.com/ → search for “perl v.5.8.0”
mod_perl	http://perl.apache.org/
SSL	
OpenSSL User's Guide	http://www.openssl.org/docs/
mod_ssl User's Guide	http://www.modssl.org/docs/
PHP	
PHP User's Guide	http://www.php.net/docs.php
PHP general information	http://www.php.net/
Webmin	
Webmin User's Guide	http://www.swelltech.com/support/webminguide/server-apache.html
Webmin Home Page	http://www.webmin.com/webmin/
LDAP authentication	
OpenLDAP	http://www.openldap.org/
auth_ldap NOTE: obsolete after June 04. Replaced by mod_auth_ldap from Apache open source	http://www.rudedog.org/auth_ldap/

For Apache and open source news, see
http://www.apacheweek.com/
http://www.oreillynet.com/apache/
http://www.serverwatch.com/stypes/servers/
http://slashdot.org/index.pl?section=apache

3 HP-UX Web Server Suite Requirements

The target web server machine must have the appropriate level of hardware and software for migration. The HP-UX Web Server Suite can be installed on the same system that currently hosts the NES or iWS web server or it can be installed on another machine.

If the target web server machine is currently running HP-UX 10.20, the operating system must be upgraded to HP-UX 11.x. HP-UX Web Server Suite runs on HP-UX 11.0, 11i, 11i Version 1.5 and 11i Version 1.6. If the target web server machine cannot run HP-UX 11.x, please contact HP for help and information on planning and performing a hardware upgrade. The HP documentation site, http://docs.hp.com/hpux/os/11.0/, has information on installing HP-UX.

The following hardware and software is necessary for installing and running the HP-UX Web Server Suite product. See the Installing Required Software section below for more information.

Table 4: HP-UX Web Server Suite Hardware and Software Requirements

HP-UX Apache-based Web Server				
HP-UX Web Server Suite Products	HP-UX Platform	Disk Space	mod_perl	apxs
HP-UX Apache-based Web Server (hpuxwsApache)	HP-UX 11.0, 11i (PA-RISC)	80-90 MB	Perl v.5.8.0	Perl 5x or higher
HP-UX Apache-based Web Server (hpuxwsApache)	HP-UX 11i Version 1.6 (11.22), and 11i Version 2 (11.23) **for 64-bit**	80-90 MB	64-bit Perl for IPF v.5.8.0	Perl 5x or higher
HP-UX Apache-based Web Server (hpuxwsApch32)	HP-UX 11i Version 2 (11.23) **for 32-bit**	80-90 MB	32-bit Perl for IPF v.5.8.0	Perl 5x or higher
HP-UX Apache-based Web Server **for IPv6** (hpuxwsApache)	HP-UX 11i (PA-RISC) IPv6 product # T1306AA	80-90 MB	Perl v.5.8.0	Perl 5x or higher

HP-UX Tomcat-based Servlet Engine			
HP-UX Web Server Suite Products	HP-UX Platform	Disk Space	Java Servlets and JSPs
HP-UX Tomcat-based Servlet Engine	HP-UX 11.0, 11i (PA-RISC) HP-UX 11i Version 1.6, 11i Version 2 (IPF)	~ 15 MB	HP JDK 1.2.2.04 or higher (JDK 1.3 or higher recommended)

HP-UX Webmin-based Admin			
HP-UX Web Server Suite Products	HP-UX Platform	Disk Space	Webmin
HP-UX Webmin-based Admin	HP-UX 11.0, 11i (PA-RISC) HP-UX 11i Version 1.6, 11i Version 2 (IPF)	~20 MB	Perl 5 or greater

HP-UX XML Web Server Tools			
HP-UX Web Server Suite Products	HP-UX Platform	Disk Space	Java
HP-UX XML Web Server Tools	HP-UX 11.0, 11i (PA-RISC) HP-UX 11i Version 1.6, 11i Version 2 (IPF)	~100 MB	HP JDK 1.3.0 for the appropriate platform

4 Installing or Upgrading HP-UX

The version of HP-UX on your web server machine may need to be updated. If the machine is currently running HP-UX 10.20 then you must upgrade to HP-UX 11.x. Check the HP-UX Web Server Suite Requirements table to determine which HP-UX platform you need. When upgrading, you will need to perform the following steps.

1. Stop your NES or iWS web server
2. Stop the Administration Server
3. Back up your system including the entire NES or iWS environment from the server root (directory structure, files, etc).
4. Install HP-UX 11.x
5. Install other hardware and software as specified in the table above, HP-UX Web Server Suite Requirements
6. Reload your files including your NES or iWS environment

5 Installing Required Software

5.1 Perl

Perl is needed for HP-UX Webmin-based Admin, mod_perl, CGI scripts written in Perl, and Perl utilities bundled with Apache (apxs).

Perl scripts and Utilities

- **HP-UX Webmin-based Admin** is a web-based GUI administrator used for managing HP-UX Apache-based Web Server. It is dependent on Perl 5 or higher and expects Perl to be installed in `/opt/perl/bin/perl`, however **HP-UX Webmin-based Admin** can be re-configured to look in any location. The Admin Guide bundled with the product, describe how to configure Webmin.

- **apxs** is a Perl script for compiling, linking, and configuring Apache modules. **apxs** expects Perl to reside in `/opt/perl/bin/perl`. The Perl path in **apxs** can be changed if Perl is installed in a different location on your machine. **apxs** on PA-RISC machines uses 32-bit Perl and on IPF machines **apxs** can use either 32-bit Perl or 64-bit Perl. More information about apxs is available at http://httpd.apache.org/docs-2.0/programs/apxs.html.

- **Perl CGI scripts** use 32-bit Perl on PA-RISC. On IPF machines, Perl CGI can use either 32-bit Perl or 64-bit Perl.

mod_perl

The **mod_perl** module is an add-on module from open source that is compiled into HP-UX Apache-based Web Server. It is not configured by default. To use **mod_perl** you must explicitly configure it. The **mod_perl** module makes CGI written in Perl run faster and it also allows Apache modules to be written entirely in Perl.

mod_perl modules are dependent on the Perl version and architecture (32 or 64-bit). For PA-RISC machines, **mod_perl** uses 32-bit Perl. For IPF systems, depending on which versions of HP-UX Apache-based Web Server is installed, **mod_perl** will require the corresponding 32-bit or 64-bit Perl. Check the release notes bundled with the product for more information.

The HP-UX Apache-based Web Server admin guide, bundled with the product, describe how to configure **mod_perl.**

5.2 Java Development Kit (JDK)

HP-UX Web Server Suite implements Java servlets and JSPs using Tomcat. Tomcat requires installation of the Java Virtual Machine through the HP-UX Java Developer's Kit.

The latest versions of the JDK can be downloaded from:
http://www.hp.com/go/java
select "`SDK and RTE 1.3`" or "`SDK and RTE 1.4`" (PA-RISC or IPF)

5.3 gcc

gcc, the GNU Open Source compiler, is needed for building Apache modules in C and C++. To download, go to the "Develop & Solution Partner Portal" at http:/www.hp.com/dspp/ and search for **gcc**.

5.4 IPv6

HP-UX Apache-based Web Server with IPv6 support is available in a separate product, which can be selected when downloading from http://software.hp.com and is only available for two versions of HP-UX.

- HP-UX 11i PA-RISC (11.11), this product requires the HP-UX 11i IPv6 product, which is bundled in Transport Optional Upgrade Release (TOUR) product. To download the TOUR product, go to http://software.hp.com and search for the TOUR product.
- For HP-UX 11i Version 2 IPF (11.23), the IPv6 product and the IPv6 related dependencies are already included.

HP-UX Apache-based Web Server may or may not work the same as in the IPv4 HP-UX Apache-based Web Server product. For example, the following components work on IPv6 but may not behave correctly for networking calls related to IPv6, due to the lack of underlying support: Perl-based CGI, mod_perl, Webmin, LDAP connectivity, Stunnel, OpenSSL, ab with SSL, and SSL clients.

For detailed information about setting up and using HP-UX Apache-based Web Server with IPv6, refer to the bundled Apache admin guide, `/opt/hpws/hp_docs/apache/apache.admin.guide.`

For more information on IPv6, see http://www.ipv6.org/

5.5 Patches

These patches are recommended for running the HP-UX Web Server Suite. Please install these patches (or supersedes) and their dependent patches. Patches can be downloaded from HP's IT Resource Center at, http://us-support.external.hp.com

1. Dynamic Shared Libraries (DSOs)

   Check the version of **ld** on your system to determine if it needs upgrading. It should be at least version B.11.37. Without this patch you may see unresolved external errors such as, "/usr/lib/dld.sl: Unresolved symbol : dlsym (code) from mod_jk.so"

   Install either PHSS_28869 for HP-UX 11.00 or PHSS_28871 for HP-UX 11i (11.11) or later to solve this problem.

   To verify the version of ld:

   $ ld –V
   ```
   92453-07 linker command s800.sgs ld PA64 B.11.37 REL 030526
   ld: 92453-07 linker linker ld B.11.37 030909
   ld: Usage:  ld [options] [flags] files
   ```

   $ what /usr/lib/libdld.sl
   ```
   /usr/lib/libdld.sl:
           92453-07 dld dld libdld.sl B.11.37 030909
   ```

2. IPv6 on HP-UX 11i (PA-RISC), and 11i Version 2 (IPF)

   See the previous section about IPv6 for details on dependencies.

NOTE: Swlist can be used to check which patches are already installed on your system. For example,
$ swlist | egrep "PHNE_|PHSS_"

sam can be used for installing patches.

6 Installing the HP-UX Web Server Suite

The HP-UX Web Server Suite product is distributed as part of the HP-UX 11i Operating Environment (OE) and on Application Release (AR) CDs. The latest versions are available for electronic download from HP Software Depot. We strongly recommend using the online versions as listed in the requirements table. The online versions are updated frequently and often contain more features and functions than HP-UX Web Server Suite versions that are part of the OE or that are distributed on CD.

HP-UX Web Server Suite is available from these sources:

- HP Software Depot, http://software.hp.com→Featured Products→HP-UX Apache-based Web Server
  or go to http://software.hp.com and search for "HP-UX Web Server Suite".
- HP-UX 11.0/11i Application Release CDs
- HP-UX 11i OE CDs

If you are receiving the HP-UX Web Server Suite as part of the HP-UX 11.x Operating Environment, the software is installed automatically as part of the OE bundle. (Note: If you are installing Apache as part of the OE update and there is a non-HP version of Apache already on the system in `/opt/hpws/`, the result is undefined and HP-UX Web Server Suite may not install.)

HP-UX Apache-based Web Server *no longer* starts up automatically after installation. HP-UX Apache-based Web Server uses port 80 and port 443 (SSL) by default. During migration, these ports can be changed to other port numbers (i.e. 8080 and 8443) to not conflict with those used by NES or iWS. Make sure to change the port to their permanent numbers when migration is complete.

It is recommended to remove any previous Apache installations (HP or non-HP) rather than installing over existing files. Although Apache from HP may be installed in /opt/apache, /opt/tomcat or /opt/hpapache2, this document only addresses the migration to HP-UX Web Server Suite, which installs into /opt/hpws/.

To remove any existing HP-UX Apache-based Web Server or HP-UX Web Server Suite from your system,

1. Use **swremove** to remove HP-UX Apache-based Web Server and completely remove it. Use **swlist** to determine which product was installed.
   $ swlist | grep -i -e apache -e tomcat -e webmin -e xml

   Based on the output, **swremove** each product.
   $ swremove <product_name>

2. Remove or move the files in `/opt/apache` and `/opt/tomcat` or `/opt/hpapache2`. Make sure to save any data you may want to keep,
   $ rm –rf /opt/apache
   or
   $ mv /opt/apache /opt/apache.xxxx

To install HP-UX Web Server Suite, use **swinstall**. The following command invokes a user interface that leads you through the installation. Each component installs under `/opt/hpws/`.
$ **/usr/sbin/swinstall &**

NOTE:
Since the installation path of HP-UX Web Server Suite is different than older versions of HP Apache-based Web Server 1.3.x, installing on top will not modify or replace any configuration files. If HP-UX Web Server Suite is being re-installed, **swinstall** retains any *modified* configuration files under the appropriate configuration directories, and replaces configuration files that have not been modified since they were installed. All new files (regardless of if they were installed or not) will be delivered in the newconfig directory under the base (e.g., `/opt/hpws/apache/newconfig/<absolute-path-to-file>`).

7 Tools and Utilities

7.1 HP Tools

These are HP tools provided with the HP-UX Web Server Suite product that cover a variety of functions. These tools are available once the suite is installed. For usage details on all HP tools, consult the utilities guide at `/opt/hpws/hp_docs/utilities.user.guide` and `/opt/hpws/hp_docs/apache/utilities.user.guide`.

7.1.1 Certmig

Certmig is a utility to migrate iPlanet 4.x certificates to HP-UX Apache-based Web Server. It is an extension of the PK12UTIL utility provided by the Mozilla community. Certmig uses Network Security Services (NSS) libraries for converting iPlanet certificates, key translations and certificate chains to those of the Apache web server. Certmig does not migrate Netscape 3.x certificates.

certmig is run using the wrapper `/opt/hpws/apache/util/test_certmig.sh`**. certmig** is installed in `/opt/hpws/apache/bin/certmig`, with documentation in `/opt/hpws/hp_docs/apache/certmig.user.guide`.

NOTE: **certmig** is not available on IPF machines since iPlanet does not run on IPF.

7.1.2 test_certmig.sh

The **test_certmig.sh** utility is a wrapper around **certmig** and is included in HP-UX Apache-based Web Server. It can be used to import, extract and list the certificates in an iPlanet 4.1.x Certificate database.

7.1.3 mkcert.sh

The **mkcert.sh** utility generates private keys, certificate signing requests and certificates.

7.1.4 Ports.sh

Ports.sh is a port list utility bundled with HP-UX Apache-based Web Server. It lists the ports being configured by the HP-UX Web Server Suite including Apache, Apache(SSL), Tomcat, mod_jk, Webmin, and LDAP. **ports.sh** resides in `/opt/hpws/util/ports.sh`.

7.1.5 Cache_util.pl

cache_util.pl is a file caching utility bundled with HP-UX Apache-based Web Server. It helps optimizes file caching by reviewing the most commonly accessed files in logs/access_log and creating a caching file list.
cache_util.pl resides in `/opt/hpws/apache/util/cache_util.pl`.

7.1.6 Altroot.sh

altroot.sh allows an administrator to relocate the entire HP-UX Web Server Suite or its individual components to a location other than the default location of /opt/hpws.
altroot.sh resides in `/opt/hpws/apache/util/altroot.sh`.

7.1.7 Chroot_os_cp.sh

chroot_os_cp.sh is a helper script that sets the stage for chroot by copying typical files needed by HP-UX Apache-based Web Server, into the specified chroot directory.
chroot_os_cp.sh resides in `/opt/hpws/apache/util/chroot_os_cp.sh`.

7.2 iPlanet Tools

7.2.1 migrateLocalDB

The utility **migrateLocalDB** can be used to migrate NES users and groups from a 3.X localdb to ldif format. The ldif can then be used to add entries into an LDAP directory server for use by HP-UX Apache-based Web Server.

```
/opt/iplanet/bin/https/admin/bin/migrateLocalDB
```

7.2.2 htconvert

iPlanet includes the **htconvert** script for converting existing .nsconfig files to .htaccess files. Apache uses .htaccess files for per-directory configuration.

htconvert is found under the plugins directory, for example,
```
/opt/iplanet/plugins/htaccess/htconvert
/opt/ns-enterprise36/plugins/htaccess/bin/htconvert
```

8 Performing the Migration

This section contains steps for performing the migration. After setting up the web server environment, execute the steps that apply to the functionality to be migrated. For example, migrate the chroot environment by executing the chroot step, migrate an `.htaccess` file by executing the .htaccess step, etc. You may not need to execute every step if you are not currently using the functionality and/or not interested in configuring it on HP-UX Web Server Suite. All steps in the Migrating the Core should be performed.

Each step includes verification to test your changes.

Most steps can be performed from either the HP-UX command line or through the HP-UX Webmin-based Admin. For example, an `.htaccess` file can be created and enabled either from the command line or from Webmin. To use Webmin, refer to the Webmin User's Guide, http://www.swelltech.com/support/webminguide/server-apache.html. The steps here show how to use the command line.

Steps are only shown for migrating to the HP-UX Web Server Suite. Previous releases of Apache by HP on HP-UX (HP Apache-based Web Server) will not be described. For complete information on the differences between HP Apache-based Web Server 1.3.x and HP-UX Web Server Suite, refer to the Migration Guide HP Apache-based Web Server Version 1.3.x to HP-UX Web Server Suite, http://www.hp.com/go/webservers→hp-ux web server suite→Technical Tips.

Table 5 Major Directories in the HP-UX Web Server Suite

HP-UX Apache-based Web Server	/opt/hpws/apache/
HP-UX Tomcat-based Servlet Engine	/opt/hpws/tomcat/
HP-UX Webmin-based Admin	/opt/hpws/webmin/
HP-UX XML Web Server tools	/opt/hpws/xmltools/
Documentation	/opt/hpws/hp_docs/
Utilities	/opt/hpws/util/
Licenses	/opt/hpws/LICENSES

8.1 HP-UX Apache-based Web Server Configuration File Overview

`httpd.conf` is HP-UX Apache-based Web Server's main configuration file. It resides in the `/opt/hpws/apache/conf` directory along with other HP-UX Apache-based Web Server configuration files.

The following table describes the important configuration files used by HP-UX Apache-based Web Server. Most additional configuration files are specified in the `httpd.conf` using the "`Include <config-file>`" directive. For example:

```
Include /opt/hpws/apache/conf/cache.conf
```

Apache server	/opt/hpws/apache/conf
Main configuration file	`httpd.conf`
mod_ssl	`ssl.conf`
mod_auth_ldap, mod_ldap, auth_ldap (deprecated June 2004)	`ldap.conf` `stunnel.conf` (for use with SSL)
mod_file_cache	`cache.conf`
mod_mime	`mime.types`
mod_mime_magic	`magic`
mod_jk / mod_jk2 for Tomcat	`mod_jk.conf` `or mod_jk2.conf`

Some helpful hints from: http://httpd.apache.org/docs-2.0/configuring.html

- Apache only recognizes changes to the main configuration files when it is started or restarted.

- Apache configuration files contain one directive per line.
- The back-slash "\" may be used as the last character on a line to indicate that the directive continues onto the next line. There must be no other characters or white space between the back-slash and the end of the line.
- Directives in the configuration files are case-insensitive, but arguments to directives are often case sensitive.
- Lines which begin with the hash character "#" are considered comments, and are ignored. Comments may not be included on a line after a configuration directive.
- Blank lines and white space occurring before a directive are ignored, so you may indent directives for clarity.
- You can check your configuration files for syntax errors without starting the server by using one of the following commands:
  $ apachectl configtest
  $ httpd -t
- All relative paths will be assumed to be relative to the `ServerRoot`.
- To include additional configuration files, use the "`Include <config-file>`" directive as described above.
- For directives multiply defined within the same `<Context> </Context>`, the last directive is used. (See below for more information on Contexts.)

The following is a sample of a directive in `httpd.conf`. There are often helpful explanations or warnings before the directive.

```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation (available
# at <URL:http://httpd.apache.org/docs-2.0/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/opt/hpws/apache"
```

Directives can be included within context tags. Contexts are contained within angular brackets "< >" and are of the format of "`<Context args> … </Context>`". See Security 5: Access Control/Access Control Lists (ACLs) for examples on using these. Contexts can be one of the following:

```
<Directory directory-path>
<DirectoryMatch regex>
<Files pattern-match>
<IfDefine parameter-name>
<IfModule mod_xxx.c>
<Limit limit-method>
<LimitExcept limit-method>
<Location URL-path>
<LocationMatch regex>
<VirtualHost address>
```

Directives that are not within a context apply to the default or global server. The following is a sample of the `<Files>` context from `httpd.conf`.

```
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
```

```
#
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

Directives defined within a context override any previous directives that were specified. Some contexts can be nested within each other.

```
#
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS PROPFIND>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

Apache can be configured by directly editing the configuration files, or by using **Webmin**. Information on using **Webmin** is in Management 2: Server Administration.

8.2 Preparation Checkpoint

Verify that the machine is ready for migration by making sure the following steps have been completed.

1. NES/iWS and Administration server is stopped.

2. All necessary hardware and software system requirements have been completed, including installation of HP-UX Web Server Suite, using the HP-UX Web Server Suite Requirements.

3. If desired, the HP-UX kernel parameters are tuned for HP-UX Web Server Suite.

4. Start HP-UX Apache-based Web Server and do a quick verification of the installation:
   a. **$ /opt/hpws/apache/bin/apachectl startssl**
   b. Check the error log**,**
      $ tail /opt/hpws/apache/logs/error_log
   c. Access the index.html page, http://yourserver.com
   d. Execute the test CGI script, http://yourserver.com/cgi-bin/test.cgi
   e. Stop the web server,
      $ **/opt/hpws/apache/bin/apachectl stop**

Look at the release documents that are bundled with the HP-UX Web Server Suite product to become familiar with the features. The bundled documentation includes user guides and configuration information:

```
/opt/hpws/hp_docs
```

8.3 Migrating the Core

See Appendix 1 : Core for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Core 1: Process Handling

Definition:

Process handling describes the process model of how a web server handles incoming requests.

NES/iWS:

Preferences → Performance Tuning
Preferences → Thread Pools

HP-UX Apache-based Web Server:

Apache 2.x implements a hybrid multi-process multi-threaded model called "worker" in which each process has a fixed number of threads. The server adjusts to handle loads by increasing or decreasing the number of processes. Each thread handles a request. More information is available at http://httpd.apache.org/docs-2.0/mod/worker.html.

The following table shows the default process and thread directives in HP-UX Apache-based Web Server.

Table 7: HP-UX Apache-based Web Server 2.x Process Directives in /opt/hpws/apache/conf/httpd.conf

Process Directive	Default Value	Description
`StartServers` `ThreadsPerChild`	2 25	`StartServers` is the number of processes to create at start-up time. However, the number of requests that can be handled is dependent on the `ThreadsPerChild` directive. Hence, in this example, the number simultaneous server requests are 2*25 = 50.
`MaxClients`	8	`MaxClients` refers to the maximum number of processes running simultaneously. In conjunction with the `ThreadsPerChild` directive, it translates to maximum number of concurrent requests served, in this case 8*25=200 requests.
`MinSpareThreads`	25	`MinSpareThreads` sets the desired minimum number of idle threads at any time. It serves as a trigger point for Apache to automatically create new threads when the number of idle threads falls below this value. Spawning of each new process directly translates into 25 (`ThreadsPerChild`) new threads.
`MaxSpareThreads`	75	`MaxSpareThreads` sets the desired maximum number of idle threads at any time. It serves as a trigger point for Apache to automatically kill spare idle threads, when the number of idle threads goes above this value. Killing of each process directly translates into `ThreadsPerChild` killed threads.
`MaxRequestsPerChild`	0	`MaxRequestsPerChild` sets the number of requests a thread will handle before it is killed (aged-out). A value of 0 means the thread never expires.

Verification:

Run "`/opt/perf/bin/gpm &`" to start Glance-Plus. Use this tool to verify the number of httpd processes and threads:

1. **$ /opt/perf/bin/gpm &**
2. Select Reports→ Select Process List→Select Configure→Select Filters, turn on filtering and filter on "httpd"
3. Select (highlight) an httpd process→Select Reports→Select Process Thread List

Core 2: Logging

Definition:

Logging is a way to record information about a client request and about the web server's response. The main kinds of logs are access logs (requests) and error logs (responses).

NES/iPlanet:

Status → Log Preferences

HP-UX Apache-based Web Server:

Apache has a built-in definition for the Common Logfile Format (CLF) standard format. It is used for Apache's `access_log` and `error_log`. New logs can be defined using Apache's `LogFormat` and `CustomLog` directives to log items such as cookies, URL referring server, etc.

In `httpd.conf:`

```
ErrorLog /opt/hpws/apache/logs/error_log
LogLevel warn
CustomLog /opt/hpws/apache/logs/access_log common
```

`ErrorLog` specifies where Apache's `error_log` is located. `LogLevel` specifies the verbosity of the messages in `error_log` (none, error, warn, info, trace, debug). `CustomLog` defines the access_log and specifies it in CLF format. However, `access_log` is not enabled by default.

The Apache **logresolve** utility will resolve IP addresses into server names in `access_log` and provide statistics for number of web server accesses,
http://httpd.apache.org/docs/programs/logresolve.html

To run **logresolve**,
$ cd /opt/hpws/apache/logs
$ opt/hpws/apache/bin/logresolve -s alog.stats -c < access_log > alog.resolved

For information on the **logresolve** utility,
http://httpd.apache.org/docs-2.0/programs/logresolve.html

For more information on logging, http://httpd.apache.org/docs-2.0/logs.html

Verification:

1. Create an error in error_log by requesting a non-existent file or URL,
   http://yourserver.com/foo.html

2. Display the entries in the `error_log` and `access_log`,
   $ tail /opt/hpws/apache/logs/error_log
   $ tail /opt/hpws/apache/logs/access_log

Core 3: IP Addresses and Port Numbers

Definition:

Network location of the web server and network ports where it listens for requests

NES/iWS:

Preferences → Network Settings

HP-UX Apache-based Web Server:

The default configuration is to listen to all IP addresses on ports 80 and 443 (SSL). These can be changed by using the `Listen` directive.

`Listen` tells the server to accept incoming requests on the specified port or address-and-port combination. If a port number only is specified only, the Apache listens to the given port on all interfaces (default). If an IP address only is given, then Apache listens on all ports for the specified IP address. Multiple `Listen` directives may be used to specify a number of addresses and ports to listen to. The server will respond to requests from any of the listed addresses and ports.

Modify the `httpd.conf` and the `ssl.conf` files to assign ports.

IP Addresses and Ports	HP-UX Apache-based Web Server
Binding to port 80 on all IP addresses	`Listen *:80`
Binding to port 80 on 111.222.333.444	`Listen 111.222.333.444:80`
Binding to ports 80 and 8080 on all IP addresses	`Listen 80` `Listen 8080`
Binding to a domain name	`Listen` uses IP addresses only.
Using extended addresses, IPv6	`Listen [fe80::1]:80`
port 80 on all IP addresses	`Listen *:80 or Listen 80`
port 80 on 111.222.333.444	`Listen 111.222.333.444:80`
ports 80 and 8080 on all IP addresses	`Listen 80` `Listen 8080`
port 80 on 111.222.333.444 and port 8080 on 555.666.777.888	`Listen 192.170.2.1:80` `Listen 192.170.2.5:8000`
Specifying a domain name	`Listen` uses IP addresses only.
Extended addresses, IPv6	`Listen [fe80::1]:80`

Notes: `<VirtualHost>` can be used to specify a different behavior for one or more of the addresses and ports. See the Enterprise 3: Virtual Servers step for setting up Virtual Hosts. The `ServerName` directive has an optional port number argument that is used in redirection URLs. See the Core 4: ServerName step for specifying a server name.

For more information see the Apache documentation,
http://httpd.apache.org/docs-2.0/bind.html
http://httpd.apache.org/docs-2.0/vhosts/
http://httpd.apache.org/docs-2.0/dns-caveats.html

Verification:

Try various URLs with ports numbers you have configured, for example,
http://yourserver.com
https://yourserver.com (SSL)

You can also run the ports utility to check the configured ports, **/opt/hpws/util/ports.sh**. More information on this utility can be found in `/opt/hpws/hp_docs/utilities.user.guide`.

Core 4: ServerName

Definition:

Sets the hostname of the server.

NES/iWS:

Preferences → Network Settings

HP-UX Apache-based Web Server:

ServerName <*fully-qualified-domain-name:port*>

`ServerName` is used when creating redirection URLs. For example, if the name of the machine hosting the Apache is `simple.example.com`, but the machine also has the DNS alias `www.example.com` and you want the web server to be identified by its alias, use:

```
ServerName www.example.com:80
```

NOTE: The port number parameter to `ServerName` is required

For more information,
http://httpd.apache.org/docs-2.0/mod/core.html#servername

Verification:

Use the Perl printenv script to display environment variables (Perl must be installed),
http://yourserver.com/cgi-bin/printenv

Look at the setting for the server name,
`SERVER_NAME="yourserver.com"`

Core 5: DNS Lookups

Definition:

Web server access to a name server for a request to resolve a client's IP address into a server name. This is often done for logging.

NES/iWS:

Preferences → Performance Tuning

HP-UX Apache-based Web Server:

Apache enables or disables DNS lookups using the `HostnameLookups` directive. It is disabled (`Off`) by default for better performance. See the Logging step for information on a bundled Apache log analyzer that can do IP to server name resolutions at log analysis time.

To turn off or on DNS lookups, edit httpd.conf:

```
HostnameLookups Off
```

Verification:

Check `the access_log` to see is IP addresses or server names are being logged,
$ tail /opt/hpws/apache/logs/access_log

Core 6: Document Root

Definition:

Location where web server documents are stored. There is at least a primary root and there may be additional locations (roots) where files are stored.

NES/iWS:

Content Management → Primary Document Directory

NES has the default primary document directory of **/opt/ns-enterprise36/docs-https-default**
iWS has the default primary document directory of **/opt/iplanet/docs**

HP-UX Apache-based Web Server:

Apache's primary document directory is the `DocumentRoot`. Additional document directories can be configured using the `Alias` directive or symbolic links.

HP-UX Apache-based Web Server has the default document directory of **/opt/hpws/apache/htdocs**

The primary document directory can be migrated in several different ways:

- Use Apache's default DocumentRoot and copy NES/iWS documents to that location,
  $ cp -R /opt/iplanet/docs/* /opt/hpws/apache/htdocs/
  $ chown -R www:other /opt/hpws/apache/htdocs/

- Change Apache's default DocumentRoot to access the NES/iWS document root. For example, in `httpd.conf`,

```
DocumentRoot "/opt/iplanet/docs"
<Directory "/opt/iplanet/docs">
   Options FollowSymLinks Indexes MultiViews
</Directory>
```

- Use Apache's default `DocumentRoot and` point to the NES/iWS documents using symbolic links,

  a. Enable symbolic links in the document directory,

```
DocumentRoot "/opt/hpws/apache/htdocs"
<Directory "/opt/hpws/apache/htdocs">
      Options FollowSymLinks Indexes MultiViews
</Directory>
```

  b. Create the symbolic link to the actual document root,
  $ cd /opt/hpws/apache
  $ ln -s /opt/iplanet/docs htdocs

- Use Apache's default `DocumentRoot and` point to the NES/iWS documents using `Alias` directives

  Set up `Alias` in the document directory,

```
DocumentRoot "/opt/hpws/apache/htdocs"
Alias /opt/iplanet/docs /opt/hpws/apache/htdocs
```

Verification:

1. Use the Perl printenv script to display environment variables to view the `DocumentRoot`,
   http://yourserver.com/cgi-bin/printenv

   `DOCUMENT_ROOT="/opt/hpws/apache/htdocs"`

2. Access the server home page
   http://yourserver.com

Core 7: Additional Document Directories

Definition:
Location of additional web server files

NES/iWS:
Content Management → Additional Document Directory

HP-UX Apache-based Web Server:
Any number of additional document directories can be configured into Apache by using the `Alias` directive.

1. Make sure the web server has at least read access to the documents,
   $ chown -R www:other /opt/iplanet/docdirectory
   $ chmod 755 /opt/iplanet/docdirectory

2. Add the Alias directive to `httpd.conf`. Since it is dependent on the mod_alias module, it can (but does not have to) be within the <IfModule></IfModule> tags.
```
<IfModule mod_alias.c>
   Alias /documents /opt/iplanet/docdirectory
</IfModule>
```

Verification:
Access a document in the aliased directory,
http://yourserver.com/documents/mydoc

Core 8: Directory Indexing

Definition:
Behavior of the web server when provided with a URL that resolves to a directory.

NES/iWS:
Content Management → Document Preferences

HP-UX Apache-based Web Server:
When Apache is given a URL that resolves to a directory, it can return a default file in the directory, generate an HTML page of the contents of the directory, or return an error message. Directory contents can also be configured to display graphics, file modification dates, and file sizes.

- To return a default file (returns first one found in the list):
```
<Location mydirectory/>
   DirectoryIndex index.html index.htm index.shtml home.html home.htm index.cgi
</Location>
```

- To generate an HTML page of directory contents:
```
<Location mydirectory/>
      Options +Indexes
</Location>
```

- To return a “Permission Denied” error:
  1. Create a CGI script to generate an error message,
     cat fake404.cgi

```
#!/bin/sh
#
#fake404.cgi
```

```
echo "Content-Type: text/html"
echo "Status: 404 Not Found"
echo ""
echo "<HTML><HEAD>"
echo "<TITLE>Permission Denied</TITLE>"
echo "</HEAD><BODY>"
echo "<H1>Permission Denied</H1>"
echo "<P>Directory access not allowed</P>"
echo "</BODY></HTML>"
```

2. Execute the CGI script when for a directory URL,

```
<Location /mydirectory/>
   DirectoryIndex index.html  /cgi-bin/fake404.cgi
</Location>
```

Verification:

Access a directory and verify that a list of files is returned,
http://yourserver.com/

Core 9: User Directories

Definition: User's home directory used for storing personal documents.

NES/iWS:

Content Management → User Document Directories

HP-UX Apache-based Web Server:

Create the user's document directory, such as `public_html`, in the user's home directory as specified in `/etc/passwd`. User document directories are set using the `UserDir` directive. This directive sets the real directory to a user's home directory when a request for user document is received.

For more information about user directories see, http://httpd.apache.org/docs-2.0/mod/mod_userdir.html.

1. Set up the user directory in `httpd.conf`,

```
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>
```

2. Set up access control for this directory

```
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
#<Directory /home/*/public_html>
#    AllowOverride FileInfo AuthConfig Limit
#    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
#    <Limit GET POST OPTIONS PROPFIND>
#        Order allow,deny
#        Allow from all
#    </Limit>
#    <LimitExcept GET POST OPTIONS PROPFIND>
#        Order deny,allow
```

```
#        Deny from all
#    </LimitExcept>
#</Directory>
```

Verification:

1. Create a file in a user's home directory,
   $ vi /home/user/public_html/file.html

2. Access the file using a URL such as the following,
   http://yourserver.com/~user/file.html

Core 10: MIME Types

Definition:

Standard for describing the content of a message

NES/iWS:

Preferences → MIME Types

HP-UX Apache-based Web Server:

1. Apache comes with the file `conf/mime.types` that includes pre-configured, standard MIME types. More types can be added to this file. Entries in `mime.types` look the following. A file with the .xml extension (i.e. foo.xml) will be treated as an xml document.

Content-type	File Extension
`text/css`	`css`
`text/html`	`html htm`
`text/plain`	`asc txt`
`text/vnd.wap.wml`	`wml`
`text/vnd.wap.wmlscript`	`wmls`
`text/xml`	`xml`

2. New MIME types can also be added to httpd.conf using the AddType directive. For example, here the .tgz file extension is treated as a tar file,

   ```
   AddType application/x-tar .tgz
   ```

Verification:

1. Create a new MIME type and extension that is displayed as plain text by the browser,
   `AddType text/plain .note`

2. Access a file with this extension and verify that it displays correctly in the browser
   http://yourserver.com/myfile.note

8.4 Migrating Security

See Appendix 2 : Security for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

For more information about Apache's authentication, authorization, and access control,
http://httpd.apache.org/docs/howto/auth.html

Security 1: SSL/TLS

Definition:

Encrypted communication between the client and the web server.

NES/iWS:

Settings are available directly through magnus.conf or via the administration server:

NES : **Preferences → Encryption On/Off**
Preferences → Encryption Preferences
Admin Server → Keys and Certificates
iWS : **Preferences → Encryption On/Off**
Preferences → Encryption Preferences
Security

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Servercomes pre-configured for SSL. SSL is configured in its own configuration file, `conf/ssl.conf`, by a set of directives that begin with SSL (SSLEngine, SSLPassPhraseDialog, SSLSessionCache, SSLSessionCacheTimeout, SSLRandomSeed, SSLMutex , SSLLog , SSLLogLevel, SSLCipherSuite, SSLCertificateFile, SSLCertificateKeyFile, etc.)

Verification:

1. Start SSL using the "startssl" option in the Apache startup script,
   $ bin/apachectl startssl

2. Use https to specify a URL (make sure to use the SSL port number, 443 by default),
   https://yourserver.com:443

NOTE: When in SSL mode, Apache accepts both encrypted (SSL) and non-encrypted (non-SSL) connections.

Security 2: Certificates

Definition:

File that validates the web server's private key on the Internet.

NES/iWS:

NES: **General Admin -> Keys & Certificates**
iWS : **Security**

HP-UX Apache-based Web Server (not for IPF):

HP-UX Apache-based Web Server supports the migration of Netscape 4.x certificates but not Netscape 3.x certificates. **/opt/hpws/apache/util/test_certmig.sh** can be used a wrapper around the **certmig** utility. If desired, **certmig** utility can be used by following the steps in `/opt/hpws/hp_docs/apache/certmig.user.guide`

- For Netscape 3.x certificates:
  a. Use the migration tool provided with iWS to migrate 3.x certificates to 4.x certificates.
  b. Use **test_certmig.sh** to migrate 4.x certificates to Apache certificates as described below.

- For Netscape 4.x certificates, use **/opt/hpws/apache/util/test_certmig.sh**
  a. Verify certificates in the desired Certificate Database,
     $ test_certmig.sh –list

This prompts the user for the directory containing the Certificate and Key Databases, and then lists the “Nick Name” of each Certificate found.

b. Extract a certificate from the list above,
$ test_certmig.sh –extract

This prompts the user for the following [defaults in brackets]

- Netscape Database directory location [/opt/hpws/apache/util]
- SSL Certificates directory location [/opt/hpws/apache/util/ssl.crt]
- Directory Location of the SSL Certificate key file [/opt/hpws/apache/util/ssl.key]
- Import/Export Pass Phrase [hp.com]
- Name of the certificate to be extracted [My Personal Certificate]
- Password or Pin for "NSS Certificate DB"

For more information, see
`/opt/hpws/hp_docs/apache/utilities.user.guide` (test_certmig.sh explained)
`/opt/hpws/hp_docs/apache/certmig.user.guide` (certmig binary explained)

Verification:

After extracting the certificate – key pairs, you should be able to “view” them using **openssl**, **/opt/hpws/apache/bin/openssl**

1. To view the Key (the following commands are one line):
$ openssl rsa –noout –text –in <*server-key*>

2. To view the Certificate:
$ openssl x509 –noout –text –in <*server-certificate*>

Security 3: Chroot

Definition:

Chroot enables a named directory to become the root directory, the starting point for path searches. A malicious user cannot get to the root file system if chroot is configured.

NES/iWS:

This cannot be configured through the administrative server. Settings are available directly through magnus.conf.

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Server bundles a script called **chroot_os_cp.sh** to create a chroot directory and copy some files and system resources (i.e. system libraries) to the chrooted directory. This script may need to be altered if you have a special configuration.

Chroot is not enabled by default. It includes SSL enhancements.

For more information, refer to the `/opt/hpws/hp_docs/apache/utilities.user.guide` as well as comments in **chroot_os_cp.sh** and in `httpd.conf`.

To set up chroot,

1. Stop HP-UX Apache-based Web Server if it is already running.
$ apachectl stop

2. Uncomment the chroot directive in `httpd.conf` and specify the chroot directory (absolute path). The default path is `/var/chroot`.

```
Chroot /var/chroot
```

3. Run **chroot_os_cp.sh** to create the chroot directory. If mkdir fails to create the directory, do it manually and rerun the script. If the specified directory exists, the chroot setup script provides an option for re-creating it or exiting the setup if you don't want to proceed.

   $ /opt/hpws/apache/util/chroot_os_cp.sh
   ```
   /var/chroot exists; remove and recreate (yes/no)?
   ```
   yes
   ```
   Creating /var/chroot . . . . . .
   Typical files needed by chrooted Apache copied to /var/chroot.
   DONE
   ```

   or exit the script instead of creating the directory,

   $ opt/hpws/apache/util/chroot_os_cp.sh
   ```
   /var/chroot exists; remove and recreate (yes/no)?
   ```
   no
   ```
   Canceling operation.
   ```

4. Copy chrooted files from iWS to HP-UX Apache-based Web Server, making sure they are readable by www:other,
   $ cp /<*iWS-chroot dir*>/* /var/chroot/
   $ chown www:other <filelist>
   $ chmod 755 <filelist>

5. Stop and start HP-UX Apache-based Web Server for chroot to take affect.

   NOTE: "**apachectl restart**" fails with chroot since it's outside the chroot. Use "**apachectl start**" when running with chroot.

Verification:

1. Create a new file under chroot's document root, for example,
   $ vi /var/chroot/opt/hpws/apache/htdocs/chroot_test.html
   ```
   <HTML>
   <HEAD>
   <TITLE>Chroot test page</TITLE>
   <HEAD>
   <BODY>
   <H1>Testing if chroot can access this page</H1>
   </BODY>
   </HTML>
   ```

2. Make sure the file is readable by HP-UX Apache-based Web Server,
   $ chown www:other chroot_test.html
   $ chmod 755 chroot_test.html

3. You should be able to read the file
   http://yourserver.com/chroot_test.html

4. Create a new file under HP-UX Apache-based Web Server's document root,
   vi /opt/hpws/apache/htdocs/root_test.html
   ```
   <HTML>
   <HEAD>
   <TITLE>Apache root test page</TITLE>
   ```

```
<HEAD>
<BODY>
<H1>Testing if Apache root can access this page</H1>
</BODY>
</HTML>
```

5. Make sure the file is readable by HP-UX Apache-based Web Server,
   $ chown www:other root_test.html
   $ chmod 755 root_test.html

6. You should not be able to read the file,
   http://yourserver.com/root_test.html

7. Verify other PHP, CGI, etc. scripts located under the `/var/chroot` directory.

Security 4: .htaccess file

Definition:

.htaccess is a file that controls access to web server resources.

NES/iWS:

iWS uses .nsconfig and .htaccess files to control resource access.

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Server uses `.htaccess` to configure access to resources within the directory where the `.htaccess` file resides. The `.htaccess` file can be renamed using the `AccessFileName` directive. Changes made in an .htaccess file take effect immediately without restarting Apache.

1. Convert any `.nsconfig` files to `.htaccess` files using the iWS **htconvert** script and specify the path to `obj.conf`. For example,

   $ /opt/iplanet/plugins/htaccess/htconvert /opt/iplanet/https-<server name>/config/obj.conf

```
Going to convert /opt/iplanet/docs/servlet
Going to convert /opt/iplanet/docs/jsp.092
Going to convert /opt/iplanet/manual/https/servlets/scripts
Going to convert /opt/iplanet/manual/https/servlets/scripts/servlet1
Going to convert /opt/iplanet/manual/https/servlets/scripts/shoes
Going to convert /opt/iplanet/manual/https/servlets/images
Going to convert /opt/iplanet/docs
```

2. Copy the .htaccess files into the directories to protect,

   $ cp .htaccess /opt/hpws/apache/htdocs/mydir/.htaccess

3. Modify `httpd.conf` so that `.htaccess` can be read,

```
<Directory "/opt/hpws/apache/htdocs/mydir">
   AllowOverride AuthConfig
</Directory>
```

4. Create a password file and add the users that are referred to in the .htaccess files. For example, create password file ".htpasswd" with the user "Ron",

   $ /opt/hpws/apache/bin/htpasswd –c /opt/hpws/apache/.htpasswd Ron
   `New password:` *<type password here>*

```
Re-type new password: <retype password here>
Adding password for user Ron
```

Verification:

1. Make sure the required user has an entry in the password file,
   $ cat /opt/hpws/apache/.htpasswd
   `Ron:vtUAT3C3KgEAg`
2. Request a file in the protected directory, for example,
   http://yourserver.com/mydir/myfile

   Enter a valid user ("Ron" in this example) and user password (from the password file) in the dialog box.

Security 5: Access Control/Access Control Lists (ACLs)

Definition:

Access control allows control over which clients can access your server and what they can access. Access control can screen out certain users, groups, or hosts to either allow or deny access to part of your server, and set up authentication so that only valid users and groups can access part of the server.

NES/iWS:

Preferences → Restrict Access

Access control is implemented through Access Control Lists (ACLs). By default, the web server uses one ACL file that contains all of the lists for access to the server in
`<server_root>/httpacl/generated.https-<servername>.acl`

Multiple ACLs can also be created and referenced in `obj.conf`. To check if multiple ACLs have been defined, look in `obj.conf` for `PathCheck fn="check-acl".` For example, the following entry restricts access to directory `/usr/ns-home/docs/test/` using the rules in `testacl`:

```
<Object ppath="/usr/ns-home/docs/test/*">
PathCheck fn="check-acl" acl="testacl"
</Object>
```

HP-UX Apache-based Web Server:

Apache implements access control through a set of directives that can be specified in `httpd.conf` (globally), in `.htaccess` (per directory), or using a combination of both.

Apache includes many directives (`<File>, <Directory>,` etc) that are used in conjunction with Apache's `Allow/Deny/Order` commands to restrict access by directory, URL, IP address, hostname, files, HTTP command, etc.

To restrict access by

1. Directory (<Directory>, <DirectoryMatch>):
   To apply access control to a directory (file system path) and its subdirectories,
   ```
   <Directory /home/www>
         Options Includes ExecCGI FollowSymLinks
         allow from all
   </Directory>
   ```

2. URL (<Location>, <LocationMatch>):
   Only allow clients in your domain to view Apache's status by accessing http://yourserver.com/apache-status,
   ```
   <Location /apache-status>
   ```

```
   SetHandler server-status
   order deny,allow
   deny from all
   allow from .yourdomain.com
</Location>
```

3. File (<Files>, <FilesMatch>):
   To prevent access to `.htaccess` by everyone,
```
<Files .htaccess>
      order deny, allow
      deny from all
</Files>
```

4. HTTP method (<Limit>, <LimitExcept>):
   To allow only POST, PUT, and OPTIONS requests from the localhost,
```
<Limit  POST PUT OPTIONS >
      order deny, allow
      deny from all
      allow from 127.0.0.1
</Limit>
```

5. Hostname:
   To allow access to all hosts except a specific web robot,
```
 <Directory />
      order deny, allow
      allow from all
      deny from robot.trouble.com
</Directory>
```

6. IP address:
   To allow access to a specific subdirectory to only hosts on the internal network,
```
<Directory /internal/>
      order deny, allow
      deny from all
      allow from 127.0.0.1 192.168.1 192.168.2
</Directory>
```

7. Environment variable:
   To lock out browsers that are using HTTP/1.0 or earlier,
   SetEnvIf Request_Protocol ^HTTP/1.1 http_11_ok
```
<Directory /internal/>
      order deny, allow
      deny from all
      allow from env=http_11_ok
</Directory>
```

Verification:

1. Set up access control to a resource and deny your client access (i.e. “deny from 192.168.100”) then verify that you cannot access the resource.

2. Change “deny from” to “allow from” and verify that you can now access the resource.

Security 6: Database Authentication

Definition:

Database authentication uses a database to store valid users, groups, and passwords to verify that someone is who they claim they are.

NES/iWS:

Preferences → Restrict Access

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Server includes basic authentication and authentication using Berkeley database files (DB files).

- Basic Authentication

  Basic authentication is not really database authentication since usernames and passwords are saved in an ascii flat file. When a resource is protected with basic authentication, Apache sends a `401 Authentication Required` header with the response to the request to notify the client that user credentials must be supplied in order for the resource to be returned as requested.

  Upon receiving a `401` response header, the client's browser will ask the user for a username and password to be sent to the server. A dialog box pops up for entering your username and password to send to the server. If the username is on the approved list, and if the password supplied is correct, the resource will be returned to the client.

  To set up basic authentication do the following,

  1. Create a password file using the **htpasswd** utility. The password file (-c to create a new file) is `.htpasswd`,

     $ /opt/hpws/apache/bin/htpasswd -c /opt/hpws/apache/.htpasswd Liza
     `New password: <password>`
     `Re-type new password: <password>`
     `Adding password for user Liza`

     $ /opt/hpws/apache/bin/htpasswd /opt/hpws/apache/.htpasswd Mohan
     `New password: <password>`
     `Re-type new password: <password>`
     `Adding password for user Mohan`

     $ cat /opt/hpws/apache/.htpasswd
     `Liza:KPFReZgxIruYE`
     `Mohan:i32RJ0L4np.2Q`

  2. Set the configuration to use this password file,
     $ chown www:other /opt/hpws/apache/.htpasswd
     $ chmod 640 /opt/hpws/apache/.htpasswd

     Add these directives to `httpd.conf` or put in a `.htaccess` file,
     ```
     AuthType Basic
     AuthName "HP-UX Web Server Suite Team"
     AuthUserFile /opt/hpws/apache/.htpasswd
     Require valid-user
     ```

  3. Optionally, create a group file
     $ vi /opt/hpws/apache/.groups
     hpws_team: Liza Mohan Ron Aida Barbara Julius Madhu Roshan Dave Tair-Shian

  4. Set the configuration to use this group file,

$ chown www:other /opt/hpws/apache/.groups
$ chmod 640 /opt/hpws/apache/.groups

Specify the group file and the required group in `httpd.conf` or `.htaccess`,

```
AuthType Basic
AuthName "HP-UX Web Server Suite Team"
AuthUserFile /opt/hpws/apache/.htpasswd
AuthGroupFile /opt/hpws/apache/.groups
Require hpws_team
```

- User Authentication Using a DBM file

This method of authentication is similar to basic authentication. It has much faster lookups of users (since they are kept in binary form) and should be used instead of basic authentication for a large set of users.

To set up DBM authentication,

1. Create the user file,
   $ /opt/hpws/apache/bindbmmanage /opt/hpws/apache/.dbmpasswd adduser Tair-Shian
   `New password: <password>`
   `Re-type new password: <password>`
   `User Tair-Shian added with password encrypted to fYcWqLRR8mr.2 using crypt`

2. Configure Apache to use that file for authentication.
   $ chown www:other /opt/hpws/apache/.dbmpasswd
   $ chmod 640 /opt/hpws/apache/.dbmpasswd

   Add these directives to `httpd.conf` or put in a `.htaccess` file,
   $ cat .htaccess
   ```
   AuthName “HP-UX Web Server Suite”
   AuthType Basic
   Require user Tair-Shian
   AuthDBMUserFile /opt/hpws/apache/.dbmpasswd
   AuthAuthoritative on
   ```

3. Optionally, create a group file,
   $ dbmmanage add /opt/hpws/apache/.dbmgroups hpws_team Ron Aida Tair-Shian

4. Set the configuration to use this group file,
   $ chown www:other /opt/hpws/apache/.dbmgroups
   $ chmod 640 /opt/hpws/apache/.dbmgroups

   Specify the group file and the required group in `httpd.conf` or `.htaccess`,
   $ cat .htaccess
   ```
   AuthName “HP-UX Web Server Suite”
   AuthType Basic
   AuthDBMUserFile /opt/hpws/apache/.dbmpasswd
   AuthDBMGroupFile /opt/hpws/apache/.dbmgroups
   AuthAuthoritative on
   Require group hpws_team
   ```

More information about basic and auth_dbm authentication is available from http://httpd.apache.org/docs-2.0/howto/auth.html.
Information about dbmmanage and htpasswd is available from,

http://httpd.apache.org/docs-2.0/programs/

Verification:

1. Request a resource protected by authentication,
   http://yourserver.com/mydir

2. Enter a valid user and password

Security 7: LDAP Authentication

Definition:

LDAP authentication uses an LDAP directory server to look up valid users and groups.

NES/iWS:

Preferences → Restrict Access
Global Admin → Global Settings → Config Directory Service

HP-UX Apache-based Web Server:

Users and groups can be authenticated in an iPlanet Directory Server or in an OpenLDAP Directory Server. Secure authentication is done over SSL to the LDAP server using the **stunnel** program for encryption.

Detailed information is available in the `/opt/hpws/hp_docs/apache/ldap.admin.guide.`

1. Enable LDAP authentication in the `httpd.conf` by uncommenting the include,
   `# Include conf/ldap.conf`
   to
   `Include conf/ldap.conf`

   If you are planning on using mod_auth_ldap/mod_ldap, make sure the following line is uncommented:
   ```
   LoadModule ldap_module modules/mod_ldap.so
   LoadModule auth_ldap_module modules/mod_auth_ldap.so
   ```

2. Edit `/opt/hpws/apache/conf/ldap.conf` to set up your auth_ldap or mod_auth_ldap configuration.

   If you are planning on using auth_ldap, make sure the following line is uncommented:
   ```
   LoadModule auth_ldap_module modules/auth_ldap.so
   ```

   For both mod_auth_ldap and auth_ldap, the following lines are required to require LDAP authentication for access to the manual directory,
   ```
   <Location /manual>
     AuthName "Restricted Area"
     AuthType Basic

     # AuthLDAPURL should point to your ldap server
     AuthLDAPURL ldap://ldap_server.com:ldap_port/o=organization.com

     # AuthLDAPStartTLS on
     require valid-user
   </Location>
   ```

3. If the LDAP Directory Server is configured for SSL (OpenLDAP uses TLS),
   a. Configure Stunnel by editing /opt/hpws/apache/conf/stunnel.conf. Stunnel must accept connections from Apache on the same port that is configured in ldap.conf (i.e. 7777).

The outgoing connection to the LDAP server must also be configured with the correct LDAP server address and port.

To accept requests on 7777 and send to myserver.com on port 636, change the "Service-level configuration" for "ldap" to:

```
# Service-level configuration
[ldap]
accept  = 127.0.0.1:7777
connect = myserver.com:636
```

b. Start Stunnel using the stunnel.init script. Each time Stunnel is started a random file is created in `/opt/hpws/apache/stunnel/.stunnel.rnd` to create a new random new seed.

$ /opt/hpws/apache/stunnel/stunnel.init start

NOTE: If you are running a LDAP server on the same machine as Apache, you may already be using port 389 or 636. Be sure to choose an unused port for Stunnel (see Troubleshooting).

4. When done, use the stunnel.init script to stop Stunnel. This kills the logs/stunnel.pid process

$ /opt/hpws/apache/stunnel/stunnel.init stop

Verification:

1. In a browser enter the URL,
   http://yourserver.com/manual/

2. Enter a user name and password,
   `Enter user name: <user name>`
   `Password: <password>`

3. If using **stunnel**, check to see if it is running,
   $ ps –aef | grep stunnel | grep –v grep
   ```
   www 28953     1  0 12:50:50 ?   0:00 /opt/hpws/apache/stunnel/sbin/stunnel
   /opt/hpws/apache/conf/stunnel.conf
   ```

8.5 Migrating Server-Side Execution

See Appendix 3: Server-Side Execution for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Server-Side 1: CGI scripts

Definition:

CGI scripts are scripts written in any number of languages that adhere to the Command Gateway Interface protocol.

NES/iWS:

CGI programs may be enabled in one of the following ways:

- File in a CGI directory:
  All files in these directories are CGI programs. To determine if configured, use the CGI Directory page in the Programs tab of the Server Manager.

- File with a specific extensions:

Files with certain extensions are treated as CGI programs, regardless of which directory they reside in. The default CGI extensions are .cgi, .bat and .exe. To determine if configured, use the CGI File Type page in the Programs tab of the Server Manager.

Programs → CGI Directory
Programs → CGI File Type

HP-UX Apache-based Web Server:

CGI programs may be enabled in one of the following ways:

- File in a CGI directory:
  Use the `ScriptAlias` directive. Any file that resides in the directory specified by `ScriptAlias` will be executed as a CGI. The default location is `/opt/hpws/apache/cgi-bin/`. To run CGI scripts from additional locations, add a `ScriptAlias` directive in `httpd.conf`,

  ```
  ScriptAlias /cgi-bin/ "/opt/hpws/apache/cgi-bin/"
  ```

- File with a specific extensions:
  Files with certain extensions are treated as CGI programs, regardless of which directory they reside in. To create CGI extensions,

  1. Use `Addhandler` in `httpd.conf`,

     ```
     AddHandler cgi-script .cgi
     ```

  2. Add `ExecCGI` to the `Options` directive in `httpd.conf`,

     ```
     <Directory "/opt/hpws/apache/cgi-bin">
         AllowOverride None
         Options +ExecCGI
         Order allow,deny
         Allow from all
     </Directory>
     ```

- CGI daemon (default enabled)
  Using a daemon speeds up CGI execution by eliminating the overhead of starting up a new process and threads for each CGI invocation. Apache communicates with this daemon using a UNIX domain socket whose name is specified in the `ScriptSock` directive.

  To configure a CGI daemon:

  1. Use `mod_cgid.so` instead of `mod_cgi.so` in `httpd.conf`,
     ```
     LoadModule cgid_module modules/mod_cgid.so
     #LoadModule cgi_module modules/mod_cgi.so
     ```

  2. View `ScriptSock` directive,
     ```
     <IfModule mod_cgid.c>
     # Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
     # for setting UNIX socket for communicating with cgid.
     Scriptsock            logs/cgisock
     </IfModule>
     ```

- suEXEC
  suEXEC provides the ability to run CGI programs under a different user ID from the user ID of the web server. Normally, CGI executes as the same user as the web server.

  For more information on suEXEC, see `/opt/hpws/hp_docs/apache/suexec.admin.guide`

To configure suEXEC:

1. "Unhide" suexec binary.
   $ mv /opt/hpws/apache/bin/suexec.hide /opt/hpws/apache/bin/suexec

2. Make sure suexec has the necessary ownership/permissions:
   $ chmod 4755 /opt/hpws/apache/bin/suexec
   $ chown root:sys /opt/hpws/apache/bin/suexec

3. Edit `httpd.conf` to set `SuexecUserGroup`
   ```
   <IfModule mod_suexec.c>
           SuexecUserGroup <your-userid> <your-group>
   </IfModule>
   ```

4. Make sure `cgi-bin` directory and `showuser.cgi` file have the necessary ownership/permissions:
   $ chmod 755 /opt/hpws/apache/cgi-bin /opt/hpws/apache/cgi-bin/showuser.cgi
   $ chown <*your-userid*>:<*your-group*> /opt/hpws/apache/cgi-bin /opt/hpws/apache/cgi-bin/showuser.cgi

Verification:

1. To verify CGI in /cgi-bin/,
   http://yourserver.com/cgi-bin/mycgi

2. To verify CGI with an extension,
   http://yourserver.com/mycgi.cgi

3. To verify suEXEC in /cgi-bin/,
   a. Access the URL,
      http://yourserver.com/cgi-bin/showuser.cgi
   b. You should see something like the following:
      `Username=<your-userid>`

4. To verify suEXEC in a user directory,
   a. Edit `httpd.conf`,
   ```
   <Directory /home/*/public_html>
      AddHandler cgi-script .cgi
      AllowOverride FileInfo AuthConfig Limit
      Options ExecCGI MultiViews Indexes SymLinksIfOwnerMatch
   IncludesNoExec
      <Limit GET POST OPTIONS PROPFIND>
         Order allow,deny
         Allow from all
      </Limit>
      <LimitExcept GET POST OPTIONS PROPFIND>
         Order deny,allow
         Deny from all
      </LimitExcept>
   </Directory>
   ```

   b. In your home directory, make a directory called "public_html"
      $ mkdir /home/<*your-userid*>/public_html

   c. Copy `showuser.cgi` into the directory you just created and change the ownership,
      $ chown <*your-userid*>:<*your-group*> /home/<*your-userid*>/public_html/showuser.cgi

d. Access the URL,
http://yourserver.com/~your-userid/showuser.cgi

e. You should see something like the following:
`Username=<your-userid>`

Server-Side 2: Parsed HTML (SHTML/SSI)

Definition:

HTML files are embedded with instructions that are read (parsed) then executed. The results are embedded into the document in the place of the instruction then sent to the client.

NES/iWS:

Content Management → Parsed HTML

HP-UX Apache-based Web Server:

SHTML/SSI is implemented by the INCLUDES filter if a document contains server-side include directives and has the extension .shtml. Apache will also activate the INCLUDES filter for any document with mime type text/x-server-parsed-html or text/x-server-parsed-html3 (output is mime type text/html).

For a tutorial on server-side includes, see http://httpd.apache.org/docs-2.0/howto/ssi.html

To configure,

1. Add the following lines in `httpd.conf`,
```
<FilesMatch "\.shtml(\..+)?$">
    SetOutputFilter INCLUDES
</FilesMatch>
```

2. In `httpd.conf` change,
```
<Directory "/opt/hpws/apache/htdocs">
      Options Indexes FollowSymLinks Multiviews
```
To,
```
<Directory "/opt/hpws/apache/htdocs">
      Options Indexes FollowSymLinks Multiviews Includes
```

Verification:

1. Create "ssi.shtml" file.
$ vi /opt/hpws/apache/htdocs/ssi.shtml
```
<HTML><BODY>
<p>Server Side Includes</P>
<br>
<p>   <!--#echo var="DATE_LOCAL" --></p>
<br>
<p>   <!--#config timefmt=" %H:%M:%S %A %B %d, %Y" -->
        Today is <!--#echo var="DATE_LOCAL" --></p>
<br>
<p>    This document last modified <!--#flastmod file="ssi.shtml" --></p>
<br>
<p><pre><!--#exec cmd="ls" --></pre></p>
</BODY></HTML>
```

2. In a browser enter the URL,
http://yourserver.com/ssi.shtml

3. The output should look similar to the following,

```
Server Side Includes

Thursday, 18-Oct-2001 09:38:50 PDT

Today is 09:38:50 Thursday October 18, 2001

This document last modified 09:35:46 Thursday October 18, 2001

README.rus
apache_pb.gif
apache_pb.png
apache_pb2.gif
apache_pb2.png
apache_pb2_ani.gif
index.html
index.html.ca
index.html.cz
...
ssi.shtml
test.php
```

Server-Side 3: Java Servlets and Java Server Pages (JSPs)

Definition:

- Java servlets are server-side Java programs that conform to the Java Servlet API specification and produce dynamic output.

- JSPs are web pages that mix regular, static HTML with dynamically generated HTML to create a servlet which is then executed.

NES/iWS:

NES : **Programs -> Java**
iWS : **Servlets**

HP-UX Apache-based Web Server:

There are two ways to run servlets, depending on whether the web application directory is already configured (3A below) or not (3B below). Also note that you have an option to use mod_jk or mod_jk2. The modules and their related configuration files are shipped with HP-UX Apache-based Web Server. Detailed information about mod_jk2 can be found in `/opt/hpws/hp_docs/tomcat/tomcat.admin.guide`.

1. Make sure that `JAVA_HOME` is set and that `$JAVA_HOME/bin` is on your path.

2. Configure the servlets engine,
   Uncomment the reference to **mod_jk's** configuration file in `httpd.conf`,
   `Include /opt/hpws/apache/conf/mod_jk.conf`

3. Set up the servlet connector to HP-UX Apache-based Web Server and a servlet context,

   A. If the web application structure is already configured:
      1. Add the following path to your servlets configuration file,
         $ vi /opt/hpws/apache/conf/mod_jk.conf

         add the line,
         `JKMount /my_servlets/* ajp13`

2. Create a tomcat context for your servlets. For example, create a file such as, `/opt/hpws/tomcat/conf/apps-my_servlets.xml` where `docBase` points to your web application directory ,
   $ cat /opt/hpws/tomcat/conf/apps-my_servlets.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <webapps>
      <!-- Setting special properties for /my_servlets
        ( as an example of overriding the defaults ) -->
     <Context path="/my_servlets"
           docBase="/opt/html-docs/my_servlets"
           debug="0"
           reloadable="true" >
     </Context>
  </webapps>
```

B. If the web application structure is not already configured:
   1. Add path to your servlets in the mod_jk configuration file:
      $ vi /opt/hpws/apache/conf/mod_jk.conf.

      add the line,
      `JkMount /my_servlets/* ajp13`

   2. Create a **Tomcat** context for your servlets, for example create a file such as `/opt/hpws/tomcat/conf/apps-my_servlets.xml` containing,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <webapps>
      <!-- Setting special properties for /my_servlets
        ( as an example of overriding the defaults ) -->
     <Context path="/my_servlets"
           docBase="/opt/html-docs/my_servlets"
           debug="0"
           reloadable="true" >
     </Context>
  </webapps>
```

   3. Create a servlets directories under the WEB-INF directory
      $ mkdir -p /opt/html-docs/my_servlets/WEB-INF
      $ mkdir -p /opt/html-docs/my_servlets/WEB-INF/classes

   4. Copy your compiled servlet to the directory you just created, for example
      $ cd /opt/html-docs/my_servlets
      $ cp MyServlet.class WEB-INF/classes/MyServlet.class

   5. Create the application context file, for example, `/opt/html-docs/my_servlets/WEB-INF/web.xml` containing,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application 2.2//EN" "http://java.sun.com/j2ee/dtds/webapp_2_2.dtd">
    <web-app>
       <servlet-mapping>
           <servlet-name>
            MyServlet
           </servlet-name>
           <url-pattern>
```

```
            /MyServlet
          </url-pattern>
        </servlet-mapping>
      </web-app>
```

6. Start Apache and Tomcat
   $ /opt/hpws/apache/bin/apachectl start
   $ /opt/hpws/tomcat/bin/startup.sh

For more information about setting up servlets, see
`/opt/hpws/hp_docs/tomcat/tomcat.migration.guide`.

For more information on **Tomcat**,
http://jakarta.apache.org/tomcat/

Verification:

1. Check Java version,
   $ java –version

2. Check that Tomcat is running and its home page is displayed,
   http://yourserver.com:8081

3. Verify that Tomcat can serve JSPs
   http://yourserver.com:8081/test/jsp/HelloWorld.jsp

   the output should be,
   `HelloWorld`

4. Verify that Tomcat can serve an example servlet (xyz can be any alphanumeric characters),
   http://yourserver.com:8081/test/foo/bar/xyz

   the output should be,
   `Servlet: Servlet1`

5. Access your own servlets from Tomcat and verify that they still run,
   http://yourserver.com:8081/my_servlets/MyServlet

6. Access your own servlets from Apache and verify they still run,
   http://yourserver.com/my_servlets/MyServlet

Server-Side 4: Java Server Pages (JSP) Custom Tag Libraries

Definition:

A tag library is a collection of custom actions written in Java that are called by JSPs as tags. It is a powerful feature of JSP v1.1 that aids in separating JSP content display from backend tiers of data access and other services.

HP-UX Tomcat-based Servlet Engine:

HP-UX Tomcat-based Servlet engine can use JSP tag libraries to implement additional functionality. For example, HP-UX Tomcat-based Servlet Engine can use the Jakarta Project's I18N Tag Library to implement I18N for localized formatting and parsing. There are also tag libraries for XML parsing and SQL access.

For more information about tag libraries,
http://jakarta.apache.org/taglibs/tutorial.html

http://jakarta.apache.org/taglibs/doc/i18n-doc/intro.html

Verification:

Execute the simple custom tag example provided with Tomcat.

1. Go to Tomcat's JSP example page, http://yourserver.com:8081/examples/jsp/index.html

2. Select "Simple custom tag example"

8.6 Migrating Management

See Appendix 4: Management for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Management 1: Server Status

Definition:

Information about the web server that enables an administrator to check on the web server.

NES/iWS:

Status → Monitor Current Activity

HP-UX Apache-based Web Server:

The `mod_status` and `mod_info` modules provide Apache status information. These are enabled in `httpd.conf`. `mod_status` creates an HTML page that displays number of children serving requests, number of idle children, average number of requests per second, time when the web server was last started, etc. `mod_info` provides a comprehensive overview of the server configuration, including all installed modules and directives in the configuration files.

To enable server status,

1. Uncomment the server-status handler and change .your-domain.com to your own domain name or address,
```
<Location /server-status>
   SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from .your-domain.com
</Location>
```

2. Optionally, enable full status information,
```
ExtendedStatus On
```

3. Enable server configuration reports, changing your-domain.com to your domain name or address,
```
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Allow from .your-domain.com
</Location>
```

For more information see, http://httpd.apache.org/docs-2.0/mod/mod_info.html and http://httpd.apache.org/docs-2.0/mod/mod_status.html

Verification:

1. View server status,
   http://yourserver.com/server-status

2. View server info,
   http://yourserver/server-info

Management 2: Server Administration

Definition:

Server administration is the configuration, monitoring, tuning, etc. of a web server.

NES/iWS:

Server administration can be done by editing configuration files from the command line or by using the Administration Server. The Administration Server is a separate server that manages the web server through a browser-based GUI with its own port number.

HP-UX Webmin-based Admin:

HP-UX Apache-based Web Server can be administered from a combination of the command line, utilities, and **HP-UX Webmin-based Admin**. Open Source **Webmin** is a web-based interface that consists of a miniserver and a number of CGI programs. The web server and CGI programs are written in Perl and use no external modules. **HP-UX Webmin-based Admin** is a general purpose UNIX and Apache administration tool that has been enhanced and customized for HP-UX Apache-based Web Server.

To start up **HP-UX Webmin-based Admin,**

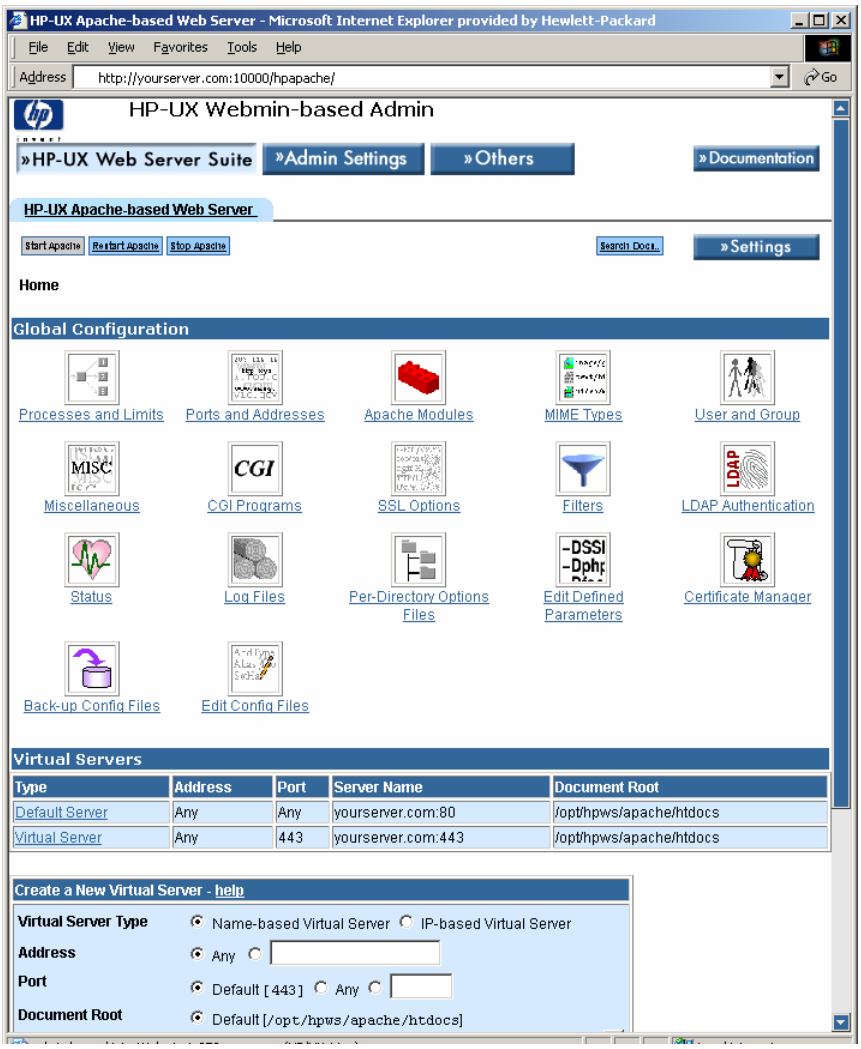
1. Run the Webmin startup script,
   $ **/opt/hpws/webmin/webmin-init start**

2. Point the browser to Webmin's port and, when prompted, enter a username and password,
   http://yourserver.com:10000
   `Login` : **admin**
   `Password` : **hp.com**

3. Initially, you will be on the HP-UX Web Server Suite page. To get to the HP-UX Apache-based Web Server administration, click on the appropriate icon.

For general information about **Webmin**, see http://www.webmin.com/webmin/
For an Open Source **Webmin** user guide for Apache, see http://www.swelltech.com/support/webminguide → Apache Webserver

For more information on **HP-UX Webmin-based Admin**, including changing **Webmin's** password, see `/opt/hpws/hp_docs/webmin/webmin.admin.guide`



Verification:

1. Check if Webmin is running,
   $ ps –aef | grep webmin | grep –v grep

   You should see something like,
   ```
   root  3140  1  0 20:54:53 ? 0:00 /opt/hpws/webmin/miniserv.pl
   /opt/hpws/webmin/miniserv
   ```

2. In a browser enter the URL,
   http://yourserver.com:10000/hpapache

   The page should have the title: “HP-UX Apache-based Web Server”

3. Click on the ”Log Files” icon within the “Global Configuration” section.

4. If there is an Error Log specified, click on the “View..” link next to the filename.
   If “default” is specified, you should also be able to “View..” it.
   It should display the last 20 lines of the log file. (The amount can be changed.)
   Do the same for the Access Log files that are specified.

5. Stop Webmin then check if is has stopped,

$ /opt/hpws/webmin/webmin-init stop
$ ps –aef | grep webmin | grep –v grep

You should not see any output from the ps/grep command.

You should see something similar to the following:
`Stopping Webmin server in /opt/hpws/webmin`

Management 3: Cluster Management

Definition:

Cluster Management is the central management of multiple remote web servers.

NES/iWS:

General Admin

HP-UX Webmin-based Admin:

HP-UX Webmin-based Admin does not have true cluster management. It can be configured to manage multiple instances of HP-UX Apache-based Web Server on the same machine.

To manage additional HP-UX Apache-based Web Servers,

1. From the `Webmin Index`, click on "`Webmin configuration`", then click on "`Webmin Modules`" icon

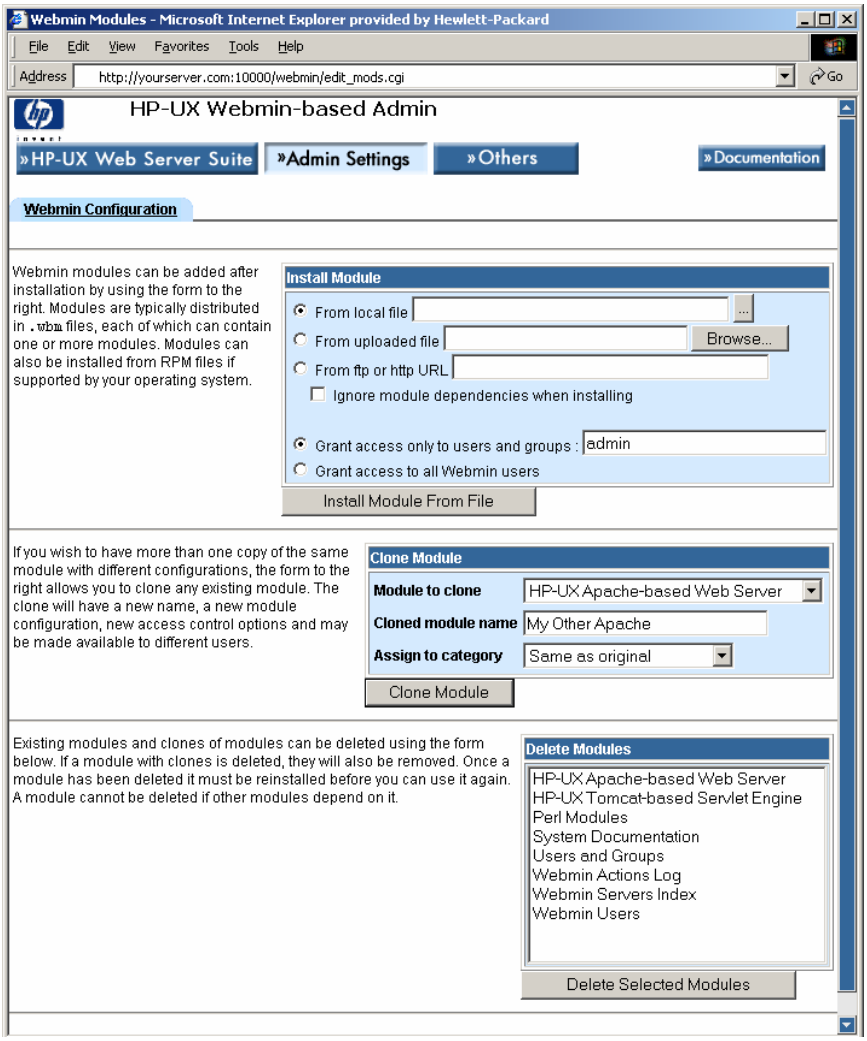
2. Under "`Clone Module`" select "`HP-UX Apache-based Web Server`", specify a unique server name, and click the "Clone Module" button.

3. Go back to `Webmin` Index, select "`Servers`" tab, select the icon of the new server, click on "`Module Config`", set the HP-UX Apache-based Web Server root, the apachectl path, the httpd path, etc.

To remove an instance of HP-UX Apache-based Web Server from HP-UX Webmin-based Admin,

1. From the `Webmin` Index, click on "`Webmin configuration`", then click on "`Webmin Modules`" icon

2. Under "`Delete Modules`" select the Apache server to remove, click on "Delete Selected Modules" button



Verification:

1. Install HP-UX Apache-based Web Server
   swinstall

2. Move HP-UX Apache-based Web Server from its default location (`/opt/hpws/apache`) to a new root using the **altroot.sh** utility
   /opt/hpws/apache/util/altroot.sh /opt/hpws/apache2.org

3. Install HP-UX Apache-based Web Server again

4. Modify `LogLevel` in both web servers and verify it the `error_log` has been affected

Management 4: Distributed Administration

Definition:

Distributed administration is the ability to create multiple administrators with different levels of administration capability.

NES/iWS:

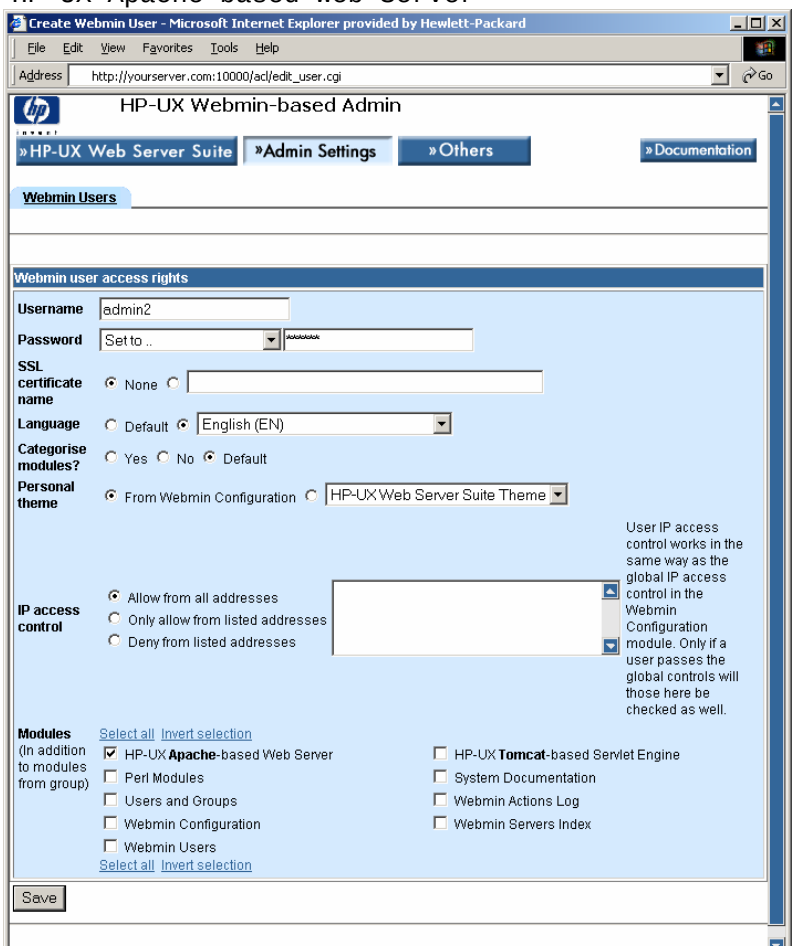
General Admin → Users and Groups

HP-UX Webmin-based Admin:

HP-UX Webmin-based Admin can be set up with multiple administrators, each with different capabilities for modifying the web server. HP-UX Webmin-based Admin is installed with user "admin" with password "hp.com" pre-configured.

- To set up additional administrators,
  1. Go to **Webmin's** main screen,
     http://yourserver.com:10000

  2. Click on "`Webmin Users`" icon, click on "`Create a new Webmin user`", under modules select "`HP-UX Apache-based Web Server`"



  3. Fill in the user name, password, etc., then click on the "`Save`" button.

  4. To remove an administrative user,
     Click on "`Webmin Users`" icon, click on the user name, click on the "`Delete`" button

Verification:

1. Create user `testadmin` with log viewing capabilities only

2. Log on to **Webmin** as `testadmin`

3. View the `error_log`

4. Try to update the `httpd.conf` file.

5. Verify that the update is not successful

Management 5: Dynamic Log Rotation

Definition:

Dynamic log rotation automatically archive log files based on some criteria such as date, time, a specified interval, etc.

NES/iWS:

Status → Archive Log

HP-UX Apache-based Web Server:

Apache logs can be rotated using an open source rotation utility, such as **rotatelogs** that is bundled with HP-UX Apache-based Web Server, or using simple cron-based scripts.

- To use the **rotatelogs** utility based on *time*:
  1. Use Apache's piped logfile feature, for example, on the `access_log`,
     ```
     CustomLog "|/opt/hpws/apache/bin/rotatelogs /opt/hpws/apache/logs/access_log 86400" common
     ```

  2. Check for files `/opt/hpws/apache/logs/access_log.nnnn` where nnnn is the system time at which the log starts (multiple of the rotation time for synchronization with cron scripts). At the end of each rotation time (here after 24 hours, 24 hrs x 60 mins/hr x 60 secs/min = 86400 secs) a new log is started.

- To use the **rotatelogs** utility based on *size*:
  1. Use Apache's piped logfile feature, for example, on the `access_log`,
     ```
     CustomLog "|/opt/hpws/apache/bin/rotatelogs /opt/hpws/apache/logs/access_log 5M" common
     ```

  2. Check for files `/opt/hpws/apache/logs/access_log.nnnn` where nnnn is the system time at which the log starts (multiple of the rotation time for synchronization with cron scripts). At the end of each rotation time (whenever the size is reached) a new log is started.

- To rotate Apache logs using a cron-based script:
  1. Create a script, for example,
     ```
     mv access_log access_log.old
     mv error_log error_log.old
     apachectl restart
     ```

  2. Optionally, compress the log files within the script by adding, for example,
     ```
     gzip access_log.old error_log.old
     ```

NOTE: If "**mv**" is done while Apache is running Apache will continue writing to the old log files since it continues to hold them open. Apache must be restarted after the log files are moved or deleted in order to open new logs. To close the old log files and open new ones, use "**apachectl stop**" then "**apachectl start**".

For more information on log rotation,
http://httpd.apache.org/docs-2.0/logs.html#rotation
http://httpd.apache.org/docs-2.0/programs/rotatelogs.html

Verification:

Look at the log files in the logs directory to verify they are being rotated,

$ **ls /opt/hpws/apache/logs**

8.7 Migrating Web Publishing

See Appendix 5: Web Publishing for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Web publishing is updating or adding content to a remote web server.

Web Publishing 1: HTTP PUT

Definition:

"PUT" is an HTTP method that is similar to the POST method except that a POST is normally directed to a script that already exists while a PUT can be directed to a resource which does not (yet) exist.

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Server handles PUT similar to the way it handles POST. It supports it, however you need to supply a script to handle it and make changes to `httpd.conf`.

For information on implementing PUT using CGI,
http://www.apacheweek.com/features/put

For information on implementing PUT using PHP,
http://teckla.corp.hp.com/reference/cgi/php/features.file-upload.put-method.html

Verification:

Check that HTTP PUT is enabled.

Web Publishing 2: WebDAV

Definition:

WebDAV ("Web-based Distributed Authoring and Versioning") is a set of extensions to the HTTP protocol that allows users to create, move, copy, and delete resources (files) and collections (directories) on a remote server. For example, **WebDAV** can be used to edit HTML files. HP-UX Apache-based Web Server provides class 1 and class 2 WebDAV using the mod_webdav module.

HP-UX Apache-based Web Server:

To configure **WebDAV**,

1. Ensure the following lines are present and not commented in `httpd.conf`,
```
LoadModule dav_module          mod_dav.so
LoadModule dav_fs_module       mod_dav_fs.so
```

2. Enable by turning on DAV in `httpd.conf` by adding the "`DAV`" directive,
```
<Directory "/opt/hpws/apache/htdocs">
# add the following line
DAV On
```

3. Set "`DAVLockDB`" in `httpd.conf` to anything except an NFS mounted files system. This is a prefix string used by **WebDAV** to create files.
```
DocumentRoot "/opt/hpws/apache/htdocs"
# add the "DAVLockDB" directive as follows
DAVLockDB /opt/hpws/apache/locks/DAVLock
```

4. Create the locks directory and change ownerships,

$ mkdir /opt/hpws/apache/locks
$ chown –R www:other /opt/hpws/apache/locks
$ chown –R www:other /opt/hpws/apache2htdocs

More information on WebDAV see, http://httpd.apache.org/docs-2.0/mod/mod_dav.html and http://www.webdav.org

Verification:

- Windows2000 ONLY
  Double click the "My Network Places" icon on the Windows 2000 desktop.
  Double click "Add Network Place".

- WindowsNT ONLY
  Double click on "Network Neighborhood".
  Click on the "Network Neighborhood" click-down list, and select "Web Folders".
  Double click on "Add Web Folder".
  Note: WindowsNT may not have complete support for "Web Folders" and hence you may not be able to fully test WebDAV.

- WindowsNT and Windows2000
  Enter the following URL: http://yourserver.com/
  Provide a simple display name for this WebDAV folder

If able to connect successfully to the WebDAV server and a web-folder is created on the Windows machine, then our WebDAV server is working.

8.8 Migrating Performance

See Appendix 6: Performance for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

More complete information on performance and sizing is available in the Performance/Sizing section of the FAQ (see Getting More Information). This is also some performance information under HP-UX Web Server SuiteTechnical Tips on HP's webserver site, http://www.hp.com/products1/unix/webservers/apache/techtips/index.html.

Performance 1: File Caching

Definition:
File Caching is the ability to save files in local storage for quicker access times.

iPlanet/NES:
File caching values are set in the nsfc.conf file.

HP-UX Apache-based Web Server:
Apache 2.x separates caching from the proxy server function. **mod_proxy** does proxying only and **mod_cache** implements caching of either local or proxied content. **mod_cache** requires one or more storage management modules. These are currently experimental and should be used with caution,

- **mod_disk_cache**
  A disk-based storage manager generally used for proxy caching. Causes the proxy server to serve cached files directly instead of sending on requests to the web server**.**
- **mod_file_cache**
  An mmap-based or file handle-based storage manager that allows the server file system to control whether the file is in memory or not.
- **mod_mem_cache** (not implemented in HP-UX Apache-based Web Server)
  An in-memory based storage manager primarily used for caching local content.

1. Specify caching in `httpd.conf`. **CacheOn** turns on caching, **CacheRoot** specifies where file handles are stored, and **CacheEnable** specifies the kind of caching (i.e. disk) and which parts of the file system may be cached (i.e. /):

```
<IfModule mod_cache.c>
   CacheOn On
      <IfModule mod_disk_cache.c>
        CacheRoot /opt/hpws/apache/proxy
        CacheEnable disk /
       </IfModule>
</IfModule>
```

2. Specify in `cache.conf` which files to cache:

```
 <IfModule mod_file_cache.c>
   CacheFile /opt/hpws/apache/htdocs/index.html
 </IfModule>
```

3. Create the directory where cached file handles will be stored:
   mkdir /opt/hpws/apache/proxy
   chown www:other /opt/hpws/apache/proxy
   chmod 755 /opt/hpws/apache/proxy

4. Optionally, use **util/cache_util.pl** to automatically save the most frequently accessed files. **cache_util.pl** will add `CacheFile` directives in `cache.conf` using the most frequently requested files listed in `logs/access_log`.

   $ **/opt/hpws/apache/util/cache_util.pl**

For more information,
http://httpd.apache.org/docs-2.0/mod/mod_cache.html
http://httpd.apache.org/docs-2.0/mod/mod_file_cache.html.

Verification:

1. There is no easy way to check if file caching is operating. Monitor the performance before and after file caching in enabled. When enabled, files should be delivered more quickly to clients.

2. If **cache_util.pl** was used, check the cached file list in `cache.conf`,
   a. **cat /opt/hpws/apache/util/cache.conf**

8.9 Migrating Scalability

See Appendix 7: Scalability for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Scalability 1: Load Balancing

Definition:

Load balancing is the distribution of client requests between multiple servers to improve reliability and performance of large, busy web sites.

HP-UX Apache-based Web Server:

Load balancing can be done using the features on `mod_rewrite` and `mod_proxy`. `mod_rewrite` can be configured to randomly select a back-end server for each client request. `mod_proxy` disguises the URL so clients are forced to go through the proxy. If considering `mod_backhand`, verify first if the module has an Apache 2.x version.

Users could also build the open source load balancing module from the Backhand Project, `mod_backhand`. When this is linked as a dynamically loadable module (DSO), it is loaded at Apache startup time. For the source code and building instructions for `mod_backhand`, go to the Backhand Project's open source site, http://www.backhand.org/mod_backhand/.

To implement load balancing with `mod_rewrite` and `mod_proxy`,

- One machine is selected to be the proxy server. This server appears to clients as the web server.
- The proxy server machine uses `mod_rewrite` directives to randomly select a back-end server for each client request.
- Caching is disabled so that back-end servers handle the load

To implement load balancing,

1. Create a proxy server, for example, `www.yourserver.com`, that randomly accesses six back-end servers named, for example, `www1.yourserver.com` through `www6.yourserver.com`.

2. Set up the proxy server,

```
ServerName www.yourserver.com
ServerAdmin webmaster@www.yourserver.com
ServerRoot  /opt/hpws/apache
DocumentRoot /opt/hpws/apache/proxy_docs
ErrorLog  /opt/hpws/apache/proxy_error_log
TransferLog /opt/hpws/apache/proxy_access_log

# This server is for proxying so disable everything else
<Directory />
Options None
AllowOverride None
</Directory>

# Turn on URL rewriting
RewriteEngine On

# Define a log for debugging (if needed)
RewriteLog logs/proxy_rewrite

# Set logging to 0 for performance unless actively debugging
```

```
RewriteLogLevel 0

# Define the server map
RewriteMap myservers rnd:/opt/hpws/apache/rewritemaps/mymap.txt

# Rewrite the URL if it matches the web server name
RewriteRule ^http://www\.(.*)$   http://mymap:www).$2 [P,L]

# Forbid any URL that doesn't match
RewriteRule  .* - [F]

#  Proxy directives
ProxyRequests  on

ProxyPassReverse   /  http://www1.myserver.com/
ProxyPassReverse   /  http://www2.myserver.com/
ProxyPassReverse   /  http://www3.myserver.com/
ProxyPassReverse   /  http://www4.myserver.com/
ProxyPassReverse   /  http://www5.myserver.com/
ProxyPassReverse   /  http://www6.myserver.com/
```

Verification:

1. Flood the proxy server with client requests. For example, test a URL 4000 times with 200 concurrently:
   /opt/hpws/apache/bin/ab -n 4000 –c 200 http://localhost/index.html

2. Determine if response time is improved by looking at the output of **ab**. **ab** is the Apache benchmarking tool bundled with the product, http://httpd.apache.org/docs/programs/ab.html

```
This is ApacheBench, Version 2.0.32 <$Revision: 1.87 $> apache-2.0
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd,
http://www.zeustech.net/
Copyright (c) 1998-2001 The Apache Software Foundation,
http://www.apache.org/

Benchmarking localhost (be patient)
Completed 400 requests
Completed 800 requests
Completed 1200 requests
Completed 1600 requests
Completed 2000 requests
Completed 2400 requests
Completed 2800 requests
Completed 3200 requests
Completed 3600 requests
Finished 4000 requests
Server Software:        HP
Server Hostname:        localhost
Server Port:            80

Document Path:          /index.html
Document Length:        4307 bytes

Concurrency Level:      200
Time taken for tests:   70.367472 seconds
Complete requests:      4000
```

```
Failed requests:        12
   (Connect: 0, Length: 12, Exceptions: 0)
Write errors:           0
Total transferred:      18352776 bytes
HTML transferred:       17176316 bytes
Requests per second:    56.84 [#/sec] (mean)
Time per request:       3.518 [ms] (mean)
Time per request:       0.018 [ms] (mean, across all concurrent requests)
Transfer rate:          254.69 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:    0     0    1.1 0 27
Processing: 216  3505 1328.8 3648 10469
Waiting:    -2878  1934 1224.3 2126 8359
Total:      216  3505 1328.8 3648 10469

Percentage of the requests served within a certain time (ms)
  50%   3648
  66%   3954
  75%   4258
  80%   4484
  90%   5475
  95%   5988
  98%   6079
  99%   8094
 100%  10469 (longest request)
```

8.10 Migrating Enterprise Capability

See Appendix 8: Enterprise Capability for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Enterprise 1: Multiple Web Server Instances

Definition:

This is the ability to run multiple copies of a web server on one machine.

NES/iWS:

General Administration

HP-UX Apache-based Web Server:

Multiple installations of HP-UX Apache-based Web Server can run on the same machine provided they use different port numbers or different IP addresses. The `/opt/hpws/util/altroot.sh` utility will move Apache from its default location (`/opt/hpws/apache/`) to an alternate directory. `altroot.sh` changes all occurrences of the current Apache root to the new Apache root in all configuration files and scripts.

To move HP-UX Apache-based Web Server to a new directory,
/opt/hpws/util/altroot.sh *</current/apache/root> </new/apache/root>*

For more information on using **altroot.sh** and its parameters see,
`/opt/hpws/hp_docs/utilities.user.guide`

Verification:

1. Install HP-UX Apache-based Web Server

2. Move it to a new root with and update the RC Scripts (default)
   $ altroot.sh /opt/hpws/apache /opt/hpws/apache_new

3. Change the port numbers,
   a. In `httpd.conf`:
   ```
   ServerName yourserver.com:8080
   Listen 8080
   Listen 4443
   ```

   b. In `ssl.conf`:
   ```
   <VirtualHost _default_:4443>
   …
   ServerName yourserver.com:4443
   ServerAdmin www@yourserver.com
   …
   </VirtualHost>
   ```

4. Install HP-UX Apache-based Web Server again

5. Start up both HP-UX Web Server Suite copies,
   $ /opt/hpws/apache/bin/apachectl start
   $ /opt/hpws/apache_new/bin/apachectl start

6. Reboot the system and verify that both webs servers auto restart
   $ reboot
   $ ps –aef | grep httpd

Enterprise 2: Internationalization (i18n)

Definition:

Encompasses support for a user's native language for input, file names, printing, displaying messages, and formatting numbers, dates, money, etc. "i18n" is a common abbreviation for internationalization.

NES/iWS:

Content Management → Document Preferences and International Characters

HP-UX Apache-based Web Server:

HP-UX Apache-based Web Server has a number of internationalization features. Native language versions of a file can be returned. Different character sets are pre-defined and new ones can be added. Client error messages are pre-defined for English, French, Spanish, and German and can customized to add new languages and alter message design. Apache's default index.html file comes in many different languages. I18N is also available through the Jakarta Project's i18n JSP Tag Library.

To get native language versions of a document,

1. Set your client to your native language. On Windows 2000, go to Control Panel, Regional Options and select "Your locale".

2. Set `Options Multiviews` in the directory(s) where you want to use native language files, for example,
   ```
   <Directory "/opt/hpws/apache/htdocs">
   # Note that "MultiViews" must be named *explicitly* --- "Options All"
   ```

```
# doesn't give it to you.
#
# The Options directive is both complicated and important.  Please see
# http://httpd.apache.org/docs-2.0/mod/core.html#options
# for more information.
#
    Options Indexes FollowSymLinks Multiviews
</Directory>
```

3. Create files using the file extension for your language. Many are already pre-configured in `httpd.conf`. For French language files, for example, use .fr
   $ vi myfile.fr

For more information on native language configuration see,
http://yourserver.com/error/README
http://httpd.apache.org/docs-2.0/misc/custom_errordocs.html
http://httpd.apache.org/docs-2.0/content-negotiation.html
http://httpd.apache.org/docs-2.0/mod/mod_negotiation.html

Verification:

1. Set your client to your native language. On Windows 2000, go to Control Panel, Regional Options and select "Your locale".

2. Access a directory that has native language files, for example, http://yourserver.com/manual

Enterprise 3: Virtual Servers

Definition:

Allows more than one server to be running on the same system. The different servers can be distinguished with different names, IP addresses or Ports.

NES/iWS:

General Administration

HP-UX Apache-based Web Server:

Apache defines IP-based virtual hosts that share a single IP address and Name-based virtual hosts where each virtual server requires its own IP address. A virtual server can use a non-standard port number (port 80 is the default non-SSL (http) port, port 443 is the default SSL (https) port.

The following example sets up two name-based virtual hosts called "foo" and "bar". You must have their names already configured on your DNS server,

```
192.168.1.1 foo www.foo.com
192.168.1.1 bar www.bar.com
```

1. Add the following lines to `httpd.conf` in "Section 3: Virtual Hosts"

```
 NameVirtualHost 192.168.1.1

<VirtualHost foo>
DocumentRoot /opt/hpws/apache/htdocs-foo
ServerName www.foo.com
</VirtualHost>

<VirtualHost bar>
DocumentRoot /opt/hpws/apache/htdocs-bar
```

```
ServerName www.bar.com
</VirtualHost>
```

2. Check the configuration,

   ```
   /opt/hpws/apache/bin/httpd -t -D DUMP_VHOSTS
   ```

   You should see the text:

   ```
   VirtualHost configuration:
       192.168.1.1:80 is a NameVirtualHost
       default server www.foo.com (/opt/hpws/apache/conf/httpd.conf:1299)
       port 80 namevhost www.foo.com (/opt/hpws/apache/conf/httpd.conf:1299)
       port 80 namevhost www.bar.com (/opt/hpws/apache/conf/httpd.conf:1304)
   ```

   For more information see,
   http://www.apache.org/docs-2.0/vhosts/

Verification:

1. Create a file under virtual host “foo”,
   $ vi /opt/hpws/apache/htdocs-foo/index.html

2. Add the following:

   ```
   <HTML>
   <HEAD>
   <TITLE>index.html for www.foo.com</TITLE>
   </HEAD>
   <BODY>
   <P>index.html for www.foo.com</P>
   </BODY>
   </HTML>
   ```

   Create a file under virtual host “bar”
   $ vi /opt/hpws/apache/htdocs-bar/index.html

3. Add the following:

   ```
   <HTML>
   <HEAD>
   <TITLE>index.html for www.bar.com</TITLE>
   </HEAD>
   <BODY>
   <P>index.html for www.bar.com</P>
   </BODY>
   </HTML>
   ```

4. Invoke virtual server foo’s home page
   http://www.foo.com

   You should see,
   “index.html for www.foo.com”

5. Invoke virtual server bar’s home page,
   http://www.bar.com

   You should see the text:
   “index.html for www.bar.com”

Enterprise 4: Proxy Server

Definition:
Proxies are intermediate servers that stand between a client and a remote server and make requests to the remote server on behalf of the client.

NES/iWS:
This is provided in a separate product.

HP-UX Apache-based Web Server:
HP-UX Web Proxy is a secure revers proxy solution that works in conjunction with the HP-UX Apache-based Web Server and ships with the HP-UX Web Server Suite. Detailed documentation is available online at http://www.hp.com/products1/unix/webservers/apache/techtips/index.html

Verification:
Verification is provided in the documentation.

8.11 Migrating Developer Support

See Appendix 9: Developer Support for a summary on the how NES and iWS differ from HP-UX Web Server Suite.

Developer 1: Database Connectivity

Definition:
Database connectivity is accessing database content via the web server to return dynamic content to a client.

NES/iWS:
Programs→Server Side Javascript
or, http://<*yourserver.com*>/appmgr

HP-UX Apache-based Web Server:
HP-UX Apache-based Web Server can connect to a variety of databases using Java JDBC, Perl's DBI, PHP, and PHP extensions (including Oracle 8.1.6 or later).

- JDBC:
  JDBC(TM) technology is an API that lets you access virtually any tabular data source from Java. It provides cross-DBMS connectivity to a wide range of SQL databases and to other tabular data sources, such as spreadsheets or flat files. It provides a standard interface to a database server. JDBC can be used with HP-UX Tomcat-based Servlet Engine's servlets or JSPs.

- Perl DBI:
  Perl DBI is an API that allows users to access multiple database types transparently such as Oracle, Informix, mySQL, and Sybase. Perl DBI can be used in conjunction with Perl CGI scripts.
  Using HP Apache mod_perl module, the perl interpreter is embedded is loaded at Apache startup so your DBI will run faster.

  To enable mod_perl:
  - Edit `httpd.conf`,
    1. Uncomment the `mod_perl LoadModule` directive.
       `LoadModule perl_module modules/mod_perl.so`

    2. Verify that the section enclosed in the

`<IfModule mod_perl.c> ... </IfModule>` tags is uncommented
This allows files ending with *.pl to run through `mod_perl`.

- PHP:
  PHP has built-in mySQL support for access to mySQL databases.

  More information about MySQL can be found at http://www.mysql.com/.
  Documentation for MySQL can be found at http://www.mysql.com/documentation/.

  To enable PHP in Apache,
  1. Enable PHP by uncommenting the following line in `httpd.conf`,
     `LoadModule php4_module modules/libphp4.so`

  A PHP extension is required for Oracle connectivity.
  To configure PHP to Oracle:
  1. Enable PHP by uncommenting the following line in `httpd.conf`,
     `LoadModule php4_module modules/libphp4.so`

  2. Install the Oracle client libraries on the same machine as HP-UX Apache-based Web Server.

  3. Uncomment the following line in `/opt/hpws/apache/conf/php.ini`,
     `extension=oci8.sl`

     This tells PHP to load the Oracle extension on startup. The `'extension_dir'` directive in the `php.ini` file points to the location of oci8.sl. It is set by default to `/opt/hpws/apache/php/libs/php/extensions/`

  4. Uncomment and set appropriate values for the following variables in **apachectl**,

     ```
     export ORACLE_HOME=/path/to/oracle-8.1.6/client/side/libraries
     export SHLIB_PATH=$SHLIB_PATH:$ORACLE_HOME/lib
     export
     LD_PRELOAD=$LD_PRELOAD:$ORACLE_HOME/JRE/lib/PA_RISC/native_threads/libjava.sl
     export ORACLE_SID=ConnectionName
     ```

     ** Note that the `LD_PRELOAD` variable is only required for PA-RISC systems.

  For more information about PHP, see the bundled PHP User's Guide `/opt/hpws/hp_docs/apache/php.admin.guide`, http://www.php.net/docs.php and PHP general information, http://www.php.net/

Verification:

mySQL

1. Enable PHP

2. Try a simple SQL example. Assuming a database with a customer table containing name and address fields,

```
<?
    @ $db = mysql_pconnect("localhost", user, password);
    if (!db)
    {
        echo "Error: Couldn't connect";
    }
    mysql_select_db("customers");
    $query = "select * from customers";
    $result = mysql_query($query);
```

```
        $num_results = mysql_num_rows($result);
        for ($i = 0; $i < $num_results; $i++)
         {
             $row = mysql_fetch_array($result);
             echo "<p;>i+1).". Name: ";
             echo $row["name"];
             echo "<br;> Address: ";
             echo $row["address"];
             echo "<p;>";
         }
    ?>
```

JDBC:

1. Enable Tomcat servlets (Server-Side 3: Java Servlets and Java Server Pages (JSPs))

2. Create a sample servlet that talks to a database such as the following

```
package test;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.sql.*;

public class Servlet1 extends HttpServlet {
  private static final String CONTENT_TYPE = "text/html";

  // initialize the driver
  static {
    try {
      Class.forName("COM.ibm.db2.jdbc.net.DB2Driver").newInstance();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

  Connection con;

  public void init() throws ServletException {
      try {
         // construct the URL
         String url = "jdbc:database://server:port/sample";
         // connect to database with userid and password
         con = DriverManager.getConnection(url, "userid", "password" );
      } catch( Exception e ) {
         e.printStackTrace();
      }
  }

  private void doSelect (PrintWriter out) {
      try {
         Statement stmt = con.createStatement();
         // we will run simple select statement
         ResultSet rs = stmt.executeQuery("SELECT * from Table");
         out.println ("<table>");
```

```
            while (rs.next()) {
               String r1= rs.getString(1);
               String r2 = rs.getString(2);
               String oneLine = "<tr><td>" + r1 + "</td><td>" + r2 +
    "</td></tr>";
            }
            stmt.close();
            out.println ("</table>");

         } catch( Exception e ) {
            e.printStackTrace();
         }

      }
      //Process the HTTP Get request
      public void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Servlet1</title></head>");
        out.println("<body>");

        doSelect (out);

        out.println("</body></html>");
      }
      //Clean up resources
      public void destroy() {
      }
    }
```

Developer 2: Plug-in APIs

Definition:

Plug-ins are code modules that are written using the web servers APIs and which extend the functionality of the web server.

NES/iWS:

NES and iPlanet use the Netscape Application Interface (NSAPI) to create plug-ins for the web server.

HP-UX Apache-based Web Server:

Apache has its own set of APIs, called the Apache Portable Runtime (APR) which are different than NSAPIs. Apache plug-ins are called modules and can be written in C, C++, or Perl (using mod_perl) and can define new Apache configuration directives and participate in any phase of request processing. For Apache 2.0, the APRs have been enhanced to provide automatic module loading and new calls with additional capabilities. HP-UX Apache modules Apache modules can be built as static objects that are compiled into Apache but are usually built as shared libraries that are dynamically loaded at Apache startup time. These modules are called Dynamic Shared Objects (DSOs).

Details about the writing Apache modules are provided on the ASF Apache web site, http://httpd.apache.org/docs-2.0/developer/

To migrate NES/iPlanet plug-ins,

1. Some plug-ins might be replaced with open source or third-party Apache modules. For open source Apache modules see the Apache Module Registry, http://modules.apache.org/. For third-party modules, contact the third-party vendor directly. For example, SpeedCard requires a plug-in which is supplied by Rainbow. An equivalent module is available from Rainbow for Apache.

2. Compare the capabilities between NSAPI and Apache APIs and consider the application's overall architecture. It may be advantageous, for example, to rewrite an application into an Apache Perl module or into a Java servlet.

3. `/opt/hpws/apache/example/mod_example` is the source code of a working Apache module that can be used as a template for creating new modules.

4. **apxs** is a bundled utility for building and installing Apache modules (DSOs). For example, to compile the source file mod_foo.c into a DSO using apxs,
   $ apxs -c mod_foo.c

   apxs can auto install a DSO in Apache's shared object directory (`/opt/hpws/apache/modules`).

   $ apxs -i -a mod_foo.c

   This adds the following line to `httpd.conf`,
   LoadModule foo_module libexec/mod_foo.so

   Complete information on **apxs** is provided in the **apxs** man page,
   http://httpd.apache.org/docs-2.0/programs/apxs.html

Verification:

1. Start HP-UX Apache-based Web Server with the new module configured,
   $ /opt/hpws/apache/bin/apachectl start

2. Check the `error_log` for module loading errors
   $ tail /opt/hpws/apache/logs/error_log

3. If the module doesn't load, turn on debugging messages for error_log and try loading it again.
   $ cat httpd.conf,
   …
   `LogLevel debug`
   …

9 Final Migration Verification

Make sure you have performed the verification for each feature that is given in its migration step. Here are additional ways to test HP-UX Web Server Suite.

1. Check the syntax of the configuration file,

   $ /opt/hpws/apache/bin/httpd –t
   ```
   Syntax OK
   ```

2. Start HP-UX Apache-based Web Server as the root user,

   $ /opt/hpws/apache/bin/apachectl startssl

3. Check the error log for Apache and for other components such as Tomcat,

   $ cat /opt/hpws/apache/logs/error_log

4. Check file permissions to make sure your files are readable, writeable, and executable (as appropriate) by HP-UX Apache-based Web Server. HP-UX Apache-based Web Server runs as,

   ```
   User www
   Group other
   ```

5. Check references in existing web pages (servlets, JSPs, HTML, CGI, etc.) to see if web server references are relative or absolute. Absolute references may no longer be valid under the HP-UX Apache-based Web Server directory structure.

6. Run your regression tests making sure to access files and applications that have been affected by the migration.

7. Run performance and load tests. Tune HP-UX Apache-based Web Server directives and HP-UX kernel parameters using the information from the HP-UX Web Server Suite FAQs, http://www.hp.com/products1/unix/webservers/apache/faqs/index.html.

   The FAQs are also bundled with HP-UX Web Server Suite

8. For troubleshooting tips, consult the HP-UX Web Server Suite FAQs, http://www.hp.com/products1/unix/webservers/apache/faqs/index.html.

9. Consult the Migration Guide HP Apache-based Web Server 1.3.x to HP-UX Web Server Suite for more information on the differences between HP Apache-based Web Server 1.3.x and HP-UX Web Server Suite, http://www.hp.com/products1/unix/webservers/apache/techtips/index.html

10. Consult the FAQs for a list of additional manual tests.

11. Remove NES or iPlanet files only after the HP-UX Web Server Suite migration has been thoroughly tested.
    NOTE: Verify the use of Alias directives and/or symlinks from HP-UX Apache-based Web Server to iPlanet before doing any file cleanup.

Appendix A Summary of Web Server Functionality Differences

This section gives a summary of how iPlanet web server functionality (NES and iWS) differs from that of HP-UX Web Server Suite. See the Performing the Migration section for more functionality details and for instructions on migrating individual components.

A.1 Core

Core functionality refers to the fundamental behavior of the web server. This is functionality that is part of a web server's basic operation.

Process Handling

Process handling is the process model of how a web server handles incoming requests.

NES/iWS: iPlanet uses a multi-process, multi-threaded model where each thread handles a request

HP Apache: HP-UX Apache-based Web Server uses a hybrid multi-process, multi-threaded model where each thread handles a request.

Logging

Logging is the saving of information for use in detecting performance problems, errors, and for evidence of potential security violations.

NES/iWS: iWS uses error and access logs. Access logs can be custom formatted or use the Common Logfile Format (CLF). The admin server displays error and access log content.

HP Apache: Apache's main logs are error and access logs, formatted using CLF. There are additional logs for configured features such as Tomcat, Rewrite, and CGI.

Log Rotation

Log rotation is a way to automatically archive log files at specific intervals.

NES/iWS: iWS' admin server can be configured to automatically archive access logs either through Internal-daemon log rotation and cron-based log rotation.

HP Apache: Apache's logs can be archived using open source rotation tools (such as the one bundled with Apache) or using cron-based log rotation.

IP Addresses and Port Numbers

IP addresses specify the network address on which a web server listens. Port numbers specify the network port(s) on which the web server listens.

NES/iWS: By default, NES and iWS listen on port 80. By default, it never tries to match IP addresses with corresponding host names.

HP Apache: By default, HP-UX Apache-based Web Server listens on port 80 (http) and port 443 (https) for all IP addresses. Default ports can be changed and additional ports can be configured. Specific IP addresses can be configured in combination with multiple ports.

DNS lookups

DNS lookups refers to web server access to a domain name server to resolve a client's IP address into a server name.

NES/iWS: Can be enabled or disabled

HP Apache: Can be enabled or disabled. It is disabled (off) by default for better performance.

Document Directories

Document directories are the directories where web pages are located.

NES/iWS: Has a Primary Document Directory. Additional document directories can be configured.

HP Apache: Apache's primary document directory is called the Document Root. Additional document directories can be configured.

Directory indexing

Directory indexing specifies which page(s) to use as an index page, and how to display directories.

NES/iWS: By default, NES and iWS assume that index.html and home.html are the preferred index. Directories can be configured to be displayed in a variety of configurations.

HP Apache: Can retrieve a specific index page, can generate a dynamic directory listing, and can create "Fancy" directory indexes that display graphics, file modification dates, etc.

File redirection

With file redirection, clients accessing a URL on a system are sent to a different location on either the same server or a different server. This is useful if a resource has moved and you want the move to be transparent to the client.

NES/iWS: Can be configured as needed.

HP Apache: Can be configured as needed. Use the "Redirect*" directives from mod_alias.

MIME Types and Settings

MIME types and settings specify the content type (HTML, picture, text, etc.) of a web page so a browser knows how to display it correctly. File extensions (.html, .gif, .txt, etc.) are mapped to a content type that is sent back to the browser. The content type can also specify a character set (charset-iso-2022-jp, etc.) to use when displaying.

NES/iWS: Uses the mime.types file

HP Apache: Uses the mime.types file that is pre-configured with standard MIME types. More types can be added.

A.2 Security

SSL/TLS

NES/iWS: iPlanet is configurable for SSL v.2 or SSL v.3 and for which ciphers each uses. The server can be enabled to request client certificates and to set the key size. SSL caching is also available.

HP Apache: HP-UX Apache-based Web Server uses mod_ssl and OpenSSL to implement SSL/TLS. These provide 128bit/168 bit strong cryptography via SSL v2, SSL v3, and TLS v1.

Certificates

NES/iWS: NES uses Netscape 3.x certificates. iWS uses Netscape 4.x certificate.

HP Apache: Uses Public-Key Cryptography Standard (PKCS) #12 developed by RSA Laboratories.

Chroot

Chroot is a security feature that provides an alternate root directory and limits a web server to directories below the alternate root. When the web server tries to access the root directory it accesses the chroot directory instead.

NES/iWS: NES using Virtual Vault for chroot, iWS has chroot built in.

HP Apache: HP-UX Apache-based Web Server has chroot built in and provides a bundled script for creating a chroot directory and copying files to that directory.

Access Control Files/Dynamic Configuration Files (.nsconfig, .htaccess)

`.nsconfig and .htaccess` files are per-directory configuration files that allow certain directives to be placed in a directory.

NES/iWS: iWS uses .nsconfig and .htaccess and refers to them as Dynamic Configuration Files. A conversion utility for converting .nsconfig to .htaccess is bundled with iWS.

HP Apache: HP-UX Apache-based Web Server uses .htaccess as the default file name for per-directory configuration files. The file name can be changed to any name.

Access Control Lists (ACLs)

ACLs are rules that control access to web server resources.

NES/iWS: NES and iWS implement ACLs in .acl files. They can restrict access to the server, a directory, a URI, a file type, a hostname, an IP address or by time of day. Access rights can be specified as read, write, execute, delete, list, and info. ACLs can authenticate users and groups in an LDAP directory.

HP Apache: HP-UX Apache-based Web Server restricts access using the containers <File>, <Directory>, etc. in conjunction with Allow/Deny commands to restrict access by directory, URI, IP address, or files. The <LIMIT> directive restricts access rights based on HTTP commands (GET, PUT, etc.). HP Apache 2.x authenticates users using an LDAP directory.

Database Authentication

Database authentication authenticates user access to a web resource from a list of users and groups in a database.

NES/iWS: iWS authenticates users and groups from customer databases such as Oracle.

HP Apache: Apache includes Basic Authentication to authenticate users and groups from an ascii flat file. A DBM database file (similar to Basic Authentication) can also be used to authenticate users and groups.

Directory-based Access Control

Directory-based access control accesses a directory server for controlling access to web server resources.

NES/iWS: NES and iWS use LDAP through the iPlanet (Netscape) Directory Server for authenticating users and groups.

HP Apache: HP-UX Apache-based Web Server authenticates users and groups via entries in an iPlanet Directory Server or an OpenLDAP Directory Server. For secure transactions, authentication can be done over SSL to an iPlanet Directory Server or over TLS to an OpenLDAP directory server.

A.3 Server-Side Execution

Common Gateway Interface (CGI)

Common Gateway Interface (CGI) programs are applications that run on the server and generate a response to the requesting client.

NES/iWS: NES and iWS support the CGI standard.

HP Apache: Apache supports the CGI standard. CGI can be written in C, C++, Java, Perl, or using shell commands. mod_perl can be configured to boost Perl CGI performance.

Parsed HTML (SHMTL)

Web servers normally send HTML back to a client exactly as it exists on disk without intervention. With SHTML files, the web server checks (parses) the disk file for special SHTML commands and modifies the file before sending it back to the client.

NES/iWS: NES and iWS uses the term “Parsed HTML”.

HP Apache: Apache refers to SHTML files as Server-Side Includes (SSI).

SHTML <SERVLET> tag

Servlets can be called from a Server-Side Includes (SHTML) document by using the <SERVLET> tag.

NES/iWS: iWS implement this functionality, NES does not.

HP Apache: In HP-UX Tomcat-based Servlet Engine, servlets are executed through a JSP only.

Web Application Interface (WAI)

WAI is an API called the Web Application Interface that extends web server functionality using the Common Object Request Broker Architecture (CORBA). WAI applications/plug-ins are ORB-compliant.

NES/iWS: NES and iWS support WAI applications written in C, C++, or Java that interact with the web server using Internet Inter-ORB Protocol (IIOP). Using servlets instead of WAI is recommended.

HP Apache: HP-UX Web Server Suite does not directly support CORBA. It does support servlets.

Java Servlets

NES/iWS: Servlets are run on an iPlanet native servlet engine.

HP Apache: HP-UX Web Server Suite only supports Tomcat and uses the mod_jk or mod_jk2 connector between HP-UX Apache-based Web Server and the HP-UX Tomcat-based Servlet Engine.

Java Server Pages (JSP)

NES/iWS: NES uses JSP 0.92 and iWS uses JSP 1.1. iWS uses the open source JSP compiler "Jasper" included with Tomcat 3.0 for compiling a JSP pages into servlets. Config/jvm12.conf file is the Java Virtual Machine (JVM) configuration file.

HP Apache: HP-UX Tomcat-based Servlet Engine supports JSP version 1.2 as implemented by the open source Jakarta Project, http://jakarta.apache.org/tomcat/index.html. JSP version 1.*x* is not backward compatible with JSP version 0.92.

Java Server Pages (JSP) Custom Tag Libraries

Tag libraries contain Java code that can be called by JSPs. Tag libraries also allow for the creation of custom JSP tags. Tags aid in the development and understanding of JSPs for HTML authors. Servers that support the JSP 1.1 specification also support tag libraries.

NES/iWS: NES uses JSP 0.92 so tag libraries are not available. IWS uses JSP 1.1 so tag libraries are available.

HP Apache: Tag libraries are supported on HP-UX Apache-based Web Server through HP-UX Tomcat-based Servlet Engine.

Server-Side JavaScript (LiveWire, LiveConnect)

Server-side JavaScript is JavaScript that is executed inside a <SERVER> tag and executes on the server.

NES/iWS: NES and iWS support server-side JavaScript. Server-side JavaScript is part of the LiveWire development tool. JavaScript applications are compiled into .web files and can access relational databases that meet the Open Database Connectivity (ODBC) standard. LiveConnect is a framework for interconnection of Java, HTML, JavaScript, CORBA objects, and plug-ins.

HP Apache: HP-UX Tomcat-based Servlet Engine supports client-side JavaScript but not server-side JavaScript.

A.4 Management

Server Status

NES/iWS: NES and iWS use the Management Information Base (MIB) to detect server status.

HP Apache: Apache uses the `mod_status` and `mod_info` modules to gather and display web server status.

Administration Server

NES/iWS: NES and iWS include a web-based GUI administration server, a separate server that manages multiple web servers through a single interface.

HP Apache: HP-UX Web Server Suite includes a web-based GUI administration called HP-UX Webmin-based Admin. It is a miniserver consisting of a number of Perl CGI programs.

Cluster Management

Administer multiple remote servers from a single administration server.

NES/iWS: Multiple web servers of the same type can be configured identically. The collection of servers that are configured from a single administration server is called a cluster.

HP Apache: HP-UX Webmin-based Admin can manage multiple remote web servers by adding the servers to its configuration.

Distributed Administration

Delegate administrative tasks without giving administrators full access.

NES/iWS: NES 3.x uses the name and password of the logged in admin server to determine what can be configured allowing others to access to a limited set of features.

HP Apache: HP-UX Webmin-based Admin can multiple administrators with different levels of capability to manage the server.

System Monitoring

System monitoring is real-time remote monitoring of networked system for checking status.

NES/iWS: NES and iWS support SNMP versions 1 and 2 for system monitoring through HP's OpenView, Tivoli/IBM's TME, BMC's Patrol, CA's UniCenter and Sun Solstice.

HP Apache: HP-UX Web Server Suite does not implement an interface for remote system monitoring.

A.5 Web Publishing

Web Publishing is the capability for clients to access and manipulate files on a remote server.

Web Publisher

NES/iWS: NES and iWS use Web Publisher. Web Publisher gives direct access to files and folders in a user's home directory and allows sharing of files from the desktop. Web Publisher is implemented as a Java applet that runs on the local client.

HP Apache: HP-UX Apache-based Web Server implements WebDAV, web-based distributed authoring and versioning, through an Apache module. Microsoft FrontPage 2002 Server Extensions are supported with HP-UX Apache-based Web Server starting in January 2004.

Search Collections and Version Control

NES/iWS: NES and iWS use the Verity Search engine and the mks versioning system. Verity Search engine can search collections (directories) for HTML and non-HTML documents, supports NNTP-clients (Network Transfer Protocol for newsgroups) and uses the mks document versioning.

HP Apache: Not available in HP-UX Web Server Suite.

A.6 Performance

File Caching

A caching server stores files for immediate return to a client without accessing a local or remote web server to retrieve it.

NES/iWS:

HP Apache: HP-UX Apache-based Web Server implements caching through the new caching modules, mod_file_cache, mod_cache, and mod_disk_cache. Also, the Network Server Accelerator (NSAHTTP), a in-kernel cache product is now available at http://software.hp.com for HP-UX.

A.7 Scalability

Load Balancing

Load balancing distributes the load of customer requests between servers for large, busy web sites.

NES/iWS: NES and iWS implement load balancing through a plug-in from Resonate. Resonate (www.resonate.com) compares resource usage with the current number of requests and balances the traffic distribution to reduce the incoming traffic volume to a particular server.

HP Apache: HP-UX Web Server Suite does not directly provide load balancing capability. A load balancing scheme can be set up using HP-UX Apache-based Web Server's mod_rewrite and mod_proxy modules or by building a third-party module such as mod_backhand.

A.8 Enterprise Capability

Multiple Web Server Instances

This is the capability to install and run multiple copies (separate binaries) of the web server. Multiple instances can be started and stopped independently.

NES/iWS: NES and iWS support installation of multiple web servers on the same box.

HP Apache: HP-UX Apache-based Web Server can run simultaneously on the same machine provided they use unique port numbers or different IP addresses. The `altroot.sh` utility can help move HP-UX Web Server Suite from its default location to prepare for multiple instances on the same system.

Internationalization (i18n)

Internationalization encompasses support for inputting characters in a users' native languages, handling files in popular encoding, using characters from a user's native language for file names and other items, printing out characters from a user's native language, displaying messages in a user's native language, formatting numbers, dates, money, etc that obeys the customs if a user's native culture.

"i18n" is a common abbreviation for internationalization because there are 18 letters between the first "i" and the last "n".

NES/iWS: i18n support is built into NES and iWS. iPlanet allows a character set assignment to the documents on the web server. The character set of a document is determined in part by the language in which it is written. A client's default character set can be overridden for a document, a set of documents, or a directory by selecting and entering a character set for that resource.

HP Apache: HP-UX Web Server Suite supports i18n through the Jakarta Project's i18n Tag Library for JSPs (HP-UX Tomcat-based Servlet Engine). It also comes with an index.html page translated into many different languages. HP-UX Apache-based Web Server error messages can also be customized into any language.

LDAP Directory Server Integration

NES/iWS: The LDAP Directory Server can be used as a general purpose LDAP directory. It can also check for valid (unrevoked) certificates. It an also can contain users and groups for authentication.

HP Apache: HP-UX Web Server Suite does not support general usage of an LDAP Directory Server. HP-UX Apache-based Web Server has LDAP integration for authentication only.

Virtual Servers

Virtual Servers are a method for running a single web server binary yet having the appearance and flexibility of running multiple web servers.

NES/iWS: NES does not implement virtual servers. iWS implements hardware virtual servers and software virtual servers. iWS Hardware Virtual Servers allow mapping of multiple IP addresses to multiple document roots. All hardware virtual servers share the same configuration information (i.e. is one virtual server has SSL enabled then all virtual servers must also have it enabled. iWS Software Virtual Servers give the ability to map a single IP address to multiple server names. Each software virtual server can have its own home page (i.e. to host multiple web sites from one IP address). Software virtual servers share the same configuration file but can have a separate document root and a separate CGI directory.

HP Apache: Apache virtual servers are called "IP-based Virtual Hosts" and "Name-based Virtual Hosts". An IP-based virtual host has its own unique IP address. It does not need to be identified by name so it will work older browsers that don't implement HTTP/1.1 protocol (specifically, don't send a `Host`: header). With name-based virtual hosts, many domain names share a single IP address and the virtual host selected depends on the `Host:` header.

Virtual Hosts can share configuration directives with the main web server or have their own configuration directives, http://httpd.apache.org/docs/vhosts/index.html. Virtual Hosts can use most of the directives available to the default (main) server.

Proxy Server

Proxy servers are intermediate servers that stand between a client and a remote server to make requests to the remote server on the client's behalf. Proxy servers can be used for load balancing or to isolate clients and servers to prevent unauthorized transactions.

NES/iWS: NES and iWS Proxy use iPlanet Proxy Server product for proxy functionality such as caching, forward proxy, reverse proxy, and logging.
HP Apache: HP Apache has built-in proxy server capability that provides caching, forward proxy, reverse proxy, and logging.

A.9 Developer Support

Database Connectivity

This is the ability for web applications to access a database.

NES/iWS: Supports native database connectivity to Oracle, IBM DB2, Sybase, and Informix using relational-database access capabilities built on native drivers.

HP Apache: Java-based applications or Java servlets can connect to a database through JDBC. Connectivity to Oracle 8.1.6 or later and mySQL are available through PHP scripts.

User Document Directories

A user document directory allows users to create and manage a home page and other documents in their own directory. Oftentimes, a user document directory resides in `/~user/public_html` and is accessed by the URL http://yourserver.com/~user/file.html. ~user is a user's home directory as specified in `/etc/passwd`.

NES/iWS: This feature is supported as described.

HP Apache: This feature is supported as described.

Server plug-ins

A server plug-in is add-on functionality that is written using a defined set of application programming interfaces (APIs). A plug-in extends and/or customizes the functionality of the web server.

NES/iWS: NES and iWS plug-ins are written using the iPlanet proprietary API (NSAPI).

HP Apache: Plug-ins are called "modules" in Apache. Modules are written using an open source API that parallels NSAPI in functionality. Apache also provides a tool for building and installing modules. Modules can be built shared libraries that are dynamically loaded at Apache startup time. These modules are called Dynamic Shared Objects (DSOs).

A.10 Application Integration

HP Virtual Vault

NES/iWS: Supported in versions 4.0 and earlier.

HP Apache: VirtualVault 4.5 uses Apache as its outside and inside web server. These implementations of Apache are not the same as the HP-UX Web Server Suite products.

Application Servers

NES/iWS: Supports Sun ONE Application Server

HP Apache: HP-UX Apache-based Web Server is certified with the BEA WebLogic Server. See the Getting More Information section for more information.

Appendix B Important HP-UX Web Server Suite Files

B.1 Configuration Files

HP-UX Apache-based Web Server	HP-UX Tomcat-based Servlet Engine
/opt/hpws/apache/conf	**/opt/hpws/tomcat/conf**
`httpd.conf`	`catalina.policy`
`httpd-std.conf`	`jk2.properties`
`cache.conf`	`server-noexamples.xml.config`
`mime.types`	`server.xml`
`magic`	`tomcat-users.xml`
`highperformance-std.conf`	`web.xml`
`highperformance.conf`	**/opt/hpws/apache/conf**
PHP	`mod_jk.conf`
/opt/hpws/apache/conf	`mod_jk2.conf`
`php.ini`	`workers.properties`
SSL	`workers2.properties`
/opt/hpws/apache/conf	
`ssl-std.conf`	
`ssl.conf`	
`ssl.crt/`	
`ssl.key/`	
LDAP	
/opt/hpws/apache/conf	
`ldap.conf`	
`stunnel.conf`	

B.2 Sample Files

HP-UX Web Server Suite has the following sample files:

Tomcat Servlets and JSPs
`/opt/hpws/tomcat/webapps/examples/` (Tomcat servlets and JSPs)
Servlets: http://yourserver.com:8081/examples/servlets
JSPs: http://yourserver.com:8081/examples/jsp

PHP
`/opt/hpws/apache/htdocs/test.php` (PHP script)
http://youserver.com/test.php

XML
```
/opt/hpws/tomcat/conf/apps-examples.xml
/opt/hpws/tomcat/conf/users/example-users.xml
```

Apache modules / Source templates
```
/opt/hpws/apache/build/examples/mod_echo.c
/opt/hpws/apache/build/examples/mod_example.c
/opt/hpws/apache/build/examples/mod_hello.cpp
```

Working Apache module
```
/opt/hpws/apache/modules/mod_echo.so
```

B.3 Bundled HP-UX Web Server Suite Documentation

All documentation that is bundled with the HP-UX Web Server Suite can be found under `/opt/hpws/hp_docs/` directory. There is a sub-directory for each (HP-UX Apache-based Web Server, HP-UX Tomcat-based Servlet Engine, HP-UX Webmin-based Admin, HP-UX XML Web Server Tools) component and the util directory. The easiest way to access documentation is to start each component and go to the home page.

B.4 HP-UX Apache-based Web Server Icons

HP-UX Apache-based Web Server has a separate directory for icons that includes a default set of .gif files:

`/opt/hpws/apache/icons`

B.5 Backup

HP-UX Web Server Suite follows the HP-UX defined standard methodology for delivering new user configurable files (henceforth referred to as just "file" or "files").

The new process is as follows:

1. The new files are delivered at:
   `/opt/hpws/<component>/newconfig/<absolute-path-to-file>`
2. During the preparation phase in an swinstall, a file from the previous installation of the component is moved to a safe location for a later review:
   `/opt/hpws/<component>/old/<absolute-path-to-file>`
3. The installation process then checks to see if the file is currently absent (implying either that this is the first installation or that the file has been removed), or that the working file is identical to the file at `/opt/hpws/<component>/old/<absolute-path-to-file>`. In both the cases, the file already on the system can be overwritten. The install process then copies the files from:
   `/opt/hpws/<component>/newconfig/<absolute-path-to-file>`
   to:
   `<absolute-path-to-file>`
   Otherwise, it leaves the file untouched. An administrator can review the files later to determine and execute the move of the file from newconfig location to the actual location manually.

Currently, even `/etc/rc.config.d/hpws_*conf` files are covered in the newconfig support.

The administrator can review the `/var/adm/sw/swagent.log` to identify the files that have been updated by the newconfig process. The following type of "NOTE:" message in the log file will assist the administrator to make this determination.

```
        NOTE:    A new version of "/opt/hpws/apache/conf/httpd.conf"
                 has been installed on the system.

   IMPORTANT: The files located at /opt/hpws/<component>/newconfig and
              /opt/hpws/<component>/old/<absolute-path-to-file> are available
              to the administrator for reference when manually editing the
              current files, but they should not be altered.  Doing so would
              make the file updates unpredictable.
```

B.6 Administration

HP-UX Webmin-based Admin has a separate subdirectory under the Apache root. HP-UX Web Server Suite utilities for automating administration tasks are located in the util subdirectory. Utilities specific to HP-UX Apache-based Web Server can be found under the apache/util subdirectory

`/opt/hpws/webmin`	(Webmin GUI tool)
`/opt/hpws/util/`	(utilities)
`/opt/hpws/apache/util/`	(Apache-specific utilities)

B.7 Installation files

HP-UX Web Server Suite is installed using swinstall, the same installation software used for installing other HP products.

B.8 Startup/Shutdown scripts

The HP-UX Apache-based Web Server (**httpd**) is started and stopped using the **apachectl** utility. This utility has various options for starting Apache. These options are described in the script itself and in the apachectl manual page, http://httpd.apache.org/docs-2.0/programs/apachectl.html.

HP-UX Webmin-based Admin and HP-UX Tomcat-based Servlet Engine each have their own startup and shutdown scripts.

HP-UX Web Server Suite:
HP-UX Apache-based Web Server
`/opt/hpws/apache/bin/apachectl start`
`/opt/hpws/apache/bin/apachectl startssl`
`/opt/hpws/apache/bin/apachectl stop`
HP-UX Tomcat-based Servlet Engine
`/opt/hpws/tomcat/bin/startup.sh`
`/opt/hpws/tomcat/bin/shutdown.sh`
HP-UX Webmin-based Admin
`/opt/hpws/webmin/webmin-init start`
`/opt/hpws/webmin/webmin-init stop`
Automatic startup scripts
`/etc/rc.config.d/hpws_apacheconf`
`/etc/rc.config.d/hpws_tomcatconf`
`/etc/rc.config.d/hpws_webminconf`
`/etc/rc.config.d/hpws_xmltoolsconf`

Appendix C Web Servers Component Reference

This table lists functionality on NES and iPlanet web servers and how it maps to HP-UX Web Server Suite.

Table C.1 NES, iWS and HP-UX Web Server Suite Functionality

NES/iWS Functionality	NES 3.x	iWS 4.x	HP-UX Web Server Suite (PA-RISC, IPF)
Core			
Static HTML	Yes	Yes	Yes
HTTP protocol	HTTP 1.0	HTTP 1.1	HTTP 1.0/1.1
Software Virtual Servers	Yes	Yes	Name-based Virtual Host
Hardware Virtual Servers	Yes	Yes	IP-based Virtual Host
Multiple instances	Yes	Yes	Yes
Alternate Installation Location		Yes	Yes
I18N		Yes	Jakarta open source I18N tag library
Data Model		32-bit	PA: 32-bit, IPF: 64-bit
Multi-Process	Yes	Yes	Yes
Multi-Threaded	1 request/proc HP-UX 10.20, 1 request/thrd HP-UX 11.x	1 request/thrd	1 request/thrd
SSI/SHMTL	Yes	Yes	Yes
SHTML <SERVLET> tag	No	Yes	JSP
LDAP Netscape Directory Server Integration	Yes	Yes LDAP SDK bundled	User authentication only
Application Development			
Java Servlets	Servlets 1.0.1 (JavaSoft)	Servlets 2.2, except WAR files	Servlets 2.3 (Tomcat 4.1.29)
Multiple servlet contexts	No	Yes	Yes
Java Server Pages (JSP)	JSP 0.92	JSP 1.1, JSP0.92 as legacy	JSP 1.2 (Tomcat 4.1.29)
JSP custom tag library		Yes. Proprietary implementation	Yes Tomcat
JDBC		JDBC 2.0	JDBC through Java 1.2,1.3
IDE for Java developers		Sun's Forte for Java	Webgain's VisualCafe
Java JDK	JDK 1.2	JDK 1.2	JDK 1.3 recommended
APIs for 3rd party	NSAPI	NSAPI	Apache 2.0 APR
API Accelerator Cache		Yes NSAPI	No, but Network Server Accelerator (NSAHTTP) is available at http://software.hp.com
Server-Side Java Script (LiveWire, LiveConnect)	Yes	JavaScript 1.4 LiveConnect 3	Servlets, PHP 4.3.4
Server-Side Java applets (HttpApplets)	Yes	Servlets	Servlets
Web App Interface (WAI) (CORBA/IIOP)	Inprise Object Request Broker (ORB)	Inprise Object Request Broker (ORB)	Servlets
CGI		Runs as server user	suEXEC
FastCGI		Extensions only	No
LiveWire Database Connectivity		Native connectivity to Oracle, Informix, IBM,	Oracle 8.1.6 and mySQL through PHP 4.3.4, JDBC through Java 1.2 and 1.3

		DB2,Sybase, JDBC	
Security			
(SSL2/SSL3) Client Authentication	Yes	Yes	mod_ssl (built-in), OpenSSL v.0.9.7c
128-bit encryption		Proprietary security libraries	Yes
Hardware Encryption		Fortezza	No
Public Key Cryptography Standard		PKCS #11	
.htaccess (Apache/NCSA-style), .nsconfig	Yes (plug-in)	Yes AuthGroupFile implemented differently. Not all directives implemented.	.htaccess
Chroot		Yes	Yes
User and Group Database Authorization	LDAP Directory Server	LDAP Directory Server Customized database access (Oracle)	auth_ldap 1.5.4 with OpenLDAP SDK 2.0.7
Server certificates	NES 3.x certificates	iWS4.x certificates Bundled tool for NES 3.x to iWS 4.x certificate migration	Bundled tool (certmig and test_certmig.sh) for iWS 4.x to Apache certificate migration (Not NES 3.x cert migration)
certmap.conf (map certificates to an identity)	Yes	Yes	No
Certification Revocation Lists (CRL)	Yes	CRL checking plug-in using NSAPI	Yes
Access Control Lists (ACLs)	Yes	Yes	<File> <Directory> <Location> <Limit> combined with Allow/Deny
Manageability			
Netscape server certificate	3.x certificates	Bundled migration utility for 3.x certificates	Bundled certmig migration tool for 4.x certificates
Monitoring/SNMP	Yes	Yes	No
Web-based GUI Administration	Netscape Server Manager	iPlanet Web Server Administrator	Webmin 1.070
Clustering		Yes	No
Distributed Administration		Administration Server	Webmin 1.070
Automatic installation		Yes	Yes
Integration with management software		CA/Unicenter, HP OpenView, IBM/Tivoli, TME,Sun Soltice	No
Content Management			
Verity Search (full-text and attribute search of HTML and non-HTML docs)		Yes	No
Multi-format collections in search	Yes	No	No
WebPub APIs		Yes	WebDAV (mod_dav)
WebPub Applet		Yes	WebDAV client
Version control and link management in web publishing	MKS	No	No
Agent email (MTA) Notification by email of document changes	Yes	No	No

Auto-catalog	Yes	No	No
FrontPage support			Yes, MSFP 2002 Server Extesions on PA-RISC only
Performance/Scalability			
Failover and High Availability		MC/Service Guard	MC/Service Guard
Load Balancing		libresonate.sl plug-in for Resonate (www.resonate.com)	Use built-in proxy server
SSL hardware accelerator		Yes	No
File Caching		Yes	Yes (mod_cache), shmem lib for SSL cache
Enterprise Capability			
Proxy server	iPlanet Proxy Server	iPlanet Proxy Server	built-in HP-UX Web Proxy support

Note:
The version numbers in the preceding table are correct for the HP-UX Web Server Suite 2.0.48 release in January 2004. For later releases see the HP-UX Web Server Suite Release Notes in the documentation directory, or on the Software Depot web site: software.hp.com/→Featured Products→HP-UX Apache-based Web Server or
http://software.hp.com/ and search for "HP-UX Web Server Suite"

Appendix D Open Source Apache versus HP-UX Web Server Suite

Table D.1 Components & Features of open source Apache (ASF Apache) and HP-UX Web Server Suite

Components & Features	Standard Open Source [1]	HP-UX Web Server Suite PA / IPF [2]
HP-UX Apache-based Web Server (Released Jan. 2004)		2.0.48
Apache Web Server	2.0.48	2.0.48
suEXEC	yes	yes
IPv6**	yes	HP-UX 11i (11.11), HP-UX 11i Version 2 (11.23)
*Auto-restart Apache, Tomcat, or Webmin		yes
*Shared Memory Caching	yes	apr_shm
*Support for C++ modules		yes
*altroot.sh		yes
*cache_util.sh		yes
*ports.sh		yes
Security		
mod_ssl	Built-in	Built-in
OpenSSL		0.9.7c
auth_ldap		1.5.4
Stunnel		4.04
*Chroot		yes
*Certmig and test_certmig.sh		PA only
*mkcert.sh		yes
*web proxy		yes
Miscellaneous		
mod_dav/mod_dav_fs	yes	yes
mod_proxy	yes	yes
Scripting		
mod_perl		1.99_10
PHP		4.3.4
PHP with Oracle extension		yes
HP-UX Tomcat-based Servlet Engine		
Tomcat Servlet Container	yes	4.1.29
Apache connector to Tomcat		mod_jk 1.2.5, mod_jk2
HP-UX Webmin-based Admin		
Webmin	yes	1.070
HP-UX XML Web Server Tools		
Xerces-J	yes	2.5.0
Xalan-J	yes	2.5.1
Cocoon	yes	2.0.4
FOP	yes	0.20.5
Batik	yes	1.5

[1] "Standard Open Source" indicates that if you were to go to the Open Source Community (e.g., http://httpd.apache.org for Apache) and only download the standard distribution and build the product, these features would be included by default.

[2] See table within the Using this Guide section detailing HP-UX Platform and product availability for both PA-RISC and IPF.

* Indicates HP added feature. Not available in Open Source.

** IPv6 is only available on PA-RISC 11i (11.11) with IPv6 product (TOUR) and 11i Version 1.6 (11.23). Since HP-UX Web Server Suite is dependent on products such as Java and Perl, some components are not completely supported. See IPv6 section for more information.

The version numbers in the preceding table are correct for HP-UX Web Server Suite 2.0.48 January 2004 release. For later releases see the Release Notes in the documentation directory, or on the Software Depot web site: software.hp.com/→ Featured Products→HP-UX Apache-based Web Server or http://software.hp.com/ and search for "HP-UX Web Server Suite"