



IBM HTTP Server - Powered by Apache

Version 9



IBM HTTP Server - Powered by Apache

Version 9

How to send your comments

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. To send comments on PDF books, you can email your comments to: wasdoc@us.ibm.com.

Your comment should pertain to specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. Be sure to include the document name and number, the IBM HTTP Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer. When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about your comments.

Using this PDF

Because the content within this PDF is designed for an online information center deliverable, you might experience broken links. You can expect the following link behavior within this PDF:

- Links to Web addresses beginning with <http://> work.
- Links that refer to specific page numbers within the same PDF book work.
- The remaining links will not work. You receive an error message when you click them.

Contents

Tables	v
-------------------------	----------

Chapter 1. Product overview and quick start **1**

What is new in this release	1
Key differences from the Apache HTTP Server	2

Chapter 2. Migrating and installing IBM HTTP Server **5**

Installing IBM HTTP Server on distributed systems	5
Installing IBM HTTP Server using the GUI	7
Migrating from previous version of IBM HTTP Server	16
Running multiple instances of IBM HTTP Server from a single install	19
Installing fix packs on IBM HTTP Server using the Installation Manager GUI	21
Installing IBM HTTP Server silently	21
Uninstalling IBM HTTP Server using the GUI	33
Uninstalling fix packs from IBM HTTP Server using the Installation Manager GUI	38
Installing and configuring IBM HTTP Server on the z/OS V2R2 system	39
Installing and configuring IBM HTTP Server on the z/OS V2R1 system	43
Migrating from IBM HTTP Server V5.3 for z/OS	47
IBM HTTP Server V5.3 for z/OS: Part 1: Planning	47
IBM HTTP Server V5.3 for z/OS: Part 2: Installing	49
IBM HTTP Server V5.3 for z/OS: Part 3: Using	49
IBM HTTP Server V5.3 for z/OS: Part 4: Basic configuration	51
IBM HTTP Server V5.3 for z/OS: Part 5: Advanced configuration	56
IBM HTTP Server V5.3 for z/OS: Part 6: Programming	60
Running multiple instances of IBM HTTP Server from a single install	61

Chapter 3. Administering and configuring IBM HTTP Server **65**

Performing required z/OS system configurations	65
Starting and stopping IBM HTTP Server	69
Using the administrative console to start IBM HTTP Server	70
Using apachectl commands to start IBM HTTP Server	71
Using Windows services to start IBM HTTP Server	72
Using JCL procedures to start IBM HTTP Server on z/OS	74
Configuring IBM HTTP Server	77

Apache modules (containing directives) supported by IBM HTTP Server	78
Apache programs supported by IBM HTTP Server	85
Apache APR and APR-util libraries supported by IBM HTTP Server	86
Apache MPM and addressing modes supported by IBM HTTP Server	87
IPv4 and IPv6 configuration for Windows operating systems	87
Enabling IBM HTTP Server for FastCGI applications	88
Learn about FastCGI	89
FastCGI directives	89
Managing IBM HTTP Server remotely with the WebSphere Application Server administrative console	100
Extending IBM HTTP Server functionality with third-party plug-in modules	101
Viable compilers for Apache and third-party plug-in modules	102
Build method options for dynamic modules	103
Considerations for building dynamic modules on Windows platforms	103
Considerations for building dynamic modules on Unix platforms	104
Configuring IBM HTTP Server for SMF recording	104
Classifying HTTP requests for WLM (z/OS operating systems)	105
WLM directives for IBM HTTP Server	107

Chapter 4. Administering and configuring the administration server . **109**

Starting and stopping the IBM HTTP Server administration server	109
Protecting access to the IBM HTTP Server administration server	110
Enabling access to the administration server using the htpasswd utility	110
Running the setupadm command for the administration server	111
Setting permissions manually for the administration server	113

Chapter 5. Securing IBM HTTP Server **115**

Securing IBM HTTP Server	115
Configure SSL between the IBM HTTP Server Administration Server and the deployment manager	116
Securing with SSL communications	118
Secure Sockets Layer (SSL) protocol	122
SSL directive considerations	127
Authentication	127
Encryption	128
Secure Sockets Layer environment variables	129
SSL directives	134

Setting advanced SSL options	160
Choosing the level of client authentication.	160
Server Name Indication	161
Choosing the type of client authentication protection	162
Defining SSL for multiple-IP virtual hosts	166
Setting up a reverse proxy configuration with SSL.	166
IBM HTTP Server certificate management	167
Managing keys with the IKEYMAN graphical interface (Distributed systems).	170
Starting the Key Management utility user interface	171
Working with key databases	171
Changing the database password.	172
Creating a new key pair and certificate request	173
Importing and exporting keys	174
Listing certificate authorities	176
Certificate expiration dates	177
Creating a self-signed certificate	177
Receiving a signed certificate from a certificate authority	178
Displaying default keys and certificate authorities	179
Storing a certificate authority certificate	180
Storing the encrypted database password in a stash file	180
Managing keys with the command line (Distributed systems).	181
Using the gskcapicmd command	182
Key Management Utility command-line interface (gskcmd) syntax	183
Creating a new key database using the command-line interface	186
Managing the database password using the command line	187
Creating a new key pair and certificate request	188
Importing and exporting keys using the command line	189
Creating a self-signed certificate	191
Receiving a signed certificate from a certificate authority	192
Displaying default keys and certificate authorities	193
Storing a certificate authority certificate	194
Storing the encrypted database password in a stash file	195
Managing keys with the native key database gskkyman (z/OS systems)	195
Getting started with the cryptographic hardware for SSL (Distributed systems)	196

Cryptographic hardware for Secure Sockets Layer	196
Using IKEYMAN to store keys on a PKCS11 device.	198
Configuring IBM HTTP Server to use nCipher and Rainbow accelerator devices and PKCS11 devices	200
Authenticating with LDAP on IBM HTTP Server using mod_ldap	201
Converting your directives from mod_ibm_ldap to mod_ldap.	203
Authenticating with SAF on IBM HTTP Server (z/OS systems).	213
SAF directives	214

Chapter 6. Troubleshooting and support: IBM HTTP Server **223**

Troubleshooting IBM HTTP Server	223
Known problems on Windows platforms	223
Known problems on z/OS platforms	225
Known problems with hardware cryptographic support on AIX.	225
Symptoms of poor server response time	226
Hints and tips for managing IBM HTTP Server using the administrative console	226
Could not connect to IBM HTTP Server administration server error	228
Experiencing an IBM HTTP Server Service logon failure on Windows operating systems	228
Viewing error messages for a target server that fails to start	229
Cache messages	230
Configuration messages	230
Handshake messages.	232
SSL initialization messages	240
I/O error messages	248
SSL stash utility messages	249

Appendix. Accessibility **255**

Accessibility features	255
Consult assistive technologies	255
Keyboard navigation of the user interface	255
Dotted decimal syntax diagrams	255

Notices **259**

Index **261**

Tables

1. Apache modules	79	8. Variables for HTTPS_CIPHER in Secure Sockets Layer V2.	131
2. MPM and addressing modes.	87	9. Attribute values for the SSLClientAuthGroup directive	145
3. Bytes ranges and their descriptions for SMF records.	105	10. Attribute values for the SSLClientAuthRequire directive	147
4. Signer Certificate information	118	11. Client authentication level	160
5. Types of access and mechanisms for SSL environment variables	130	12. Medium and high strength TLS ciphers	164
6. SSL handshake environment variables	130	13. Other TLS ciphers	165
7. Variables for HTTPS_KEYSIZE and HTTPS_SECRETKEYSIZE in Secure Sockets Layer V3 and Transport Layer Security V1.	131	14. Actions for gskcmd command objects	183
		15. LDAP configuration directives conversion	203

Chapter 1. Product overview and quick start

Distributed operating systems

z/OS

This section provides shortcuts to information for obtaining a high level understanding of the product.

What is new in this release

IBM® HTTP Server contains some new functions. Review this topic to find out about what is new in this release.

Distributed operating systems

Apache HTTP Server modules after V8.5.5

The following Apache HTTP Server modules have been added since V8.5.5:

- mod_proxy_fcgi
- mod_substitute
- mod_lua
- mod_authn_certificate
- mod_remoteip
- mod_macro

Distributed operating systems

Enhancements to existing modules

- mod_ibm_ssl has added support for certificate selection, based on TLS Server Name Indication (SNI) extensions.
- A new configuration option, `<if>`, allows complex conditions to enclose Apache directives.
- A number of directives now support a robust expression language.
- You can now group authorization directives (`Require`) in `<RequireAny>` or `<RequireAll>` and parameters are interpolated with the expression language.
- The event MPM is available on Linux.
- The `ProxyRemote` directive now works with `SSLProxyEngine`.
- The bundled PCRE is updated.
- Logging changes.
- You can specify the `LogLevel` directive per-directory, per-module, and per-request.
- Log levels TRACE1 through TRACE8 have been added.
- Many Apache HTTP Server messages now include an AHnnnnn identifier.
- The error log format is user configurable (see `ErrorLogFormat`).
- Higher precision timestamps are available in the error log.

Distributed operating systems

Logging changes

- You can specify the `LogLevel` directive per-directory, per-module, and per-request.

- Log levels TRACE1 through TRACE8 have been added.
- Many Apache HTTP Server messages now include an AHnnnnn identifier.
- The error log format is user configurable (see ErrorLogFormat).
- Higher precision timestamps are available in the error log.

Distributed operating systems

Default changes

The following default changes apply to IBM HTTP Server Version 9:

- RC4 disabled by default.
- The default configuration has reduced the `mod_mpmstats ReportInterval` directive from 600 to 300.
- **z/OS** On z/OS platforms, `mod_mpmstats` now displays the number of keepalive requests, even though they do not occupy a thread. In prior releases, `mpmstats` would always show a value of zero.
- **z/OS** On z/OS platforms, TLS1.1, TLS1.2, and ECC ciphers are enabled by default if ICSF is configured at startup.
- **z/OS** On z/OS platforms, `mod_deflate.so` uses offload by default and `mod_deflate_z.so` is no longer provided.

AIX HP-UX Solaris Linux z/OS

Modules that have been removed

The following modules have been removed from IBM HTTP Server Version 9:

- The `mod_mem_cache` module is provided for transition purposes only and has been removed from Apache HTTP Server 2.4.
- `mod_ibm_ldap`
- WebSphereCE modules (`mod_proxy_balancer` and `mod_proxy_ajp`)
- 32-bit builds have been removed for AIX, Linux on PPC, Linux on S390, Solaris on SPARC.

Related information:

Chapter 2, “Migrating and installing IBM HTTP Server,” on page 5

Learn how to establish the product in new and existing environments, including planning, preparing for, and completing product installations.

Key differences from the Apache HTTP Server

Distributed operating systems

z/OS

This section takes a high-level look at the main differences between IBM HTTP Server and the Apache HTTP Server.

IBM HTTP Server is based on Apache HTTP Server 2.4.9, with additional fixes. For behaviors affected by changes in Apache HTTP Server 2.2.9 and later, refer to the bundled copy of the Apache HTTP Server manual which will describe these changes, where applicable, in terms of IBM HTTP Server maintenance levels. See related reference: Apache modules (containing directives) supported by IBM HTTP Server for more information.

The Apache Web server can be built with many different capabilities and configuration options. IBM HTTP Server includes a set of features from the available options. For information about Apache Web server features supported in

IBM HTTP Server, see the information center topics about Apache modules (containing directives), programs, Apache Portable Runtime (APR) and APR-util libraries, and Multi-processing module (MPM) and addressing modes.

Related reference:

“Apache modules (containing directives) supported by IBM HTTP Server” on page 78

This section provides information on Apache modules that are supported by IBM HTTP Server. The directives defined within the supported Apache modules can be used to configure IBM HTTP Server.

“Apache programs supported by IBM HTTP Server” on page 85

This section provides information on Apache programs that are supported by IBM HTTP Server. These supported Apache programs can be used to configure IBM HTTP Server.

“Apache APR and APR-util libraries supported by IBM HTTP Server” on page 86

This section provides information about the Apache Portable Runtime (APR) and APR-util libraries that are supported by IBM HTTP Server. IBM HTTP Server supports only the APR and APR-util libraries installed with the product. Copies of the libraries cannot be substituted.

“Apache MPM and addressing modes supported by IBM HTTP Server” on page 87

This section provides information about Apache Multi-processing module (MPM) and addressing modes supported by IBM HTTP Server.

Chapter 2. Migrating and installing IBM HTTP Server

Learn how to establish the product in new and existing environments, including planning, preparing for, and completing product installations.

Installing IBM HTTP Server on distributed systems

Distributed operating systems

IBM Installation Manager is a common installer for many IBM software products that you use to install, update, roll back, and uninstall IBM HTTP Server.

Before you begin

Restrictions:

- Before you can successfully install IBM HTTP Server, ensure that your environment meets the prerequisites for the application server. For more information, see the Preparing the operating system for product installation topic.
- **Linux** For any Linux system that is enabled for Security Enhanced Linux (SELinux), such as Red Hat Enterprise Linux Version 5 or SUSE Linux Enterprise Server Version 11, you must identify the Java™ shared libraries in the Installation Manager 1.4.2 or later installation image to the system. Also, you must identify the Java shared libraries in the Installation Manager 1.4.2 or later installation after it has been installed. For example:

```
chcon -R -t texrel_shlib_t ${IM_Image}/jre_5.0.3.sr8a_20080811b/jre/bin
chcon -R -t texrel_shlib_t ${IM_Install_root}/eclipse/jre_5.0.3.sr8a_20080811b/jre/bin
```

About this task

Complete one of these procedures to install, update, roll back, or uninstall IBM HTTP Server using Installation Manager.

Procedure

- “Installing IBM HTTP Server using the GUI” on page 7
- “Installing IBM HTTP Server silently” on page 21
- “Installing fix packs on IBM HTTP Server using the Installation Manager GUI” on page 21
- “Uninstalling fix packs from IBM HTTP Server using the Installation Manager GUI” on page 38
- “Uninstalling IBM HTTP Server using the GUI” on page 33

Results

Notes on logging and tracing:

- An easy way to view the logs is to open Installation Manager and go to **File > View Log**. An individual log file can be opened by selecting it in the table and then clicking the **Open log file** icon.
- Logs are located in the logs directory of Installation Manager's application data location. For example:

- **Windows** **Administrative installation:**

C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager

- **Windows** **Non-administrative installation:**

C:\Documents and Settings\user_name\Application Data\IBM\Installation Manager

- **AIX** **HP-UX** **Linux** **Solaris** **Administrative installation:**

/var/ibm/InstallationManager

- **AIX** **HP-UX** **Linux** **Solaris** **Non-administrative installation:**

user_home/var/ibm/InstallationManager

- The main log files are time-stamped XML files in the logs directory, and they can be viewed using any standard Web browser.
- The log.properties file in the logs directory specifies the level of logging or tracing that Installation Manager uses.

Notes on troubleshooting:

- **HP-UX** By default, some HP-UX systems are configured to not use DNS to resolve host names. This could result in Installation Manager not being able to connect to an external repository.

You can ping the repository, but nslookup does not return anything.

Work with your system administrator to configure your machine to use DNS, or use the IP address of the repository.

- In some cases, you might need to bypass existing checking mechanisms in Installation Manager.

- On some network file systems, disk space might not be reported correctly at times; and you might need to bypass disk-space checking and proceed with your installation.

To disable disk-space checking, specify the following system property in the config.ini file in *IM_install_root/eclipse/configuration* and restart Installation Manager:

```
cic.override.disk.space=sizeunit
```

where *size* is a positive integer and *unit* is blank for bytes, k for kilo, m for megabytes, or g for gigabytes. For example:

```
cic.override.disk.space=120 (120 bytes)
cic.override.disk.space=130k (130 kilobytes)
cic.override.disk.space=140m (140 megabytes)
cic.override.disk.space=150g (150 gigabytes)
cic.override.disk.space=true
```

Installation Manager will report a disk-space size of Long.MAX_VALUE. Instead of displaying a very large amount of available disk space, N/A is displayed.

- To bypass operating-system prerequisite checking, add `disableOSPrereqChecking=true` to the config.ini file in *IM_install_root/eclipse/configuration* and restart Installation Manager.

If you need to use any of these bypass methods, contact IBM Support for assistance in developing a solution that does not involve bypassing the Installation Manager checking mechanisms.

- Installation Manager might display a warning message during the uninstallation process.

Uninstalling IBM HTTP Server using Installation Manager requires that the data repositories remain valid and available.

A warning message is displayed by Installation Manager to alert you when repositories are not available or connected. A similar warning message might display after you add or modify data repository connection preferences in Installation Manager.

If Installation Manager detects missing data repositories or fails to connect to repositories during the uninstallation process, complete the following actions:

1. Click **Cancel** to end the uninstallation task.
 2. Select **File > Preferences > Repositories**, and add the appropriate data repositories that you can connect to successfully.
 3. Exit Installation Manager.
 4. Restart Installation Manager.
 5. Uninstall IBM HTTP Server.
- For more information on using Installation Manager, read the IBM Installation Manager Information Center.

Read the release notes to learn more about the latest version of Installation Manager. To access the release notes, complete the following task:

- **Windows** Click **Start > Programs > IBM Installation Manager > Release Notes®**.
- **AIX** **HP-UX** **Linux** **Solaris** Go to the documentation subdirectory in the directory where Installation Manager is installed, and open the `readme.html` file.

Installing IBM HTTP Server using the GUI

Distributed operating systems

You can use the Installation Manager GUI to install IBM HTTP Server.

Before you begin

Install Installation Manager:

1. Perform one of the following procedures:
 - If you want to use the Installation Manager that is included with this product, perform the following actions:
 - a. Obtain the necessary files.

There are two basic options for obtaining and installing Installation Manager and the product.

 - **Download the files from the Beta site, and use local installation**

You can download the necessary product repositories from the Beta site.

 - 1) Download the files from the Beta site.
 - 2) Install Installation Manager on your system.

You can install Installation Manager using a file obtained from the Beta site.
 - 3) Use Installation Manager to install the product from the downloaded repositories.
 - **Access the live repositories, and use web-based installation**

You can install the product from the web-based repositories.

 - 1) Install Installation Manager on your system.

You can install Installation Manager using a file obtained from the Beta site.

- 2) Use Installation Manager to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product.

There are three basic options for obtaining and installing Installation Manager and the product.

– **Access the physical media, and use local installation**

You can access Installation Manager and the product repositories on the product media. You can install Installation Manager on your system and use it to install the product from the product repositories on the media.

– **Download the files from the Passport Advantage® site, and use local installation**

Licensed customers can download Installation Manager as well as the necessary product repositories from the Passport Advantage site. You can then install Installation Manager on your system and use it to install the product from the repositories.

– **Download a file from the Installation Manager website, and use web-based installation**

You can download and unpack a compressed file containing Installation Manager from the IBM Installation Manager website. You can then install Installation Manager on your local system and use it to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product.

- b. Change to the location containing the Installation Manager installation files, and run one of the following commands:

Administrative installation:

- **Windows** install.exe
- **AIX** **HP-UX** **Linux** **Solaris** ./install

Non-administrative installation:

- **Windows** userinst.exe
- **AIX** **HP-UX** **Linux** **Solaris** ./userinst

Group-mode installation:

- **AIX** **HP-UX** **Linux** **Solaris** ./groupinst

Notes on group mode:

- Group mode allows multiple users to use a single instance of IBM Installation Manager to manage software packages.
This does not mean that two people can use the single instance of IBM Installation Manager at the same time.
- **Windows** Group mode is not available on Windows operating systems.
- If you do not install Installation Manager using group mode, you will not be able to use group mode to manage any of the products that you install later using this Installation Manager.
- Make sure that you change the installation location from the default location in the current user's home directory to a location that is accessible by all users in the group.
- Set up your groups, permissions, and environment variables as described in the Group mode road maps in the IBM Installation Manager Information Center before installing in group mode.
- For more information on using group mode, read the Group mode road maps in the IBM Installation Manager Information Center.

The installer opens an **Install Packages** window.

- c. Make sure that the Installation Manager package is selected, and click **Next**.
 - d. Accept the terms in the license agreements, and click **Next**.
The program creates the directory for your installation.
 - e. Click **Next**.
 - f. Review the summary information, and click **Install**.
 - If the installation is successful, the program displays a message indicating that installation is successful.
 - If the installation is not successful, click **View Log File** to troubleshoot the problem.
- If you already have the Installation Manager Version 1.5.2 Beta Installation Manager Version 1.5.2 or later installed on your system and you want to use it to install and maintain the product, obtain the necessary product files from the physical media or the web.

There are two basic options for installing the product.

– **Download the files from the Beta site, and use local installation**

You can download the necessary product repositories from the Beta site.

- a. Download the product repositories from the Beta site.
- b. Use Installation Manager to install the product from the downloaded repositories.

– **Access the live repositories, and use web-based installation**

You can install Installation Manager on your local system and use it to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product.

Whenever possible, you should use the remote web-based repositories so that you are accessing the most up-to-date installation files.

There are three basic options for installing the product.

– **Access the physical media, and use local installation**

You can access the product repositories on the product media. Use your existing Installation Manager to install the product from the product repositories on the media.

– **Download the files from the Passport Advantage site, and use local installation**

Licensed customers can download the necessary product repositories from the Passport Advantage site. You can then use your existing Installation Manager to install the product from the repositories.

– **Access the live repositories, and use web-based installation**

You can install Installation Manager on your local system and use it to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product.

Whenever possible, you should use the remote web-based repositories so that you are accessing the most up-to-date installation files.

2. Add the product repository to your Installation Manager preferences.

a. Start Installation Manager.

b. Go to **File > Preferences**.





c. Select **Repositories**.

d. Perform the following actions:

1) Click **Add Repository**.

2) Enter the path to the repository.config file in the location containing the repository files.

For example:

-  C:\repositories*product_name*\local-repositories
-     /var/repositories/*product_name*/local-repositories

or

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

3) Click **OK**.

e. Deselect any locations listed in the Repositories window that you will not be using.

f. Click **Apply**.

g. Click **OK**.

h. Click **File > Exit** to close Installation Manager.

About this task

Complete this procedure to use the Installation Manager GUI to install IBM HTTP Server.

Procedure

1. Start Installation Manager.
2. Click **Install**.

Note: If you are prompted to authenticate, use the IBM ID and password that you registered with on the program website.

Installation Manager searches its defined repositories for available packages.

3. In the **Install Packages** window, complete the appropriate actions.
 - a. Select **IBM HTTP Server** and the appropriate version.

Note: If you are installing the trial version of this product, select **IBM HTTP Server Trial**.

If you already have IBM HTTP Server installed on your system, a message displays indicating that IBM HTTP Server is already installed. To create another installation of IBM HTTP Server in another location, click **Continue**.

- b. Click **Next**.

Note: If you try to install a newer level of IBM HTTP Server with a previous version of Installation Manager, Installation Manager might prompt you to update to the latest level of Installation Manager when it connects to the repository. Update to the newer version before you continue if you are prompted to do so. Read *Installing updates in the Installation Manager information center* for information about automatic updates.

4. Accept the terms in the license agreements, and click **Next**.
5. On the Location panel, specify the installation root directory for the product binaries, which are also referred to as the core product files or system files. The panel also displays the shared resources directory and disk-space information.

The core product files do not change unless you install maintenance.

Restrictions:

- Deleting the default target location and leaving an installation-directory field empty prevents you from continuing.
- Do not use symbolic links as the destination directory. Symbolic links are not supported.
- Be careful when using spaces in the name of the installation directory. Some operating systems allow spaces in the specification of the installation directory. Other operating systems do not allow spaces. For example:

C:\Program Files (x86)	Windows supported
/opt/IBM/HTTP Server	Unix NOT supported
/opt/IBM/HTTPServer	Unix supported

- Do not use a semicolon in the directory name.

IBM HTTP Server cannot install properly if the target directory includes a semicolon.

Windows A semicolon is the character used to construct the class path on Windows systems.

- **Windows** The maximum path length on the Windows XP, Windows Vista, and Windows 7 operating systems is 60 characters.

6. Click **Next**.

7. **AIX** **Linux** **Solaris** If you are installing on a 64-bit system, choose between a 32-bit or 64-bit HTTP Server environment and click **Next**.

Notes:

- This option displays only if you are installing on a 64-bit system.
 - This does not apply to Solaris x86 64-bit systems.
 - You must select one of the two options.
 - You cannot modify this installation later and change this selection.
8. Click **Next** to display the Configuration for IBM HTTP Server panel,
 9. **Windows** On the Configuration for IBM HTTP Server panel, specify your Web server configuration.
 - Specify a port number on which IBM HTTP Server will communicate. The default port is 80.
 - Choose whether to use a Windows service to run IBM HTTP Server.

Note: You have the option to create a Windows service for IBM HTTP Server on this panel. You can configure the services to run as local system account or a user ID that you specify. The user ID requires the following advanced user rights:

- Act as part of the operating system
- Log on as a service

Important: If you do not select **Run IBM HTTP Server as a Windows Service**, this instance of IBM HTTP Server cannot be stopped or started by the WebSphere Application Server administrative console. At any time after installation, you can create a new service by running the following command:

```
ihp_root/bin/httpd.exe -n new_service_name -k install
```

and then updating the web server definition in the administrative console to reflect the new service name.

- Determine if your startup type will be automatic or manual.
10. **AIX** **HP-UX** **Solaris** On the Configuration for IBM HTTP Server panel, specify your Web server configuration.

Specify a port number on which IBM HTTP Server will communicate. The default port is 80.
 11. Click **Next**.
 12. Review the summary information, and click **Install**.
 - If the installation is successful, the program displays a message indicating that installation is successful.

Note: The program might also display important post-installation instructions as well.

 - If the installation is not successful, click **View Log File** to troubleshoot the problem.
 13. Click **Finish**.
 14. Click **File > Exit** to close Installation Manager.

Results

If the installation is successful, the IBM HTTP Server product is installed and the log file is located in the `/logs/install/` directory. However, if the product installation fails, see the `log.txt` file in either the `/logs/install/` directory or the `$USER/ihslogs/` directory.

What to do next

Set up IBM HTTP Server administration authentication, using the `htpasswd` utility.

You can get started using Secure Sockets Layer (SSL) connections by making only a few configuration changes, as described in [Securing with SSL communications](#).

AIX **Windows** You can configure the Fast Response Cache Accelerator to boost performance.

You can also make other configuration changes with Apache directives.

Mounting CD-ROMS on AIX, HP-UX, Linux and Solaris systems

AIX **HP-UX** **Linux** **Solaris**

This section describes how to mount the CD-ROM for IBM HTTP Server on AIX®, HP-UX, Linux and Solaris operating systems.

Before you begin

After inserting a CD-ROM into a drive, some operating systems require you to mount the drive.

About this task

Use these procedures to mount the product discs for IBM HTTP Server.

Procedure

- **AIX** **Mount the CD-ROM using the System Management Interface Tool (SMIT) as follows:**
 1. Log in as a user with root authority.
 2. Insert the CD-ROM in the drive.
 3. Create a CD-ROM mount point by entering the `mkdir -p /cdrom` command, where `cdrom` represents the CD-ROM mount point directory.
 4. Allocate a CD-ROM file system using SMIT by entering the `smit storage` command.
 5. After SMIT starts, click **File Systems > Add / Change / Show / Delete File Systems > CDRom File Systems > Add CDRom File System**.
 6. In the Add a File System window:
 - Enter a device name for your CD-ROM file system in the **DEVICE Name** field. Device names for CD-ROM file systems must be unique. If there is a duplicate device name, you may need to delete a previously-defined CD-ROM file system or use another name for your directory. The example uses `/dev/cd0` as the device name.
 - Enter the CD-ROM mount point directory in the **MOUNT POINT** window. In our example, the mount point directory is `/cdrom`.

- In the **Mount AUTOMATICALLY at system restart** field, select **yes** to enable automatic mounting of the file system.
 - Click **OK** to close the window, then click **Cancel** three times to exit SMIT.
7. Next, mount the CD-ROM file system by entering the **smit mountfs** command.
 8. In the Mount a File System window:
 - Enter the device name for this CD-ROM file system in the **FILE SYSTEM name** field. In our example, the device name is `/dev/cd0`.
 - Enter the CD-ROM mount point in the **Directory over which to mount** field. In our example, the mount point is `/cdrom`.
 - Enter `cdrfs` in the **Type of Filesystem** field. To view the other kinds of file systems you can mount, click **List**.
 - In the **Mount as READ-ONLY system** field, select **yes**.
 - Accept the remaining default values and click **OK** to close the window.

Your CD-ROM file system is now mounted. To view the contents of the CD-ROM, place the disk in the drive and enter the `cd /cdrom` command where `cdrom` is the CD-ROM mount point directory.

- **HP-UX** **Mount the CD-ROM.** Because WebSphere® Application Server contains several files with long file names, the mount command can fail. The following steps let you successfully mount your WebSphere Application Server product CD-ROM.

1. Log in as a user with root authority.
2. In the `/etc` directory, add the following line to the `pfs_fstab` file:


```
/dev/dsk/c0t2d0 mount_point pfs-rrip ro,hard
```

where *mount_point* represents the mount point of the CD-ROM.

3. Start the `pfs` daemon by entering the following commands (if they are not already running):

```
/usr/sbin/pfs_mountd &
/usr/sbin/pfsd 4 &
```

4. Insert the CD-ROM in the drive and enter the following commands:

```
mkdir /cdrom
/usr/sbin/pfs_mount /cdrom
```

The `/cdrom` variable represents the mount point of the CD-ROM.

5. Log out.

- **Linux** **Mount the CD-ROM using the following steps.**

1. Log in as a user with root authority.
2. Insert the CD-ROM in the drive and enter the following command:

```
mount -t iso9660 -o ro /dev/cdrom /cdrom
```

The `/cdrom` variable represents the mount point of the CD-ROM.

3. Log out.

Some window managers can automatically mount your CD-ROM for you. Consult your system documentation for more information.

- **Solaris** **Mount the CD-ROM using the following steps.**

1. Log in as a user with root authority.
2. Insert the CD-ROM into the drive.
3. If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

The `/cdrom/unnamed_cdrom` variable represents the CD-ROM mount directory and the `/dev/dsk/c0t6d0s2` represents the CD-ROM drive device.

If you are mounting the CD-ROM drive from a remote system using NFS, the CD-ROM file system on the remote machine must be exported with root access. You must also mount that file system with root access on the local machine.

If the Volume Manager (vold) is running on your system, the CD-ROM is automatically mounted as:

```
/cdrom/unnamed_cdrom
```

4. Log out.

What to do next

Return to the installation procedure to continue.

Creating multiple instances of IBM HTTP Server on Windows operating systems

Windows

On Windows operating systems, you can create multiple instances of IBM HTTP Server by manually creating additional service names.

Before you begin

About this task

When you install IBM HTTP Server, you create one IBM HTTP Server as a Windows service with a default name. If you need to run with more than one IBM HTTP Server instance, you can manually create additional service names.

Procedure

1. Install a new service name. Use the `httpd.exe` program, which is located in the `bin` directory of the IBM HTTP Server installation. The command syntax for installing a new service name is:

```
httpd -k install -n <new_service_name> -f
    <path_to_new_configuration_file>
```

This command allows you to associate a unique configuration file with each service name.

2. Specify different IP addresses or ports in the Listen directives of each configuration file and specify different log file names.
3. Optional: Change settings of the new service using the Windows Services control panel. The new service name will have "Log On" set to "Local System Account" and will have "Startup Type" set to "Automatic". You can change these default settings using the Windows Services control panel. It might be necessary to change the "Log On" setting of the new service name to match the "Log On" of the main installation in order to ensure that file permissions will allow the new service name to run.
4. Disable the Fast Response Cache Accelerator (FRCA). When running multiple instances of IBM HTTP Server, you must disable the FRCA (AFPA directives) in all configuration files.

What to do next

After creating a new service name, you can add it to the WebSphere Administration Server administrative console by creating a new Web server definition and specifying the new service name and the path to the new configuration file.

The syntax for uninstalling an existing service name is:

```
httpd -k uninstall -n <service_name>
```

Migrating from previous version of IBM HTTP Server

This section provides information about upgrading from a previous version of IBM HTTP Server.

About this task

IBM HTTP Server Version can coexist with earlier versions if you install the most recent version into a different directory. You can also upgrade earlier versions of IBM HTTP Server by installing the recent version into the directory where an earlier version of IBM HTTP Server is located. Using the same installation path on the same system for the new IBM HTTP Server version preserves the validity of the WebSphere Application Server web server definition (with a minor exception for a Windows server where the service name needs to be modified in the server definition to the service name used for the new version).

This procedure covers migrating from the previous major release. If you are migrating from an IBM HTTP Server earlier than a previous major release, read the product documentation for the interim IBM HTTP Server version(s) and review the migration information because there might be additional steps you need to complete that are not documented here.

Procedure

- Upgrade IBM HTTP Server from your previous installation.

When you upgrade IBM HTTP Server from a previous version, complete the following steps to install the new version in the same directory location as the previous version. If the new version is installed in a different directory, you do not need to complete Steps 1 - 4. Whether you need to complete the remaining steps depends on how similar you want to make the current configuration to the configuration of a previous version of IBM HTTP Server.

1. Stop the IBM HTTP Server and the IBM HTTP Server administration server.
2. Copy the existing installation directory to a new location.

This action preserves your configuration, keys, and content.

Issue the following command to copy the previous installation:

```
HP-UX Linux Solaris  
cp -rp current_install_directory new_directory_name
```

```
Windows  
xcopy current_install_directory new_directory_name /s /e /k /i
```

3. Uninstall the previous IBM HTTP Server version.
4. Remove the previous installation directory.

Because the uninstall leaves behind some files, such as modified and added files, fixpack files, and uninstall files, you must manually remove the

previous installation directory to complete the uninstall process. If you had any uninstall issues, review and backup the uninstall log files in the `http_server_install/logs/uninstall` directory before proceeding.

Issue the following command to remove the installation directory:

AIX

HP-UX

Linux

Solaris

```
rm -r current_install_directory
```

Windows

```
rd /s current_install_directory
```

5. Install IBM HTTP Server.

If upgrading your existing version, install into the directory where the previous installation was located.

If installing the new version alongside an existing version, install the new version into a different directory.

6. Run the Plug-ins Configuration Tool, the `pct` tool, to configure your web server plug-ins. Refer to the [Configuring a web server plug-in using the `pct` tool](#) topic for information on running the `pct` tool.

7. Restore any custom configurations that were made to your previous version of IBM HTTP Server and IBM HTTP Server administration server.

- Identify your previous customizations.

If you used the `httpd.conf` configuration files provided with the previous version of IBM HTTP Server as the starting point for your configuration files, compare the content of each configuration file, with its corresponding `.default` file, within the directory containing your previous IBM HTTP Server installation. For example, if you compare the content of the `httpd.conf` file with the `httpd.conf.default` file you should see any customization that were made to the `httpd.conf` file since the original installation. Then perform similar comparisons for the other configuration files.

If you did not use the `httpd.conf` configuration files that are provided with the previous version of IBM HTTP Server as the starting point for your configuration files, you must complete a more manual analysis to determine your previous settings. In this scenario, you might want to compare the settings in the `httpd.conf.default` file that is provided with the new IBM HTTP Server, with the settings in the `httpd.conf.default` file that is provided with the previous IBM HTTP Server version. This comparison enables you to identify configuration differences in the two `httpd.conf.default` files. You can then use this information to modify your customized configuration file to work with the current IBM HTTP Server.

Compare the `bin/envvars` file to the `bin/envvars-std` file within the directory containing your previous IBM HTTP Server installation. This identifies what customizations, if any, that were made to this file.

- Merge the customizations into the newly installed IBM HTTP Server configuration and `envvars` files.

After you identify the configuration customizations you made to your previous version of IBM HTTP Server, make these same changes, when applicable, to the configuration files for the current IBM HTTP Server.

If the configuration files contain WebSphere Application Server plug-in statements from previous versions, remove them to not cause duplicates.

If you do not remove these statements, when the HTTP Server attempts to

start the current plug-in binary module, an error might occur that indicates that the module is already loaded.

The configuration file might also contain duplicate entries for accessing WebSphere Application Server samples. Remove any aliases for previous versions and retain the current entries:

- Use a configuration file from IBM HTTP Server V7.0, V8.0, or V8.5.5.
- 8. Restore HTML content. If your web page content was previously stored under your IBM HTTP Server installation directory, copy those content files from the directory that contains your prior version of IBM HTTP Server into the installation directory for the new version.
- 9. Copy any SSL KeyFiles, that might be within the installation directory of the previous IBM HTTP Server, into the new installation directory

- Change port assignments for coexisting IBM HTTP Servers.

If you installed the IBM HTTP Server into a new directory and retained your previous version of the IBM HTTP Server, by default the administration server and the Web Server use the same ports as the previous version administration server and Web Server. If you ever run both versions of the IBM HTTP Server simultaneously, port conflicts will occur unless you change the port numbers for one of the server versions.

To modify the port numbers for one of the IBM HTTP Servers, edit the server configuration files for that IBM HTTP Server. These files are located in the *http_server_install/conf* directory.

- Upgrade Apache plug-in modules.

There are no Apache API changes from the previous major release so there should be no need to rebuild modules that worked with the previous release. However, if you use modules from third party vendors, then you should contact your vendors to verify they support the module with the version of IBM HTTP Server to which you are upgrading.

Apache plug-in modules from sources other than the current IBM HTTP Server installation must be built to support Apache 2.4. The distributors of modules used with older versions of IBM HTTP Server might need to recompile the modules to support Apache 2.4.

- WebSphere Application Server provides a new plug-in for Apache 2.4 and IBM HTTP Server.
 - If you use modules from third party vendors, contact your vendor for a version of the module that works with the Apache 2.4 API (application programming interface).
 - If you use modules developed in-house, you must rebuild your modules to support Apache 2.4. The modules might also require some modifications.
- Update the IBM HTTP Server service name.

Update the IBM HTTP Server service name in the WebSphere Application Server web server definition if the following conditions apply:

- You are using a Windows server
- You installed IBM HTTP Server into the same directory where an earlier version was located
- You are using a web server definition from that prior installation

For an IBM HTTP Server on a Windows server system, use 'Services' to determine the name used for the new IBM HTTP Server service, and then update the web server definition to use this service name.

Running multiple instances of IBM HTTP Server from a single install

Distributed operating systems

z/OS

Run multiple, independent instances of IBM HTTP Server from a single installation. It is seldom necessary to run multiple instances, as features like virtual hosts allow a single instance to efficiently serve many sites, but in some cases it is necessary. If you need to securely administer your sites by different administrators, for example, you must run separate instances that each use their own configuration files.

Before you begin

This topic is primarily for AIX, HP-UX, Linux, Solaris, and Windows operating systems. On the z/OS® platform, the `install_ihs` command creates a separate directory for each instance without creating another copy of the product. See the z/OS topic for configuring IBM HTTP Server for more information.

Before configuring multiple instances, consider if your problem can be solved by using virtual hosts and/or having IBM HTTP Server listen on multiple addresses and ports. The advantage of a single instance is that it uses less resources to serve the same requests as multiple instances.

Note: When you follow the examples, change "this_instance" to a unique name for each instance.

Procedure

1. Create a separate main configuration file, normally the `httpd.conf` file, for each instance.

Note: To reduce duplication, store common directives in common files and import these into the separate, main configuration files with the *Include* directive.

We'll call the configuration file `conf/this_instance.conf` for the rest of these steps.

Here is a simple example of a configuration file for an instance:

```
Listen 10.0.0.1:80
PidFile instance1/httpd.pid
ErrorLog instance1/error.log
CustomLog instance1/access.log common
# Other directives that make this instance behave uniquely
Include conf/common.conf
```

A real configuration file would have more directives in it to make this instance behave differently than the other instances.

2. Configure the port settings in the configuration files. You cannot use a combination of listen port and listen IP address for more than one instance. Check the Listen directives in each configuration file, and verify that they are unique. See information on the Listen directive for Apache HTTP Server for more information.
3. Configure settings for logging and other special files. Any files that are normally stored in the `install_root/logs` directory cannot be shared between instances. Each instance must have unique values for the following directives:

PidFile

Applicable to all configurations. See the information on the PidFile directive for Apache HTTP Server.

ScriptSock

Applicable to non-Windows configurations with mod_cgid enabled.

ErrorLog

Applicable to all configurations. See the information on the ErrorLog directive for Apache HTTP Server.

CustomLog or TransferLog

Applicable to all configurations. See the information on the CustomLog directive or the TransferLog directive for Apache HTTP Server.

SSLCachePortFilename

Applicable to all non-Windows configurations with SSL enabled. See the information on the SSLCachePortFilename directive.

SSLCachePath

Applicable when all of the following conditions are true:

- Platform is not Windows.
- SSL is enabled.
- SSLCacheDisable directive is not configured.
- bin/apachectl has been modified to specify a different -d flag, or bin/apachectl is launched with an explicit -d flag.
- The directory specified by the -d flag does not contain the file bin/sidd.

See the information on the SSLCachePath directive for Apache HTTP Server. See information on the SSLCachePath directive.

Other optional directives that specify a file path, like logging or tracing.

4. **AIX** **Windows** Ensure that no more than one IHS instance has the fast response cache accelerator (FRCA), or AFPA, enabled.

Note: FRCA/AFPA has been deprecated starting with V7.0 and its use is discouraged. There is no support for Windows Vista, Windows 2008, or any later Windows operating systems.

5. Start or stop the IHS server instance.
 - **AIX** **HP-UX** **Linux** **Solaris** Use these commands to start and stop IHS:

```
# cd /install_dir
# bin/apachectl -k start -f conf/this_instance.conf
# bin/apachectl -k stop -f conf/this_instance.conf
```

Alternatively, you can create a copy of apachectl for each instance, and update the commands in each copy to include "-f conf/this_instance.conf".

- **Windows** Use these commands to setup a new instance:

```
cd \install_dir
bin\Apache.exe -f conf/this_instance.conf -k install -n IHS-this_instance
```

Choose one of these commands to start and stop IHS:

- Use this command:

```
net start IHS-this_instance
```
- Use this command:

```
cd \install_dir
bin\Apache.exe -k install -n IHS-this_instance.conf
```

- Find IHS-this_instance in the Services interface for Microsoft Windows.

See the topic on starting and stopping IBM HTTP Server for more information.

Installing fix packs on IBM HTTP Server using the Installation Manager GUI

Distributed operating systems

You can use Installation Manager to update IBM HTTP Server to a later version.

Before you begin

Make sure that the web-based or local service repository location is listed and checked or that the **Search service repositories during installation and updates** option is selected on the Repositories panel in your Installation Manager preferences. For more information on using service repositories with Installation Manager, read the Installation Manager Information Center.

About this task

Perform this procedure to use Installation Manager to update IBM HTTP Server.

Procedure

1. Start Installation Manager.
2. Click **Update**.

Note: If you are prompted to authenticate, use the IBM ID and password that you use to access protected IBM software websites.

3. Select the package group to update.
4. Click **Next**.
5. Select the version to which you want to update under **IBM HTTP Server**.
6. Click **Next**.
7. Accept the terms in the license agreements, and click **Next**.
8. Review the summary information, and click **Update**.
 - If the installation is successful, the program displays a message indicating that installation is successful.
 - If the installation is not successful, click **View Log File** to troubleshoot the problem.
9. Click **Finish**.
10. Click **File > Exit** to close Installation Manager.

Installing IBM HTTP Server silently

Distributed operating systems

You can use Installation Manager to install IBM HTTP Server silently.

Before you begin

Install Installation Manager on each of the systems onto which you want to install the product.

1. Perform one of the following procedures:

- If you want to use the Installation Manager that is included with this product, perform the following actions:

a. Obtain the necessary files.

There are two basic options for obtaining and installing Installation Manager and the product.

– **Download the files from the Beta site, and use local installation**

You can download the necessary product repositories from the Beta site.

1) Download the files from the Beta site.

2) Install Installation Manager on your system.

You can install Installation Manager using a file obtained from the Beta site.

3) Use Installation Manager to install the product from the downloaded repositories.

– **Access the live repositories, and use web-based installation**

You can install the product from the web-based repositories.

1) Install Installation Manager on your system.

You can install Installation Manager using a file obtained from the Beta site.

2) Use Installation Manager to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product.

There are three basic options for obtaining and installing Installation Manager and the product.

– **Access the physical media, and use local installation**

You can access Installation Manager and the product repositories on the product media. You can install Installation Manager on your system and use it to install the product from the product repositories on the media.

– **Download the files from the Passport Advantage site, and use local installation**

Licensed customers can download Installation Manager as well as the necessary product repositories from the Passport Advantage site. You can then install Installation Manager on your system and use it to install the product from the repositories.

– **Download a file from the Installation Manager website, and use web-based installation**

You can download and unpack a compressed file containing Installation Manager from the IBM Installation Manager website. You can then install Installation Manager on your local system and use it to install the product from the web-based repository located at

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v85>

- b. Change to the location containing the Installation Manager installation files, and run one of the following commands to install Installation Manager silently:

Administrative installation:

- **Windows** `installc.exe -acceptLicense -log log_file_path_and_name`
- **AIX** **HP-UX** **Linux** **Solaris** `./installc -acceptLicense -log log_file_path_and_name`

Non-administrative installation:

- **Windows** `userinstc.exe -acceptLicense -log log_file_path_and_name`
- **AIX** **HP-UX** **Linux** **Solaris** `./userinstc -acceptLicense -log log_file_path_and_name`

Group-mode installation:

- **AIX** **HP-UX** **Linux** **Solaris** `./groupinstc -acceptLicense -dataLocation application_data_location -log log_file_path_and_name`

Notes on group mode:

- Group mode allows multiple users to use a single instance of IBM Installation Manager to manage software packages.
This does not mean that two people can use the single instance of IBM Installation Manager at the same time.
 - **Windows** Group mode is not available on Windows operating systems.
 - If you do not install Installation Manager using group mode, you will not be able to use group mode to manage any of the products that you install later using this Installation Manager.
 - Make sure that you change the installation location from the default location in the current user's home directory to a location that is accessible by all users in the group.
 - Set up your groups, permissions, and environment variables as described in the Group mode road maps in the IBM Installation Manager Information Center before installing in group mode.
 - For more information on using group mode, read the Group mode road maps in the IBM Installation Manager Information Center.
- If you already have the Installation Manager Version 1.5.2 Beta Installation Manager Version 1.5.2 or later installed on your system and you want to use it to install and maintain the product, obtain the necessary product files from the physical media or the web.

There are two basic options for installing the product.

- **Download the files from the Beta site, and use local installation**

You can download the necessary product repositories from the Beta site.

a. Download the product repositories from the Beta site.

b. Use Installation Manager to install the product from the downloaded repositories.

- **Access the live repositories, and use web-based installation**

You can install Installation Manager on your local system and use it to install the product from the web-based repository located at <http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product. Whenever possible, you should use the remote web-based repositories so that you are accessing the most up-to-date installation files.

There are three basic options for installing the product.

– **Access the physical media, and use local installation**

You can access the product repositories on the product media. Use your existing Installation Manager to install the product from the product repositories on the media.

– **Download the files from the Passport Advantage site, and use local installation**

Licensed customers can download the necessary product repositories from the Passport Advantage site. You can then use your existing Installation Manager to install the product from the repositories.

– **Access the live repositories, and use web-based installation**

You can install Installation Manager on your local system and use it to install the product from the web-based repository located at <http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v85>

Note: This location does not contain a web page that you can access using a web browser. This is a remote web-based repository location that you must add to your Installation Manager preferences before the Installation Manager GUI can access the files in this repository to install the product. Whenever possible, you should use the remote web-based repositories so that you are accessing the most up-to-date installation files.

2. Add the product repository to your Installation Manager preferences.
 - a. Start Installation Manager.
 - b. In the menu, click **File > Preferences**.
 - c. Select **Repositories**.
 - d. Perform the following actions:
 - 1) Click **Add Repository**.
 - 2) Enter the path to the repository.config file in the location containing the repository files.

For example:

-  C:\repositories*product_name*\local-repositories
-     /var/repositories/*product_name*/local-repositories

or

<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>

- 3) Click **OK**.
- e. Deselect any locations listed in the Repositories window that you will not be using.
- f. Click **Apply**.

- g. Click **OK**.
- h. Click **File > Exit** to close Installation Manager.

Restriction: You can only configure the HTTP port (that is, the `user.ihs.httpPort`) in IMCL/Silent/GUI mode on AIX or other UNIX platforms. The administrative configuration cannot be performed along with the installation of IHS. You need to install IHS first; then, you can use the GUI-based Plug-in Configuration Tool or the `pct` command-line tool to perform the administrative configuration.

About this task

Complete this procedure to install IBM HTTP Server silently.

Procedure

1. **Record a response file to install IBM HTTP Server:** On one of your systems, complete the following actions to record a response file that will install IBM HTTP Server.
 - a. From a command line, change to the `eclipse` subdirectory in the directory where you installed Installation Manager.
 - b. Start Installation Manager from the command line using the `-record` option.

For example:

- **Windows Administrator or non-administrator:**
`IBMIM.exe -skipInstall "C:\temp\imRegistry" -record C:\temp\install_response_file.xml`
- **AIX HP-UX Linux Solaris Administrator:**
`./IBMIM -skipInstall /var/temp/imRegistry -record /var/temp/install_response_file.xml`
- **AIX HP-UX Linux Solaris Non-administrator:**
`./IBMIM -skipInstall user_home/var/temp/imRegistry -record user_home/var/temp/install_response_file.xml`

Tip: When you record a new response file, you can specify the `-skipInstall` parameter. Using this parameter has the following benefits:

- No files are actually installed, and this speeds up the recording.
- If you use a temporary data location with the `-skipInstall` parameter, Installation Manager writes the installation registry to the specified data location while recording. When you start Installation Manager again without the `-skipInstall` parameter, you then can use your response file to install against the real installation registry.

The `-skipInstall` operation should not be used on the actual agent data location used by Installation Manager. This is unsupported. Use a clean writable location, and re-use that location for future recording sessions.

For more information, read the IBM Installation Manager Information Center.

- c. Add the appropriate repositories to your Installation Manager preferences.
 - 1) In the menu, click **File > Preferences**.
 - 2) Select **Repositories**.
 - 3) Complete the following actions for each repository:
 - a) Click **Add Repository**.
 - b) Enter the path to the `repository.config` file in the remote Web-based repository or the local directory into which you unpacked the repository files.

For example:

- Remote repositories:
https://downloads.mycorp.com:8080/WAS_85_repository
 or
<http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.beta.v85>
- Local repositories:
 - **Windows** C:\repositories\ihs\local-repositories
 - **AIX** **HP-UX** **Linux** **Solaris**
 /var/repositories/ihs/local-repositories

c) Click **OK**.

4) Click **Apply**.

5) Click **OK**.

d. Click **Install**.

Note: If you are prompted to authenticate, use the IBM ID and password that you registered with on the program website.

Installation Manager searches its defined repositories for available packages.

e. In the **Install Packages** window, complete the appropriate actions.

1) Select **IBM HTTP Server** and the appropriate version.

Note: If you are installing the trial version of this product, select **IBM HTTP Server Trial** and the appropriate version.

If you already have IBM HTTP Server installed on your system, a message displays indicating that IBM HTTP Server is already installed. To create another installation of IBM HTTP Server in another location, click **Continue**.

2) Click **Next**.

f. Accept the terms in the license agreements, and click **Next**.

g. On the Location panel, specify the installation root directory for IBM HTTP Server binaries, which are also referred to as the core product files or system files.

The panel also displays the shared resources directory and disk-space information.



The core product files do not change unless you install maintenance.

Restrictions:


- Deleting the default target location and leaving an installation-directory field empty prevents you from continuing.
- Do not use symbolic links as the destination directory.
Symbolic links are not supported.
- Do not use spaces in the name of the installation directory.
These spaces are not supported.
- Do not use a semicolon in the directory name.
IBM HTTP Server cannot install properly if the target directory includes a semicolon.

Windows A semicolon is the character used to construct the class path on Windows systems.

- **Windows** The maximum path length on the Windows XP, Windows Vista, and Windows 7 operating systems is 60 characters.

- h. Click **Next**.
- i.    If you are installing on a 64-bit system, choose between a 32-bit or 64-bit HTTP Server environment and click **Next**.

Notes:

- This option displays only if you are installing on a 64-bit system.
 - This does not apply to Solaris x86 64-bit systems.
 - You must select one of the two options.
 - You cannot modify this installation later and change this selection.
- j. Click **Next** to display the Configuration for IBM HTTP Server panel,
- k.  On the Configuration for IBM HTTP Server panel, specify your Web server configuration.
- Specify a port number on which IBM HTTP Server will communicate. The default port is 80.
 - Choose whether to use a Windows service to run IBM HTTP Server.




Note: You have the option to create a Windows service for IBM HTTP Server on this panel. You can configure the services to run as local system account or a user ID that you specify. The user ID requires the following advanced user rights:

- Act as part of the operating system
- Log on as a service

Important: If you do not select **Run IBM HTTP Server as a Windows Service**, this instance of IBM HTTP Server cannot be stopped or started by the WebSphere Application Server administrative console. At any time after installation, you can create a new service by running the following command:

```
ihp_root/bin/httpd.exe -n new_service_name -k install
```

and then updating the web server definition in the administrative console to reflect the new service name.

- Determine if your startup type will be automatic or manual.
- l.    On the Configuration for IBM HTTP Server panel, specify your Web server configuration.
- Specify a port number on which IBM HTTP Server will communicate. The default port is 80.
- m. Click **Next**.
- n. Review the summary information, and click **Install**.
- If the installation is successful, the program displays a message indicating that installation is successful.
- Note:** The program might also display important post-installation instructions as well.
- If the installation is not successful, click **View Log File** to troubleshoot the problem.
- o. Click **Finish**.
- p. Click **File > Exit** to close Installation Manager.
- q. Optional: If you are using an authenticated remote repository, create a keyring file for silent installation.

- 1) From a command line, change to the eclipse subdirectory in the directory where you installed Installation Manager.
- 2) Start Installation Manager from the command line using the `-record` option.

For example:

- **Windows Administrator or non-administrator:**

```
IBMIM.exe -skipInstall "C:\temp\imRegistry"
-keyring C:\IM\im.keyring
-record C:\temp\keyring_response_file.xml
```

- **AIX HP-UX Linux Solaris Administrator:**

```
./IBMIM -skipInstall /var/temp/imRegistry
-keyring /var/IM/im.keyring
-record /var/temp/keyring_response_file.xml
```

- **AIX HP-UX Linux Solaris Non-administrator:**

```
./IBMIM -skipInstall user_home/var/temp/imRegistry
-keyring user_home/var/IM/im.keyring
-record user_home/var/temp/keyring_response_file.xml
```

- 3) When a window opens that requests your credentials for the authenticated remote repository, enter the correct credentials and **save** them.
- 4) Click **File > Exit** to close Installation Manager.

For more information, read the IBM Installation Manager Information Center.

2. Use the response files to install IBM HTTP Server silently:

- a. **Optional: Use the response file to install the keyring silently:** Go to a command line on each of the systems on which you want to install IBM HTTP Server, change to the `eclipse/tools` subdirectory in the directory where you installed Installation Manager, and install the keyring silently.

For example:

- **Windows Administrator or non-administrator:**

```
imcl.exe -acceptLicense
-input C:\temp\keyring_response_file.xml
-log C:\temp\keyring_log.xml
```

- **AIX HP-UX Linux Solaris Administrator:**

```
./imcl -acceptLicense
-input /var/temp/keyring_response_file.xml
-log /var/temp/keyring_log.xml
```

- **AIX HP-UX Linux Solaris Non-administrator:**

```
./imcl -acceptLicense
-input user_home/var/temp/keyring_response_file.xml
-log user_home/var/temp/keyring_log.xml
```

- b. **Use the response file to install IBM HTTP Server silently:** Go to a command line on each of the systems on which you want to install IBM HTTP Server, change to the `eclipse/tools` subdirectory in the directory where you installed Installation Manager, and install IBM HTTP Server silently.

For example:

- **Windows Administrator or non-administrator:**

```
imcl.exe -acceptLicense
-input C:\temp\install_response_file.xml
-log C:\temp\install_log.xml
-keyring C:\IM\im.keyring -password password
-repositories C:\IHS_REPOSITORY\ -installationDirectory C:\IHS -sharedResourcesDirectory C:\IM\IMShared\ -acceptLicense
-properties "user.ihs.httpPort=80,user.ihs.allowNonRootSilentInstall=true"
```

- **AIX** **HP-UX** **Linux** **Solaris** **Administrator:**

```
./imcl -acceptLicense
-input /var/temp/install_response_file.xml
-log /var/temp/install_log.xml
-keyring /var/IM/im.keyring -password password
-repositories /root/IHS/ -installationDirectory /QIBM/ProdData/IHS -sharedResourcesDirectory
/QIBM/UserData/InstallationManager/IMShared/ -acceptLicense -properties "user.ihs.httpPort=80,user.ihs.allowNonRootSilentInstall=true"
```

- **AIX** **HP-UX** **Linux** **Solaris** **Non-administrator:**

```
./imcl -acceptLicense
-input user_home/var/temp/install_response_file.xml
-log user_home/var/temp/install_log.xml
-keyring user_home/var/IM/im.keyring -password password
-repositories /root/IHS/ -installationDirectory /QIBM/ProdData/IHS -sharedResourcesDirectory
/QIBM/UserData/InstallationManager/IMShared/ -acceptLicense -properties "user.ihs.httpPort=80,user.ihs.allowNonRootSilentInstall=true"
```

Notes:

- The relevant terms and conditions, notices, and other information are provided in the license-agreement files in the `lafiles` or `product_name/lafiles` subdirectory of the installation image or repository for this product.
- The program might write important post-installation instructions to standard output.

Read the IBM Installation Manager Information Center for more information.

Windows

Example

The following is an example of a response file for silently installing IBM HTTP Server.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ##### Copyright #####
# Licensed Materials - Property of IBM (c) Copyright IBM Corp. 2011.
# All Rights Reserved. US Government Users Restricted Rights-Use, duplication
# or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
##### -->

<!-- ##### Frequently Asked Questions #####
# The latest information about using Installation Manager is
# located in the online Information Center. There you can find
# information about the commands and attributes used in
# silent installation response files.
#
# Installation Manager Information Center can be found at:
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp
#
# Question 1. How do I record a response file using Installation Manager?
# Answer 1. Start Installation Manager from the command line under the
# eclipse subdirectory with the record parameter and it will generate a
# response file containing actions it performed, repositories it used, and
# its preferences settings. Optionally use the -skipInstall parameter if
# you do not want the product to be installed to the machine. Specify a
# new agentDataLocation location value when doing a new installation. Do
# not use an existing agentDataLocation for an installation because it might
# damage the installation data and prevent you from modifying, updating,
# rolling back, or uninstalling the installed packages.
#
# Windows: IBMIM -record <responseFile> -skipInstall <agentDataLocation>
# Linux or UNIX: ./IBMIM -record <responseFile> -skipInstall <agentDataLocation>
#
# For example:
# Windows = IBMIM.exe -record c:\temp\responsefiles\WASv8.install.Win32.xml
# -skipInstall c:\temp\skipInstall\WebSphere_Temp_Registry
# Linux or UNIX = ./IBMIM -record /home/user/responsefiles/WASv8.install.RHEL64.xml
# -skipInstall c:\temp\skipInstall\WebSphere_Temp_Registry
#
# Question 2. How do I run Installation Manager silently using response file?
# Answer 2. Create a silent installation response file and run the following command
# from the eclipse\tools subdirectory in the directory where you installed
# Installation Manager:
#
# Windows = imcl.exe -acceptLicense -showProgress
# input <response_file_path_and_name> -log <log_file_path_and_name>
```

```

# Linux, UNIX, IBM i and z/OS = ./imcl -acceptLicense -showProgress
#   input <response_file_path_and_name> -log <log_file_path_and_name>
#
# For example:
#   Windows = imcl.exe -acceptLicense -showProgress
#             input c:\temp\responsefile\WASv8.install.Win32.xml
# Linux, UNIX, IBM i and z/OS = ./imcl -acceptLicense -showProgress
#   input /home/user/responsefile/WASv8.install.RHEL64.xml
#
# The -acceptLicense command must be included to indicate acceptance of all
#   license agreements of all offerings being installed, updated or modified.
# The -showProgress command shows progress when running in silent mode.
# Additional commands can be displayed by requesting help: IBMIM -help
#
# Question 3. How do I store and pass credentials to repositories that
#   require authentication?
# Answer 3. Installation Manager uses a key ring file to store encrypted
#   credentials for authenticating with repositories. Follow this two-step
#   process for creating and using a key ring file with Installation Manager.
#
# First, create a key ring file with your credentials by starting
#   Installation Manager from the command line under eclipse subdirectory
#   with the keyring parameter.
# Use the optional password parameter to password protect your file.
#
#   Windows = IBMIM.exe -keyring <path and file name> -password <password>
#   Linux, UNIX, IBM i and z/OS = ./IBMIM -keyring <path and file name>
#                                     -password <password>
#
# Installation Manager will start in graphical mode. Verify that the
#   repositories to which you need to authenticate are included in the
#   preferences, File / Preferences / Repositories. If they are not
#   listed, then click Add Repositories to add the URL or UNC path.
# Installation Manager will prompt for your credentials. If the repository
#   is already in the list, then any attempt to access the repository location,
#   such as clicking the Test Connections button, will also prompt for your
#   credentials. Enter the correct credential and check the Save password
#   checkbox. The credentials are saved to the key ring file you specified.
#
# Second, when you start a silent installation, run imcl under eclipse/tools
#   subdirectory, and provide Installation Manager with the location of the key
#   ring file and the password if the file is protected. For example:
#
#   Windows = imcl.exe -acceptLicense -showProgress
#             input <path and file name of response file>
#             -keyring <path and name of key ring file> -password <password>
# Linux, UNIX, IBM i and z/OS = ./imcl -acceptLicense -showProgress
#   input <path and file name of response file>
#   -keyring <path and name of key ring file> -password <password>
#
##### -->
<!-- ##### Agent Input #####
#
# Note that the "acceptLicense" attribute has been deprecated.
# Use "-acceptLicense" command line option to accept license agreements.
#
# The clean and temporary attributes specify the repositories and other
# preferences Installation Manager uses and whether those settings
# should persist after the installation finishes.
#
# Valid values for clean:
#   true = only use the repositories and other preferences that are
#         specified in the response file.
#   false = use the repositories and other preferences that are
#         specified in the response file and Installation Manager.
#
# Valid values for temporary:
#   true = repositories and other preferences specified in the
#         response file do not persist in Installation Manager.
#   false = repositories and other preferences specified in the
#         response file persist in Installation Manager.
#
##### -->
<agent-input clean="true" temporary="true">

<!-- ##### Repositories #####
# Repositories are locations that Installation Manager queries for
# installable packages. Repositories can be local (on the machine
# with Installation Manager) or remote (on a corporate intranet or
# hosted elsewhere on the internet).
#
# If the machine using this response file has access to the internet,
# then include the IBM WebSphere Live Update Repositories in the list
# of repository locations.
#
# If the machine using this response file cannot access the internet,
# then comment out the IBM WebSphere Live Update Repositories and
# specify the URL or UNC path to custom intranet repositories and

```

```

# directory paths to local repositories to use.
#
##### -->

<server>
<!-- ##### IBM WebSphere Live Update Repositories #####
# These repositories contain IBM HTTP Server offerings,
# and updates for those offerings
#
# To use the secure repository (https), you must have an IBM ID,
# which can be obtained by registering at: http://www.ibm.com/account
# or your Passport Advantage account.
#
# And, you must use a key ring file with your response file.
##### -->
<!-- repository location="http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v80" /> -->
<!-- <repository location="https://www.ibm.com/software/rational/repositorymanager/repositories/websphere" /> -->

<!-- ##### Custom Repositories #####
# Uncomment and update the repository location key below
# to specify URLs or UNC paths to any intranet repositories
# and directory paths to local repositories to use.
##### -->
<!-- <repository location='https:\\w3.mycompany.com\\repositories\\' /> -->
<!-- <repository location='/home/user/repositories/websphere/' /> -->

<!-- ##### Local Repositories #####
# Uncomment and update the following line when using a local
# repository located on your own machine to install a
# IBM HTTP Server offering.
##### -->
<!-- <repository location='insert the full directory path inside single quotes' /> -->
<repository location='C:\Documents and Settings\Administrator\DownloadDirector\V8IHS' />
</server>

<!-- ##### Install Packages #####
#
# Install Command
#
# Use the install command to inform Installation Manager of the
# installation packages to install.
#
# The modify attribute is optional and can be paired with an install
# command to add features or paired with an uninstall command to
# remove commands. If omitted, the default value is set to false.
# false = indicates not to modify an existing install by adding
# or removing features.
# true = indicates to modify an existing install by adding or
# removing features.
#
# The offering ID attribute is required because it specifies the
# offering to be installed. The offering listed must be present in
# at least one of the repositories listed earlier. The example
# command below contains the offering ID for the IBM HTTP Server.
#
# The version attribute is optional. If a version number is provided,
# then the offering will be installed at the version level specified
# as long as it is available in the repositories. If the version
# attribute is not provided, then the default behavior is to install
# the latest version available in the repositories. The version number
# can be found in the repository.xml file in the repositories.
# For example, <offering ... version='8.0.0.20110617_2222'>.
#
# The profile attribute is required and typically is unique to the
# offering. If modifying or updating an existing installation, the
# profile attribute must match the profile ID of the targeted installation
# of IBM HTTP Server.
#
# The features attribute is optional. Offerings always have at least
# one feature; a required core feature which is installed regardless
# of whether it is explicitly specified. If other feature names
# are provided, then only those features will be installed.
# Features must be comma delimited without spaces.
#
# The feature values for IBM HTTP Server include:
# arch.32bit,arch.64bit
#
# The installFixes attribute indicates whether fixes available in
# repositories are installed with the product. By default, all
# available fixes will be installed with the offering.
#
# Valid values for installFixes:
# none = do not install available fixes with the offering.
# recommended = installs all available recommended fixes with the offering.
# all = installs all available fixes with the offering.
#
# Interim fixes for offerings also can be installed while they
# are being installed by including the offering ID for the interim
# fix and specifying the profile ID. A commented out example is
# provided in the install command below.

```

```

#
# Installation Manager supports installing multiple offerings at once.
# Additional offerings can be included in the install command,
# with each offering requiring its own offering ID, version, profile value,
# and feature values.
#
# Profile Command
#
# A separate profile command must be included for each offering listed
# in the install command. The profile command informs Installation
# Manager about offering specific properties or configuration values.
#
# The installLocation specifies where the offering will be installed.
# If the response file is used to modify or update an existing
# installation, then ensure the installLocation points to the
# location where the offering was installed previously.
#
# The eclipseLocation data key should use the same directory path to
# IBM HTTP Server as the installationLocation attribute.
#
# Include data keys for product specific profile properties.
#
##### -->
<install modify='false'>
<offering id='com.ibm.websphere.IHS.v80'
  profile='IBM HTTP Server V8.0'
  features='core.feature,arch.32bit' installFixes='none'>
<!-- <offering id='PM12345_WAS80' profile='IBM HTTP Server for WebSphere Application Server V8.0'> -->
</install>

<profile id='IBM HTTP Server V8.0'
  installLocation='C:\Program Files\IBM\HTTPServer'>
<data key='eclipseLocation' value='C:\Program Files\IBM\HTTPServer'>
<data key='user.import.profile' value='false'>
<data key='user.ihs.http.server.service.name' value='none'> <!-- Always none if
user.ihs.installHttpService = false Otherwise Unique Windows service name -->
<data key='user.ihs.httpPort' value='80'>
<data key='user.ihs.installHttpService' value='false'>
<!-- data key='user.ihs.http.server.service.name.key' value='Unique Windows service registry key'
Windows Only - Required if user.ihs.installHttpServer = true
data key='user.ihs.win.serverServiceStartType' value='auto | demand'> Windows Only
data key='user.ihs.win.serverServiceLogOnAsLocalSystem' value='true | false'> Windows Only
data key='user.ihs.win.serverServiceUser' value='local user name'> Windows Only
data key='user.ihs.win.serverServicePassword' value='local user password'> Windows Only
Required if data key='user.ihs.win.serverServiceLogOnAsLocalSystem' = false
password value can be encrypted using Installation Manager
utility program, <installationManagerRoot>/eclipse/tools/imutilsc
-->
<data key='cic.selector.nl' value='en'>
</profile>

<!-- ##### Shared Data Location #####
# Uncomment the preference for eclipseCache to set the shared data
# location the first time you use Installation Manager to do an
# installation.
#
# Eclipse cache location can be obtained from the installed.xml file found in
# Linux/Unix: /var/ibm/InstallationManager
# Windows: C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager
# from the following property:
# <property name='cacheLocation' value='C:\Program Files\IBM\IMShared'>
#
# Open the installed.xml file in a text editor because the style sheet
# might hide this value if opened in a web browser.
# For further information on how to edit preferences, refer to the public library at:
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp?topic=/com.ibm.silentinstall2.doc/topics/r_silent_prefs.html
#
# After the shared data location is set, it cannot be changed
# using a response file or the graphical wizard.
#
# Ensure that the shared data location is a location that can be written
# to by all user accounts that are expected to use Installation Manager.
#
# By default, Installation Manager saves downloaded artifacts to
# the shared data location. This serves two purposes.
#
# First, if the same product is installed a more than once to the machine,
# then the files in the shared data location will be used rather than
# downloading them again.
#
# Second, during the rollback process, the saved artifacts are used.
# Otherwise, if the artifacts are not saved or are removed, then
# Installation Manager must have to access the repositories used to
# install the previous versions.
#
# Valid values for preserveDownloadedArtifacts:
# true = store downloaded artifacts in the shared data location
# false = remove downloaded artifacts from the shared data location
#

```



```
##### -->
<!--
<preference name='com.ibm.cic.common.core.preferences.eclipseCache' value='C:\Program Files\IBM\IMShared' />
<preference name='com.ibm.cic.common.core.preferences.preserveDownloadedArtifacts' value='true' />
-->

<!-- ##### Preferences Settings #####
# Additional preferences for Installation Manager can be specified.
# These preference correspond to those that are located in the graphical
# interface under File / Preferences.
#
# If a preference command is omitted from or commented out of the response
# file, then Installation Manager uses the preference value that was
# previously set or the default value for the preference.
#
# Preference settings might be added or deprecated in new versions of
# Installation Manager. Consult the online Installation Manager
# Information Center for the latest set of preferences and
# descriptions about how to use them.
#
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp
#
##### -->

<!--
<preference name='com.ibm.cic.common.core.preferences.connectTimeout' value='30' />
<preference name='com.ibm.cic.common.core.preferences.readTimeout' value='45' />
<preference name='com.ibm.cic.common.core.preferences.downloadAutoRetryCount' value='0' />
<preference name='offering.service.repositories.areUsed' value='true' />
<preference name='com.ibm.cic.common.core.preferences.ssl.nonsecureMode' value='false' />
<preference name='com.ibm.cic.common.core.preferences.http.disablePreemptiveAuthentication' value='false' />
<preference name='http.ntlm.auth.kind' value='NTLM' />
<preference name='http.ntlm.auth.enableIntegrated.win32' value='true' />
<preference name='com.ibm.cic.common.core.preferences.keepFetchedFiles' value='false' />
<preference name='PassportAdvantageIsEnabled' value='false' />
<preference name='com.ibm.cic.common.core.preferences.searchForUpdates' value='false' />
<preference name='com.ibm.cic.agent.ui.displayInternalVersion' value='false' />
-->

</agent-input>
```

Uninstalling IBM HTTP Server using the GUI

Distributed operating systems

Use the Installation Manager GUI to uninstall IBM HTTP Server.

Procedure

Uninstall IBM HTTP Server.

1. Start Installation Manager.
2. Click **Uninstall**.
3. In the **Uninstall Packages** window, perform the following actions.
 - a. Select **IBM HTTP Server** and the appropriate version.

Note: If you are uninstalling the trial version of this product, select **IBM HTTP Server Trial** and the appropriate version.

- b. Click **Next**.
4. Review the summary information.
5. Click **Uninstall**.
 - If the uninstallation is successful, the program displays a message that indicates success.
 - If the uninstallation is not successful, click **View log** to troubleshoot the problem.
6. Click **Finish**.
7. Click **File > Exit** to close Installation Manager.

Uninstalling IBM HTTP Server silently

Distributed operating systems

You can use Installation Manager to uninstall IBM HTTP Server silently.

Before you begin

Optional: Complete or record the installation of Installation Manager and installation of IBM HTTP Server to a temporary installation registry on one of your systems so that you can use this temporary registry to record the uninstallation without using the standard registry where Installation Manager is installed.

Read the following for more information:

- “Installing IBM HTTP Server using the GUI” on page 7
- “Installing IBM HTTP Server silently” on page 21

Procedure

1. **Record a response file to uninstall IBM HTTP Server:** On one of your systems, complete the following actions to record a response file that will uninstall IBM HTTP Server:
 - a. From a command line, change to the eclipse subdirectory in the directory where you installed Installation Manager.
 - b. Start Installation Manager from the command line using the `-record` option.

For example:

- **Windows Administrator or non-administrator:**

```
IBMIM.exe -skipInstall "C:\temp\imRegistry" -record C:\temp\uninstall_response_file.xml
```

- **AIX HP-UX Linux Solaris Administrator:**

```
./IBMIM -skipInstall /var/temp/imRegistry -record /var/temp/uninstall_response_file.xml
```

- **AIX HP-UX Linux Solaris Non-administrator:**

```
./IBMIM -skipInstall user_home/var/temp/imRegistry -record user_home/var/temp/uninstall_response_file.xml
```

Tip: If you choose to use the `-skipInstall` parameter with a temporary installation registry created as described in the “Before you begin” section. Installation Manager uses the temporary installation registry while recording the response file. It is important to note that when the `-skipInstall` parameter is specified, no packages are installed or uninstalled. All of the actions that you complete in Installation Manager simply update the installation data that is stored in the specified temporary registry. After the response file is generated, it can be used to uninstall IBM HTTP Server, removing IBM HTTP Server files and updating the standard installation registry.

The `-skipInstall` operation should not be used on the actual agent data location used by Installation Manager. This is unsupported. Use a clean writable location, and re-use that location for future recording sessions. For more information, read the IBM Installation Manager Information Center.

- c. Click **Uninstall**.
- d. In the **Uninstall Packages** window, complete the following actions.
 - 1) Select IBM HTTP Server and the appropriate version.

Note: If you are uninstalling the trial version of this product, select IBM HTTP Server Trial and the appropriate version.

2) Click **Next**.

e. Review the summary information.

f. Click **Uninstall**.

- If the uninstallation is successful, the program displays a message that indicates success.
- If the uninstallation is not successful, click **View log** to troubleshoot the problem.

g. Click **Finish**.

h. Click **File > Exit** to close Installation Manager.

2. Use the response file to uninstall IBM HTTP Server silently: From a command line on each of the systems from which you want to uninstall IBM HTTP Server, change to the `eclipse/tools` subdirectory in the directory where you installed Installation Manager and use the response file that you created to silently uninstall IBM HTTP Server.

For example:

- **Windows** Administrator or non-administrator:

```
imcl.exe
-input C:\temp\uninstall_response_file.xml
-log C:\temp\uninstall_log.xml
```

- **AIX** **HP-UX** **Linux** **Solaris** Administrator:

```
./imcl
-input /var/temp/uninstall_response_file.xml
-log /var/temp/uninstall_log.xml
```

- **AIX** **HP-UX** **Linux** **Solaris** Non-administrator:

```
./imcl
-input user_home/var/temp/uninstall_response_file.xml
-log user_home/var/temp/uninstall_log.xml
```

Go to the IBM Installation Manager Information Center.

Windows **Example**

The following is an example of a response file for silently uninstalling IBM HTTP Server.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ##### Copyright #####
# Licensed Materials - Property of IBM (c) Copyright IBM Corp. 2011.
# All Rights Reserved. US Government Users Restricted Rights-Use, duplication
# or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
##### -->

<!-- ##### Frequently Asked Questions #####
# The latest information about using Installation Manager is
# located in the online Information Center. There you can find
# information about the commands and attributes used in
# silent installation response files.
#
# Installation Manager Information Center can be found at:
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp
#
# Question 1. How do I record a response file using Installation Manager?
# Answer 1. Start Installation Manager from the command line under the
# eclipse subdirectory with the record parameter and it will generate a
# response file containing actions it performed, repositories it used, and
# its preferences settings. Optionally use the -skipInstall parameter if
# you do not want the product to be installed to the machine. Specify a
# new agentDataLocation location value when doing a new installation. Do
# not use an existing agentDataLocation for an installation because it might
# damage the installation data and prevent you from modifying, updating,
# rolling back, or uninstalling the installed packages.
```

```

#
# Windows: IBMIM -record <responseFile> -skipInstall <agentDataLocation>
# Linux or UNIX: ./IBMIM -record <responseFile> -skipInstall <agentDataLocation>
#
# For example:
# Windows = IBMIM.exe -record c:\temp\responsefiles\WASv8.install.Win32.xml
# -skipInstall c:\temp\skipInstall\WebSphere_Temp_Registry
# Linux or UNIX = ./IBMIM -record /home/user/responsefiles/WASv8.install.RHEL64.xml
# -skipInstall c:\temp\skipInstall\WebSphere_Temp_Registry
#
# Question 2. How do I run Installation Manager silently using response file?
# Answer 2. Create a silent installation response file and run the following command
# from the eclipse\tools subdirectory in the directory where you installed
# Installation Manager:
#
# Windows = imcl.exe -acceptLicense -showProgress
# input <response_file_path_and_name> -log <log_file_path_and_name>
# Linux, UNIX, IBM i and z/OS = ./imcl -acceptLicense -showProgress
# input <response_file_path_and_name> -log <log_file_path_and_name>
#
# For example:
# Windows = imcl.exe -acceptLicense -showProgress
# input c:\temp\responsefile\WASv8.install.Win32.xml
# Linux, UNIX, IBM i and z/OS = ./imcl -acceptLicense -showProgress
# input /home/user/responsefile/WASv8.install.RHEL64.xml
#
# The -acceptLicense command must be included to indicate acceptance of all
# license agreements of all offerings being installed, updated or modified.
# The -showProgress command shows progress when running in silent mode.
# Additional commands can be displayed by requesting help: IBMIM -help
#
##### -->

<!-- ##### Agent Input #####
# The clean and temporary attributes specify the repositories and other
# preferences Installation Manager uses and whether those settings
# should persist after the uninstall finishes.
#
# Valid values for clean:
# true = only use the repositories and other preferences that are
# specified in the response file.
# false = use the repositories and other preferences that are
# specified in the response file and Installation Manager.
#
# Valid values for temporary:
# true = repositories and other preferences specified in the
# response file do not persist in Installation Manager.
# false = repositories and other preferences specified in the
# response file persist in Installation Manager.
#
##### -->

<agent-input clean='true' temporary='true'>

<!-- ##### Repositories #####
# Repositories are locations that Installation Manager queries for
# installable packages. Repositories can be local (on the machine
# with Installation Manager) or remote (on a corporate intranet or
# hosted elsewhere on the internet).
#
# If the machine using this response file has access to the internet,
# then include the IBM WebSphere Live Update Repositories in the list
# of repository locations.
#
# If the machine using this response file cannot access the internet,
# then comment out the IBM WebSphere Live Update Repositories and
# specify the URL or UNC path to custom intranet repositories and
# directory paths to local repositories to use.
#
##### -->

<server>
  <!-- ##### IBM WebSphere Live Update Repositories #####
  # These repositories contain IBM HTTP Server offerings,
  # and updates for those offerings
  #
  # To use the secure repository (https), you must have an IBM ID,
  # which can be obtained by registering at: http://www.ibm.com/account
  # or your Passport Advantage account.
  #
  # And, you must use a key ring file with your response file.
  ##### -->
  <repository location="http://www.ibm.com/software/repositorymanager/com.ibm.websphere.IHS.v85" />
  <!-- <repository location="https://www.ibm.com/software/rational/repositorymanager/repositories/websphere" /> -->

  <!-- ##### Custom Repositories #####
  # Uncomment and update the repository location key below
  # to specify URLs or UNC paths to any intranet repositories
  # and directory paths to local repositories to use.
  ##### -->

```

```

<!-- <repository location='https:\w3.mycompany.com\repositories\'/> -->
<!-- <repository location='/home/user/repositories/websphere/'/> -->

<!-- ##### Local Repositories #####
# Uncomment and update the following line when using a local
# repository located on your own machine to install a
# IBM HTTP Server offering.
##### -->
<!-- <repository location='insert the full directory path inside single quotes'/> -->
</server>

<!-- ##### Uninstall Packages #####
#
# Uninstall Command
#
# Use the uninstall command to inform Installation Manager of the
# installation packages to uninstall.
#
# The modify attribute is optional and can be paired with an install
# command to add features or paired with an uninstall command to
# remove commands. If omitted, the default value is set to false.
# false = indicates not to modify an existing install by adding
# or removing features.
# true = indicates to modify an existing install by adding or
# removing features.
#
# The offering ID attribute is required because it specifies the
# offering to be uninstalled. The example command below contains the
# offering ID for IBM HTTP Server.
#
# The version attribute is optional. If a version number is provided,
# then the offering will be uninstalled at the version level specified
# If the version attribute is not provided, then the default behavior is
# to uninstall the latest version. The version number can be found in
# the repository.xml file in the repositories.
# For example, <offering ... version='8.5.0.20110617_2222'>.
#
# The profile attribute is required and must match the package group
# name for the offering to be uninstalled.
#
# The features attribute is optional. If there is no feature attribute,
# then all features are uninstalled. If features are specified, then
# only those features will be uninstalled.
# Features must be comma delimited without spaces.
#
# The feature values for IBM HTTP Server include:
# arch.32bit,arch.64bit
#
# Installation Manager supports uninstalling multiple offerings at once.
# Additional offerings can be included in the uninstall command,
# with each offering requiring its own offering ID, version, profile value,
# and feature values.
#
# Profile Command
#
# A separate profile command must be included for each offering listed
# in the install command. The profile command informs Installation
# Manager about offering specific properties or configuration values.
#
# The installLocation specifies where the offering will be installed.
# If the response file is used to modify or update an existing
# installation, then ensure the installLocation points to the
# location where the offering was installed previously.
#
# The eclipseLocation data key should use the same directory path to
# IBM HTTP Server as the installationLocation attribute.
#
# Include data keys for product specific profile properties.
#
##### -->
<uninstall modify='false'>
  <offering id='com.ibm.websphere.IHS.v85'
    profile='IBM HTTP Server for WebSphere Application Server V8.5'
    features='core.feature,arch.32bit'/>
</uninstall>

<profile id='IBM HTTP Server for WebSphere Application Server V8.5'
  installLocation='C:\Program Files\IBM\HTTPServer'>
  <data key='eclipseLocation' value='C:\Program Files\IBM\HTTPServer'/>
  <data key='user.import.profile' value='false'/>
  <data key='user.ihs.http.server.service.name' value='none'/>
  <data key='user.ihs.httpPort' value='80'/>
  <data key='user.ihs.installHttpService' value='false'/>
  <data key='user.ihs.http.admin.service.name' value='none'/>
  <data key='user.ihs.runSetupAdmin' value='false'/>
  <data key='user.ihs.createAdminAuth' value='false'/>
  <data key='user.ihs.installAdminService' value='false'/>
  <data key='user.ihs.win.adminServiceLogOnAsLocalSystem' value='false'/>
  <data key='user.ihs.createAdminUserGroup' value='false'/>
  <data key='user.ihs.adminPort' value=''/>

```

```

<data key='cic.selector.nl' value='en'/>
</profile>

<!-- ##### Shared Data Location #####
# Uncomment the preference for eclipseCache to set the shared data
# location the first time you use Installation Manager to do an
# installation.
#
# Eclipse cache location can be obtained from the installed.xml file found in
# Linux/Unix: /var/ibm/InstallationManager
# Windows: C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager
# from the following property:
# <property name='cacheLocation' value='C:\Program Files\IBM\IMShared'/>
#
# Open the installed.xml file in a text editor because the style sheet
# might hide this value if opened in a web browser.
# For further information on how to edit preferences, refer to the public library at:
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp?topic=/com.ibm.silentinstall12.doc/topics/r_silent_prefs.html
#
# After the shared data location is set, it cannot be changed
# using a response file or the graphical wizard.
#
# Ensure that the shared data location is a location that can be written
# to by all user accounts that are expected to use Installation Manager.
#
# By default, Installation Manager saves downloaded artifacts to
# the shared data location. This serves two purposes.
#
# First, if the same product is installed a more than once to the machine,
# then the files in the shared data location will be used rather than
# downloading them again.
#
# Second, during the rollback process, the saved artifacts are used.
# Otherwise, if the artifacts are not saved or are removed, then
# Installation Manager must have to access the repositories used to
# install the previous versions.
#
# Valid values for preserveDownloadedArtifacts:
# true = store downloaded artifacts in the shared data location
# false = remove downloaded artifacts from the shared data location
#
##### -->

<!--
<preference name='com.ibm.cic.common.core.preferences.eclipseCache' value='C:\Program Files\IBM\IMShared'/>
<preference name='com.ibm.cic.common.core.preferences.preserveDownloadedArtifacts' value='true'/>
-->

<!-- ##### Preferences Settings #####
# Additional preferences for Installation Manager can be specified.
# These preference correspond to those that are located in the graphical
# interface under File / Preferences.
#
# If a preference command is omitted from or commented out of the response
# file, then Installation Manager uses the preference value that was
# previously set or the default value for the preference.
#
# Preference settings might be added or deprecated in new versions of
# Installation Manager. Consult the online Installation Manager
# Information Center for the latest set of preferences and
# descriptions about how to use them.
#
# http://publib.boulder.ibm.com/infocenter/install/v1r4/index.jsp
#
##### -->

<!--
<preference name='com.ibm.cic.common.core.preferences.connectTimeout' value='30'/>
<preference name='com.ibm.cic.common.core.preferences.readTimeout' value='45'/>
<preference name='com.ibm.cic.common.core.preferences.downloadAutoRetryCount' value='0'/>
<preference name='offering.service.repositories.areUsed' value='true'/>
<preference name='com.ibm.cic.common.core.preferences.ssl.nonsecureMode' value='false'/>
<preference name='com.ibm.cic.common.core.preferences.http.disablePreemptiveAuthentication' value='false'/>
<preference name='http.ntlm.auth.kind' value='NTLM'/>
<preference name='http.ntlm.auth.enableIntegrated.win32' value='true'/>
<preference name='com.ibm.cic.common.core.preferences.keepFetchedFiles' value='false'/>
<preference name='PassportAdvantageIsEnabled' value='false'/>
<preference name='com.ibm.cic.common.core.preferences.searchForUpdates' value='false'/>
<preference name='com.ibm.cic.agent.ui.displayInternalVersion' value='false'/>
-->

```

```
</agent-input>
```

Uninstalling fix packs from IBM HTTP Server using the Installation Manager GUI

Distributed operating systems

You can use Installation Manager to roll back IBM HTTP Server to an earlier version.

Before you begin

During the rollback process, Installation Manager must access files from the earlier version of the package. By default, these files are stored on your computer when you install a package. If you change the default setting or delete the files using the **Remove Saved Files** option, Installation Manager requires access to the repository that was used to install the earlier version.

About this task

Complete this procedure to use Installation Manager to roll back IBM HTTP Server to an earlier version.

Procedure

1. Start Installation Manager.
2. Click **Roll Back**.
3. Select the package group to roll back.
4. Click **Next**.

Note: If you are prompted to authenticate, use the IBM ID and password that you registered with on the program Web site.

5. Select the version to which you want to roll back under **IBM HTTP Server**.
6. Click **Next**.
7. Review the summary information, and click **Roll Back**.
 - If the rollback is successful, the program displays a message indicating that the rollback is successful.
 - If the rollback is not successful, click **View Log File** to troubleshoot the problem.
8. Click **Finish**.
9. Click **File > Exit** to close Installation Manager.

Installing and configuring IBM HTTP Server on the z/OS V2R2 system

z/OS

You can configure an instance of IBM HTTP Server on the z/OS operating system. IBM HTTP Server Version 9 for WebSphere Application Server for z/OS is a base element of z/OS V2R2 and later.

Before you begin

Attention: IBM HTTP Server now installs with the base operating system on z/OS V2R2 and later. No separate installation is required. IBM HTTP Server Version 9 installs in the `/usr/lpp/ihsa_zos` directory.

Prior to using the installer program:

- If you are installing the product for the first time, then create a System Authorization Facility (SAF) user ID and group for IBM HTTP Server. For information, see the topic about required z/OS system configurations.

The examples that follow in this topic assume a server user ID of WWWSERV and a server group of WWWGROUP.

- Create an installation directory for the configuration files for the server instance. For more information, see the topic about migrating and installing IBM HTTP Server on z/OS systems.

The examples that follow in this topic assume an installation directory of /etc/websrv1. Set the directory permissions to 770 and the directory ownership to the server user ID and group:

```
mkdir /etc/websrv1
chown WWWSERV:WWWGROUP /etc/websrv1
chmod 770 /etc/websrv1
```

- If you are installing the product for the first time, then enable the administrative console to modify the httpd.conf file by adding the WebSphere Application Server control region user ID to the IBM HTTP Server group using SAF. For example, to add a user ASCR1 to the group WWWGROUP, type the following command:

```
CONNECT ASCR1 GROUP (WWWGROUP) OWNER (WWWGROUP)
```

Attention: IBM HTTP Server on z/OS V2R2 installs with the base operating system and no separate installation is required. For older releases of z/OS, follow the IBM Installation Manager information in this topic.

About this task

Using the installer program, perform the following tasks to install a running instance of IBM HTTP Server for z/OS on your machine.

Procedure

1. Log in to the z/OS UNIX System Services shell with the user ID that runs the installer. (See the *Before you begin* section for this topic.) Change the directory to the IBM HTTP Server product code directory:

```
cd /usr/lpp/ihsa_zos
```

2. Set the umask value to 022 by specifying `umask 022`. To verify that the umask value is set to 022, run the `umask` command.
3. Run the installer program to install the product files into the installation directory, perform initial customization, and create symbolic links from the installation directory to the product directory.

```
bin/install_ihs -admin server_installation_directory server_port
```

Three parameters can be used to invoke the installer program.

- Optional: The `-admin` keyword, which allows you to use the administrative console to modify the `httpd.conf` file.
- The installation directory for the server instance. This must not be the same as the product directory.
- Optional: The non-SSL port for the web server. The default port is 80. You can also change the port on the `Listen` directive.

The following examples invoke the installer program from the administrative console. You can invoke the command with or without support for modifying the `httpd.conf` file. For both examples, /etc/websrv1 is the installation directory, and 80 is the non-SSL port for the Web server.

- This example invokes the command with support for modifying the `httpd.conf` file.

```
bin/install_ihs -admin /etc/websrv1 80
```


- This example invokes the command without support for modifying the `httpd.conf` file.

```
bin/install_ihs /etc/websrv1 80
```

Note: If your product directory path contains symbolic links, point the symbolic links to the following default product directory: `/usr/lpp/ihsa_zos`. If you do not use the default product directory, you must invoke the installation script using its absolute path, such as `/WebSphere/8.5/SMPE/bin/install_ihs`. If you do not use of the two options, IBM HTTP Server creates physical links, not logical links, when it creates the symbolic links for the installation directory.

4. Optional: This step is optional unless the administrative console is configured to start and stop IBM HTTP Server. You can start the IBM HTTP Server instance from the MVS™ console by creating a JCL cataloged procedure for the instance. For more information, see the topic about using JCL procedures to start IBM HTTP Server on z/OS. Ensure that the JCL procedure is assigned to the user and group you defined for IBM HTTP Server, as described in the topic about performing required z/OS system configurations.

Note: The PARM value on your JCL cataloged procedure is limited to 100 characters. Since the PARM value contains the installation directory (`&DIR`), the total length could exceed the 100 character limit if the directory path is too long. The path name length needs to be taken into consideration when choosing the installation directory. If the installation directory path name is too long, it is possible to use a shorter named path in the JCL that is symbolically linked to the original installation directory path name.

5. Optional: You can create multiple instances of IBM HTTP Server by running the IBM HTTP Server installer program more than once. However, you must specify a different installation directory each time you run the installer program.

Results

Perform the following steps to confirm that you have successfully installed a running version of the product on your machine:

1. Log in to the OMVS shell using the server user ID. Verify that the server user ID has a non-zero UID value. Change the directory to the server instance's installation directory:

```
cd /etc/websrv1
```

2. Run the following commands to verify the installation of the program:

```
apachectl -v and apachectl configtest
```

The following sample output is an example of a successful program installation:

```
# bin/apachectl -v
Server version: IBM_HTTP_Server/9.0.0.0 (Unix)
Server built:   Mar 27 2015 12:38:02
# bin/apachectl configtest
Syntax OK
```

The actual version string and build date varies.

3. Start IBM HTTP Server.

```
bin/apachectl start
```

4. Point a web browser to the IP name or address of your z/OS system, using either the non-SSL port number you specified when running the installer program, or the default port of 80. You should see the IBM HTTP Server default home page.
5. Stop IBM HTTP Server by running the following command:


```
bin/apachectl stop
```

What to do next

- Install and configure the WebSphere Application Server plug-in for IBM HTTP Server.
- For information about editing the IBM HTTP Server configuration file, `httpd.conf`, and information about supported Apache modules, see the topic about configuring IBM HTTP Server.

Typical changes that you can make to the configuration file are:

- Edit the `DocumentRoot` directive to point to the Web pages for your site.
- Enable the WebSphere Application Server plug-in for IBM HTTP Server by adding the following directives to the end of `httpd.conf`:

```
LoadModule was_ap24_module <plugin_config_hfs>/bin/mod_was_ap24_http.so
WebSpherePluginConfig /path/to/existing/plugin-cfg.xml
```

If the plug-in configuration file has been used with a WebSphere Application Server Version 5.0 or 5.1 plug-in, then the file is in EBCDIC. Before using the file with this WebSphere Application Server Version 6.0 or higher plug-in, you need to convert it to ASCII. The following example is for converting the plug-in configuration file from EBCDIC to ASCII:

```
$ iconv -f IBM1047 -t ISO8859-1 < /path/to/existing/plugin-cfg.xml \
> /path/to/ascii/plugin-cfg.xml
```

- Enable SSL support by adding the following directives to the end of `httpd.conf`:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 443
<VirtualHost *:443>
SSLEnable
</VirtualHost>
SSLDisable
Keyfile /saf saf-keyring-name
```

The `Keyfile` directive can instead specify an HFS file name using the syntax: `Keyfile /path/to/keyfile.kdb`. The `.sth` file must be in the same directory as the `.kdb` file. For more information, see “Securing with SSL communications” on page 118 and “SSL directives” on page 134.

- Enable `mod_status` by removing the comment delimiter in the default configuration file highlighted in the following example:

```
<IfModule mod_status.c>
ExtendedStatus On
</IfModule>
...
#<Location /server-status>
#   SetHandler server-status
#   Order deny,allow
#   Deny from all
#   Allow from .example.com
#</Location>
```

If you want to restrict access to specific networks, uncomment the sample `mod_access` configuration, but modify the *Allow from* directive to specify the proper domain or network.

- You can install the Web server to an HFS shared R/W by multiple hosts in a sysplex.

There are special configuration requirements for components of the Web server which utilize AF_UNIX sockets. AF_UNIX sockets are not supported by an HFS which are shared R/W, so configuration directives are used to place the AF_UNIX sockets on a filesystem owned by the host on which the Web server runs.

- If mod_ibm_ssl is loaded, use the SSLCachePortFilename directive to specify a file on a filesystem owned by the local host.
- If mod_fastcgi is loaded, use the FastCGIpcDir directive to specify a directory on a filesystem owned by the local host.
- Add support for the administrative console after the initial installation.
 - Run the bin/enable_admin script to set the permissions needed to modify the httpd.conf file from the administrative console.
 - To modify the httpd.conf file from the administrative console, you must add the control region user ID to the IBM HTTP Server group using SAF. For example, to add a user *ASCR1* to the group *WWWGROUP*, type the following command:


```
CONNECT ASCR1 GROUP (WWWGROUP) OWNER (WWWGROUP)
```
 - To use the administrative console to start and stop IBM HTTP Server, you must create a cataloged JCL procedure. For information, see the topic about using JCL procedures to start IBM HTTP Server on z/OS. Ensure that the JCL procedure is assigned to the user and group you defined for IBM HTTP Server, as described in the topic about performing required z/OS system configurations.

Installing and configuring IBM HTTP Server on the z/OS V2R1 system

z/OS

You can configure an instance of IBM HTTP Server on the z/OS operating system after installing IBM HTTP Server code using IBM Installation Manager.

Before you begin

Prior to using the installer program:

- Ensure that your environment meets the prerequisites for the application server. For more information, see the Preparing the base operating system topic.
- Install the IBM HTTP Server product code using IBM Installation Manager.
- Mount the file system containing this directory on the z/OS system where the IBM HTTP Server instance will run.
- Perform the z/OS system configurations that are required for IBM HTTP Server.
- If you are installing the product for the first time, then create a System Authorization Facility (SAF) user ID and group for IBM HTTP Server. For information, see the topic about required z/OS system configurations.

The examples that follow in this topic assume a server user ID of *WWWSERV* and a server group of *WWWGROUP*.

- Create an installation directory for the configuration files for the server instance. For more information, see the topic about migrating and installing IBM HTTP Server on z/OS systems.

The examples that follow in this topic assume an installation directory of `/etc/websrv1`. Set the directory permissions to 770 and the directory ownership to the server user ID and group:

```
mkdir /etc/websrv1
chown WWWSERV:WWWGROUP /etc/websrv1
chmod 770 /etc/websrv1
```

- If you are installing the product for the first time, then enable the administrative console to modify the `httpd.conf` file by adding the WebSphere Application Server control region user ID to the IBM HTTP Server group using SAF. For example, to add a user `ASCR1` to the group `WWWGROUP`, type the following command:

```
CONNECT ASCR1 GROUP (WWWGROUP) OWNER (WWWGROUP)
```

About this task

Using the installer program, perform the following tasks to install a running instance of IBM HTTP Server for z/OS on your machine.

Procedure

1. Log in to the z/OS UNIX System Services shell with the user ID that runs the installer. (See the *Before you begin* section for this topic.) Change the directory to the IBM HTTP Server product code directory:

```
cd /usr/lpp/IHSA/V8R5
```

2. Set the `umask` value to 022 by specifying `umask 022`. To verify that the `umask` value is set to 022, run the `umask` command.
3. Run the installer program to install the product files into the installation directory, perform initial customization, and create symbolic links from the installation directory to the product directory.

```
bin/install_ihs -admin server_installation_directory server_port
```

Three parameters can be used to invoke the installer program.

- Optional: The `-admin` keyword, which allows you to use the administrative console to modify the `httpd.conf` file.
- The installation directory for the server instance. This must not be the same as the product directory.
- Optional: The non-SSL port for the web server. The default port is 80. You can also change the port on the `Listen` directive.

The following examples invoke the installer program from the administrative console. You can invoke the command with or without support for modifying the `httpd.conf` file. For both examples, `/etc/websrv1` is the installation directory, and 80 is the non-SSL port for the Web server.

- This example invokes the command with support for modifying the `httpd.conf` file.

```
bin/install_ihs -admin /etc/websrv1 80
```

- This example invokes the command without support for modifying the `httpd.conf` file.

```
bin/install_ihs /etc/websrv1 80
```

Note: If your product directory path contains symbolic links, point the symbolic links to the following default product directory: `/usr/lpp/IHSA/V8R5`. If you do not use the default product directory, you must invoke the installation script using its absolute path, such as `/WebSphere/8.5/SMPE/bin/`

install_ihs. If you do not use of the two options, IBM HTTP Server creates physical links, not logical links, when it creates the symbolic links for the installation directory.

- Optional: This step is optional unless the administrative console is configured to start and stop IBM HTTP Server. You can start the IBM HTTP Server instance from the MVS console by creating a JCL cataloged procedure for the instance. For more information, see the topic about using JCL procedures to start IBM HTTP Server on z/OS. Ensure that the JCL procedure is assigned to the user and group you defined for IBM HTTP Server, as described in the topic about performing required z/OS system configurations.

Note: The PARM value on your JCL cataloged procedure is limited to 100 characters. Since the PARM value contains the installation directory (&DIR), the total length could exceed the 100 character limit if the directory path is too long. The path name length needs to be taken into consideration when choosing the installation directory. If the installation directory path name is too long, it is possible to use a shorter named path in the JCL that is symbolically linked to the original installation directory path name.

- Optional: You can create multiple instances of IBM HTTP Server by running the IBM HTTP Server installer program more than once. However, you must specify a different installation directory each time you run the installer program.

Results

Perform the following steps to confirm that you have successfully installed a running version of the product on your machine:

- Log in to the OMVS shell using the server user ID. Verify that the server user ID has a non-zero UID value. Change the directory to the server instance's installation directory:

```
cd /etc/websrv1
```

- Run the following commands to verify the installation of the program:
apachectl -v and **apachectl configtest**

The following sample output is an example of a successful program installation:

```
# bin/apachectl -v
Server version: IBM_HTTP_Server/8.5.0.0 (Unix)
Server built:   Jan  9 2012 11:20:34
# bin/apachectl configtest
Syntax OK
```

The actual version string and build date varies.

- Start IBM HTTP Server.

```
bin/apachectl start
```
- Point a web browser to the IP name or address of your z/OS system, using either the non-SSL port number you specified when running the installer program, or the default port of 80. You should see the IBM HTTP Server default home page.
- Stop IBM HTTP Server by running the following command:

```
bin/apachectl stop
```

What to do next

- Install and configure the WebSphere Application Server plug-in for IBM HTTP Server.

- For information about editing the IBM HTTP Server configuration file, `httpd.conf`, and information about supported Apache modules, see the topic about configuring IBM HTTP Server.

Typical changes that you can make to the configuration file are:

- Edit the `DocumentRoot` directive to point to the Web pages for your site.
- Enable the WebSphere Application Server plug-in for IBM HTTP Server by adding the following directives to the end of `httpd.conf`:

```
LoadModule was_ap22_module <plugin_config_hfs>/bin/mod_was_ap22_http.so
WebSpherePluginConfig /path/to/existing/plugin-cfg.xml
```

If the plug-in configuration file has been used with a WebSphere Application Server Version 5.0 or 5.1 plug-in, then the file is in EBCDIC. Before using the file with this WebSphere Application Server Version 6.0 or higher plug-in, you need to convert it to ASCII. The following example is for converting the plug-in configuration file from EBCDIC to ASCII:

```
$ iconv -f IBM1047 -t ISO8859-1 < /path/to/existing/plugin-cfg.xml \
> /path/to/ascii/plugin-cfg.xml
```

- Enable SSL support by adding the following directives to the end of `httpd.conf`:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 443
<VirtualHost *:443>
SSLEnable
</VirtualHost>
SSLDisable
Keyfile /saf saf-keyring-name
```

The `Keyfile` directive can instead specify an HFS file name using the syntax: `Keyfile /path/to/keyfile.kdb`. The `.sth` file must be in the same directory as the `.kdb` file. For more information, see “Securing with SSL communications” on page 118 and “SSL directives” on page 134.

- Enable `mod_status` by removing the comment delimiter in the default configuration file highlighted in the following example:

```
<IfModule mod_status.c>
ExtendedStatus On
</IfModule>
...
#<Location /server-status>
#   SetHandler server-status
#   Order deny,allow
#   Deny from all
#   Allow from .example.com
#</Location>
```

If you want to restrict access to specific networks, uncomment the sample `mod_access` configuration, but modify the *Allow from* directive to specify the proper domain or network.

- You can install the Web server to an HFS shared R/W by multiple hosts in a sysplex.

There are special configuration requirements for components of the Web server which utilize `AF_UNIX` sockets. `AF_UNIX` sockets are not supported by an HFS which are shared R/W, so configuration directives are used to place the `AF_UNIX` sockets on a filesystem owned by the host on which the Web server runs.

- If `mod_ibm_ssl` is loaded, use the `SSLCachePortFilename` directive to specify a file on a filesystem owned by the local host.

- If `mod_fastcgi` is loaded, use the `FastCGIIPCDir` directive to specify a directory on a filesystem owned by the local host.
- Add support for the administrative console after the initial installation.
 - Run the `bin/enable_admin` script to set the permissions needed to modify the `httpd.conf` file from the administrative console.
 - To modify the `httpd.conf` file from the administrative console, you must add the control region user ID to the IBM HTTP Server group using SAF. For example, to add a user `ASCR1` to the group `WWWGROUP`, type the following command:


```
CONNECT ASCR1 GROUP (WWWGROUP) OWNER (WWWGROUP)
```
 - To use the administrative console to start and stop IBM HTTP Server, you must create a cataloged JCL procedure. For information, see the topic about using JCL procedures to start IBM HTTP Server on z/OS. Ensure that the JCL procedure is assigned to the user and group you defined for IBM HTTP Server, as described in the topic about performing required z/OS system configurations.

Migrating from IBM HTTP Server V5.3 for z/OS

z/OS

Learn about key differences between your current IBM HTTP Server V5.3 for z/OS server environment and the IBM HTTP Server powered by Apache environment. Understanding these key differences can help you migrate to IBM HTTP Server powered by Apache.

The content in this migration guide corresponds to parts and chapters in the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

Related information:

 [Migrating from Domino-powered to Apache-powered](#)

IBM HTTP Server V5.3 for z/OS: Part 1: Planning

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences in installation and migration that are important for planning.

The part and chapters correspond to the part and chapters in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The following topics are applicable to chapter 1:

- “Configuration and administration” on page 48
- “z/OS UNIX System Services” on page 48
- “DLL considerations” on page 48
- “Fast Response Cache Accelerator” on page 48
- “MVSDS DLL z/OS data set considerations” on page 48
- “Performance” on page 48
- “Maximum active threads” on page 48
- “How to code directives to improve web server performance” on page 49
- “Workload management” on page 49

The following topic is applicable to chapter 2:

- “Migration considerations for installation” on page 49

Configuration and administration

You can administer IBM HTTP Server only by updating the EBCDIC configuration files.

z/OS UNIX System Services

z/OS UNIX System Services user ID and UID

IBM HTTP Server runs in a UNIX process under a user ID of your choice, such as WWWSERV. Ensure that the z/OS UNIX System Services UID is not 0 . Read the “Performing required z/OS system configurations” on page 65 topic for information about the user ID and UID.

BPX.SERVER facility

If you change the thread identity of logged in users by using the SAFRunAs directive, list the web server ID in the BPX.SERVER facility.

DLL considerations

Add the DLL path to the LIBPATH statement in the *install_root/bin/envvars* file.

Fast Response Cache Accelerator

The Fast Response Cache Accelerator is not supported on the z/OS operating system for IBM HTTP Server. As alternatives, use the *mod_cache*, *mod_disk_cache*, and *mod_file_cache* modules for caching.

Note: Cache remote or dynamic files instead of static files. An exception is the caching of static files with the *mod_file_cache* module. This module speeds up static file access.

For more information about the modules, read the Apache HTTP Server documentation.

MVSDS DLL z/OS data set considerations

IBM HTTP Server does not cache or preload MVS data sets.

Performance

Tune your server by using the performance and tuning information for the IBM HTTP Server.

Maximum active threads

IBM HTTP Server V5.3 for z/OS uses the *MaxActiveThreads* directive to set the maximum number of threads that can be active at one time. IBM HTTP Server uses a hybrid model with multiple thread processes. This hybrid model uses multiple directives to manage the processes.

The *MAXClients* directive determines the limit for the total number of process threads. The *ThreadLimit* and *ThreadsPerChild* directives affect how many threads are created for each process. The *ServerLimit*, *MaxClients*, and *ThreadsperChild* directives control the maximum number of processes created.

Note: Use the configuration in the `httpd.conf` default configuration file until your server needs to process more concurrent requests. When you change the `MaxClients` directive and `ServerLimit` directives, increase or decrease their values by the same factor.

For more information about the directives, read the Apache HTTP Server documentation.

How to code directives to improve web server performance

The following directives are disabled in the default `httpd.conf` configuration file for IBM HTTP Server. Use caution when you enable these directives as they can have an affect on your web server performance.

```
HostnameLookups
IdentityCheck
MimeMagicFile
MaxRequestsPerChild
AllowOverride
Options FollowSymlinks
SMFReportInterval
```

For more information about these directives, read the Apache HTTP Server documentation.

Workload management

Follow the directions in the topic on classifying HTTP requests for workload management (WLM) to implement WLM for IBM HTTP Server.

Migration considerations for installation

Whether you are initially installing IBM HTTP Server or are migrating to a later version after your initial installation, follow the instructions for the “Installing fix packs on IBM HTTP Server using the Installation Manager GUI” on page 21 topic.

IBM HTTP Server V5.3 for z/OS: Part 2: Installing

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences that are important for installing.

The part and chapter correspond to the part and chapter in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The UID for the web server

Always define a non-zero UID for the web server. For more information, read the “Performing required z/OS system configurations” on page 65 topic.

IBM HTTP Server V5.3 for z/OS: Part 3: Using

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences that are important for using a web server.

The part and chapters correspond to the part and chapters in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The following topics apply to chapter 4:

- “Starting the server from the z/OS UNIX System Services shell”
- “Starting the server from a procedure”
- “Starting multiple instances of the server”
- “Restarting the server”

No topics apply to chapter 5.

The following topic applies to chapter 6:

- “Stopping the server”

Starting the server from the z/OS UNIX System Services shell

Start your server from the z/OS UNIX system shell by running the **apachectl** command with either the `start` or `-k start` parameters. For more information, read the “Using apachectl commands to start IBM HTTP Server” on page 71 topic.

Starting the server from a procedure

Start your server with a procedure by using the “Using JCL procedures to start IBM HTTP Server on z/OS” on page 74 topic as a guide.

Starting multiple instances of the server

Start multiple instances of your server by running the `bin/install_ihs` script multiple times as described in the Configuring an instance of IBM HTTP Server on the z/OS system topic.

Restarting the server

You can either restart the server or restart the server gracefully. To restart the server, pass the `restart` parameter to the **apachectl** command. To restart the server gracefully, pass the `graceful` parameter to the **apachectl** command. You can alternatively run the equivalent MVS console command. Read the “Using JCL procedures to start IBM HTTP Server on z/OS” on page 74 topic for more information.

Stopping the server

Stop your server from the z/OS UNIX system shell by running the **apachectl** command with the `stop` parameter. For more information, read the “Using apachectl commands to start IBM HTTP Server” on page 71 topic.

You can stop your server from the MVS console, but you cannot cancel it from the console. Read the “Using JCL procedures to start IBM HTTP Server on z/OS” on page 74 topic for more information.

IBM HTTP Server V5.3 for z/OS: Part 4: Basic configuration

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences in the basic configuration of the two web servers.

The part and chapters correspond to the part and chapters in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The following topics apply to chapter 7:

- “How to serve files”
- “How to serve directory listings” on page 52
- “How to configure the server” on page 52
- “Files to back up” on page 52

The following topics apply to chapter 8:

- “Encryption support” on page 52
- “Hardware encryption” on page 53
- “How to check whether hardware encryption is used for web server encryption” on page 53
- “Checklist for setting up a secure server” on page 54

The following topics apply to chapter 9:

- “How to set up protection for server resources” on page 54
- “Rules for specifying user names, group names, and address templates” on page 56
- “Use of group files in protection setups” on page 56
- “Access Control List files” on page 56

How to serve files

IBM HTTP Server can serve static files or run CGI script files. These files can be in default directories or directories that you specify. You can use various directives to serve these files. Use the Directory section to group the directives together and specify that they apply to a particular directory.

Static files are in the *install_root/htdocs* directory by default. You can specify an alternative directory on the Alias directive to map the alternative directory to a web address prefix. Then, you can create or copy a Directory section and have it point to the alternative directory. For example, you can copy the Directory directive that specifies the *install_root/htdocs* default directory and change from the default directory to the *install_root/static* directory.

CGI scripts run from the *install_root/cgi-bin/* directory by default. You can specify an alternative directory on the ScriptAlias directive to map the alternative directory to a web address prefix. Then, you can create or copy a Directory section and have it point to the alternative directory. For example, you can copy the Directory directive that specifies the *install_root/cgi-bin/* default directory and change from the default directory to the *install_root/cgi2* directory.

For more information about the directives, read the Apache HTTP Server documentation.

How to serve directory listings

Because the DirectoryIndex directive is set to index.html in the default httpd.conf file, IBM HTTP Server serves the directory index file of index.html for directory requests. You can set the DirectoryIndex directive to other files for IBM HTTP Server to serve. You can also add the Options directive with the Indexes argument to a new or existing Directory section so that the web server returns information for that directory. If you include a + in front of the Indexes argument, then the Directory section inherits arguments that are set on other Options directives. If neither of the DirectoryIndex and Options directives is set, the web server returns a 403 error.

For more information about the directives, read the Apache HTTP Server documentation.

How to configure the server

You can administer IBM HTTP Server only by updating the EBCDIC configuration files.

The default IBM HTTP Server configuration file is *install_root/conf/httpd.conf*. If you want to review or recover the shipped defaults, you can find them in the *install_root/conf/httpd.conf.default* file.

Files to back up

Periodically back up the following files:

- The configuration file, which by default is the *install_root/conf/httpd.conf* file
- The environment variable file, which is the *install_root/bin/envvars* file
- Secure Sockets Layer (SSL) files, such as the following files:
 - Key database files, which have a kdb extension
 - Stash files, which have an sth extension
 - Request database files, which have an rdb extension
 - Certificate revocation list files, which have a crl extension
 - Certificate files, which have an arm extension
- Output from commands like the *install_root/bin/htpasswd* command, which you can use for access control
- Hand-edited group lists
- Any content that is served in HTTP requests, such as HTML files, images, Java scripts, cascading style sheets, and CGI scripts

Encryption support

The US government and governments outside the US regulate products that are used for encryption and prohibit their export unless their key size is strictly limited. As the US government updates their export laws and governments outside the US update their import rules, the supported key lengths and cipher specifications can change.

The IBM HTTP Server supports the SSL ciphers that are listed in the “SSL cipher specifications” on page 163 topic.

Hardware encryption

You can use hardware encryption to improve the performance of SSL sessions between the client and the server. By far, the biggest gain in performance for the web server is in the SSL handshake. The handshake uses asymmetric keys and functions. The web server uses RSA technology to implement the asymmetric capability. When you implement SSL without hardware encryption, the asymmetric functions are much slower than symmetric functions. So when you implement hardware encryption with the web server, make sure that you set up your asymmetric Master Keys properly. Use the Integrated Cryptographic Services Facility (ICSF) software so that you can take advantage of the performance boost. The asymmetric Master Keys are not the same as the RSA keys of the web server.

Data Encryption Standard (DES) cipher specifications and Triple-DES cipher specifications use symmetric keys in to handle data transmission. Data transmission might or might not be faster in hardware. Whether data transmission is faster in hardware or software depends on the size of the data stream. SSL should be sending relatively small streams of data, usually 4K bytes, or less. Smaller streams of data tend to be faster in software. Mid-range streams can be faster in hardware or software. Very large streams are faster in hardware.

When you implement hardware encryption, keep in mind these points:

- The web server uses RSA technology for the SSL handshake. The handshake is an asymmetric capability and uses RSA public-private key pairs. You can generate RSA keys in software or hardware.
- If you generate the RSA keys in software, you can use RACF[®] commands or the **gskkyman** utility.
- Define RACF commands to permit user IDs and the web server ID to profiles in the CSFSERV general resource class. The CSFSERV general resource class controls the use of ICSF software.

For information about how to implement hardware encryption, read the appropriate manuals. For example, read the *z/OS Processor Resource/Systems Management Planning Guide* on the IBM support portal.. Additionally, you can read the *z/OS Cryptographic Services ICSF Administrator's Guide* and the *z/OS Cryptographic Services ICSF System Programmer's Guide*, which are available on the z/OS Internet Library.

How to check whether hardware encryption is used for web server encryption

ICSF is the software interface to the cryptographic hardware. Use this checklist to determine if your web server is working with hardware encryption.

- Verify that user IDs and the web server ID have access to ICSF.
- Check that the ICSF started task is active.
- Do one or both of the following tasks through the ICSF TSO panels to ensure that ICSF is working properly:
 - Check that ICSF has PKA Master Keys defined.
 - Generate a PKA Master Key successfully.

Checklist for setting up a secure server

To enable Transport Layer Security (TLS), use the SSL virtual host example in the `conf/httpd.conf.default` file. The example contains elements that are required to enable TLS, including a `Listen` directive, the `SSLEnable` directive, and a `mod_ibm_ssl` module.

IBM HTTP Server uses CMS SSL keystore files, which have a `kdb` extension. You can use the `gskkyman` utility or the RACF `RACDCERT` command to create and administer a keystore file.

Attention: Do not share these keystore files between z/OS and distributed platforms.

How to change the default order of the encryption levels that the web server uses

You can use the `SSLCipherSpec` directive to control order of the encryption levels. IBM HTTP Server always enforces the order of preference. Read about the `SSLCipherSpec` directive in the topic on SSL directives.

How to set up protection for server resources

The following steps are in the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS. The information that is associated with each step is information that is needed to do the step in IBM HTTP Server.

- Step 1. Activate protection on the server.
You have nothing to do for this step because IBM HTTP Server by default loads the common modules that limit access to resources
- Step 2. Specify which requests you want the server to accept.
Use configuration sections to enclose protection-related configuration directives. Read about configuration sections in the Apache HTTP Server documentation.
For resources in the hierarchical file system (HFS), use the `<Directory>` and `<DirectoryMatch>` directives to enclose protection directives. For other resources that are not in the HFS, such as those resources that plug-ins serve, use the `<Location>` and `<LocationMatch>` directives.
- Step 3. Decide which protection options to use.
IBM HTTP Server offers a number of different protection mechanisms from which you can choose:
 - Host-based access control via the `mod_authz_host` module. The `mod_authz_host` module permits or denies individual IP addresses or subnets.
 - Various modules interoperate to provide user ID and password authentication. These capabilities include HTTP basic authentication for files databases, and LDAP; HTTP digest authentication; and SSL client certificate authentication.
 - Various modules interoperate to provide authorization. These capabilities include groups, Lightweight Directory Access Protocol (LDAP), and SSL client certificates.

The server processes requests by first checking host-based access control. Then, it checks authentication and access control. If the `Satisfy` directive is set to `any`, the request only must meet either host-based access control or authorization

requirements. Any match of a Require authorization directive allows access. However, granting access based on the match of multiple Require directives is not possible.

CAUTION: You can use the <Limit> and <LimitExcept> directives to constrain protection methods to individual HTTP request methods, but carefully test this approach.

- Step 4. Create protection setups.

You can check IBM HTTP Server passwords against user and group password files. However, if you want to check IBM HTTP Server passwords against the local system, then specify the AuthBasicProvider SAF directive. You can optionally change the SAF user ID under which a request is served by specifying the SAFRunAs directive.

If you want to request SSL client authentication on a virtual host basis, then specify the SSLClientAuth required directive. Use the SSLClientAuthRequire directive to specify attribute values, or groups of attribute values, that must be validated against a client certificate before the server allows access to the protected resource.

Use the following examples to guide you in creating your protection setups:

- Control access to resources by using the Order, allow, and deny directives:

```
Alias /my-app /opt/my-app/htdocs
```

```
<Directory /opt/my-app/htdocs>
# Allow requests that match the allow directives. Then, deny requests that match the deny directives.
# Then, deny requests that do not match the allow or deny directives.
Order allow,deny
# Allow access only to those users from the local host.
Allow from 127.0.0.1
</Directory>
```

- Control access to resources by using the order, allow, and deny directives. Additionally, use basic authentication so that a user provides a user ID and password to access the resources. Specify the file that contains the user IDs and passwords.

```
<Directory /opt/my-app/htdocs/members-only>
Order allow,deny
Allow from 127.0.0.1
# Add HTTP basic authentication.
AuthType Basic
AuthBasicProvider file
AuthName "Login with your example.com user ID."
# Use the htpasswd utility in the <install_root>/bin/htpasswd file to maintain the passwords.
# Store the userid and password file in a directory other than the one that it is protecting.
AuthUserFile /opt/my-app/users.passwd
Require valid-user
</Directory>
```

- Allow only the user ID of administrator to access the resources.

```
<Directory /opt/my-app/htdocs/admin>
...
Require user administrator
</Directory>
```

- Allow only the user group of admins to access the resources. Specify the file that contains the user groups.

```
<Directory /opt/my-app/htdocs/admin>
...
# text file with multiple group-name: member1 member2... lines
# Store the group file in a directory other than the one that it is protecting.
AuthzGroupFile /auth/groups
Require group admins
</Directory>
```

- Allow the local host to access resources as if it is an administrator.

```
<Directory /opt/my-app/htdocs/admin>
...
Require group admins
Satisfy any
Order allow,deny
Allow from 127.0.0.1
</Directory>
```

- Step 5. Limit access to individual files.

You can limit the files that a user accesses by nesting the <Files> directive or the <FilesMatch> directive inside of the <Directory> directive or the <DirectoryMatch> directive.

Rules for specifying user names, group names, and address templates

You cannot allow access based on a combination of a user name and an address, such as bob@192.168.1.1 and steve@192.168.2.2, without writing your own Apache module for authorization.

Use of group files in protection setups

A group file in IBM HTTP Server is only a mapping from a group name to a list of users. It cannot have nested definitions or include address specifications.

Access Control List files

IBM HTTP Server does not have access control list files. You can use .htaccess files to limit access to resources. However, avoid using .htaccess files if you can update the httpd.conf file because using .htaccess files slows down your server. As an alternative, include directives in a <Directory> directive and put all the directives in the httpd.conf file.

IBM HTTP Server V5.3 for z/OS: Part 5: Advanced configuration

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences in the advanced configuration of the two web servers.

The part and chapters correspond to the part and chapters in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The following topic applies to chapter 10:

- “Caching” on page 57

The following topics apply to chapter 11:

- “Log types” on page 57
- “Log maintenance” on page 58
- “Filters for the access log” on page 58
- “Reports for logs” on page 58
- “System Management Facility record types” on page 58

The following topics apply to chapter 12:

- “The HTCounter and other CGI programs” on page 58

- “Server-side includes” on page 58
- “Server-side image maps” on page 58

The following topics apply to chapter 13:

- “Modes of operation” on page 59
- “Server activity monitor” on page 59
- “Simple Network Management Protocol (SNMP)” on page 59
- “z/OS operator console modify command for System Management Facilities” on page 59
- “SMF record formats” on page 59
-

The following topic applies to chapter 14:

- “Website ratings with Platform for Internet Content Selection (PICS)” on page 59

The following topic applies to chapter 15:

- “Lightweight Directory Access Protocol (LDAP) information retrieval” on page 59

The following topic applies to chapter 16:

- “Your server as a proxy” on page 60

The following topic applies to chapter 17:

- “Multiple IP addresses or virtual hosts for your server” on page 60

Caching

The Fast Response Cache Accelerator is supported in IBM HTTP Server, but is not supported for the z/OS operating system. Use these alternatives for caching:

- Use the `mod_expires` module to set browser cache headers. Read the `mod_expires` module topic in the Apache HTTP Server documentation for further information.
- Use the `mod_cache` module and the `CacheEnable` directive to cache local files. Read the `mod_cache` module topic in the Apache HTTP Server documentation for further information.

Note: Do not use in-memory or on-disk caching for any static content.

Note: Use caching for generated, proxied, or dynamic content.

Log types

IBM HTTP Server has three main types of logs:

Error logs

Have a fixed format. Configure them by using the `ErrorLog` directive.

Access logs

Have custom formats. IBM HTTP Server updates them on a per-request basis. Configure them by using the `LogFormat` and `CustomLog` directives. Read the Apache HTTP Server documentation for information about custom log formats.

Module-specific diagnostic logs

Have logs for particular modules. Examples include the rewriting log file

for the `mod_rewrite` module and the script error log file for the `mod_cgi` module. Read the log files topic in the Apache HTTP Server documentation for further information.

Read the log files topic in the Apache HTTP Server documentation for further information.

Log maintenance

You can pipe access and error logs to an external program for rotation or maintenance. IBM HTTP Server includes a piped logger that does simple time based and size based rotation. IBM HTTP Server does not manage the rotated log files. You must manage these logs external to the server or through the custom pipe logger. Read the topic in the Apache HTTP server documentation about the `Rotatelogs` program for more information.

Filters for the access log

IBM HTTP Server uses conditional logging for the access log instead of filters. Read about conditional logging in the log files topic in the Apache HTTP Server documentation.

Reports for logs

IBM HTTP Server by default produces logs in the Apache HTTP Server standard formats. Various tools can use these logs to generate reports.

System Management Facility record types

You can use two different System Management Facility (SMF) record types to record IBM HTTP Server data to SMF.

- Record aggregate server statistics periodically by using the `mod_mpmstats` module and setting the `SMFReportInterval` directive to a non-zero value.
- Record access log like data by using the `mod_smf` module and setting the `SMFRecord` directive to `on`. You can set the `SMFRecord` directive in any scope of the `Location` directive or the `Directory` directive in the `httpd.conf` file.

For further information, read the topic about configuring the server for SMF recording. The topic includes record format information.

The HTCounter and other CGI programs

IBM HTTP Server does not include an `HTCounter` program or any other CGI program.

Server-side includes

You can use server-side includes with IBM HTTP Server. Implement server-side includes by using the `mod_include` module.

Server-side image maps

You can use server-side image maps with IBM HTTP Server. However, the technology is deprecated. Implement server-side image maps by using the `mod_imagemap` module.

Modes of operation

IBM HTTP Server has one mode of operation, which is a multithreaded, multiprocess server. The IBM HTTP Server parent process starts servers dynamically in response to thread utilization instead of using Workload Management (WLM).

Server activity monitor

You can implement a web-accessible monitoring interface or periodically collect server statistics in an error log.

To implement the web-accessible monitoring interface, use the `mod_status` module.

To periodically collect server statistics, use the `mod_mpmstats` module. The module generates messages in the error logs. The messages contain the statistics.

Simple Network Management Protocol (SNMP)

IBM HTTP Server does not include an SNMP subagent or provide any SNMP data.

z/OS operator console modify command for System Management Facilities

You cannot use the z/OS operator console `modify` command to manage your System Management Facilities (SMF). Instead, use IBM HTTP Server directives. The `SMFReportInterval` directive controls how often aggregate server statistics are recorded in SMF. The `SMFRecord` directive controls which URL patterns record access-log like details in SMF.

You can create your own SMF custom module to manage SMF. Use the sample SMF custom module as a guide.

SMF record formats

Use the `mod_mpmstats` module to record type 103 subtype 13 records. Read the topic about configuring the server for SMF recording.

Use the `mod_smf` module to record type 103 subtype 14 records. Read the topic about the `mod_smf` module.

Website ratings with Platform for Internet Content Selection (PICS)

Platform for Internet Content Selection (PICS) is not supported in IBM HTTP Server.

Lightweight Directory Access Protocol (LDAP) information retrieval

Use the `mod_ldap` module and the `mod_authnz_ldap` module to do LDAP authentication and authorization.

Your server as a proxy

You can use IBM HTTP Server as a forward proxy or as a reverse proxy. To implement a proxy, use the `mod_proxy` module.

To configure IBM HTTP Server as a forward proxy, generally use the `<Proxy>` container and the `ProxyRequests` directive. As a forward proxy, IBM HTTP Server supports Secure Sockets Layer (SSL) tunneling for SSL clients. Use the `mod_proxy_connect` module to do SSL tunneling.

IBM HTTP Server can be a reverse proxy to HTTPS, HTTP, and FTP origin servers. However, reverse proxy support to FTP origin servers is deprecated. To configure IBM HTTP Server as a reverse proxy, use directives that begin with `ProxyPass`. Additionally, set the `SSLProxyEngine` directive to `on` to use IBM HTTP Server as a reverse proxy to an HTTPS origin server.

Proxy capability is turned off by default.

When you use IBM HTTP Server as a web server, the Apache HTTP Server caching capability and the Fast Response Cache Accelerator capability are supported. However, when you use IBM HTTP Server as a proxy, the Apache HTTP Server caching capability is supported, but the Fast Response Cache Accelerator capability is not.

Multiple IP addresses or virtual hosts for your server

The documentation on Apache virtual hosts provides comprehensive documentation for IP and name-based virtual hosts.

IBM HTTP Server V5.3 for z/OS: Part 6: Programming

z/OS

Various capabilities in IBM HTTP Server V5.3 for z/OS are available in IBM HTTP Server, but implemented differently. Learn about key differences in programming for the two web servers.

The part and chapters correspond to the part and chapters in publication number SC34-4826-09 of the *z/OS HTTP Server Planning, Install, and Using* guide for IBM HTTP Server V5.3 for z/OS.

The following topic applies to chapter 18:

- “CGI and FastCGI programs”

The following topic applies to chapter 19:

- “Go Webserver Application Programming Interface (GWAPI)” on page 61

The following topic applies to chapter 20:

- “Lightweight Directory Access Protocol (LDAP)” on page 61

CGI and FastCGI programs

IBM HTTP Server supports Common Gateway Interface (CGI) and FastCGI programs, but does not include any information about how to write code for them.

Go Webserver Application Programming Interface (GWAPI)

IBM HTTP Server provides Apache APIs that are similar to GWAPI in IBM HTTP Server V5.3 for z/OS, but the binary and sources are not compatible. These Apache APIs are part of the open source Apache HTTP Server, on which IBM HTTP Server is based.

IBM HTTP Server includes source for one example module and public headers that define the Apache API.

- The example is in the <install_root>/example_module/mod_example.c file path.
- The public headers are in the <install_root>/include/ subdirectory.

Apache HTTP Server includes source for a number of modules.

- You can download the source and review these modules for pointers on how to accomplish a task in a plug-in.
- Generally, use the apxs tool to compile and install Apache modules.

Learn more about Apache modules by reading white papers and publicly available books. Two useful white papers document the process of writing simple Apache modules for the z/OS operating system. One of the white papers contains information about the classification of URL requests in IBM HTTP Server by using WLM. Another white paper contains information about extending IBM HTTP Server with custom modules.

Attention: REXX plug-ins are not supported. You cannot develop Apache modules by using REXX.

Lightweight Directory Access Protocol (LDAP)

The plug-ins that you develop for IBM HTTP Server by default have limited access to LDAP data. If you manage your own connections, you can have more access.

Use the AuthLDAPUrl directive to define attributes. IBM HTTP Server adds each attribute that you define as an environment variable that is internal to the server itself. When IBM HTTP Server adds an attribute, it puts a prefix of AUTHENTICATE_ on the attribute name.

Use the mod_ldap module so that your LDAP modules can use connection pooling. For further information, consult the include/util_ldap.h file in the Apache HTTP Server source.

Running multiple instances of IBM HTTP Server from a single install

Distributed operating systems

z/OS

Run multiple, independent instances of IBM HTTP Server from a single installation. It is seldom necessary to run multiple instances, as features like virtual hosts allow a single instance to efficiently serve many sites, but in some cases it is necessary. If you need to securely administer your sites by different administrators, for example, you must run separate instances that each use their own configuration files.

Before you begin

This topic is primarily for AIX, HP-UX, Linux, Solaris, and Windows operating systems. On the z/OS platform, the `install_ihs` command creates a separate directory for each instance without creating another copy of the product. See the z/OS topic for configuring IBM HTTP Server for more information.

Before configuring multiple instances, consider if your problem can be solved by using virtual hosts and/or having IBM HTTP Server listen on multiple addresses and ports. The advantage of a single instance is that it uses less resources to serve the same requests as multiple instances.

Note: When you follow the examples, change "this_instance" to a unique name for each instance.

Procedure

1. Create a separate main configuration file, normally the `httpd.conf` file, for each instance.

Note: To reduce duplication, store common directives in common files and import these into the separate, main configuration files with the *Include* directive.

We'll call the configuration file `conf/this_instance.conf` for the rest of these steps.

Here is a simple example of a configuration file for an instance:

```
Listen 10.0.0.1:80
PidFile instance1/httpd.pid
ErrorLog instance1/error.log
CustomLog instance1/access.log common
# Other directives that make this instance behave uniquely
Include conf/common.conf
```

A real configuration file would have more directives in it to make this instance behave differently than the other instances.

2. Configure the port settings in the configuration files. You cannot use a combination of listen port and listen IP address for more than one instance. Check the Listen directives in each configuration file, and verify that they are unique. See information on the Listen directive for Apache HTTP Server for more information.
3. Configure settings for logging and other special files. Any files that are normally stored in the `install_root/logs` directory cannot be shared between instances. Each instance must have unique values for the following directives:

PidFile

Applicable to all configurations. See the information on the PidFile directive for Apache HTTP Server.

ScriptSock

Applicable to non-Windows configurations with `mod_cgid` enabled.

ErrorLog

Applicable to all configurations. See the information on the ErrorLog directive for Apache HTTP Server.

CustomLog or TransferLog

Applicable to all configurations. See the information on the CustomLog directive or the TransferLog directive for Apache HTTP Server.

SSLCachePortFilename

Applicable to all non-Windows configurations with SSL enabled. See the information on the SSLCachePortFilename directive.

SSLCachePath

Applicable when all of the following conditions are true:

- Platform is not Windows.
- SSL is enabled.
- SSLCacheDisable directive is not configured.
- bin/apachectl has been modified to specify a different -d flag, or bin/apachectl is launched with an explicit -d flag.
- The directory specified by the -d flag does not contain the file bin/sidd.

See the information on the SSLCachePath directive for Apache HTTP Server. See information on the SSLCachePath directive.

Other optional directives that specify a file path, like logging or tracing.

4. **AIX** **Windows** Ensure that no more than one IHS instance has the fast response cache accelerator (FRCA), or AFPA, enabled.

Note: FRCA/AFPA has been deprecated starting with V7.0 and its use is discouraged. There is no support for Windows Vista, Windows 2008, or any later Windows operating systems.

5. Start or stop the IHS server instance.
 - **AIX** **HP-UX** **Linux** **Solaris** Use these commands to start and stop IHS:

```
# cd /install_dir
# bin/apachectl -k start -f conf/this_instance.conf
# bin/apachectl -k stop -f conf/this_instance.conf
```

Alternatively, you can create a copy of apachectl for each instance, and update the commands in each copy to include "-f conf/this_instance.conf".

- **Windows** Use these commands to setup a new instance:

```
cd \install_dir
bin\Apache.exe -f conf/this_instance.conf -k install -n IHS-this_instance
```

Choose one of these commands to start and stop IHS:

- Use this command:

```
net start IHS-this_instance
```
- Use this command:

```
cd \install_dir
bin\Apache.exe -k install -n IHS-this_instance.conf
```
- Find IHS-this_instance in the Services interface for Microsoft Windows.

See the topic on starting and stopping IBM HTTP Server for more information.

Chapter 3. Administering and configuring IBM HTTP Server

Distributed operating systems

z/OS

Learn how to administer and configure IBM HTTP Server, including: Secure Socket Layer (SSL), Key management, Lightweight Directory Access Protocol (LDAP) and System Authorization Facility (SAF) for z/OS systems

Performing required z/OS system configurations

Before starting IBM HTTP Server, there are required z/OS system configurations that you must set up.

About this task

In order to run IBM HTTP Server, you must set the following z/OS system configurations:

- Set the memlimit parameter.
- Configure a mechanism for allowing access to low ports.
- Required System Authorization Facility (SAF) configurations.
 - Create a user ID and group for IBM HTTP Server.
 - Set program control for required MVS data sets.
 - Set program control for HFS files.
 - Set program control for z/OS System SSL.
 - Access to SAF key rings.
 - Permitting user IDs to CSFSERV for hardware encryption.
 - Using cryptographic hardware for key storage (optional).
- Setting environment variable * _BPX_JOBNAME (optional).

Procedure

- **Set the MEMLIMIT parameter.** The **MEMLIMIT** parameter controls the amount of virtual memory higher than two gigabytes for a particular address space. The default setting for **MEMLIMIT** is 0. However, all binary programs provided with IBM HTTP Server are 64-bit applications, and these applications will not be operational with the default setting for **MEMLIMIT**.

The **MEMLIMIT** parameter can be set:

- In the OMVS segment of the user ID used to run the server:
`ALTUSER WWWSERV OMVS(MEMLIMIT(512M))`
- In the parmlib member **SMFPRMxx**. Setting the parmlib member **SMFPRMxx** will establish the system-wide **MEMLIMIT** default.

For a complete description of how to set **MEMLIMIT**, refer to the section “Limiting the use of memory objects” in *z/OS MVS Programming Extended Addressability Guide* (SA22-7614). You can link to this document from the *z/OS Internet Library*.

IBM HTTP Server requires approximately 5.4 megabytes of 64-bit virtual memory per thread. The minimum recommended **MEMLIMIT** setting for proper IBM HTTP Server operation is: $6 * (\text{ThreadsPerChild} + 3)$ megabytes.

- Configure a mechanism for allowing access to low ports. The Web server user ID must have access to the TCP ports on which it will handle client connections. If

port values less than 1024 are used, such as Web server ports 80 and 443, special configuration is required to allow the Web server to bind to the port.

You can use one of the following mechanisms to allow access to low ports:

- Set the PORT directive in the TCP/IP configuration.
- Disable RESTRICTLOWPORTS in the TCP/IP configuration.
- Code the Web server job name on a PORT statement in the TCP/IP configuration.
- Code a wildcard for the job name on a PORT statement in the TCP/IP configuration.
- Code SAF and a safname value on the PORT statement in the TCP/IP configuration, and permit the Web server user ID read access to the SAF FACILITY class profile EZB.PORTACCESS.sysname.stackname.safname.

For more information on configuration methods for allowing access to low ports, refer to the sections "Port access control" and "Setting up reserved port number definitions in PROFILE.TCPIP" in *z/OS Communications Server IP Configuration Guide* (SC31-8775). You can link to this document from the *z/OS Internet Library*.

For an explanation of how Unix System Services jobnames (such as those for IBM HTTP Server instances) are determined, refer to the section "Generating jobnames for OMVS address spaces" in *z/OS UNIX System Services Planning* (GA22-7800). Link to this document from the *z/OS Internet Library*.

- Required System Authorization Facility (SAF) configurations.

- Create a user ID and group for IBM HTTP Server.

You can use a new or existing user ID. It must have an OMVS segment and the UID cannot be zero. The following example contains RACF commands to create a new user and group.

Password example

```
ADDGROUP WWWGROUP OMVS(GID(999))
ADDUSER WWWSERV DFLTGRP(WWWGROUP) OMVS(UID(999)) PASSWORD(password)
```

Password phrase example

```
ADDGROUP WWWGROUP OMVS(GID(999))
ADDUSER WWWSERV DFLTGRP(WWWGROUP) OMVS(UID(999)) PHRASE('my0users@99#701_workgroup')
```

The security administrator should define the password for the Web server user ID, instead of allowing it to default, to prevent an unauthorized user from being able to log in with that user ID. The ALTUSER command can be used to modify the password of an existing user ID.

Note: If you use a JCL cataloged procedure to start an IBM HTTP Server instance, create a SAF STARTED profile to assign the server user ID and group ID to the server started task. For example, to use a cataloged procedure named WEBSRV1:

```
RDEFINE STARTED WEBSRV1.* STDATA(USER(WWWSERV) GROUP(WWWGROUP) TRACE(YES))
```

- Set program control for required MVS data sets.

Ensure that program control is turned on for the following MVS data sets. For *hlq*, enter the high level qualifier for your system installation, for example: SYS1.LINKLIB.

- hlq.LINKLIB
- hlq.SCEERUN
- hlq.SCEERUN2
- hlq.SCLBDLL

The following example shows how to turn on program control using RACF commands. If you are using another security product, refer to that product's documentation for instructions. If you are turning on program control for the first time, you should use RDEFINE statements instead of RALTER statements:

```
RALTER PROGRAM * ADDMEM('hlq.LINKLIB'//NOPADCHK) UACC(READ)
RALTER PROGRAM * ADDMEM('hlq.SCEERUN'//NOPADCHK) UACC(READ)
RALTER PROGRAM * ADDMEM('hlq.SCLBDLL') UACC(READ)
SETROPTS WHEN(PROGRAM) REFRESH
```

In this example, an asterisk (*) is used to specify all programs in the data set.

- Set program control for HFS files.

The SMP/E installation logic enables the program control bit for the provided libraries and executable files that need it. If you install custom plug-in modules, use the **extattr** command to enable the APF and Program Control flags. For example:

```
# extattr +ap /opt/IBM/HTTPServer/modules/mod_jauth.so
```

In this example, substitute the IBM HTTP Server installation location for /opt/IBM/HTTPServer/. (You can build custom plug-in modules using the apxs script that is provided.)

- Set program control for z/OS System SSL.

If you set up your IBM HTTP Server to provide secure communications over the Internet, IBM HTTP Server uses z/OS System Secure Sockets Layer (SSL) to establish the secure connections. Before IBM HTTP Server can use System SSL, you must:

- Add the System SSL load library (hlq.SIEALNKE) to the system link list or to the STEPLIB DD concatenation in the HTTP Server cataloged procedure
- Set program control hlq.SIEALNKE in RACF.

The variable *hlq* is the high level qualifier for your system installation, for example: SYS1.SIEALNKE.

To turn on program control using RACF, issue the following command:

```
RALTER PROGRAM * ADDMEM('hlq.SIEALNKE'//NOPADCHK) UACC(READ)
SETROPTS WHEN(PROGRAM) REFRESH
```

If you are turning on program control for the first time, use the RDEFINE statements instead of the RALTER statements. If you are using another security product, refer to that product's documentation for instructions.

- Access to SAF key rings.

The SSL and LDAP authentication support can optionally use certificates stored in SAF key rings. This requires that the Web server user ID have certain SAF permissions. Specifically, the Web server user ID must be permitted to the IRR.DIGTCERT.LISTRING facility in order to use key rings. Here are the general steps required:

1. Define the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources with universal access of None.
2. Permit the Web server user ID read access to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources in the FACILITY class.
3. Activate the FACILITY general resource class.
4. Refresh the FACILITY general resource class.

The following commands are RACF commands. Replace **WWWSEV** with the actual user ID under which IBM HTTP Server is started.

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
PE IRR.DIGTCERT.LIST CLASS(FACILITY) ID(WWWSERV) ACCESS(READ)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(WWWSERV) ACCESS(READ)
SETR CLASSACT(FACILITY)
SETR RACLIST(FACILITY) REFRESH
```

For a complete guide to RACF commands, refer to *z/OS Security Server RACF Security Administrator's Guide (SA22-7683)*. You can link to this document from the *z/OS Internet Library*.

- Permitting user IDs to CSFSERV for hardware encryption:

Integrated Cryptographic Services Facility (ICSF) is the software interface to the cryptographic hardware. If you plan to run **IBM** HTTP Server with cryptographic hardware capability, you can restrict the use of ICSF services. To restrict the use of ICSF services, you can permit user IDs to certain profiles in the **CSFSERV** general resource class. **CSFSERV** controls the use of ICSF software. If you have defined your IBM HTTP Server to execute with a nonzero user ID, you can give the nonzero user ID READ access to **CSFSERV**. If you are using a security product other than RACF, refer to that product's documentation for instructions.

If you want to restrict the use of ICSF services, issue RACF commands similar to the commands in the following examples. If you have applications other than IBM HTTP Server that are using ICSF, you must customize the examples. Otherwise, the other applications will no longer have access to ICSF services.

The following example shows how to permit the **WWWSERV** ID and the **PUBLIC** ID access to profiles in **CSFSERV**.

```
SETROPTS RACLIST(CSFSERV) GENERIC(CSFSERV)
RDEFINE CSFSERV CSF* UACC(NONE)
PERMIT CSF%* CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
PERMIT CSF%* CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
PERMIT CSF%* CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
SETR CLASSACT(CSFSERV)
SETR RACLIST(CSFSERV) GENERIC(CSFSERV) REFRESH
```

The following example shows how to give user IDs and the **WWWSERV** ID access to profiles in **CSFSERV**.

```
SETROPTS RACLIST(CSFSERV) GENERIC(CSFSERV)
RDEFINE CSFSERV CSF%* UACC(READ)
RDEFINE CSFSERV CSF%* UACC(READ)
RDEFINE CSFSERV CSF%* UACC(READ)
SETR CLASSACT(CSFSERV)
SETR RACLIST(CSFSERV) GENERIC(CSFSERV) REFRESH
```

- Using cryptographic hardware for key storage (optional):

To perform key storage on cryptographic devices refer to the section "Integrated Cryptographic Service Facility (ICSF) Considerations" in *z/OS Security Server RACF Security Administrator's Guide (SA22-7683)*.

For information on ICSF options refer to the section "Using Hardware Cryptographic Features with System SSL" in *z/OS Cryptographic Services System Secure Sockets Layer (SSL) Programming (SC24-5901)*.

You can link to both of these documents from the *z/OS Internet Library*.

- Setting environment variable * **_BPX_JOBNAME** (optional):

IBM HTTP Server provides the file `<installroot>/bin/envvars` for setting environment variables for the `httpd` processes. You can set the environmental variable * **_BPX_JOBNAME** to give the server a distinct jobname. This allows you to:

- See the server in MVS operator commands and System Display and Search Facility (SDSF).
- Categorize the server in workload management (WLM) to give web traffic adequate priority.
- Use syslogd isolation for the server.
- Use PORT statements in the TCP/IP configuration that select by job name.

A typical setting is: **export _BPX_JOBNAME=HTTPD**. The default is to append an incrementing integer to your jobname, such as HTTPD1, HTTPD2, HTTPD3. For more information refer to the section “Generating jobnames for OMVS address spaces” in *z/OS UNIX System Services Planning* (GA22-7800). Link to this document from the *z/OS Internet Library*.

If you use the `_BPX_JOBNAME` variable to set the jobname, the user ID which you use to run the server must have read access to the SAF FACILITY profile `BPX.JOBNAME`. For example:

```
RDEFINE FACILITY BPX.JOBNAME UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
PERMIT BPX.JOBNAME CLASS(FACILITY) ACCESS(READ) ID(wwwserv)
SETROPTS RACLIST(FACILITY) REFRESH
RLIST FACILITY BPX.JOBNAME ALL
```

For more information refer to the section “Setting up the BPX.* FACILITY class profiles” in *z/OS UNIX System Services Planning* (GA22-7800). Link to this document from the *z/OS Internet Library*.

Starting and stopping IBM HTTP Server

You can start or stop IBM HTTP Server using the WebSphere Application Server administrative console or using other methods depending on your platform.

Before you begin

For installation information, refer to:

- **Distributed operating systems** “Installing IBM HTTP Server on distributed systems” on page 5
- **z/OS** Migrating and installing IBM HTTP Server on z/OS systems

HP-UX **Linux** **Solaris** You can configure your operating system to allow the log file for the IBM HTTP Server plug-in to exceed the typical two gigabytes size limit. To enable this functionality, add the `USEPLUGINLARGEFILE` environment variable to your operating system configuration settings, and set it to true, before you start the IBM HTTP Server. If you do not add this environment variable to your operating system settings, or if you set this environment variable to false, the log file is limited to 2 gigabytes.

gotcha: **HP-UX** **Linux** **Solaris** Because not limiting the size of the log file might cause storage resources to be exhausted, if you decide to use this environment variable, you should periodically monitor the size of this log file.

Important: **z/OS** Before starting IBM HTTP Server, there are required z/OS system configurations that you must perform.

About this task

You can choose the following methods to start and stop IBM HTTP Server:

Procedure

- Use the WebSphere Application Server administrative console.
- **AIX** **HP-UX** **Linux** **Solaris** **z/OS** Use the command line interface.
- **Windows** Use the Windows service.
- **Windows** For debugging purposes, run the server in the foreground by leaving off the **-k** arguments.
On a command prompt, type
httpd.exe
- **z/OS** Using JCL procedures from the system console

Results

IBM HTTP Server starts successfully.

Using the administrative console to start IBM HTTP Server

You can use the WebSphere Application Server administrative console to start and stop IBM HTTP Server.

About this task

Distributed operating systems You can administer IBM HTTP Server through the WebSphere Application Server administrative console using the WebSphere Application Server node agent or using the IBM HTTP Server administration server. An IBM HTTP Server that is defined to a deployment manager (dmgr) managed node is administered using the node agent. An IBM HTTP Server that is defined to an unmanaged node is administered using the administration server.

z/OS You can administer IBM HTTP Server through the WebSphere Application Server administrative console using the WebSphere Application Server node agent. In order to enable the WebSphere Application Server administrative console for administering IBM HTTP Server, you need to specify the **-admin** option during installation of IBM HTTP Server. Or, if you already installed IBM HTTP Server without specifying the **-admin** option, you can run the **bin/enable_admin** script. For more information about enabling the administrative console for administering IBM HTTP Server on z/OS, see “Installing and configuring IBM HTTP Server on the z/OS V2R2 system” on page 39.

Important: You must start the IBM HTTP Server administration server with the same user ID that you used to start IBM HTTP Server. Also, the user ID that you used to start the IBM HTTP Server administration server must be the same as defined on the **Admin.conf** directive:

- User <admin>
- Group <admgroup>

Procedure

1. Launch the WebSphere administrative console.
2. Click **Servers > Web servers**.
3. Select your server by clicking the check box.
4. Click **Start**.
5. You can stop IBM HTTP Server by clicking **Stop**.

Results

To confirm that IBM HTTP Server started successfully, open a browser and type in your server name in the URL box.

If you are going to run Application Response Measurement (ARM) agents, make sure you have the authority to run ARM agents when you start IBM HTTP Server.

What to do next

You can configure your server for:

-
- Secure Sockets Layer (SSL)
- Lightweight Directory Access Protocol (LDAP)
- **AIX** **Windows** Fast Response Cache Accelerator (FRCA)

Using `apachectl` commands to start IBM HTTP Server

This topic describes how to start and stop IBM HTTP Server using the `apachectl` commands.

About this task

To start and stop IBM HTTP Server, use the `apachectl` command.

The `apachectl` command is located in the `bin` subdirectory within the IBM HTTP Server installation directory. If that directory is not in your `PATH`, the full path should be given on the command line.

z/OS Log on as the Web server user ID. This user ID must have an OMVS segment defined and a UID which is not zero. Verify that both the IBM HTTP Server product directory and the installation directory for the server instance are mounted and available.

Procedure

- **Starting and stopping IBM HTTP Server using the default configuration file.**

To start IBM HTTP Server using the default `httpd.conf` configuration file, run the `apachectl start` command.

To stop IBM HTTP Server using the default `httpd.conf` configuration file, run the `apachectl stop` command.

AIX **HP-UX** **Linux** **Solaris** Issue the commands from the default installation directories, based on your operating system.

- **AIX** `/usr/IBM/HTTPServer/bin/apachectl start|stop`
- **HP-UX** `/opt/IBM/HTTPServer/bin/apachectl start|stop`
- **Linux** `/opt/IBM/HTTPServer/bin/apachectl start|stop`
- **Solaris** `/opt/IBM/HTTPServer/bin/apachectl start|stop`

z/OS Issue the commands from the installation directory of the IBM HTTP Server instance.

- `<IHS_install_dir>/bin/apachectl start|stop`

For example, if the `apachectl` command is not in your `PATH`, the IBM HTTP Server installation directory is `/usr/IBM/HTTPServer`, and the default configuration file is used:

```
# /usr/IBM/HTTPServer/bin/apachectl start
# /usr/IBM/HTTPServer/bin/apachectl stop
```

- **Starting and stopping IBM HTTP Server using an alternate configuration file.**

To start IBM HTTP Server using an alternate configuration file, run the following command:

```
- apachectl -k start -f <path_to_configuration_file>
```

To stop IBM HTTP Server using an alternate configuration file, run the following command:

```
- apachectl -k stop -f <path_to_configuration_file>
```

For example, the **apachectl** command is not in your PATH, the IBM HTTP Server installation directory is /opt/IBM/HTTPServer, and an alternate configuration file, /opt/IBM/HTTPServer/conf/nodeb.conf, is used:

```
# /opt/IBM/HTTPServer/bin/apachectl -k start -f /opt/IBM/HTTPServer/conf/nodeb.conf
# /opt/IBM/HTTPServer/bin/apachectl -k stop -f /opt/IBM/HTTPServer/conf/nodeb.conf
```


Results

To confirm that IBM HTTP Server started successfully, open a browser and type in your server name in the URL box.

If you are going to run Application Response Measurement (ARM) agents, make sure you have the authority to run ARM agents when you start IBM HTTP Server.

What to do next

You can configure your server for:

- Secure Sockets Layer (SSL)
- Lightweight Directory Access (LDAP)
-  Fast Response Cache Accelerator (FRCA)

For more **apachectl** command options see Apache Hypertext Transfer Protocol Server.

Using Windows services to start IBM HTTP Server

This topic provides information about getting started with IBM HTTP Server on Windows operating systems.

Before you begin

Microsoft has introduced additional security into Windows operating systems newer than Windows 2003 via the User Account Control (UAC) feature. This additional security affects starting and stopping IBM HTTP Server from the start menu items. In order to use these menu items from an account that is not explicitly the administrator account, do one of the following actions even if the user ID being used is an administrator.

- To invoke the menu item, right-click the menu item and select **Run as administrator** each time that you want to use it.
- Configure the menu item to run as an administrator:
 1. Right click **Start HTTP Server** and select **Properties**.
 2. Select **Compatibility**.
 3. Select the check box for **Run this program as an administrator**.
 4. Click **OK**.

The previous actions automatically set the same flag on the **Stop HTTP Server** menu item.

Using either of these options allows the IBM HTTP Server to be started and stopped using the menu items. However, the User Access Control will prompt the user on each invocation for permission to run the item. If you do not want to be prompted for permission at each use, then you can alter the Local Security Policy to allow the program to be run without prompting. Be aware that making this alteration is done at a system level and will affect all other applications for which this prompting occurs. Carefully consider any security ramifications before making this change. If you want to make this change, you can complete the following actions:

1. Click **Control Panel > Administrative Tools > Local Security Policy**.
2. Expand **Local Policies**.
3. Select **Security Options**.
4. Double-click the policy of **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode**.
5. Change the setting to **Elevate without prompting**.
6. Click **OK**.

About this task

Use this task to start IBM HTTP Server as a Windows service.

Procedure

1. Start the IBM HTTP Server.
 - a. If you changed the menu item properties as described in the Before you begin section of this topic or you are using the Administrator account, click **Start > All Programs > IBM HTTP Server V8.5 > Start HTTP Server**.
 - a. If you did not change the menu item properties, click **Start > All Programs > IBM HTTP Server V8.5 > Start HTTP Server > Run as administrator**.
2. If you are prompted to run the application by User Access Control, then allow it to run.
3. To confirm that IBM HTTP Server started successfully by opening a browser window and type in your server name in the URL box. If you use the non-Administrator installation option, then the IBM HTTP Server does not install as a service. You have to run the httpd.exe file from a command line.

If IBM HTTP Server does not start:

 - a. Go to **Services** in the Control Panel.
 - b. Double-click **IBM HTTP Server** to start the server.
 - c. To confirm that IBM HTTP Server started successfully, open a browser and type in your server name in the URL box.

If you are going to run Application Response Measurement (ARM) agents, make sure you have the authority to run ARM agents when you start IBM HTTP Server.

Results

IBM HTTP Server starts successfully.

What to do next

You can configure your server for Secure Sockets Layer (SSL), Lightweight Directory Access Protocol (LDAP), and Fast Response Cache Accelerator (FRCA).

Using JCL procedures to start IBM HTTP Server on z/OS

You can prepare JCL procedures to start and stop IBM HTTP Server from the MVS system console.

By using a JCL cataloged procedure to issue the `apachectl` start and stop commands, you can start and stop an IBM HTTP Server instance from the MVS system console. Other `apachectl` commands can be issued from the MVS system console using the same procedure.

Copy the following sample JCL procedure from SHAPJCL(HAPAPROC) to your system procedure library:

```
/*-----  
//IHSAPACH PROC ACTION='start',  
//          DIR='/path/to/IHS/runtime/directory',  
//          CONF='conf/httpd.conf'  
/*-----  
//IHS      EXEC PGM=BPXBATCH,  
// PARM='SH &DIR/bin/apachectl -k &ACTION -f &CONF -DNO_DETACH',  
// MEMLIMIT=512M  
//STDOUT  DD PATH='&DIR/logs/proc.output',  
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),  
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)  
//STDERR  DD PATH='&DIR/logs/proc.errors',  
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),  
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)  
//          PEND
```

Note: The PARM value is limited to 100 characters. Since the PARM value contains the installation directory (`&DIR`), the total length could exceed the 100 character limit if the directory path is too long. The path name length needs to be taken into consideration when choosing the installation directory. If the installation directory path name is too long, it is possible to use a shorter named path in the JCL that is symbolically linked to the original installation directory path name.

If you require a PARM value greater than 100 characters, you can use the `/PARMIN DD *,SYMBOLS=JCLONLY` JCL card as illustrated in the example below:

```
/*-----  
//WEBFTNEI JOB (KOMA-Y98),'Apache NZX2      ',MSGCLASS=T,  
// TIME=NOLIMIT  
//          EXPORT SYMLIST=**  
/*-----  
// SET ACTION='start'  
// SET DIR='/home/ihsa/install/webft'  
// SET CONF='../../httpd/webftnei.conf'  
/*-----  
//IHS EXEC PGM=BPXBATCH,PARMDD=PARMIN,  
// MEMLIMIT=512M  
//PARMIN  DD *,SYMBOLS=JCLONLY  
PGM &DIR/bin/apachectl  
PGM &DIR/bin/apachectl  
      -k &ACTION  
      -f &CONF  
      -DNO_DETACH  
/*-----  
//STDOUT  DD SYSOUT=T  
/*
```

```

//STDERR DD SYSOUT=T
//*
//
//STDOUT DD PATH='&DIR/logs/proc.output',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//*
//STDERR DD PATH='&DIR/logs/proc.errors',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//
/*-----

```

A description of the **apachectl** command used in the sample JCL can be found at the Apache HTTP Server Control Interface Web site.

The default jobname for the IBM HTTP Server instance will be the same as the member name of the cataloged procedure. The Web server is a multi-process server, and each additional process that is created will have a generated jobname that is based on the original jobname. If the original jobname is 8 characters long, then all the additional processes will have the same jobname. If the original jobname is less than 8 characters then the additional processes will have jobnames that are composed of the original jobname with an added digit as a suffix. If the Web server is started from the UNIX environment using the `bin/apachectl` command, then the default jobname will be the userid that the command is running under. As with the jobname, if the userid is 8 characters long, then all the additional processes will have the same jobname. If the userid is less than 8 characters then the processes will have jobnames that are composed of the userid with an added digit as a suffix.

In the following examples, a procedure name of WEBSRV1 is used. Edit the new cataloged procedure by replacing `/path/to/IHS/runtime/directory` with the actual installation directory for this instance of IBM HTTP Server. Create a SAF STARTED profile to associate the server user ID and group with the Web server started task:

```

RDEFINE STARTED WEBSRV1.* STDATA(USER(WWWSERV) GROUP(WWWGROUP) TRACE(YES))
SETROPTS RACLIST(STARTED) GENERIC(STARTED) REFRESH

```

- To start the server from the MVS system console, enter:

```
S WEBSRV1
```

Note: The Web server name can be changed by adding **jobname** to the start command, for example:

```
S WEBSRV1,JOBNAME=HTTPDWS1
```

Best Practice 1: Use an eight character jobname. Using an eight character jobname ensures that all the processes created for this instance of the Web server will have the same jobname.

To stop the server, enter:

```
P WEBSRV1
```

Note: When using When using SDSF, you must use the System Command Extension (command entry) screen to enter the command to stop the server.

- At the command prompt, type a forward slash (/) and then hit enter to access the System Command Extension window.
- From the System Command Extension window, enter the **S WEBSRV1,ACTION='stop'** command. Make sure that stop is in lowercase.
- To issue other apachectl commands, enter:

```
S WEBSRV1,ACTION='<command>'
```

- To restart the server, enter:

```
S WEBSRV1,ACTION=restart
```

You can restart the server from the z/OS console.

- To gracefully restart the server, enter:

```
S WEBSRV1,ACTION=graceful
```

You can gracefully restart the server from the z/OS console.

The output files for the start and stop commands are located in the files specified on the STDOUT and STDERR DD JCL statements in the procedure

The output files for the start and stop commands are located in the files specified on the STDOUT and STDERR DD JCL statements in the procedure.

Using the zos_cmds module: If zos_cmds module is active, then you can use the z/OS STOP and MODIFY console commands.

Important: In the httpd.conf file, the following line must be added to activate the zos_cmds module:

```
LoadModule zos_cmds_module modules/mod_zos_cmds.so
```

Note: If you are not using a consistent eight character jobname for all the processes, you must determine the jobname of the process that is handling the z/OS operator commands is instance of the Web server. When the Web server is started, it will issue a message to the operator console that identifies the job that is handling the operator commands.

```
BPMX023I (WASTST1) IHS is active. Use jobname HTTPDWS1 for MVS commands.
```

where WASTST1 is the userid that the Web server is running under. An entry will be written to the error log that identifies the jobname as well. Notice that the mod_zos_cmds daemon jobname is HTTPDWS1. When a MODIFY command is entered, it gets targeted to every job with the specified jobname. Only one of the Web server processes will accept the command. The system will issue the following message for each of the other jobs of the same jobname.

```
IEE342I MODIFY REJECTED-TASK BUSY
```

- To stop the server using the stop command:

```
P HTTPDWS1
```

- To restart the server using the modify command, enter:

```
F HTTPDWS1,appl='restart'
```

- To gracefully restart the server using the modify command, enter:

```
F HTTPDWS1,appl='graceful'
```

When the Web server stops with the zos_cmds module active, it will issue the following message to the operator console.

```
BPMX023I (WASTST1) IHS is stopping
```

where WASTST1 is the userid that the Web server is running under.

Best Practice 2: The output files are overwritten each time the procedure is used. They might contain warning messages about the configuration or error messages for startup failures. If you want to retain a log of these messages across multiple uses of the procedure, modify the two occurrences of the PATHOPTS option in the

sample procedure to **PATHOPTS=(OCREAT,OAPPEND,OWRONLY)**. For more information on the PATHOPTS option, refer to the *z/OS MVS JCL Reference (SA22-7597)*. Link to this document from the *z/OS Internet Library*.

Best Practice 3: The STDENV DD statement is not recommended. You might consider adding environment variable settings to the bin/envvars file within the runtime directory so that the variables are active whether IBM HTTP Server is started from JCL or from the UNIX environment.

Best Practice 4: The SH parameter of BPXBATCH is recommended instead of the PGM parameter. Processing for the PGM parameter bypasses system default settings in the /etc/profile file, including the umask setting, and files created by IBM HTTP Server do not have the correct permissions.

Configuring IBM HTTP Server

To configure the IBM HTTP Server, edit the `httpd.conf` configuration file. To successfully configure the server, the name of the configuration file must be retained as **httpd.conf**.

Procedure

- **Locating the default and sample configuration files.**

The `httpd.conf` configuration file is in the `conf` directory of your server installation. There is also an `httpd.conf.default` file, in case you need to use another copy of the original file.

Restriction: To successfully configure the server, the name of the configuration file must be retained as **httpd.conf**.

IBM HTTP Server also provides the following configuration files:

- **Distributed operating systems** `admin.conf.default`
- `magic.default`
- `mime.types.default`

- **Special considerations for IBM HTTP Server.** The following items regarding the configuration file should be known when using IBM HTTP Server:

- Configuration files that only support single-byte characters (SBCS) are:
 - `httpd.conf` (IBM HTTP Server configuration file)

Restriction: To successfully configure the server, the name of the configuration file must be retained as **httpd.conf**.

- **Distributed operating systems** `admin.conf` (Administration server configuration file)
- **Windows** The forward slash character (/) should be used as a path separator in the configuration file, instead of the backward slash character (\).
- **HP-UX** **Linux** **Solaris** Configure your operating system to allow large log files for the IBM HTTP Server plug-in.

You can configure your operating system to allow the log file for the IBM HTTP Server plug-in to exceed the typical two gigabytes size limit. To enable this functionality, add the `USEPLUGINLARGEFILE` environment variable to your operating system configuration settings, and set it to true, before you start the IBM HTTP Server. If you do not add this environment variable to your operating system settings, or if you set this environment variable to false, the log file is limited to 2 gigabytes.



gotcha: Because not limiting the size of the log file might cause storage resources to be exhausted, if you decide to use this environment variable, you should periodically monitor the size of this log file.

Apache modules (containing directives) supported by IBM HTTP Server

This section provides information on Apache modules that are supported by IBM HTTP Server. The directives defined within the supported Apache modules can be used to configure IBM HTTP Server.

Supported Apache modules

The following Apache modules were changed in Version 9 (Apache HTTP Server 2.4):

- The `mod_proxy_balancer` and `mod_proxy_ajp` modules are no longer included. Previous releases included these modules in the WebSphereCE directory on some platforms.
- The `mod_mem_cache` module has been removed from the Apache distribution. Use the `mod_disk_cache` module, instead.
-   The event MPM is supported on z/OS and Linux platforms only.
- The `mod_ibm_ldap` module has been removed.
- The following modules have been added:
 - `mod_access_compat`
 - `mod_proxy_fcgi`
 - `mod_substitute`
 - `mod_lua`
 - `mod_authn_certificate`
 - `mod_remoteip`
 - `mod_macro`

The modules that are supported by IBM HTTP Server V9.0 are listed in the Apache HTTP Server V 2.4 Directive Index.

The following Apache modules were changed or are not supported in Version 7 (Apache HTTP Server V2.2):

- The `mod_file_cache` module is provided with this release of IBM HTTP Server for compatibility with previous releases, however, you must migrate existing configurations to use the `mod_mem_cache` module to ensure future support for your LDAP configuration. These modules provide equivalent function in the memory instead of on a disk.
- The `mod_mime_magic` module is provided with this release of IBM HTTP Server for compatibility with previous releases, but might not be available in a future release. No replacement will be provided for this module.
- The `mod_proxy_ftp` module is provided with this release of IBM HTTP Server for compatibility with previous releases, but might not be available in a future release. No replacement will be provided for this module.
- The `mod_cern_meta` module is not supported. Instead use the `mod_headers` module.

- The mod_imap module was renamed to mod_imagemap. The LoadModule directive for the mod_imap module must be changed to refer to the new module name for an existing configuration file.
- You must set the EnableExceptionHook directive value to On for the mod_backtrace and mod_whatkilledus diagnostic modules.
- You may set the McacheMinObjectSize directive value to a minimum of 1 for the mod_mem_cache module. In previous releases, the minimum value was zero.
- The Compression_Level directive for the mod_deflate module was renamed to DeflateCompressionLevel.
- The configurations for the mod_ldap and the mod_auth_ldap modules have changed. See the following procedure about migrating from the mod_ldap and mod_auth_ldap module configurations.
- The Apache mod_example source is installed in the <i>hinst</i>/example_module directory.
- The AddOutputFilterByType directive now applies to proxy requests.
- Directory listings created by the mod_autoindex module now have a default character set which can be modified using the IndexOptions directive. If you rely on browser detection of character sets for correct display of directory listings, you might need to specify the correct character set using the IndexOptions directive.

Best Practice: Distributed operating systems If you are using the mod_ibm_ldap module for your LDAP configuration, consider migrating your mod_ibm_ldap directives to use the mod_ldap module. The mod_ibm_ldap module is provided with this release of IBM HTTP Server for compatibility with previous releases, however, you must migrate existing configurations to use the mod_authnz_ldap and mod_ldap modules to ensure future support for your LDAP configuration.

The following table contains a list of Apache modules supported for IBM HTTP Server.

Table 1. Apache modules. The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
core	Core Apache HTTP Server features	http://publib.boulder.ibm.com/httserv/manual24/mod/core.html
Windows mpm_winnt	Multi-processing module (MPM)	http://publib.boulder.ibm.com/httserv/manual24/mod/mpm_winnt.html
AIX HP-UX Linux Solaris worker	MPM	http://publib.boulder.ibm.com/httserv/manual24/mod/worker.html
z/OS Linux event	MPM	http://publib.boulder.ibm.com/httserv/manual24/mod/event.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_actions	Provides for executing CGI scripts, based on media type or request method.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_actions.html
mod_alias	Provides for mapping different parts of the host file system in the document tree and for URL redirection.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_actions.html
mod_asis	Sends files that contain their own HTTP headers.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_asis.html
mod_auth_basic	Basic authentication	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_auth_basic.html
mod_authn_anon	Allows anonymous user access to authenticated areas.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authn_anon.html
mod_authn_dbm	User authentication using DBM files.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authn_dbm.html
mod_authn_default	Authentication fallback module	
mod_authn_file	User authentication using text files	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authn_file.html
mod_authnz_ldap	Allows an LDAP directory to be used to store the database for HTTP basic authentication.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authnz_ldap.html
mod_authz_dbm	Group authorization using DBM files.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authz_dbm.html
mod_authz_default	Authorization fallback module	
mod_authz_groupfile	Group authorization using text files	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authz_groupfile.html
mod_authz_host	Group authorizations based on host, such as host name or IP address	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authz_host.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_authz_user	User authorization	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_authz_user.html
mod_autoindex	Generates directory indexes automatically. This is similar to ls command on the UNIX platform or the Win32 dir shell command.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_autoindex.html
mod_cache	Content cache keyed to URIs	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_cache.html
mod_cgi	Execution of CGI scripts	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_cgi.html
AIX HP-UX Linux Solaris mod_cgid	Execution of CGI scripts using an external CGI daemon.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_cgid.html
z/OS mod_charset_lite	Specifies character set translation or recoding.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_charset_lite.html
Distributed operating systems mod_dav	Distributed Authoring and Versioning (WebDAV) functionality. Tip: z/OS Although mod_dav and mod_dav_fs are not supported, IBM HTTP Server and the WebSphere plug-in can pass through WebDAV requests to WebSphere.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_dav.html
Distributed operating systems mod_dav_fs	File system provider for mod_dav.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_dav_fs.html
mod_deflate	Compress content before it is delivered to the client.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_deflate.html
mod_dir	Provides for "trailing slash" redirects and serving directory index files.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_dir.html
mod_disk_cache	Implements a disk based storage manager. It is primarily of use in conjunction mod_cache.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_disk_cache.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_env	Modifies the environment which is passed to CGI scripts and SSI pages.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_env.html
mod_expires	Generation of Expires and Cache Control HTTP headers according to user-specified criteria.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_expires.html
mod_ext_filter	Pass the response body through an external program before delivery to the client.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_ext_filter.html
Distributed operating systems mod_file_cache	Caches a static list of files in memory. This module is provided with this release for compatibility with previous releases. Begin using mod_mem_cache or mod_cache to ensure compatibility with future releases of IBM HTTP Server. Tip: The recommended caching mechanism for file handling is the CacheEnable feature of the mod_cache module.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_file_cache.html
mod_filter	Specifies the context-sensitive smart filter configuration module.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_filter.html
mod_ibm_ssl	Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocol support for IBM HTTP Server.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_ibm_ssl.html
mod_headers	Customization of HTTP request and response headers.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_headers.html
mod_imagemap	Server-side image map processing.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_imagemap.html
mod_include	Server-parsed HTML documents (Server Side Includes).	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_include.html
mod_info	Provides a comprehensive overview of the server configuration.	http://publib.boulder.ibm.com/httpserv/manual24/mod/mod_info.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_ldap	Provides LDAP connection pooling and result caching services for use by other LDAP modules.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_ldap.html
mod_log_config	Logging of the requests made to the server.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_log_config.html
mod_logio	Logging of input and output bytes per request.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_logio.html
mod_mem_cache	Content cache keyed to URIs.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_mem_cache.html
mod_mime	Associates the requested file extensions with the behavior of the file (handlers and filters), and content (mime-type, language, character set and encoding).	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_mime.html
mod_mpmstats	MPM/thread monitoring module for IBM HTTP Server.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_mpmstats.html
Distributed operating systems mod_mime_magic	Determines the MIME type of a file by looking at a few bytes of its contents. This module is provided with this release of IBM HTTP Server for compatibility with previous releases, but will not be supported in a future release. No replacement will be provided for this module. Important: Using mod_mime_magic can decrease performance because the file must be read and compared to a set of patterns to determine the content- type.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_mime_magic.html
z/OS mod_mvds	Serve MVS (z/OS) Datasets	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_mvds.html
mod_negotiation	Provides for content negotiation.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_negotiation.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_proxy	HTTP, 1.1 proxy, and gateway server	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_proxy.html
mod_proxy_connect	Specifies the mod_proxy module extension for CONNECT request handling.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_proxy_connect.html
Distributed operating systems mod_proxy_ftp	Provides FTP support for the mod_proxy module. This module is provided with this release of IBM HTTP Server for compatibility with previous releases, but will not be supported in a future release. No replacement will be provided for this module.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_proxy_ftp.html
mod_proxy_http	Provides HTTP support for the mod_proxy module.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_proxy_http.html
mod_rewrite	Provides a rule-based rewriting engine to rewrite requested URLs.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_rewrite.html
z/OS mod_smf	Record SMF entries for HTTP requests.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_smf.html
mod_setenvif	Enables the setting of environment variables based on characteristics of the request.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_setenvif.html
mod_so	Loading of executable code and modules into the server at start or restart time.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_so.html
mod_speling	Attempts to correct mistaken URLs that users might have entered by ignoring capitalization and by allowing up to one misspelling.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_speling.html
mod_status	Provides information on server activity and performance.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_status.html
mod_suexec	Allows CGI scripts to run as the specified user or group.	http://publib.boulder.ibm.com/httserv/manual24//mod/mod_suexec.html

Table 1. Apache modules (continued). The table lists the Apache module, a brief description of the module, and a web address to a detailed description of each module.

Module	Description	Web address
mod_unique_id	Provides an environment variable with a unique identifier for each request.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_unique_id.html
mod_userdir	User-specific directories.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_userdir.html
mod_usertrack	Clickstream logging of user activity on a site.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_usertrack.html
mod_vhost_alias	Provides for dynamically configured mass virtual hosting.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_vhost_alias.html
z/OS mod_wlm	z/OS WLM classification of HTTP requests.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_wlm.html
z/OS mod_zos_cmds	This module allows the server to respond to STOP and MODIFY z/OS console commands.	http://publib.boulder.ibm.com/httserv/manual24/mod/mod_zos_cmds.html

Apache programs supported by IBM HTTP Server

This section provides information on Apache programs that are supported by IBM HTTP Server. These supported Apache programs can be used to configure IBM HTTP Server.

Supported Apache programs

The following table contains a list of Apache commands supported for IBM HTTP Server.

Note: **Windows** The apache.exe command was replaced with the httpd.exe command. The apache.exe command is provided with this release of IBM HTTP Server for compatibility with previous releases. Migrate existing scripts and procedures to use the httpd.exe command to ensure future support for this functionality.

Program	Description	URL
ab	Provides benchmarking functionality for the Web server	http://publib.boulder.ibm.com/httserv/manual24/programs/ab.html

Program	Description	URL
<p>Windows</p> <p>Linux</p> <p>AIX HP-UX Solaris</p> <p>z/OS</p> <p>apachectl</p>	Provides start, stop, and restart functionality for the Web server.	http://publib.boulder.ibm.com/httpserv/manual24/programs/apachectl.html
<p>AIX HP-UX</p> <p>Linux Solaris</p> <p>z/OS</p> <p>Windows</p> <p>httpd.exe</p>	Provides start, stop, and restart functionality for the Web server.	http://publib.boulder.ibm.com/httpserv/manual24/programs/httpd.html
<p>Linux</p> <p>AIX HP-UX Solaris</p> <p>z/OS</p> <p>apxs</p>	Builds plug-in modules.	http://publib.boulder.ibm.com/httpserv/manual24/programs/apxs.html
dbmmanage	Creates and updates user authentication files in DBM format for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual24/programs/dbmmanage.html
htdbm	Creates and updates user authentication files in DBM format for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual24/programs/htdbm.html
htpasswd	Creates and updates user authentication files for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual24/programs/htpasswd.html
htt2dbm	Creates DMB files for use with RewriteMap.	http://publib.boulder.ibm.com/httpserv/manual24/programs/htt2dbm.html
logresolve	Resolves host names for IP addresses in Apache log files.	http://publib.boulder.ibm.com/httpserv/manual24/programs/logresolve.html
rotatlogs	Rotates log files without having to stop the server.	http://publib.boulder.ibm.com/httpserv/manual24/programs/rotatlogs.html

Apache APR and APR-util libraries supported by IBM HTTP Server

This section provides information about the Apache Portable Runtime (APR) and APR-util libraries that are supported by IBM HTTP Server. IBM HTTP Server supports only the APR and APR-util libraries installed with the product. Copies of the libraries cannot be substituted.

Supported APR and APR-util libraries

The APR and APR-util libraries installed with IBM HTTP Server are provided for only IBM and third-party plug-in modules loaded into IBM HTTP Server. Use of these libraries by stand-alone applications or commands, other than those provided with IBM HTTP Server, is not supported.

The following build-time features of APR and APR-util are not provided on all platforms.

- random number support
- native atomic operation support
- il8n translation
- DBD support is not provided for any platform
- LDAP support is not provided for any platform.

The only supported APR-util library database management type is SDBM. SDBM affects the `htdbm` and `httxt2dbm` commands. It also affects the `mod_authn_dbm`, `mod_authz_dbm`, and `mod_rewrite` modules for DBM map files and the `mod_dav` module for the lock database.

Apache MPM and addressing modes supported by IBM HTTP Server

This section provides information about Apache Multi-processing module (MPM) and addressing modes supported by IBM HTTP Server.

The following table contains a list of platforms and the MPM and addressing modes supported on those platforms by IBM HTTP Server.

Table 2. MPM and addressing modes. The table lists the platform, addressing mode, and MPM.

Platform	Addressing mode	MPM
AIX	64-bit	worker MPM
AIX HP-UX Linux Solaris z/OS HP-UX/ia64	64-bit	worker MPM
Linux/x86	32-bit and 64-bit	worker and event MPM
Linux/PPC	32-bit	worker and event MPM
Linux on System z [®]	32-bit	worker and event MPM
Solaris/SPARC	64-bit	worker MPM
Solaris/x64	64-bit	worker MPM
Windows	32-bit	WinNT MPM
z/OS	64-bit	event MPM

IPv4 and IPv6 configuration for Windows operating systems

IBM HTTP Server supports IPv6 on Windows XP and 2003 operating systems. It does not support IPv6 on the Windows 2000 operating system.

Support for IPv6 on Windows operating systems is configured differently than other supported platforms. The `Listen` directive on Windows operating systems

should always include either an IPv4 address or an IPv6 address. Any existing Listen directives that are not qualified with an IP address should be updated to include one, even if Windows IPv6 networking is not configured.

Use 0.0.0.0 for the default IPv4 address and [::] for the default IPv6 address. Add the following line in httpd.conf configuration file to listen on IPv6 port 80:
Listen [::]:80

If you want to accept connections over IPv4, configure Listen 0.0.0.0:80 or AfpPort 80. Advanced fast path architecture (AFPA) is only supported for IPv4.

Configure Windows IPv6 networking before enabling the Listen directive for IPv6.

Enabling IBM HTTP Server for FastCGI applications

FastCGI applications use TCP or UNIX sockets to communicate with the Web server. This scalable architecture enables applications to run on the same platform as the Web server, or on many machines scattered across an enterprise network.

About this task

You can port FastCGI applications to other Web server platforms. Most popular Web servers support FastCGI directly, or through commercial extensions.

FastCGI applications run fast because of their persistency. These applications require no per-request startup and initialization overhead. This persistency enables the development of applications, otherwise impractical within the CGI paradigm, like a huge Perl script, or an application requiring a connection to one or more databases.

Procedure

1. Load the mod_fastcgi module into the server.
LoadModule fastcgi_module modules/mod_fastcgi.so
2. Configure FastCGI using the FastCGI directives.

Example

Windows In the following configuration example, the c:/Program Files/IBM/HTTPServer/fcgi-bin/ directory contains FastCGI echo.exe applications. Requests from Web browsers for the /fcgi-bin/echo.exe URI will be handled by the FastCGI echo.exe application :

```
LoadModule fastcgi_module modules/mod_fastcgi.so
<IfModule mod_fastcgi.c>
    AllowOverride None
    Options +ExecCGI
    SetHandler fastcgi-script
</Directory>
```

```
FastCGIServer "C:/Program Files/IBM/HTTPServer/fcgi-bin/echo.exe" -processes 1
</IfModule>
```

AIX **HP-UX** **Linux** **Solaris** **z/OS** In the following configuration example, the /opt/IBM/HTTPServer/fcgi-bin/ directory contains FastCGI applications, including the echo application. Requests from Web browsers for the /fcgi-bin/echo URI will be handled by the FastCGI echo application :


```

LoadModule fastcgi_module modules/mod_fastcgi.so
<IfModule mod_fastcgi.c>
ScriptAlias /fcgi-bin/ "/opt/IBM/HTTPServer/fcgi-bin/"

<Directory> "/opt/IBM/HTTPServer/fcgi-bin/"
    AllowOverride None
    Options +ExecCGI
    SetHandler fastcgi-script
</Directory>

FastCGIServer "/opt/IBM/HTTPServer/fcgi-bin/echo" -processes 1
</IfModule>

```

Learn about FastCGI

FastCGI is an interface between Web servers and applications which combines some of the performance characteristics of native Web server modules with the Web server independence of the Common Gateway Interface (CGI) programming interface.

FastCGI is an open extension to CGI that is language independent and is a scalable architecture. FastCGI provides high performance and persistence without the limitations of server-specific APIs. The FastCGI interface is described at <http://www.fastcgi.com/>.

IBM HTTP Server provides FastCGI support with the `mod_fastcgi` module. The `mod_fastcgi` module implements the capability for IBM HTTP Server to manage FastCGI applications and to allow them to process requests.

A FastCGI application typically uses a programming library such as the FastCGI development kit from <http://www.fastcgi.com/>. IBM HTTP Server does not provide a FastCGI programming library for use by FastCGI applications.

FastCGI applications are not limited to a particular development language. FastCGI application libraries currently exist for Perl, C/C++, Java, Python and the transmission control layer (TCL).

For more information on FastCGI, visit the FastCGI Web site. To receive FastCGI related announcements and notifications of module updates, send mail to fastcgi-announce-request@idle.com with `subscribe` in the Subject field. To participate in the discussion of `mod_fastcgi` and FastCGI application development, send mail to fastcgi-developers-request@idle.com with `subscribe` in the Subject field.

The IBM HTTP Server Fast CGI plug-in provides an alternative method of producing dynamic content.

FastCGI directives

These configuration parameters control the FastCGI feature in IBM HTTP Server.

- “FastCGIAccessChecker directive” on page 90
- “FastCGIAccessCheckerAuthoritative directive” on page 90
- “FastCGIAuthenticator directive” on page 91
- “FastCGIAuthenticatorAuthoritative directive” on page 92
- “FastCGIAuthorizer directive” on page 92
- “FastCGIAuthorizerAuthoritative directive” on page 93
- “FastCGIConfig directive” on page 93
- “FastCGIExternalServer directive” on page 95
- “FastCGIIpcDir directive” on page 98

- “FastCGIServer directive” on page 98
- **Distributed operating systems** “FastCGISuEXEC directive” on page 100

FastCGIAccessChecker directive

The FastCGIAccessChecker directive defines a FastCGI application as a per-directory access validator.

Directive	Description
Syntax	FastCGIAccessChecker file name [-compat]
Scope	directory, location
Default	Directory
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	File name

The Apache Access phase precedes user authentication and the HTTP headers submitted with the request determine the decision to enable access to the requested resource. Use FastCGI-based authorizers when a dynamic component exists as part of the access validation decision, like the time, or the status of a domain account.

If the FastCGI application file name does not have a corresponding static or external server definition, the application starts as a dynamic FastCGI application. If the file name does not begin with a slash (/), then the application assumes that the file name is relative to the ServerRoot.

Use the FastCgiAccessChecker directive within Directory or Location containers. For example:

```
<Directory htdocs/protected>
FastCgiAccessChecker fcgi-bin/access-checker
</Directory>
```

Mod_fastcgi sends nearly all of the standard environment variables typically available to CGI and FastCGI request handlers. All headers returned by a FastCGI access-checker application in a successful response (Status: 200), pass to subprocesses, or CGI and FastCGI invocations, as environment variables. All headers returned in an unsuccessful response pass to the client. Obtain FastCGI specification compliant behavior by using the -compat option.

Mod_fastcgi sets the environment variable FCGI_APACHE_ROLE to ACCESS_CHECKER, to indicate the Apache-specific authorizer phase performed.

The HTTP Server does not support custom failure responses from FastCGI authorizer applications. See the ErrorDocument directive for a workaround. A FastCGI application can serve the document.

FastCGIAccessCheckerAuthoritative directive

The FastCGIAccessCheckerAuthoritative directive enables access checking passing to lower-level modules.

Directive	Description
Syntax	FastCGIAccessCheckerAuthoritative On Off
Scope	directory, location

Directive	Description
Default	FastCGIAccessCheckerAuthoritative On
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	On or off

Setting the FastCgiAccessCheckerAuthoritative directive explicitly to Off, enables access checking passing to lower level modules, as defined in the Configuration and modules.c files, if the FastCGI application fails to enable access.

By default, control does not pass on and a failed access check results in a forbidden reply. Consider the implications carefully before disabling the default.

FastCGIAuthenticator directive

The FastCGIAuthenticator directive defines a FastCGI application as a per-directory authenticator.

Directive	Description
Syntax	FastCGIAuthenticator file name [-compat]
Scope	directory
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	File name

Authenticators verify the requester by matching the user name and password that is provided against a list or database of known users and passwords. Use FastCGI-based authenticators when the user database is maintained within an existing independent program, or resides on a machine other than the Web server.

If the FastCGI application file name does not have a corresponding static or external server definition, the application starts as a dynamic FastCGI application. If the file name does not begin with a slash (/), then the file name is assumed to be relative to the ServerRoot.

Use the FastCgiAuthenticator directive within directory or location containers, along with an AuthType and AuthName directive. This directive only supports the basic user authentication type. This authentication type needs a require, or FastCgiAuthorizer directive, to work correctly.

```

/Directory htdocs/protected>
AuthType Basic
AuthName ProtectedRealm
FastCgiAuthenticator fcgi-bin/authenticator
require valid-user
</Directory>

```

The Mod_fastcgi directive sends nearly all of the standard environment variables that are typically available to CGI and FastCGI request handlers. All headers returned by a FastCGI authentication application in a successful response (Status: 200) pass to subprocesses, or CGI and FastCGI invocations, as environment variables. All headers returned in an unsuccessful response are passed to the client. Obtain FastCGI specification compliant behavior by using the -compat option.

The `Mod_fastcgi` directive sets the `FCGL_APACHE_ROLE` environment variable to `AUTHENTICATOR`, indicating the Apache-specific authorizer phase performed.

This directive does not support custom failure responses from FastCGI authorizer applications. See the `ErrorDocument` directive for a workaround. A FastCGI application can serve the document.

FastCGIAuthenticatorAuthoritative directive

The `FastCGIAuthenticatorAuthoritative` directive enables authentication passing to lower level modules defined in the configuration and `modules.c` files, if explicitly set to `off` and the FastCGI application fails to authenticate the user.

Directive	Description
Syntax	<code>FastCGIAuthenticatorAuthoritative On Off</code>
Scope	directory
Default	<code>FastCgiAuthenticatorAuthoritative On</code>
Module	<code>mod_fastcgi</code>
Multiple instances in the configuration file	yes
Values	On or off

Use this directive in conjunction with a well protected `AuthUserFile` directive, containing a few administration-related users.

By default, control does not pass on and an unknown user results in an Authorization Required reply. Consider implications carefully before disabling the default.

FastCGIAuthorizer directive

The `FastCGIAuthorizer` directives defines a FastCGI application as a per-directory authorizer.

Directive	Description
Syntax	<code>FastCgiAuthorizer file name [-compat]</code>
Scope	directory
Default	None
Module	<code>mod_fastcgi</code>
Multiple instances in the configuration file	yes
Values	File name

Authorizers validate whether an authenticated user can access a requested resource. Use FastCGI-based authorizers when a dynamic component exists as part of the authorization decision, such as the time, or currency of the user's bills.

If the FastCGI application file name does not have a corresponding static or external server definition, the application starts as a dynamic FastCGI application. If the file name does not begin with a slash (/) then the file name is assumed relative to the `ServerRoot`.

Use `FastCgiAuthorizer` within `Directory` or `Location` containers. Include an `AuthType` and `AuthName` directive. This directive requires an authentication directive, such as `FastCgiAuthenticator`, `AuthUserFile`, `AuthDBUserFile`, or `AuthDBMUserFile` to work correctly.

```

<Directory htdocs/protected>
AuthType Basic
AuthName ProtectedRealm
AuthDBMUserFile conf/authentication-database
FastCgiAuthorizer fcgi-bin/authorizer
</Directory>

```

The `Mod_fastcgi` directive sends nearly all of the standard environment variables typically available to CGI and FastCGI request handlers. All headers returned by a FastCGI authentication application in a successful response (Status: 200) pass to subprocesses, or CGI and FastCGI invocations, as environment variables. All headers returned in an unsuccessful response pass on to the client. Obtain FastCGI specification compliant behavior by using the `-compat` option.

The `Mod_fastcgi` directive sets the environment variable `FCGI_APACHE_ROLE` to `AUTHORIZER`, to indicate the Apache-specific authorizer phase performed.

This directive does not support custom failure responses from FastCGI authorizer applications. See the `ErrorDocument` directive for a workaround. A FastCGI application can serve the document.

FastCGIAuthorizerAuthoritative directive

The `FastCGIAuthorizerAuthoritative` directive enables authentication passing to lower level modules, as defined in the configuration and `modules.c` files, when explicitly set to `Off`, if the FastCGI application fails to authenticate the user.

Directive	Description
Syntax	<code>FastCgiAuthorizerAuthoritative</code> file name On Off
Scope	directory
Default	<code>FastCgiAuthorizerAuthoritative</code> file name On
Module	<code>mod_fastcgi</code>
Multiple instances in the configuration file	yes
Values	On or off

Use this directive in conjunction with a well protected `AuthUserFile` containing a few administration-related users.

By default, control does not pass on and an unknown user results in an Authorization Required reply. Consider the implications carefully before disabling the default.

FastCGIConfig directive

The `FastCGIConfig` directive defines the default parameters for all dynamic FastCGI applications.

Directive	Description
Syntax	<code>FastCgiConfig</code> option option...
	The <code>FastCgiConfig</code> directive does not affect static or external applications.
Scope	directory
Default	None
Module	<code>mod_fastcgi</code>
Multiple instances in the configuration file	yes

**Directive
Values**

Description

Dynamic applications start upon demand. Additional application instances start to accommodate heavy demand. As demand fades, the number of application instances decline. Many of the options govern this process.

Option can include one of the following (case insensitive):

- **appConnTimeout n (0 seconds)**. The number of seconds to wait for a connection to the FastCGI application to complete or 0, to indicate use of a blocking connect(). If the timeout expires, a SERVER_ERROR results. For non-zero values, this amount of time used in a select() to write to the file descriptor returned by a non-blocking connect(). Non-blocking connect(s) are troublesome on many platforms. See also -idle-timeout; this option produces similar results, but in a more portable manner.
- **idle-timeout n (30 seconds)**. The number of seconds of FastCGI application inactivity allowed before the request aborts and the event logs at the error LogLevel. The inactivity timer applies only when a pending connection with the FastCGI application exists. If an application does not respond to a queued request within this period, the request aborts. If communication completes with the application, but not with the client (a buffered response), the timeout does not apply.
- **autoUpdate none**. This option causes the mod_fastcgi module to check the age of the application on disk before processing each request. For recent applications, this function notifies the process manager and stops all running instances of the application. Build this type of functionality into the application. A problem can occur when using this option with -restart.
- **gainValue n (0.5)**. A floating point value between 0 and 1 that is used as an exponent in the computation of the exponentially decayed connection times load factor of the currently running dynamic FastCGI applications. Old values are scaled by (1 - gainValue), so making values smaller, weights them more heavily compared to the current value, which is scaled by gainValue.
- **initial-env name[=value] none**. A name-value pair passed in the initial environment when instances of the application spawn. To pass a variable from the Apache environment, do not provide the "=" (if the variable is not actually in the environment, it is defined without a value). To define a variable without a value, provide the "=" without any value. This option is repeatable.
- **init-start-delay n (1 second)**. The minimum number of seconds between the spawning of instances of this application. This delay decreases the demand placed on the system at server initialization.
- **killInterval n (300 seconds)**. The killInterval determines how often the dynamic application instance killing policy is implemented within the process manager. Lower numbers result in a more aggressive policy, while higher numbers result in a less aggressive policy.
- **listen-queue-depth n (100)**. The depth of the listen() queue, also known as the backlog, shared by all instances of this application. A deeper listen queue allows the server to cope with transient load fluctuations without rejecting requests; it does not increase throughput. Adding additional application instances can increase throughput and performance, depending upon the application and the host.
- **maxClassProcesses n (10)**. The maximum number of dynamic FastCGI application instances allowed to run for any one FastCGI application.

- **maxProcesses n (50).** The maximum number of dynamic FastCGI application instances allowed to run at any time.
- **minProcesses n (5).** The minimum number of dynamic FastCGI application instances the process manager allows to run at any time, without killing them due to lack of demand.
- **multiThreshold n (50).** An integer between 0 and 100 used to determine whether to terminate any instance of a FastCGI application. If the application has more than one instance currently running, this attribute helps to decide whether to terminate one of them. If only one instance remains, singleThreshold is used instead.
- **pass-header header none.** The name of an HTTP Request Header passed in the request environment. This option makes the contents of headers available to a CGI environment.
- **priority n (0).** The process priority assigned to the application instances using setpriority().
- **processSlack n (5 seconds).** If the sum of all currently running dynamic FastCGI applications exceeds maxProcesses - processSlack, the process manager invokes the killing policy. This action improves performance at higher loads, by killing some of the most inactive application instances before reaching the maxProcesses value.
- **restart none.** This option causes the process manager to restart dynamic applications upon failure, similar to static applications.
- **Restart-delay n (5 seconds).** The minimum number of seconds between the respawning of failed instances of this application. This delay prevents a broken application from soaking up too much of the system.
- **singleThreshold n (0).** An integer between 0 and 100, used to determine whether the last instance of a FastCGI application can terminate. If the process manager computed load factor for the application is lower than the specified threshold, the last instance is terminated. Specify a value closer to 1, to make your executables run in the idle mode for a long time. If memory or CPU time is a concern, a value closer to 100 is more applicable. A value of 0, prevents the last instance of an application from terminating; this value is the default. Changing this default is not recommended, especially if you set the -appConnTimeout option.
- **startDelay n (3 seconds).** The number of seconds the Web server waits while trying to connect to a dynamic FastCGI application. If the interval expires, the process manager is notified with hope that another instance of the application starts. Set the startDelay value smaller than the appConnTimeout value, to be effective.
- **updateInterval n (300 seconds).** The updateInterval decides how often statistical analysis is performed to determine the fate of dynamic FastCGI applications.

FastCGIExternalServer directive

The FastCGIExternalServer defines file name as an external FastCGI application.

It operates the same as the Fastcgiserver directive, except that the CGI application is running in another process outside of the Web server.

Directive	Description
Syntax	FastCgiExternalServer file name -host hostnameport [-appConnTimeout n] FastCgiExternalServer file name -socket file name [-appConnTimeout n]

Directive	Description
Scope	Server configuration
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes

**Directive
Values**

Description

- **appConnTimeout *n* (0 seconds).** The number of seconds to wait for a connection to the FastCGI application to complete, or 0, to indicate use of a blocking connect() method. If the timeout expires, a SERVER_ERROR results. For non-zero values, this indicator is the amount of time used in a select() method to write to the file descriptor returned by a non-blocking connect() method. Non-blocking connect() methods are troublesome on many platforms. See also -idle-timeout; this option produces similar results, but in a more portable manner.
- **Idle-timeout *n* (30 seconds).** The number of seconds of FastCGI application inactivity allowed before the request aborts and the event is logged (at the error LogLevel). The inactivity timer applies only as long as a connection is pending with the FastCGI application. If a request is queued to an application, but the application does not respond by writing and flushing within this period, the request aborts. If communication is complete with the application but incomplete with the client (a buffered response), the timeout does not apply.
- **flush none.** Force a write to the client as data is received from the application. By default, the mod_fastcgi option buffers data to free the application quickly.
- **host hostname:port none.** The hostname, or IP address and TCP port number (1-65535) the application uses for communication with the Web server. The -socket and -host options are mutually exclusive.
- **Pass-header header none.** The name of an HTTP Request Header passed in the request environment. This option makes the header contents available, to a CGI environment.
- **socket file name none.**
 - **On UNIX operating systems.** The file name of the UNIX domain socket the application uses for communication with the Web server. The file name is relative to the FastCgiIpcDir option. The -socket and -port options are mutually exclusive.
 - **On Windows operating systems.** The name of the pipe the application uses for communicating with the Web server. The name is relative to the FastCgiIpcDir option. The -socket and -port options are mutually exclusive.

FastCGIpcDir directive

The FastCGIpcDir directive specifies directory as the place to store the UNIX socket files used for communication between the applications and the Web server.

Directive	Description
Syntax	<ul style="list-style-type: none">• On UNIX platforms - FastCgiIpcDir <i>directory</i>• On Windows operating systems - FastCgiIpcDir <i>name</i>
Scope	Server configuration
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	directory or name

AIX **HP-UX** **Linux** **Solaris** The FastCgiIpcDir directive specifies directory as the place to store and find, in the case of external FastCGI applications, the UNIX socket files that are used for communication between the applications and the Web server. If the directory does not begin with a slash (/) then it is assumed to be relative to the ServerRoot. If the directory does not exist, the function attempts to create the directive with appropriate permissions. Specify a directory on a local file system. If you use the default directory, or another directory within /tmp, mod_fastcgi breaks if your system periodically deletes files from the /tmp directory.

Windows The FastCgiIpcDir directive specifies *name* as the root for the named pipes used for communication between the application and the Web server. Define the name in the form >\\.\pipe*pipename*. . The pipename syntax can contain any character other than a backslash.

The FastCgiIpcDir directive must precede any FastCgiServer or FastCgiExternalServer directives, which make use of UNIX sockets. Ensure a readable, writeable, and executable directory by the Web server. No one should have access to this directory.

FastCGIServer directive

The FastCGIServer directive defines file name as a static FastCGI application.

The Process Manager starts one instance of the application with the default configuration specified in parentheses below. Should a static application instance die for any reason, the mod_fastcgi module spawns another instance for replacement and logs the event at the warn LogLevel.

Directive	Description
Syntax	FastCgiServer file name [options]
Scope	Server configuration
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	directory or name

You can use one of the following case-insensitive options:

- **appConnTimeout *n* (0 seconds)**. The number of seconds to wait for a connection to the FastCGI application to complete, or 0, to indicate use of a blocking connect(). If the timeout expires, a SERVER_ERROR results. For non-zero values, this indicator is the amount of time used in a select() to write to the file descriptor returned by a non-blocking connect(). Non-blocking connect()s prove troublesome on many platforms. See the -idle-timeout option; it produces similar results but in a more portable manner.
- **Idle-timeout *n* (30 seconds)**. The number of seconds of FastCGI application inactivity allowed before the request aborts and the event logs at the error LogLevel. The inactivity timer applies only when a pending connection with the FastCGI application exists. If an application does not respond to a queued request within this period, the request aborts. If communication completes with the application, but does not complete with the client (a buffered response), the timeout does not apply.
- **initial-env name [=value] none]none**. A name-value pair passed in the FastCGI application initial environment. To pass a variable from the Apache environment, do not provide the "=" (variables not actually in the environment, are defined without a value). To define a variable without a value, provide the "=" without a value. You can repeat this option.
- **init-start-delay *n* (1 second)**. The minimum number of seconds between the spawning of instances of this application. This delay decreases the demand placed on the system at server initialization.
- **Flush none**. Force a write to the client as data arrives from the application. By default, mod_fastcgi buffers data to free the application quickly.
- **Listen-queue-depth *n* (100)**. The depth of the listen() queue, also known as the backlog, shared by all of the instances of this application. A deeper listen queue enables the server to cope with transient load fluctuations, without rejecting requests; this option does not increase throughput. Adding additional application instances can increase throughput and performance, depending upon the application and the host.
- **Pass-header header none**. The name of an HTTP Request Header passed in the request environment. This option makes the contents of headers available to a CGI environment.
- **processes *n* (1)**. The number of application instances to spawn at server initialization.
- **Priority *n* (0)**. The process priority assigned to the application instances, using setpriority().
- **port *n* none**. The TCP port number (1-65535) the application uses for communication with the Web server. This option makes the application accessible from other machines on the network. The -socket and -port options are mutually exclusive.
- **Restart-delay *n* (5 seconds)**. The minimum number of seconds between the respawning of failed instances of this application. This delay prevents a broken application from using too many system resources.
- **Socket file name:**
 - On UNIX platforms: The file name of the UNIX domain socket that the application uses for communication with the Web server. The module creates the socket within the directory specified by FastCgiIpcDir. This option makes the application accessible to other applications, for example, cgi-fcgi on the same machine, or through an external FastCGI application definition, FastCgiExternalServer. If neither the -socket nor the -port options are given, the module generates a UNIX domain socket file name. The -socket and -port options are mutually exclusive.

- On Windows operating systems: The name of the pipe for the application to use for communication with the Web server. The module creates the named pipe off the named pipe root specified by the FastCgiIpcDir directive. This option makes the application accessible to other applications, like cgi-fcgi on the same machine or through an external FastCGI application definition, FastCgiExternalServer. If neither the -socket nor the -port options are given, the module generates a name for the named pipe. The -socket and -port options are mutually exclusive. If the file name does not begin with a slash (/), then this file name is assumed relative to the ServerRoot.

Distributed operating systems

FastCGIsuEXEC directive

The FastCGIsuEXEC directive supports the suEXEC-wrapper.

Directive	Description
Syntax	FastCgiSuexec On Off file name
Scope	Server configuration
Default	FastCgiSuexec Off
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	The FastCgiSuexec directive requires suEXEC enabling in Apache for CGI. To use the same suEXEC-wrapper used by Apache, set FastCgiSuexec to On. To use a different suEXEC-wrapper, specify the file name of the suEXEC-wrapper. If the file name does not begin with a slash (/), then the file name is assumed relative to the ServerRoot.

When you enable the FastCgiSuexec directive, the location of static or external FastCGI application definitions becomes important. These differences inherit their user and group from the User and Group directives in the virtual server in which they were defined. User and Group directives should precede FastCGI application definitions. This function does not limit the FastCGI application to the virtual server in which it was defined. The application can service requests from any virtual server with the same user and group. If a request is received for a FastCGI application, without an existing matching definition running with the correct user and group, a dynamic instance of the application starts with the correct user and group. This action can lead to multiple copies of the same application running with a different user and group. If this causes a problem, preclude navigation to the application from other virtual servers, or configure the virtual servers with the same user and group.

See the Apache documentation for more information about suEXEC and the security implications.

Managing IBM HTTP Server remotely with the WebSphere Application Server administrative console

You can remotely administer and configure IBM HTTP Server using the WebSphere Application Server administrative console.

About this task

After you define a Web server definition in the WebSphere repository to represent the installed IBM HTTP Server, an administrator can administer and configure IBM HTTP Server through the WebSphere Application Server administrative console.

Administration includes the ability to start and stop the IBM HTTP Server. You can display and edit the IBM HTTP Server configuration file, and you can view the IBM HTTP Server error and access logs. The plug-in configuration file can be generated for IBM HTTP Server and propagated to the remote, or locally-installed, IBM HTTP Server.

Note: **z/OS** Administration and configuration using the WebSphere Application Server administrative console is available if IBM HTTP Server is on a managed node only. The node agent must be present to perform administration because there is no support for the IBM HTTP Server administration server.

Procedure

- **IBM HTTP Server remote administration with managed nodes:** When you install IBM HTTP Server on a remote machine with a managed node, the administration interface that handles requests between the administrative console and the IBM HTTP Server is the network deployment node agent.
 - **Windows** If you are planning on managing an IBM HTTP Server on a managed node (through nodeagent), configure the Windows service for IBM HTTP Server to *log on as local system account*. You can specify this during the installation using the create services panel.
- **Distributed operating systems** **IBM HTTP Server remote administration with unmanaged nodes:** When you install IBM HTTP Server on a remote machine *without* a managed node, the **administration server** is necessary for remote administration. The IBM HTTP Server installation includes the administration server, which installs by default during a typical IBM HTTP Server installation.

The administration server is the interface that handles requests between the administrative console and the remote IBM HTTP Server defined on the unmanaged node. The administration server must be started by a root user and defined to an unmanaged WebSphere Application Server node.
- **Distributed operating systems** **IBM HTTP Server remote administration using WebSphere Application Server Express and Base:** Administration function for IBM HTTP Server with the WebSphere Application Server Express or Base product requires installation and configuration of the administration server.

Related tasks:

Distributed operating systems “Starting and stopping the IBM HTTP Server administration server” on page 109

This topic describes how to start and stop the IBM HTTP Server administration server on distributed platforms.

Extending IBM HTTP Server functionality with third-party plug-in modules

This section contains topics on using third-party plug-in modules with IBM HTTP Server.

Distributed operating systems

Before you begin

Modules that are loaded into IBM HTTP Server, whether distributed by IBM or a third-party vendor, must comply with the following specifications:

- The openssl library cannot be loaded by IBM HTTP Server plug-in modules.
- Plug-in modules provided by IBM may use the Global Security Kit (GSKit) library for SSL communications. These plug-in modules must comply with the GSKit restrictions for using a local GSKit installation to interoperate with the current release of IBM HTTP Server.

About this task

You can build third-party plug-in modules (dynamic shared object modules) for execution with IBM HTTP Server. IBM HTTP Server ships as an installation image with executables that you cannot rebuild because the source does not ship with the installation image. However, IBM HTTP Server does ship the header files necessary to compile and build third-party plug-in modules that execute as an IBM HTTP Server module.

Important: The use of third-party plug-in modules does not prevent IBM HTTP Server from being supported, but IBM cannot support the third-party plug-in module itself. If a problem occurs when the third-party plug-in module is loaded, IBM support might ask for the problem to be reproduced without the third-party plug-in module loaded, in order to determine if the problem is specific to the configuration with the third-party plug-in module. If a problem is specific to the configuration with the third-party plug-in module, the provider of that module might need to help determine the cause of the problem. IBM cannot resolve such problems without the involvement of the provider of the module, as this requires understanding of the implementation of the module, particularly with regard to its use of the Apache APIs.

Procedure

- Identify viable compilers. Apache and third-party plug-in module testing incorporated the compilers and compiler levels that are listed in this topic.
- **AIX** **HP-UX** **Linux** **Solaris** **z/OS** Determine the method to use to build the dynamic modules. Two common options for building dynamic modules are described in this topic.
- **Windows** Considerations for building dynamic modules. Restrictions apply when building a module to run with IBM HTTP Server. This topic describes the restrictions.

Viable compilers for Apache and third-party plug-in modules

There are many viable compilers and compiler levels, which have been tested, that you can use for Apache and third-party plug-in modules.

Apache modules and third-party module testing incorporated the compilers and compiler levels that are included in the following list. Other compilers may work, but testing was limited to the following environments:

- **AIX** xlc V9
- **HP-UX** HP_UXaC++ Compiler (A.03.xx)
- **Linux** Linux platforms:
 - Linux on Intel: gcc 3.3.3
 - Linux on POWER®: gcc 3.3.3

- Linux on zSeries: gcc 3.3.3
- **Solaris** SunWorkShop V5.0
- **Windows** Microsoft Visual C++ 6.0
- **z/OS** z/OS V1R6.0 C/C++

The primary concern with determining if a different compiler can be used is when the third-party module, or libraries it uses, are implemented in C++. Different compiler versions may use different C++ application binary interfaces (ABI), in which case the behavior is undefined.

Build method options for dynamic modules

There are two common methods you can use to build dynamic modules: Apache extension tool (apxs) and module-provided configuration scripts.

The two common options for building dynamic modules are:

- **Apache extension tool (apxs).** IBM HTTP Server provides the apxs tool for building dynamic modules. You can build and install most modules with apxs. Here is an example:

```
# /usr/IBMIHS/bin/apxs -ci mod_example.c
```

To use the apxs tool, verify that Perl V5.004 or later is installed and that the path to the Perl executable on the first line of apxs is correct. See Apache APXS for more information.

- **Module-provided configuration scripts.** Some complex modules cannot be built directly with apxs, and instead provide their own configuration scripts for building the module. Consult the documentation provided with the module for detailed instructions. Check for special configuration options that must point to the IBM HTTP Server installation directory, or the apxs program installed with IBM HTTP Server.

The configuration scripts for some modules check specifically for the use of Apache HTTP Server and will not work properly with IBM HTTP Server. In that case, install Apache V2.2.8 and build the module for Apache V2.2.8, then use the resulting dynamic module (mod_example.so) with IBM HTTP Server.

IBM HTTP Server customers occasionally try to use third-party modules which do not build or run properly on their platform with either Apache HTTP Server or IBM HTTP Server. Whenever there are build or run-time concerns with third-party modules, first verify that it builds and operates properly with Apache HTTP Server on the same machine. If problems are encountered with Apache HTTP Server, the module cannot be expected to work with IBM HTTP Server.

Considerations for building dynamic modules on Windows platforms

There are restrictions that you must consider when building dynamic modules for Windows platforms.

Note: IBM HTTP Server is provided exclusively as a 32-bit application on the Windows platform, and is therefore only able to load 32-bit Apache HTTP Server modules and libraries.

There are restrictions that you must consider when building dynamic modules for Windows platforms.

The following restrictions apply when building a module to run with IBM HTTP Server:

- Link your dynamic module to the libraries that are contained in `lib` directory where the server is installed.
- The Apache HTTP Server module API is defined by the header files that are contained in the `include` directory where the server is installed. Your module should include any of these header files as needed.
- You must not modify any file or data structure that is contained in any file in the `include` directory where the server is installed.

Considerations for building dynamic modules on Unix platforms

There are restrictions that you must consider when building dynamic modules for Unix platforms.

There are restrictions that you must consider when building dynamic modules for Unix platforms.

The following restrictions apply when building a module to run with IBM HTTP Server:

- Modules and libraries added to IBM HTTP Server must match the webserver's architecture.
- For IBM HTTP Server 7.0 and earlier, IBM HTTP Server is a 64-bit application on HP-UX/ia64 and Solaris/x64. IBM HTTP Server is a 32-bit application on other platforms, regardless of which WebSphere Application Server supplement CD it was installed from.
- IBM HTTP Server 8.0 adds the ability to choose at installation time between 32-bit and 64-bit binaries on Linux (all architectures), AIX, and Solaris/SPARC.

Configuring IBM HTTP Server for SMF recording

Use System Management Facilities (SMF) to record operational statistics for IBM HTTP Server.

About this task

You can enable the recording of SMF type 103 subtype 13 records for IBM HTTP Server after you load the appropriate modules, set the appropriate directives, and update the `SMFPRMxx` parmlib member.

Procedure

1. Verify that the `mod_mpmstats` and `mod_status` modules are loaded into the server.
2. Enable the `ExtendedStatus` directive.
The `ExtendedStatus` directive is set to `on` by default.
3. Set the `SMFReportInterval` directive to the number of seconds between reports.
For example,

```
SMFReportInterval 600
```


If the server is not idle, the server writes records every interval. The default is 0. When the value for the directive is 0, no SMF records are written.
4. Enable the recording of SMF type 103 subtype 13 records by editing the `SMFPRMxx` parmlib member.

Table 3. Bytes ranges and their descriptions for SMF records

Bytes	Description
0-3	Process id
4-7	Number of ready threads
8-11	Number of busy threads
12-15	Number of reading threads
16-19	Number of writing threads
20-23	Number of logging threads
24-27	Number of domain name server threads
28-31	Number of closing threads
32-35	Number of keepalive threads
36-43	Number of kilobytes transferred
44-51	Number of requests served
52-55	Length of serve name
56+length of server name	Server name

Classifying HTTP requests for WLM (z/OS operating systems)

Classify HTTP requests for workload management (WLM) by first enabling WLM support in IBM HTTP server. Then, map HTTP requests to one or more WLM transaction classes.

Before you begin

- Have an understanding of workload management on the z/OS operating system, including goal achievement, throughput, response time, and turnaround time. Read the topic about what workload management is and the topic about managing workloads on the z/OS operating system.
- Install and configure IBM HTTP Server.
- Authorize the user ID that the IBM HTTP Server runs under to the BPX.WLMSEVER RACF resource in the FACILITY class. Give the user ID at least READ access.

About this task

First, enable WLM support. Then, map HTTP requests to one or more WLM transaction classes. There are various ways to map HTTP requests. Three examples are provided.

Procedure

1. Enable WLM support by loading the mod_wlm module into the server.

Append the following statement to the httpd.conf file:

```
LoadModule wlm_module modules/mod_wlm.so
```

2. Map HTTP requests to one or more WLM transaction classes.

You can map the HTTP requests in various ways. This step provides three example substeps.

To classify your requests, add directives to the httpd.conf file.

In all the examples, the value of the wlmSubSysType directive corresponds to a subsystem type defined in WLM. This example uses CB, since CB is defined in

WLM and is reserved for WebSphere Application server. This directive can occur in the `httpd.conf` file only once. The scope is global only. The directive cannot exist within any other directives.

```
wlmSubSysType CB
```

- Map all HTTP requests to one WLM transaction class.

Applying all HTTP requests to one WLM transaction class is the simplest approach. The collection name of IHS corresponds to the collection name defined in the Name heading Qualifier part of the WLM ISPF panels. All the HTTP requests run in a WLM enclave associated with the WLM transaction class of IHSDEFLT.

```
wlmSubSysType CB
wlmCollectionName IHS
wlmTranClass IHSDEFLT
```

- Map two applications to two WLM transaction classes.

You can assign different WLM transaction classes to requests for different applications.

A virtual host is defined to port 9080. Two `LocationMatch` directives are defined, one for requests for the `appABC` application, and one for requests for the `appXYZ` application. Within each `LocationMatch` directive, the `wlmTranClass` directive is defined with different WLM transaction class names.

HTTP requests for the `appABC` application run in WLM enclaves associated with the `IHSABCG1` WLM transaction class. HTTP requests for the `appXYZ` application run in WLM enclaves associated with the `IHSXYZG1` WLM transaction class. Other requests are mapped to `IHSDEFLT`.

```
<VirtualHost *:9080>
ServerName example.com
wlmSubSysType CB
wlmCollectionName IHS
wlmTranClass IHSDEFLT
<VirtualHost *:9080>
<LocationMatch "/wlmSample/appABC/(extra|special)/data">
wlmTranClass IHSABCG1
</LocationMatch>
<LocationMatch "/wlmSample/appXYZ/(extra|special)/data">
wlmTranClass IHSXYZG1
</LocationMatch>
</VirtualHost>
```

- Map requests for a specific domain to WLM transaction classes.

You can assign different WLM transaction classes to requests that apply to a specific domain name and application.

A virtual host is defined to port 9080. Only HTTP requests that have a domain name of `example.com` can have a WLM transaction class assigned because the `ServerName` directive limits the requests to the domain name of `example.com`.

One `LocationMatch` directive is defined for requests for the `appABC` application. Within the `LocationMatch` directive, the `wlmCollectionName` and `wlmTranClass` directives are defined. Requests for the `appABC` application run in WLM enclaves associated with the `IHSABCP1WLM` transaction class.

A `wlmCollectionName` directive and a `wlmTranClass` directive are also defined outside the `LocationMatch` directive. Any requests that have a domain name of `example.com` in the URL but are not for the `appABC` application run in WLM enclaves associated with the WLM transaction class of `IHSWSCG1`. The `IHSWSCG1` WLM transaction class has no corresponding WLM collection name. Thus, the `wlmCollectionName` directive is set to `NA`. A

value of NA tells the IBM HTTP Server WLM module to not set any WLM collection name when creating the enclave.

```
wlmSubSysType CB
<VirtualHost *:9080>
  ServerName example.com
  <LocationMatch "/wlmSample/appABC">
    wlmCollectionName IHSMGT
    wlmTranClass IHSABCP1
  </LocationMatch>
  wlmCollectionName NA
  wlmTranClass IHSWSCG1
</VirtualHost>
```

WLM directives for IBM HTTP Server

Use the `wlmSubSysType`, `wlmCollectionName`, and `wlmTranClass` directives to classify HTTP requests for workload management (WLM).

- “`wlmSubSysType` directive”
- “`wlmCollectionName` directive”
- “`wlmTranClass` directive” on page 108

`wlmSubSysType` directive

The `wlmSubSysType` directive specifies a subsystem type defined in WLM.

The directive can occur in the `httpd.conf` file only once.

Note: If you specify the directive more than once, you receive a message in the `proc.errors` file. Your server cannot start.

Directive item	Description
Syntax	<code>wlmSubSysType wlm_subsystem</code>
Scope	You can use a value of CB for <code>wlm_subsystem</code> since the CB value is defined in WLM and is reserved for WebSphere Application Server. The scope is global only. The directive cannot exist within any other directives.
Default	None

`wlmCollectionName` directive

The `wlmCollectionName` directive specifies the collection name defined in the Name heading Qualifier part of the WLM ISPF panels.

The directive must occur at least once in the `httpd.conf` file for classification of HTTP requests for WLM.

Directive item	Description
Syntax	<code>wlmCollectionName collection_name</code>
Scope	You can optionally set the value to NA so that the WLM module does not set the WLM collection name when creating the enclave. The directive can exist within the <code>VirtualHost</code> , <code>Directory</code> , and <code>Location</code> directives. The directive can also exist outside a directive.

Directive item	Description
Default	None

wlmTranClass directive

The wlmTranClass directive specifies the transaction class defined in the Name heading Qualifier part of the WLM ISPF panels.

The directive must occur at least once in the httpd.conf file for classification of HTTP requests for WLM.

Directive item	Description
Syntax	wlmTranClass <i>transaction_class</i>
Scope	The directive can exist within the VirtualHost, Directory, and Location directives. The directive can also exist outside a directive.
Default	None

Related tasks:

“Classifying HTTP requests for WLM (z/OS operating systems)” on page 105
 Classify HTTP requests for workload management (WLM) by first enabling WLM support in IBM HTTP server. Then, map HTTP requests to one or more WLM transaction classes.

Chapter 4. Administering and configuring the administration server

Learn how to administer and configure the administration server.

Starting and stopping the IBM HTTP Server administration server

This topic describes how to start and stop the IBM HTTP Server administration server on distributed platforms.

Before you begin

You can set up the IBM HTTP Server administration server when you install IBM HTTP Server. For more information see “Installing IBM HTTP Server using the GUI” on page 7.

About this task

Start the IBM HTTP Server administration server as follows.

Note: Do not enable the IBM HTTP administration server in security-sensitive environments.

Procedure

- **Windows** From the Start menu:
 - Click **Start > Programs > IBM HTTP Server > Start Administration Server**. A message box displays that indicates the server has started.
 - If the IBM HTTP Server administration server does not start, complete the following steps:
 1. Open the Control Panel.
 2. Click **Services**.
 3. Double-click IBM HTTP Server Administration Server to start the server.

Confirm that IBM HTTP Server administration server started successfully by checking the `admin_error.log` file for a "start successful" message. If you use the developer installation option, then the IBM HTTP Server administration server does not install as a service. You have to run the `httpd.exe` file from a command line with the `-f` option. From the default directory, type:

```
httpd -f conf\admin.conf
```

- **AIX** **HP-UX** **Linux** **Solaris** The `adminctl` command starts and stops the IBM HTTP Server administration server. You can find the `adminctl` command in the `bin` subdirectory, within the IBM HTTP Server installation directory. If that directory is not in your `PATH`, the full path should be given on the command line. Start or stop the IBM HTTP Server administration server using the default `admin.conf` configuration file as follows:
 1. Run the `adminctl start` command to start the server or run the `adminctl stop` command to stop the server. Issue the commands from the default directories, based on your operating system:
 - **AIX** `/usr/IBM/HTTPServer/bin/adminctl start|stop`

```
- HP-UX Linux Solaris /opt/IBM/HTTPServer/bin/adminctl  
start|stop
```

For example, The **adminctl** command is not in your PATH, the IBM HTTP Server installation directory is /usr/IBM/HTTPServer, and the default configuration file is used as follows:

```
# /usr/IBM/HTTPServer/bin/adminctl start  
# /usr/IBM/HTTPServer/bin/adminctl stop
```

Important: The admin.conf configuration file supports single-byte characters (SBCS) only.

2. Confirm that IBM HTTP Server administration server started successfully by checking the admin_error.log.

Protecting access to the IBM HTTP Server administration server

This section describes topics on controlling access to the administration server in order to protect IBM HTTP Server configuration files.

About this task

The WebSphere Application Server administrative console can administer a remote IBM HTTP Server, on an unmanaged node, using IBM HTTPS Server administration server as the interface. Refer to the following topics for controlling access to the administration server in order to protect IBM HTTP Server configuration files.

Note: Do not enable the IBM HTTP administration server in security-sensitive environments.

Procedure

- Enable access to the administration server using the htpasswd utility
- **AIX** **HP-UX** **Linux** **Solaris** Run the setupadm script for the administration server
- **AIX** **HP-UX** **Linux** **Solaris** Set permissions manually for the administration server

Enabling access to the administration server using the htpasswd utility

The administration server is installed with authentication enabled. This means that the administration server will not accept a connection without a valid user ID and password. This is done to protect the IBM HTTP Server configuration file from unauthorized access.

Procedure

Launch the **htpasswd** utility that is shipped with the administration server. This utility creates and updates the files used to store user names and password for basic authentication of users who access your Web server. Locate **htpasswd** in the bin directory.

- **Windows** htpasswd -cm <install_dir>\conf\admin.passwd [login name]
- **AIX** **HP-UX** **Linux** **Solaris** ./htpasswd -cm <install_dir>/conf/admin.passwd [login name]

where *<install_dir>* is the IBM HTTP Server installation directory and *[login name]* is the user ID that you use to log into the administration server.

Results

The password file is referenced in the `admin.conf` file with the `AuthUserFile` directive. For further information on authentication configuration, see the Apache Authentication, Authorization and Access Control documentation.

Running the `setupadm` command for the administration server

Run the `setupadm` command if you need to configure the administration server manually or you need to modify its configuration.

Before you begin

The plugin configuration tool (PCT) is the supported way to configure the IBM HTTP Server administration server. If you cannot run PCT, manual instructions are provided in this topic.

gotcha: Prior to running `setupadm`, you must first run:

```
cd /path/to/IHS
bin/postinst -t setupadm -i $PWD
```

Procedure

1. Optional: Change the user ID and password that WebSphere Application Server uses to authenticate to the administration server. If you need to change the user ID and password, use the `htpasswd` utility. For more information, see the documentation about enabling access to the administration server using the `htpasswd` utility.
2. Use the `IHS_HOME/bin/setupadm` command to set up the administration server. You can set up the administration server to run in the following scenarios:
 - A non-root user and group, which the IBM HTTP Server Administration Server will run as when started by root
 - A non-root group
 - The path to the IBM HTTP Server configuration file
 - The path to the IBM HTTP Server administration server configuration file
 - The plug-in configuration file for WebSphere Application Server

Command syntax

```
setupadm [-silent] [-create] -usr user_name
-grp group_name -cfg IBM_HTTP_Server_configuration_file
[-plg plug-in_configuration_file]
-adm administration_server_configuration_file
```

-silent

This parameter enables the `setupadm` command to run without message text.

-create

This parameter specifies that you want to create a user and group. If you do not specify this parameter, the values for the `-usr` and `-grp` parameters must exist.

-usr

This parameter specifies the user ID that will run the administration server. This user ID value is updated in the `<User>` directive within the administration server configuration file, which is called `admin.conf`.

- grp** This parameter specifies the group name that will run the administration server. When you specify a value, it is used to change the file permissions for the configuration files and the user or group authentication files. This group name value is updated in the <Group> directive within the administration server configuration file, which is called `admin.conf`.

Ensure that you specify a unique group name for the administration server.
- cfg** This parameter defines the fully qualified path to the IBM HTTP Server web server configuration file. Within this file, the permission and group information is updated.

Note: The administration server requires both read and write access to IBM HTTP Server configuration files.
- plg** This parameter specifies the fully qualified path to the `plugin-cfg.xml` configuration file. Within this file, permissions are changed.
- adm** This parameter specifies the fully qualified path to the administration server configuration file. If you do not specify this parameter, a default administration configuration file is used that is based on the `install_root/conf/admin.conf` file.

Results

When you run the **setupadm** command, the following actions occur:

- Creates a new user and group is created on the system if you specify the **-create** parameter
- Changes the group owner of the configuration files to the group name that you specify and grants group write permissions to those files. This process allows the administration server to modify those configuration files.
- Updates the administration server configuration file with the user name and group name.
- Creates a backup file each time that you run the command.

What to do next

Complete the steps to start the IBM HTTP Server administration server. For more information, see the documentation about starting and stopping the IBM HTTP Server administration server.

The IBM HTTP Server administration server has to be started under the same user ID as the IBM HTTP Server to be able to restart it with **apachectl restart**.

Related tasks:

“Enabling access to the administration server using the `htpasswd` utility” on page 110

The administration server is installed with authentication enabled. This means that the administration server will not accept a connection without a valid user ID and password. This is done to protect the IBM HTTP Server configuration file from unauthorized access.

“Starting and stopping the IBM HTTP Server administration server” on page 109
This topic describes how to start and stop the IBM HTTP Server administration server on distributed platforms.

Setting permissions manually for the administration server

For IBM HTTP Server administration server, the setupadm script creates users and groups and sets file permissions for them. This topic describes how to do this manually.

About this task

Perform the following steps to create users and groups and set file permissions.

Procedure

- Create a new user and unique group for the IBM HTTP Server administration server.
 - **AIX**
 1. Launch SMIT.
 2. Click **Security and Users**.
 3. Click **Groups > Add a Group**.
 4. Enter the group name, for example, **admingrp**.
 5. Click **OK**. Go back to **Security and Users**.
 6. Click **Users > Add a User**.
 7. Enter the user name, for example, **adminuser**.
 8. Enter the primary group you just created.
 9. Click **OK**.
 - **HP-UX** **Linux**
 - Run the following command from a command line:

```
groupadd <group_name>
useradd -g <group_name> <user_ID>
```
 - **Solaris**
 1. Launch the administration tool.
 2. Click **Browse > Groups**.
 3. Click **Edit > Add**.
 4. Enter the group name, for example, **admingrp**.
 5. Click **OK**.
 6. Click **Browse > Users**.
 7. Click **Edit > Add**.
 8. Enter the user name, for example, **adminuser** and the primary group name, for example, **admingrp**.
 9. Click **OK**.
- **AIX** **HP-UX** **Linux** **Solaris** Updating file permissions.

Once you have created a user and group, set up file permissions as follows:

 1. Update the permissions for the targeted IBM HTTP Server conf directory.
 - a. At a command prompt, change to the directory where you installed IBM HTTP Server.
 - b. Type the following commands:

```
chgrp <group_name> <directory_name>
chmod g+rw <directory_name>
```
 2. Update the file permission for the targeted IBM HTTP Server configuration files.

- a. At a command prompt, change to the directory that contains the configuration files.
 - b. Type the following commands:

```
chgrp <group_name> <file_name>
chmod g+rw <file_name>
```
3. Update the `admin.conf` configuration file for the IBM HTTP Server administration server.
 - a. Change to the IBM HTTP Server administration server `admin.conf` directory.
 - b. Search for the following lines in the `admin.conf` file:

```
User nobody
Group nobody
```
 - c. Change those lines to reflect the user ID and unique group name you created. For example:

```
User userID
Group group_name
```
 4. Update the file permission for the targeted plug-in configuration files.
 - a. At a command prompt, change to the directory that contains the plug-in configuration files.
 - b. Type the following commands:

```
chgrp <group_name> <file_name>
chmod g+rw <file_name>
```

Results

You have set up read and write access for the configuration and authentication files. Now you can perform Web server configuration data administration.

Chapter 5. Securing IBM HTTP Server

Learn about IBM HTTP Server security, including: Secure Socket Layer (SSL), Key management, Lightweight Directory Access Protocol (LDAP) and System Authorization Facility (SAF) for z/OS systems.

Securing IBM HTTP Server

Distributed operating systems

z/OS

This section lists topic overviews for securing IBM HTTP Server.

About this task

The following topics describe specific tasks for you to secure IBM HTTP Server.

Procedure

- “Configure SSL between the IBM HTTP Server Administration Server and the deployment manager” on page 116
- “Securing with SSL communications” on page 118. For secure communication, you can set up the Secure Sockets Layer (SSL) directives in the default `httpd.conf` configuration file.
- “Setting advanced SSL options” on page 160. More advanced SSL options to secure your IBM HTTP Server are also available. Advanced SSL options include: setting the level and type of client authentication, setting cipher specifications, defining SSL for multiple-IP virtual hosts, and configuring reverse proxy setup with SSL.
- **Distributed operating systems** “Managing keys with the IKEYMAN graphical interface (Distributed systems)” on page 170. You can set up the Key Management utility (IKEYMAN) with IBM HTTP Server to create key databases, public and private key pairs and certificate requests. Use the IKEYMAN graphical user interface rather than using the command line interface.
- **Distributed operating systems** “Managing keys with the command line (Distributed systems)” on page 181. You can use IKEYCMD, which is the Java command line interface to IKEYMAN. Use the command line only if you are unable to use the graphical user interface.
- **z/OS** “Managing keys with the native key database gskkyman (z/OS systems)” on page 195 You can use the native z/OS key management (gskkyman key database) with IBM HTTP Server to create key databases, public and private key pairs and certificate requests.
- “Getting started with the cryptographic hardware for SSL (Distributed systems)” on page 196. You can use cryptographic hardware for SSL. The IBM 4758 requires the PKCS11 software for the host machine and internal firmware.
- **Distributed operating systems** Authenticating with LDAP on IBM HTTP Server using **mod_ibm_ldap** (Distributed systems). You can configure LDAP to protect files on IBM HTTP Server.
- **z/OS** “Authenticating with LDAP on IBM HTTP Server using **mod_ldap**” on page 201 You can configure LDAP to protect files on IBM HTTP Server.

- **z/OS** “Authenticating with SAF on IBM HTTP Server (z/OS systems)” on page 213. You can provide IBM HTTP Server with user authentication using the System Authorization Facility security product.

Results

Your IBM HTTP Server is secured.

Configure SSL between the IBM HTTP Server Administration Server and the deployment manager

Distributed operating systems **z/OS**

Configure Secure Sockets Layer (SSL) between the deployment manager for WebSphere Application Server and the IBM HTTP Server administration server, which is called `adminctl`.

About this task

The Application Server has new SSL management functions that need to be managed properly in order for IBM HTTP Server to connect with an SSL request. In earlier releases, SSL connections used default dummy certificates that were exchanged between IBM HTTP Server and the Application Server. In WebSphere Application Server, you must configure the Application Server to accept a self-signed certificate from IBM HTTP Server so SSL connections are accepted and transactions are completed.

If the Application Server and the IBM HTTP Server administration server are not configured correctly, the Application Server shows any errors that are received in the log file for the deployment manager. In situations where the IBM HTTP Server administration server is attempting to connect through SSL and the Application Server is not configured, you might receive an error that is similar to the following message:

```
-CWPKI0022E: SSL HANDSHAKE FAILURE: A signer with
SubjectDN "CN=localhost" was sent from target host:port "null:null".
The signer may need to be added to local trust store "c:/619/app2/profiles/Dmgr01/config/cells/rjrCe
located in SSL configuration alias "CellDefaultSSLSettings"
loaded from SSL configuration file "security.xml".
The extended error message from the SSL handshake
exception is: "No trusted certificate found".
```

```
-IOException javax.net.ssl.SSLHandshakeException:
com.ibm.jsse2.util.h: No trusted certificate found
```

Procedure

1. Obtain a server certificate. You can generate a new self-signed certificate or use the existing certificate from the IBM HTTP Server Web server plugin.
 - Use the existing self-signed certificate from the IBM HTTP Server Web server plugin.
 - Create a CMS key database file and a self-signed server certificate. Use the iKeyman utility for distributed operating systems and the gskkyman tool for z/OS operating systems. This step and later steps will assume that you are using the iKeyman utility.

- **Distributed operating systems** Use the IBM HTTP Server iKeyman utility graphical user interface or command line to create a CMS key database file and a self-signed server certificate.

Use the iKeyman utility to create a self-signed certificate for the IBM HTTP Server Administration Server and save the certificate as /conf/admin.kdb.

Note: Make note of the password and select **Stash password to a file**. The following fields are required for the certificate:

Label adminselfSigned

Common Name

fully_qualified_host_name

- **z/OS** IBM HTTP Server uses the z/OS gskkyman tool for key management to create a CMS key database file, public and private key pairs, and self-signed certificates. Alternatively, you can create a SAF keyring in place of a CMS key database file.
 - For information on gskkyman, see Key management using the native z/OS key database.
 - For information on creating SAF keyrings, see Authenticating with SAF on IBM HTTP Server and SSL keyfile directive.

2. Extract the certificate to a file using iKeyman utility.
 - a. Select the certificate that you created in Step 1. For example, adminselfSigned.
 - b. Click **Extract Certificate**. The recommended file name for extraction is C:\Program Files\IBM\HTTPServer\conf\cert.arm.

Note: Do not change the data type.

3. Modify the Administration Server configuration File, which is named admin.conf.
 - a. Configure the file to load the IBM SSL module. Uncomment the following line:


```
LoadModule ibm_ssl_module    modules/mod_ibm_ssl.so
```
 - b. Enable SSL and define a key file to use. Uncomment the following lines to enable SSL and define a key file to use:


```
SSLEnable
SSLServerCert default
Keyfile "C:/Program Files/IBM/HTTPServer5/conf/admin.kdb"
```

Note: Be aware of the following:

- The key file directive must match the name and location of a valid key file that is installed on your system.
 - You must have IBM SSL support installed for this to work.
 - The "default" in SSLServerCert is the label, or name, of the self-signed certificate that is created when the plugin-key.kdb file was created.
 - The previous example uses SSLServerCert because the default self-signed certificate in the plugin-key.kdb is not flagged as the default certificate.
4. Start the administration server for IBM HTTP Server. Verify that the log file does not contain GSKIT errors.
 5. Configure WebSphere Application Server.

- a. Log into the Administrative Console for the Application Server and start the deployment manager.
- b. Select **Security > SSL certificate and key management**.
- c. Select **Manage endpoint security configurations**. You are directed to a list of inbound and outbound endpoints.
- d. Select the outbound cell (cellDefaultSSLSettings,null). Select outbound cells because, in this setup, the Administration Console for the Application Server is the client, and the IBM HTTP Server Administration Server is the server.

Note: This setup is the opposite configuration from an SSL setup with the IBM HTTP Server plugin and the Application Server.

- e. In the Related Items section, click **Key stores and certificates**.
- f. Click **CellDefaultTrustStore**.
- g. In the Additional Properties section, click **Signer Certificates**.
- h. FTP the certificate file to the Application Server. Do not change the data type.
- i. In the collection panel for Signer Certificates, click **Add**. Enter the following information in the fields.

Table 4. Signer Certificate information

Name	Value
Alias	adminselfSigned
File name	<i>file_name</i> For example, enter the following: c:\program files\ibm\httpserver\conf\cert.arn

- j. Save the configuration changes to the administrative console.
- k. Stop the deployment manager.
- l. Start the deployment manager.

Results

The IBM HTTP Server administration server and Application Server are now configured to use SSL transactions.

Securing with SSL communications

Distributed operating systems **z/OS**

This section provides information to help you set up Secure Sockets Layer (SSL), using the default httpd.conf configuration file.

About this task

For each virtual host, set the cipher specification to use during secure transactions. The specified cipher specifications validate against the level of the Global Security Kit (GSK) toolkit that is installed on your system. Invalid cipher specifications cause an error to log in the error log. If the client issuing the request does not support the ciphers specified, the request fails and the connection closes to the client.

IBM HTTP Server has a built-in list of cipher specifications to use for communicating with clients over Secure Sockets Layer (SSL). The actual cipher specification that is used for a particular client connection is selected from those cipher specifications that both IBM HTTP Server and the client support.

Some cipher specifications provide a weaker level of security than others, and might need to be avoided for security reasons. Some of the stronger cipher specifications are more computationally intensive than weaker cipher specifications and might be avoided if required for performance reasons. You can use the `SSLCipherSpec` directive to provide a customized list of cipher specifications that are supported by the Web server in order to avoid the selection of cipher specifications that are considered too weak or too computationally intensive.

If you do not specify cipher specifications using the `SSLCipherSpec` directive, IBM HTTP Server Version 8.0 and later uses a conservative set of default ciphers. The default set of ciphers excludes SSL Version 2, null ciphers, and weak ciphers. The weak ciphers include export-grade ciphers. These defaults can be viewed at runtime in the error log by enabling **LogLevel debug** and **SSLTrace**.

Procedure

1. **Distributed operating systems** Use the IBM HTTP Server `IKEYMAN` utility (graphical user interface) or `IKEYMAN` utility (command line) to create a CMS key database file and server certificate.
2. **z/OS** IBM HTTP Server uses the `z/OS gskkyman` tool for key management to create a CMS key database file, public and private key pairs, and server certificates. Or, you can create a SAF keyring in place of a CMS key database file.
 - For information on `gskkyman`, see *Key management using the native z/OS key database*.
 - For information on creating SAF keyrings, see “*Authenticating with SAF on IBM HTTP Server (z/OS systems)*” on page 213 and `SSL keyfile` directive.
3. Enable SSL directives in the IBM HTTP Server `httpd.conf` configuration file.
 - a. Uncomment the `LoadModule ibm_ssl_module modules/mod_ibm_ssl.so` configuration directive.
 - b. Create an SSL virtual host stanza in the `httpd.conf` file using the following examples and directives.

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 443
<VirtualHost *:443>
    SSLEnable
</VirtualHost>
SSLDisable
KeyFile "c:/Program Files/IBM HTTP Server/key.kdb"
```

This second example assumes that you are enabling a single Web site to use SSL, and the server name is different from the server name that is defined in the global scope for non-SSL (port 80). Both host names must be registered in a domain name server (DNS) to a separate IP address, and you must configure both IP addresses on local network interface cards.

```
Listen 80
ServerName www.mycompany.com

<Directory "c:/Program Files/IBM HTTP Server/htdocs">
Options Indexes
AllowOverride None
order allow,deny
```

```

allow from all
<Directory>

DocumentRoot "c:/program files/ibm http server/htdocs"
DirectoryIndex index.html

<VirtualHost 192.168.1.103:80>
ServerName www.mycompany2.com
<Directory "c:/Program Files/IBM HTTP Server/htdocs2">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs2"
DirectoryIndex index2.html
</VirtualHost>

Listen 443
<VirtualHost 192.168.1.103:443>
ServerName www.mycompany2.com
SSLEnable
SSLClientAuth None
<Directory "c:/Program Files/IBM HTTP Server/htdocs2">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs2"
DirectoryIndex index2.html
</VirtualHost>

SSLDisable
KeyFile "c:/program files/ibm http server/key.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000

```

This third example assumes that you are enabling multiple Web sites to use SSL. All host names must be registered in the domain name server (DNS) to a separate IP address. Also, you must configure all of the IP addresses on a local network interface card. Use the SSLServerCert directive to identify which personal server certificate in the key database file passes to the client browser during the SSL handshake for each Web site. If you have not defined the SSLServerCert directive, IBM HTTP Server passes the certificate in the key database file that is marked (*) as the "default key".

```

Listen 80
ServerName www.mycompany.com

<Directory "c:/Program Files/IBM HTTP Server/htdocs">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>

DocumentRoot "c:/program files/ibm http server/htdocs"
DirectoryIndex index.html

<VirtualHost 192.168.1.103:80>
ServerName www.mycompany2.com
<Directory "c:/Program Files/IBM HTTP Server/htdocs2">
Options Indexes
AllowOverride None
order allow,deny

```



```

allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs2"
DirectoryIndex index2.html
</VirtualHost>

<VirtualHost 192.168.1.104:80>
ServerName www.mycompany3.com
<Directory "c:/Program Files/IBM HTTP Server/htdocs3">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs3"
DirectoryIndex index3.html
</VirtualHost>

Listen 443
<VirtualHost 192.168.1.102:443>
ServerName www.mycompany.com
SSLEnable
SSLClientAuth None
SSLServerCert mycompany
<Directory "c:/Program Files/IBM HTTP Server/htdocs">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs"
DirectoryIndex index.html
</VirtualHost>

<VirtualHost 192.168.1.103:443>
ServerName www.mycompany2.com
SSLEnable
SSLClientAuth None
SSLServerCert mycompany2
<Directory "c:/Program Files/IBM HTTP Server/htdocs2">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs2"
DirectoryIndex index2.html
</VirtualHost>

<VirtualHost 192.168.1.104:443>
ServerName www.mycompany3.com
SSLEnable
SSLClientAuth None
SSLServerCert mycompany3
<Directory "c:/Program Files/IBM HTTP Server/htdocs3">
Options Indexes
AllowOverride None
order allow,deny
allow from all
</Directory>
DocumentRoot "c:/program files/ibm http server/htdocs3"
DirectoryIndex index3.html
</VirtualHost>

```

```
SSLDisable
KeyFile "c:/program files/ibm http server/key.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
```

Secure Sockets Layer (SSL) protocol

Distributed operating systems z/OS

The Secure Sockets Layer (SSL) protocol was developed by Netscape Communications Corporation.

SSL ensures the data that is transferred between a client and a server remains private. This protocol enables the client to authenticate the identity of the server.

When your server has a digital certificate, SSL-enabled browsers can communicate securely with your server, using SSL. With SSL, you can easily establish a security-enabled Web site on the Internet, or on your private intranet. A browser that does not support HTTP over SSL cannot request URLs using HTTPS. The non-SSL browsers do not allow submission of forms that require secure communications.

SSL uses a *security handshake* to initiate a secure connection between the client and the server. During the handshake, the client and server agree on the security keys to use for the session and the algorithms to use for encryption. The client authenticates the server; optionally, the server can request the client certificate. After the handshake, SSL encrypts and decrypts all the information in both the HTTPS request and the server response, including:

- The URL requested by the client
- The contents of any submitted form
- Access authorization information, like user names and passwords
- All data sent between the client and the server

HTTPS represents a unique protocol that combines SSL and HTTP. Specify `https://` as an anchor in HTML documents that link to SSL-protected documents. A client user can also open a URL by specifying `https://` to request an SSL-protected document.

Because HTTPS (HTTP + SSL) and HTTP are different protocols and use different ports (443 and 80, respectively), you can run both SSL and non-SSL requests simultaneously. This capability enables you to provide information to users without security, while providing specific information only to browsers making secure requests. With this functionality, a retail company on the Internet can support users looking through their company merchandise without security, but then fill out order forms and send their credit card numbers using security.

Certificates

Distributed operating systems z/OS

This topic provides information on Secure Sockets Layer certificates.

Distributed operating systems Use the IBM HTTP Server IKEYMAN utility to create a CMS key database file and server certificate.

z/OS For IBM HTTP Server, use the native z/OS key management (gskkyman key database) to create a CMS key database file and server certificate.

Production Web servers must use signed certificates purchased from a Certificate Authority that supports IBM HTTP Server such as VeriSign or Thawte. The default certificate request file name is `certreq.arm`. The certificate request file is a PKCS 10 file, in Base64-encoded format.

Distributed operating systems You can use the IKEYMAN Key Management utility or IKEYMAN Key Management utility command line interface that is provided with IBM HTTP Server to create server certificates.

z/OS You can use the native z/OS key management (gskkyman key database) to create server certificates.

Self-signed certificates are useful for test purposes but should not be used in a production Web server.

For your convenience, IBM HTTP Server includes several default signer certificates. Be aware that these default signer certificates have expiration dates. It is important to verify the expiration dates of all your certificates and manage them appropriately. When you purchase a signed certificate from a CA, they will provide you access to their most recent signer certificates.

List of trusted certificate authorities on the IBM HTTP Server:

Distributed operating systems **z/OS**

Associate your public key with a digitally signed certificate from a certificate authority (CA) that is designated as a trusted root CA on your server. You can buy a signed certificate by submitting a certificate request to a certificate authority provider. The default certificate request file name is `certreq.arm`. The certificate request file is a PKCS 10 file, in Base64-encoded format.

You can create a new `.kdb` keystore file and view the list of designated trusted certificate authorities (CAs). If you are using a personal certificate and the signer is not in the list, you must obtain a signer certificate from the associated trusted certificate authority. IBM HTTP Server supports the following certificate authority (CA) software:

- Any X.509-compliant certificate authority
- Entrust
- Netscape Certificate Server
- Tivoli® PKI
- XCert

Certificate expiration dates: **Distributed operating systems**

You can display expiration dates of certificates in your key database by viewing the certificate information with the IKEYMAN Key Management utility GUI or using the `gskcmd` command.

The following is an example of how to use the `gskcmd` command to display the validity dates on all certificates in the `key.kdb` certificate key file that will expire within 1825 days (5 years):

```
<ihsinst>/bin/gskcmd -cert -list all -expiry 1825 -db key.kdb -pw <password>
```

```
Certificates in database: key.kdb
VeriSign Class 1 CA Individual Subscriber-Persona Not Validated
Validity
Not Before: Mon May 11 20:00:00 EDT 1998
Not After: Mon May 12 19:59:59 EDT 2008
```

where *<password>* is the password you specified when creating the `key.kdb` key database file.

SSL certificate revocation list: Distributed operating systems z/OS

This section provides information on identifying directives for certificate revocation list (CRL) and those supported in global servers and virtual hosts.

Certificate revocation provides the ability to revoke a client certificate given to IBM HTTP Server by the browser when the key becomes compromised or when access permission to the key gets revoked. CRL represents a database which contains a list of certificates revoked before their scheduled expiration date.

If you want to enable certificate revocation in IBM HTTP Server, publish the CRL on a Lightweight Directory Access Protocol (LDAP) server. Once the CRL is published to an LDAP server, you can access the CRL using the IBM HTTP Server configuration file. The CRL determines the access permission status of the requested client certificate. Be aware, however, that it's not always possible to determine the revocation status of a client certificate if the backend server, the source of revocation data, is not available or not communicating properly with IBM HTTP Server.

Identifying directives needed to set up a certificate revocation list. The `SSLClientAuth` directive can include two options at once:

- `SSLClientAuth 2 crl`
- `SSLClientAuth 1 crl`

The `CRL` option turns CRL on and off inside an SSL virtual host. If you specify `CRL` as an option, then you elect to turn CRL on. If you do not specify `CRL` as an option, then CRL remains off. If the first option for `SSLClientAuth` equals `0/none`, then you cannot use the second option, `CRL`. If you do not have client authentication on, then CRL processing does not take place.

Identifying directives supported in global or server and virtual host. Global server and virtual host support the following directives:

- `SSLCRLHostname`: The IP Address and host of the LDAP server, where the CRL database resides. Currently, you must configure any static CRL repositories to allow for checking of other URI forms in the `CRLDistributionPoint` fields.
z/OS Only an explicitly configured LDAP server can be queried for CRL, and the SSL handshake fails if the backend server is not reachable.
- `SSLCRLPort`: The port of the LDAP server where the CRL database resides; the default equals 389.
- `SSLCRLUserID`: The user ID to send to the LDAP server where the CRL database resides; defaults to anonymous if you do not specify the bind.
- `SSLStashfile`: The fully qualified path to file where the password for the user name on the LDAP server resides. This directive is not required for an anonymous bind. Use when you specify a user ID.

Use the **sslstash** command, located in the bin directory of IBM HTTP Server, to create your CRL password stash file. The password you specify using the **sslstash** command should equal the one you use to log in to your LDAP server.

Usage:

```
sslstash [-c] &lt;directory_to_password_file_and_file_name>; <function_name> <password>
```

where:

- **-c**: Creates a new stash file. If not specified, an existing file updates.
 - **File**: Represents the fully qualified name of the file to create, or update.
 - **Function**: Indicates the function for which to use the password. Valid values include `crl`, or `crypto`.
 - **Password**: Represents the password to stash.
- **Distributed operating systems** **SSLUnknownRevocationStatus**: This directive allows you to configure how IBM HTTP Server will respond when fresh Certificate Revocation List (CRL) information or OCSP (Online Certificate Status Protocol) information is not available, and the client certificate that is currently offered is not known to be revoked from a previous query. Certificates are presumed not to be revoked, by default, which means they are valid, and a temporary failure to obtain CRL or OCSP information does not automatically result in an SSL handshake failure. This directive is provided to respond to circumstances in which a certificate has been accepted without IBM HTTP Server being able to reliably confirm the revocation status.

This directive has an effect only when all of these conditions are true:

- IBM HTTP Server is configured to accept client certificates with the `SSLClientAuth` directive.
 - IBM HTTP Server is configured with one of the following directives: `SSLOCSPEnable`, `SSLOCSPUrl`, or `SSLCRLHostname`.
 - An SSL client certificate is provided.
 - IBM HTTP Server does not receive a valid OCSP or CRL response from the configured backend server, and the client certificate does not appear as revoked in a cached, but expired, CRL response.
- IBM HTTP Server uses a cached CRL that is beyond its published expiration time when a current version is not available. When a certificate has been revoked in such an expired CRL, this will result in a direct SSL handshake failure that is outside the scope of the `SSLUnknownRevocationStatus` directive.

See the “SSL directives” on page 134 topic for more information.

CRL checking follows the `URIDistributionPoint X509` extension in the client certificate as well as trying the DN constructed from the issuer of the client certificate. If the certificate contains a CRL Distribution Point (CDP), then that information is given precedence. The order in which the information is used is as follows:

1. CDP LDAP X.500 name
2. CDP LDAP URI
3. Issuer name combined with the value from the `SSLCRLHostname` directive

gotcha: If your certificates use the LDAP or HTTP URI forms of the `CertificateDistributionPoint` or `AIA` extensions, be sure that the IBM HTTP Server system can establish outgoing connections of this type; you might need to adjust the settings for your firewall.

Obtaining certificates: Distributed operating systems z/OS

This section provides information to help you get started with secure connections on the Web server. Obtaining certificates is the first step in securing your Web server.

About this task

When you set up secure connections, associate your public key with a digitally-signed certificate from a certificate authority (CA) that is designated as a trusted CA on your server.

Procedure

- **Buy a certificate from an external certificate authority provider.** You can buy a signed certificate by submitting a certificate request to a CA provider. The IBM HTTP Server supports several external certificate authorities. By default, many CAs exist as trusted CAs on the IBM HTTP Server. See “List of trusted certificate authorities on the IBM HTTP Server” on page 123.

Use the key management utility to create a new key pair and certificate request to send to an external CA, then define SSL settings in the `httpd.conf` file.

- Distributed operating systems IKEYMAN graphical user interface. If you are unable to use the IKEYMAN interface, use the command line interface `gskcmd` command.
- z/OS Native z/OS key management (`gskkyman` key database).
- **Create a self-signed certificate.** Use the key management utility or purchase certificate authority software from a CA provider.

Public Key Infrastructure

Distributed operating systems z/OS

A Public Key Infrastructure (PKI) represents a system of digital certificates, certificate authorities, registration authorities, a certificate management service, and X.500 directories.

A PKI verifies the identity and the authority of each party that is involved in an Internet transaction, either financial or operational, with requirements for identity verification. Examples of these transactions include confirming the origin of proposal bids, or the author of e-mail messages.

A PKI supports the use of *certificate revocation lists* (CRLs). A CRL is a list of revoked certificates. CRLs provide a more global method for authenticating client identity by certificate, and can verify the validity of trusted CA certificates.

An X.500 directory server stores and retrieves CRLs and trusted CA certificates. The protocols used for storing and retrieving information from an X.500 directory server include Directory Access Protocol (DAP) and Lightweight Directory Access Protocol (LDAP). The IBM HTTP Server supports LDAP.

You can distribute information on multiple directory servers over the Internet and intranets, enabling an organization to manage certificates, trust policy, and CRLs from either a central location, or in a distributed manner. This capability makes the trust policy more dynamic because you can add or delete trusted CAs from a network of secure servers, without having to reconfigure each of the servers.

Session ID cache

Distributed operating systems z/OS

IBM HTTP Server caches secure sockets layer (SSL) session IDs when Web clients establish secure connections with the Web server. Cached session IDs enable subsequent SSL session requests to use a shortened SSL handshake during session establishment. Session ID caching is enabled by default on all supported platforms.

AIX HP-UX Linux Solaris z/OS The session ID cache is implemented as a daemon process named **sidd**. You will see this process running when IBM HTTP Server is started with SSL enabled.

Distributed operating systems In most cases, you will not need to take an additional configuration steps to effectively use SSL session ID caching in IBM HTTP Server.

SSL directive considerations

Distributed operating systems z/OS

When using SSL directives, you should consider the following: Limiting encryption to 128 bits or higher, rewriting HTTP (port 80) requests to HTTPS (port 443), logging SSL request information in the access log, and enabling certificate revocation lists (CRL).

You should consider the following when you want to enable SSL directives in the IBM HTTP Server `httpd.conf` configuration file:

- **Limiting IBM HTTP Server to encrypt at only 128 bits or higher.** There are several methods of configuring IBM HTTP Server to restrict and limit SSL to allow only 128 bit browsers and 128,168 bit ciphers access to Web content. For complete information, refer to [Limiting IBM HTTP Server to encrypt at only 128 bits or higher](#).
- **How to rewrite HTTP (port 80) requests to HTTPS (port 443).** The `mod_rewrite.c` rewrite module provided with IBM HTTP Server can be used as an effective way to automatically rewrite all HTTP requests to HTTPS. For complete information refer to [How to rewrite HTTP \(port 80\) requests to HTTPS \(port 443\)](#).
- **Logging SSL request information in the access log for IBM HTTP Server.** The IBM HTTP Server implementation provides Secure Sockets Layer (SSL) environment variables that are configurable with the `LogFormat` directive in the `httpd.conf` configuration file. For complete information refer to [Logging SSL request information in the access log for IBM HTTP Server](#).
- **Enabling certificate revocation lists (CRL) in IBM HTTP Server.** Certificate revocation provides the ability to revoke a client certificate given to the IBM HTTP Server by the browser when the key is compromised or when access permission to the key is revoked. CRL represents a database that contains a list of certificates revoked before their scheduled expiration date. For complete information refer to [“SSL certificate revocation list” on page 124](#).

Authentication

Distributed operating systems z/OS

Authentication verifies identity.

The server uses authentication in two ways:

- **Digital signature.** A digital signature represents a unique mathematically computed signature that ensures accountability. Think of a digital signature as similar to a credit card, on which your photo displays. To verify the identity of the person that is sending you a message, look at the digital certificate of the sender.
- **Digital certificate.** A digital certificate, or digital ID, is similar to having a credit card with a picture of the bank president with his arm around you. A merchant trusts you more because not only do you look like the picture on the credit card, the bank president trusts you, too.

You base your trust of the sender authenticity on whether you trust the third party, a person, or agency that certified the sender. The third party issuing digital certificates is called a certificate authority (CA) or *certificate signer*.

A digital certificate contains:

- The public key of the person getting certified
- The name and address of the person or organization getting certified, also known as the *distinguished name*
- The digital signature of the CA
- The issue date of the certificate
- The expiration date of the certificate

You enter your distinguished name as part of a certificate request. The digitally signed certificate includes your distinguished name and the distinguished name of the CA.

You can request one of the following certificates:

- A server certificate to do commercial business on the Internet from VeriSign or some other CA. For a list of supported CAs, see *Buying a certificate from an external CA provider*.
- A server certificate that you create for your own private Web network.

CAs broadcast their public key and distinguished name bundled together so that people add them to their Web servers and browsers, as a trusted CA certificate. When you designate the public key and certificate from a CA to become a trusted CA certificate, your server trusts anyone who has a certificate from that CA. You can have many trusted CAs as part of your server. The HTTP Server includes several default trusted CA certificates.

Distributed operating systems You can add or remove trusted CAs using the IBM Key Management utility (ikeyman) that is included with your server.

z/OS You can add or remove trusted CAs using the native z/OS key management (gskkyman).

To communicate securely, the receiver in a transmission must trust the CA who issued the sender certificate. This situation remains true whether the receiver is a Web server or a browser. When a sender signs a message, the receiver must have the corresponding CA-signed certificate and public key designated as a trusted CA certificate.

Encryption

Distributed operating systems **z/OS**

Encryption in its simplest form involves scrambling a message so that no one can read the message until it is unscrambled by the receiver.

The sender uses an algorithmic pattern, or a key to scramble, or encrypt the message. The receiver has the decryption key. Encryption ensures privacy and confidentiality in transmissions sent over the Internet.

Use two different kinds of keys for encryption:

Asymmetric keys. You create a key pair with asymmetric keys. The key pair consists of a public key and a private key, which differ from each other. The private key holds more of the secret encryption pattern than the public key. Do not share your private key with anyone.

The server uses its private key to sign messages to clients. The server sends its public key to clients so that they can encrypt messages to the server, which the server decrypts with its private key. Only you can decrypt a message that is encrypted with your public key because only you have the private key. Key pairs are stored in a key database that is protected by a password.

Symmetric keys. Symmetric keys follow an older model of the sender and receiver sharing some kind of pattern. The sender uses this same pattern to encrypt the message and the receiver uses this pattern to decrypt the message. The risk involved with symmetric keys centers around finding a safe transportation method to use, when sharing your secret key with the people to which you want to communicate.

The Secure Sockets Layer (SSL) protocol uses both asymmetric and symmetric key exchange. Use asymmetric keys for the *SSL handshake*. During the handshake, the master key, encrypted with the receiver public key passes from the client to the server. The client and server make their own session keys using the master key. The session keys encrypt and decrypt data for the remainder of the session. Symmetric key exchange occurs during the exchange of the cipher specification, or encryption level.

The server needs a *digital certificate*, which is an encrypted message that authenticates Web content, to send its public key to clients. A certificate authority (CA), which signs all certificates that it issues with a private key, issues this certificate and verifies the identity of the server.

Secure Sockets Layer environment variables

Distributed operating systems

z/OS

The `mod_ibm_ssl` parameter provides access to information about an Secure Sockets Layer (SSL) session by setting variables in the Apache API `subprocess_env` table for the active request. These variables are considered environment variables because of how information is accessed when the variables are passed to CGI applications.

You can categorize SSL environment variables into three types based on the type of information that is accessed when the variable is passed to the application.

- Variables for information regarding the SSL handshake
- Variables for exposing the server certificate information
- Variables for exposing client certificate information, when client authentication is enabled.

The following table provides the types of access to information as well as the mechanisms used to access information using SSL environment variables.

Table 5. Types of access and mechanisms for SSL environment variables

Access type	Mechanism
access from a CGI or FastCGI application	The information is passed to the CGI application as an environment variable. Use the method provided by the implementation language for accessing environments, such as <code>getenv ("HTTPS")</code> in C or <code>\$ENV{'HTTPS'}</code> in Perl. For a SSL environment variable to be used in CGI or FastCGI, there must be a corresponding <code>PassEnv</code> directive.
access from a plug-in module	The information is available in the <code>subprocess_env</code> table after the quick handler has run. Access it with a call such as <code>apr_table_lookup (r->subprocess_env, "HTTPS")</code>
logging in the access log with other information about the request	Use the following <code>%{varname}</code> example. <code>LogFormat "%h %l %u %t \ "%r\ " %>s %b %{HTTPS}e" ssl-custom</code> If the information is not available, <code>mod_log_config</code> logs a dash (-) for the field.
use with the <code>setenvif</code> variable	# Silly example, don't compress SSL connections <code>SetEnvIf HTTPS no-gzip</code>
use as part of a <code>mod_rewrite</code> rule variable	<code>RewriteEngine On</code> <code>RewriteCond %{ENV:HTTPS} ^OFF\$</code> <code>RewriteRule .* /no-ssl.html</code>
access in an SSI document	In order for an SSL environment variable to be used in an SSI document, there must be a corresponding <code>PassEnv</code> directive. <code>SSL is <!--#echo var="HTTPS" --></code>
access control	Allow from env=HTTPS

SSL handshake environment variables

Distributed operating systems z/OS

Secure Sockets Layer (SSL) handshake environment variables are used to access server certificate information. When an SSL handshake is successfully completed, the SSL handshake environment variables are automatically set.

Variables

Table 6. SSL handshake environment variables. The table provides a list of SSL handshake environment variables with their descriptions and values.

SSL handshake environment variable	Description	Value
HTTPS	Indicates SSL connection	String contains either ON, for an SSL connection, or OFF, if not.

Table 6. SSL handshake environment variables (continued). The table provides a list of SSL handshake environment variables with their descriptions and values.

SSL handshake environment variable	Description	Value
HTTPS_CIPHER	Contains the cipher used in the SSL handshake.	See the following table.
HTTPS_KEYSIZE	Indicates the size of the key.	See the following table.
HTTPS_SECRETKEYSIZE	Indicates the strength of the key.	See the following table.
SSL_PROTOCOL_VERSION	Contains the protocol version.	<p>z/OS String contains either SSLV2, SSLV3, or TLSV1 for Transport Layer Security (TLS) Version 1.0).</p> <p>Distributed operating systems String contains SSLV2, SSLV3, TLSV1 for TLS Version 1.0, or TLSV1.1 for TLS Version 1.1.</p>

Table 7. Variables for HTTPS_KEYSIZE and HTTPS_SECRETKEYSIZE in Secure Sockets Layer V3 and Transport Layer Security V1. The table provides the cipher suite, the key size, and the secret key size.

Cipher suite	Key size	Secret key size
SSL_RSA_WITH_NULL_MD5	0	0
SSL_RSA_WITH_NULL_SHA	0	0
SSL_RSA_EXPORT_WITH_RC4_40_MD5	128	40
SSL_RSA_WITH_RC4_128_MD5	128	128
SSL_RSA_WITH_RC4_128_SHA	128	128
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	128	40
SSL_RSA_WITH_DES_CBC_SHA	64	56
SSL_RSA_WITH_3DES_EDE_CBC_SHA	192	168
SSL_NULL_WITH_NULL_NULL	0	0
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	56	20
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	56	20

Table 8. Variables for HTTPS_CIPHER in Secure Sockets Layer V2.. The table provides the cipher suite, the key size, and the secret key size.

Cipher suite	Key size	Secret key size
RC4_128_WITH_MD5	128	128
RC4_128_EXPORT40_WITH_MD5	128	40
RC2_128_CBC_WITH_MD5	128	128
RC2_128_CBC_EXPORT40_WITH_MD5	128	40
DES_64_CBC_WITH_MD5	64	56
DES_192_EDE3_CBC_WITH_MD5	192	168

Server certificate environment variables

Distributed operating systems z/OS

Server certificate environment variables are used to access server certificate information. The server certificate environment variables are automatically set. If client authentication is not configured, references to these values are empty.

Variables

The following table provides a list of server certificate environment variables with their descriptions and values.

Server certificate environment variable	Description	Value
SSL_SERVER_C	Contains the country attribute of the server certificate	String
SSL_SERVER_CN	Contains the common name attribute of the server certificate	String
SSL_SERVER_DN	Contains the distinguished name of the server certificate used in the IP-based virtual host which received the request	String
SSL_SERVER_EMAIL	Contains the e-mail attribute of the server certificate	String
SSL_SERVER_L	Contains the locality attribute of the server certificate	String
SSL_SERVER_O	Contains the organization attribute of the server certificate	String
SSL_SERVER_OU	Contains the organizational unit attribute of the server certificate	String
SSL_SERVER_ST	Contains the state or province attribute of the server certificate	String

Client certificate environment variables

Distributed operating systems z/OS

Client certificate environment variables are used to access client certificate information when client authentication is enabled. If client authentication is not enabled, references to these values are empty.

Variables

The following table provides a list of client certificate environment variables and their descriptions and values.

SSL client certificate environment variable	Description	Value
SSL_CLIENT_C	Contains the client certificate country	String
SSL_CLIENT_CERTBODY	Contains the client certificate	This value is the unformatted body of the client certificate, if a certificate was provided by the client
SSL_CLIENT_CERTBODYLEN	Contains the length of the client certificate	Integer
SSL_CLIENT_CN	Contains the client certificate common name	String
SSL_CLIENT_DN	Contains the distinguished name from the client certificate	String
SSL_CLIENT_EMAIL	Contains the client certificate e-mail	String
SSL_CLIENT_IC	Contains the country name of the client certificate issuer	String
SSL_CLIENT_ICN	Contains the common name of the client certificate issuer	String
SSL_CLIENT_IDN	Contains the distinguished name of the client certificate issuer	String
SSL_CLIENT_IEMAIL	Contains the e-mail address of the client certificate issuer	String
SSL_CLIENT_IL	Contains the locality of the client certificate issuer	String
SSL_CLIENT_IO	Contains the organization name of the client certificate issuer	String
SSL_CLIENT_IOU	Contains the organizational unit name of the client certificate issuer	String
SSL_CLIENT_IPC	Contains the postal code of the client certificate issuer	String
SSL_CLIENT_IST	Contains the state or province of the client certificate issuer	String
SSL_CLIENT_L	Contains the client certificate locality	String
SSL_CLIENT_NEWSESSIONID	Indicates whether this session ID is new	String. This value must be TRUE or FALSE.
SSL_CLIENT_O	Contains the client certificate organization	String
SSL_CLIENT_OU	Contains the client certificate organizational unit	String
SSL_CLIENT_PC	Contains the client certificate postal code	String
SSL_CLIENT_SERIALNUM	Contains the client certificate serial number	String

SSL client certificate environment variable	Description	Value
SSL_CLIENT_SESSIONID	Contains the session ID	String
SSL_CLIENT_ST	Contains the client certificate state or province	String

SSL directives

Distributed operating systems z/OS

Secure Sockets Layer (SSL) directives are the configuration parameters that control SSL features in IBM HTTP Server.

Most SSL directives in IBM HTTP Server have the same behavior. A directive specified for a given virtual host configuration overrides a directive specified in the base server configuration. Also, a directive specified for a child directory overrides a directive specified for its parent directory. However, there are exceptions.

For example, when no directive is specified for a virtual host, the directive specified in the base server configuration might be copied to the virtual host configuration. In this case, the directive in the base server configuration overrides the virtual host configuration.

Attention: The SSLEnable directive should not be specified in the base server configuration if you do not want the directive automatically copied to a given virtual host configuration.

Also, a directive specified for a child directory might be appended to the directive specified for its parent directory. In this case, the directive for the parent directory does not override the directive for the child directory, but instead is appended to it and both directives are applied to the child directory.

The following list contains the SSL directives for IBM HTTP Server.

- “SSLOCSPResponderURL” on page 135
- “SSLOCSPEnable” on page 136
- “Keyfile directive” on page 136
- “SSLAcceleratorDisable directive” on page 138
- Distributed operating systems “SSLAllowNonCriticalBasicConstraints directive” on page 138
- AIX HP-UX Linux Solaris z/OS “SSLCacheDisable directive” on page 139
- AIX HP-UX Linux Solaris z/OS “SSLCacheEnable directive” on page 139
- AIX HP-UX Linux Solaris z/OS “SSLCacheErrorLog directive” on page 139
- AIX HP-UX Linux Solaris z/OS “SSLCachePath directive” on page 139
- AIX HP-UX Linux Solaris z/OS “SSLCachePortFilename directive” on page 140
- AIX HP-UX Linux Solaris z/OS “SSLCacheTraceLog directive” on page 140
- “SSLCipherBan directive” on page 141
- “SSLCipherRequire directive” on page 141
- “SSLCipherSpec directive” on page 141
- “SSLClientAuth directive” on page 143

- “SSLClientAuthGroup directive” on page 145
- “SSLClientAuthRequire directive” on page 146
- “SSLClientAuthVerify directive” on page 148
- “SSLCRLHostname directive” on page 149
- “SSLCRLPort directive” on page 149
- “SSLCRLUserID directive” on page 150
- “SSLDisable directive” on page 150
- “SSEnable directive” on page 151
- “SSLFakeBasicAuth directive” on page 151
- **Distributed operating systems** “SSLFIPSDisable directive” on page 151
- “SSLFIPSEnable directive” on page 152
- “SSLInsecureRenegotiation directive” on page 152
- “SSLPKCSDriver directive” on page 152
- “SSLProtocolDisable directive” on page 153
- “SSLProtocolEnable directive” on page 154
- “SSLProxyEngine directive” on page 154
- “SSLRenegotiation directive” on page 154
- “SSLServerCert directive” on page 155
- “SSLSNIMap” on page 155
- “SSLStashfile directive” on page 156
- “SSLSuiteBMode” on page 157
- “SSLTrace directive” on page 157
- **Distributed operating systems** “SSLUnknownRevocationStatus” on page 158
- “SSLV2Timeout directive” on page 159
- “SSLV3Timeout directive” on page 159
- “SSLVersion directive” on page 159

SSLOCSPResponderURL

The SSLOCSPResponderURL directive enables checking of client certificates through a statically configured online certificate status protocol (OCSP) responder.

Name	Description
Syntax	
	Distributed operating systems
	SSLOCSPResponderURL<URL>
Scope	Virtual host
Default	Disabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	One per virtual host
Values	A fully qualified URL that points to an OCSP responder, for example, http://hostname:2560/.

Even if CRL checking is configured, OCSP checking is performed before any CRL checking. CRL checking occurs only if the result of the CRL is unknown or inconclusive.

If SSLOCSPResponderURL is set, IBM HTTP Server uses the supplied URL to check for certificate revocation status when an SSL client certificate is provided.

If both SSLOCSPEnable and SSLOCSPResponderURL are configured, the responder defined by SSLOCSPResponderURL is checked first. If the revocation status is unknown or inconclusive, IBM HTTP Server checks OCSP responders for SSLOCSPEnable.

Note: In some cases IBM HTTP Server might not be able to determine the revocation status of a client certificate, because the backend server, which is the source of the revocation data, is not available. You should be aware that:

- A static CRL repository (SSLCRLHost) must be configured to enable checking of other URI forms in the CRLDistributionPoint fields.
- If your certificates use the LDAP or HTTP URI forms of the CertificateDistributionPoint or AIA extensions, be sure that the IBM HTTP Server system can establish outgoing connections of this type; you must adjust the settings for your firewall.
- **Distributed operating systems** The SSLUnknownRevocationStatus directive is provided for cases in which recoverable errors occur in IBM HTTP Server when it is communicating with the backend server, and the IBM HTTP Server cannot determine the revocation status of a certificate. The default behavior is to continue processing the handshake unless the backend server can successfully indicate that the certificate is revoked.
- **z/OS** Only an explicitly configured LDAP server can be queried for CRL, and the SSL handshake fails if the backend server is not reachable.

SSLOCSPEnable

The SSLOCSPEnable directive enables checking of client certificates through OCSP responders defined in the Authority Information Access (AIA) extension of their certificate.

Name	Description
Syntax	Distributed operating systems SSLOCSPEnable
Scope	Virtual host
Default	Disabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance permitted for each virtual host
Values	None

If SSLOCSPEnable is set, and an SSL client certificate chain contains an AIA extension, IBM HTTP Server contacts the OCSP responder indicated by the AIA extension to check revocation status of the client certificate.

If both OCSP and CRL checking is configured, OCSP checking is performed before any CRL checking. CRL checking occurs only if the result of the OCSP checking is unknown or inconclusive.

If both SSLOCSPEnable and SSLOCSPResponderURL are configured, the responder defined by SSLOCSPResponderURL is checked first. If the revocation status is unknown or inconclusive, IBM HTTP Server checks OCSP responders for SSLOCSPEnable.

Keyfile directive

The keyfile directive sets the key file to use.

Attention: This directive might be overridden by the base server configuration.

Name
Syntax

Description

AIX

Solaris

HP-UX

Linux

Windows

Keyfile [/prompt]
*/fully qualified path to key
file/keyfile.kdb*

Attention: **z/OS** The /prompt function is only supported when running from a USS shell, not from a JCL started job. If you attempt to use the /prompt function from a JCL started job, then a configuration error occurs.

z/OS You can use a keyring stored in the Hierarchical File System (HFS) or in the System Authorization Facility (SAF). To use a keyring stored in HFS:

- Keyfile */fully qualified path to key file/keyfile.kdb*

To use a keyring stored in SAF:

- Keyfile /saf WASKeyring

Important: With SAF keyrings:

- There is no stash file when using SAF, and access is controlled by SAF rules. Therefore, if you attempt to use the keyfile/prompt/saf argument, the argument is not supported. An attempt to use this argument results in a configuration error.
- The ID that is used to start IBM HTTP Server must have access to the keyring named in this directive. If the ID does not have access, SSL initialization fails.

Global base and virtual host

None

mod_ibm_ssl

One instance per virtual host and global server

File name of the key file.

Distributed operating systems Use the prompt option to enable the HTTP server to prompt you for the Key file password at start time.

z/OS File system protection can be used to limit access. Use the SAF (System Authorization Facility) keyrings for limiting access to SSL certificates.

Scope

Default

Module

Multiple instances in the configuration file

Values

Important: **z/OS** The z/OS system does not support key database files created on other platforms. Key database files used for z/OS systems must be created on the z/OS platform.

You can use only one of the following configurations for the key file type:

- **Distributed operating systems** Certificate Management Services (CMS)

- **z/OS** CMS or Resource Access Control Facility (RACF)

SSLAcceleratorDisable directive

The SSLAcceleratorDisable directive disables the accelerator device.

Name	Description
Syntax	SSLAcceleratorDisable
Scope	Virtual and global
Default	Accelerator device is enabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host.
Values	None. Place this directive anywhere inside of the configuration file, including inside a virtual host. During initialization, if the system determines that an accelerator device is installed on the machine, the system uses that accelerator to increase number of secure transactions. This directive does not take arguments.

Distributed operating systems

SSLAllowNonCriticalBasicConstraints directive

The SSLAllowNonCriticalBasicConstraints directive enables compatibility with one aspect of the GPKI specification from the government of Japan that conflicts with RFC3280.

Name	Description
Syntax	SSLAllowNonCriticalBasicConstraints <i>on off</i>
Scope	Global server or virtual host
Default	Off
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server
Values	None. This directive changes the behavior of the certificate validation algorithm such that a non-critical basic constraints extension on an issuer certificate authority (CA) certificate does not cause a validation failure. This enables compatibility with one aspect of the GPKI specification from the government of Japan that conflicts with RFC3280.

Attention: RFC3280 states that this extension must appear as a critical extension in all CA certificates that contain public keys used to validate digital signatures on certificates.

AIX

HP-UX

Linux

Solaris

z/OS

SSLCacheDisable directive

The SSLCacheDisable directive disables the external SSL session ID cache.

Name	Description
Syntax	SSLCacheDisable
Scope	One per physical Apache server instance, enabled only outside of virtual host stanzas.
Default	None
Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.
Values	None.

AIX HP-UX Linux Solaris z/OS

SSLCacheEnable directive

The SSLCacheEnable directive enables the external SSL session ID cache.

Name	Description
Syntax	SSLCacheEnable
Scope	One per physical Apache server instance, enabled only outside of virtual host stanzas.
Default	None
Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.
Values	None.

AIX HP-UX Linux Solaris z/OS

SSLCacheErrorLog directive

The SSLCacheErrorLog directive sets the file name for session ID cache.

Name	Description
Syntax	SSLCacheErrorLog /usr/HTTPServer/logs/sidd_logg
Scope	Server configuration outside of virtual host.
Default	None
Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.
Values	Valid file name.

AIX HP-UX Linux Solaris z/OS

SSLCachePath directive

The SSLCachePath directive specifies the path to the session ID caching daemon. Unless multiple instances of IHS, with multiple **ServerRootor -d** parameters are sharing one installation, this directive is not required to be specified.

When multiple instances of IHS are being used with an alternate server root as previously described, this directive should be used to point this instance of IHS at the path to the bin/sidd binary in the single installation root instead of the separate server roots which are used by default.

There is no practical reason to copy the bin/sidd binary around, or to use this directive to specify anything other than the bin/sidd installed under the server root when multiple instances are used. The value of this directive does not have to vary between instances of IHS sharing the same binaries.

Name	Description
Syntax	SSLCachePath /usr/HTTPServer/bin/sidd
Scope	Server configuration outside of virtual host.
Default	<server-root>/bin/sidd
Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.
Values	Valid path name.

AIX
HP-UX
Linux
Solaris
z/OS

SSLCachePortFilename directive

The SSLCachePortFilename directive sets the file name for the UNIX domain socket that is used for communication between the server instances and the session ID cache daemon. You must set this directive if you run two instances of IBM HTTP Server from the same installation directory and both instances are configured for SSL. Otherwise, you do not need to set this directive.

Name	Description
Syntax	SSLCachePortFilename /usr/HTTPServer/logs/siddport
Scope	Server configuration outside of virtual host.
Default	If this directive is not specified and the cache is enabled, the server attempts to use the <server-root>/logs/siddport file.

NOTES

- For AIX: the default is /usr/HTTPServer/logs/siddport.
- For Solaris: the default is /opt/IBMHTTPD/logs/siddport .
- Not valid on Windows NT

Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.
Values	Valid path name. The web server deletes this file during startup; do not use an existing filename.

AIX
HP-UX
Linux
Solaris
z/OS

SSLCacheTraceLog directive

The SSLCacheTraceLog directive specifies the file to which the session ID trace messages are written. Without this directive, tracing is disabled.

Name	Description
Syntax	SSLCacheTraceLog /usr/HTTPServer/logs/sidd-trace.log
Scope	Server configuration outside of virtual host.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	Not permitted.

Name	Description
Values	Valid path name.

SSLCipherBan directive

The SSLCipherBan directive denies access to an object if the client has connected using one of the specified ciphers. The request fails with a 403 status code.

Attention: This directive, when specified for a child directory, does not override the directive specified for the parent directory. Instead, both directories are applied to the child directory.

Name	Description
Syntax	SSLCipherBan < <i>cipher_specification</i> >
Scope	Multiple instances per directory stanza.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	Permitted per directory stanza. Order of preference is top to bottom.
Values	See the SSL cipher specification topic for values.

SSLCipherRequire directive

The SSLCipherRequire directive restricts access to objects to clients that have connected using one of the specified ciphers. If access is denied, the request fails with a '403' status code.

Attention: This directive, when specified for a child directory, does not override the directive specified for the parent directory. Instead, both directories are applied to the child directory.

Name	Description
Syntax	SSLCipherRequire < <i>cipher_specification</i> >
Scope	Multiple instances per directory stanza.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	Permitted per directory stanza.
Values	See the SSL cipher specification topic for values

SSLCipherSpec directive

The SSLCipherSpec directive enables you to customize the SSL ciphers supported during the handshake. You can customize the set of SSL ciphers and the order of preference of the SSL ciphers.

Distributed operating systems On distributed platforms, each protocol has its own ordered list of ciphers. The supported protocols are SSL version 2, SSL version 3, TLS version 1.0, TLS version 1.1, and TLS version 1.2.

z/OS On z/OS, there are only two lists of enabled ciphers, one for SSL version 2 and one for the other protocols. The supported protocols are SSL version 2, SSL version 3, and TLS version 1.0.

SSL Version 2 ciphers default to no ciphers, which means that the protocol is disabled. The other protocols default to a set of SSL ciphers that excludes null ciphers, export ciphers, and weak ciphers.

When you use the single-argument form of SSLCipherSpec, the given cipher is enabled in all protocols for which it is valid. The first time such a change is made for each protocol, the default ciphers for the protocol are discarded.

When you use the multiple-argument form of SSLCipherSpec, specifying the name of an SSL protocol (or "ALL") as the first argument, you can use an enhanced syntax with the following benefits:

- Multiple ciphers can be listed with each occurrence of SSLCipherSpec
- Individual ciphers can be removed from the current set of enabled ciphers by prefixing the cipher name with "-".
- The first time a given protocol cipher list is being modified, the given cipher can be added to the end of the defaults, instead of replacing them, by prefixing the cipher name with "+".

If you provide a protocol name of "ALL", then the adding or removing specified for each cipher name is applied to each protocol where that cipher is valid.

As a special case, to empty all the cipher lists with a single command, you can use SSLCipherSpec ALL NONE. Using this command is a good way to start a configuration anytime you do not want to use the default ciphers.

Name	Description
z/OS Syntax	SSLCipherSpec <i>short name</i> or SSLCipherSpec <i>long name</i>
Distributed operating systems Syntax	SSLCipherSpec [<i>protocol_name</i>] [+ -] <i>short name</i> <i>long name</i> [[+ -] <i>short name</i> <i>long name</i> ...]
Scope	Server config, virtual host.
Default	If nothing is specified, the server uses all non-NULL, non-export, non-weak cipher specifications available.
Module	mod_ibm_ssl
Multiple instances in the configuration file	Permitted. Order of preference is top to bottom, first to last.
Distributed operating systems Values for protocol name on distributed platforms	SSLv2, SSLv3, TLSv10, TLSv11, TLSv12, ALL
z/OS Values for protocol name on z/OS	SSLv2, SSLv3, TLSv10, ALL
Values for cipher names	See the SSL cipher specification topic for values
Example 1	<p>If you want to select just a few ciphers, it is best to start by resetting all the cipher lists, then adding the ones you want to use:</p> <pre># Delete all ciphers from the cipher lists SSLCipherSpec ALL NONE # Add a few specific ciphers SSLCipherSpec ALL +SSL_RSA_WITH_3DES_EDE_CBC_SHA SSLCipherSpec ALL +TLS_RSA_WITH_AES_256_CBC_SHA</pre>

Name
Example 2

Description

If you want to use most of the defaults, but there are one or two ciphers that you do not want, you can remove those from any lists that they are currently in:

```
# Delete some specific ciphers from the protocols  
where they are valid  
SSLCipherSpec ALL -SSL_RSA_WITH_RC4_128_MD5  
SSLCipherSpec ALL -SSL_RSA_WITH_RC4_128_SHA
```

SSLClientAuth directive

The SSLClientAuth directive sets the mode of client authentication to use (none (0), optional (1), or required (2)).

Name

Description

Syntax

SSLClientAuth <level required> [cr!]

Scope

Virtual host.

Default

SSLClientAuth none

Module

mod_ibm_ssl

Multiple instances in the configuration file

One instance per virtual host.

Name
Values

Description

- 0/None: No client certificate requested.
- 1/Optional: Client certificate requested, but not required.
- 2/Required: Valid client certificate required.
- Required_reset: The server requires a valid certificate from all clients, and if no certificate is available, the server sends an SSL alert to the client. This alert enables the client to understand that the SSL failure is client-certificate related, and causes browsers to re-prompt for client certificate information about subsequent access. This option requires GSKit version 7.0.4.19 or later, or z/OS V1R8 or later.
Important: No numeric option is provided so it will not look like the existing options.
- CRL: Turns certificate revocation list (CRL) on and off inside an SSL virtual host. If you use CRL, you must specify crl as a second argument for SSLClientAuth. For example: SSLClientAuth 2 crl. If you do not specify crl, you cannot perform CRL in an SSL virtual host.
- noverify: Enables SSL handshake to succeed and establish a connection, even if the certificate provided by the client fails validation (for example, the certificate is expired or revoked). Use this directive with SSLClientAuthVerify to provide a user-friendly web page, instead of the default browser error message. This option is only valid with Optional. Use SSLClientAuthVerify to fail requests received on the connection with the invalid client certificate.

If you specify the value 0/None, you cannot use the CRL option.

Required_reset

The server requires a valid certificate from all clients, and if no certificate is available, the server sends an SSL alert to the client. This enables the client to understand that the SSL failure is client-certificate related, and causes browsers to re-prompt for client certificate information about subsequent access. This option requires GSKit version 7.0.4.19 or later, or z/OS V1R8 or later.

Attention: In some cases IBM HTTP Server might not be able to determine the revocation status of a client certificate, because the backend server, which is the source of the revocation data, is not available. You should be aware that:

- A static CRL repository (SSLCRLHost) must be configured to enable checking of other URI forms in the CRLDistributionPoint fields.

- If your certificates use the LDAP or HTTP URI forms of the CertificateDistributionPoint or AIA extensions, be sure that the IBM HTTP Server system can establish outgoing connections of this type; you must adjust the settings for your firewall.
- **Distributed operating systems** The SSLUnknownRevocationStatus directive is provided for cases in which recoverable errors occur in IBM HTTP Server when it is communicating with the backend server, and the IBM HTTP Server cannot determine the revocation status of a certificate. The default behavior is to continue processing the handshake unless the backend server can successfully indicate that the certificate is revoked.
- **z/OS** Only an explicitly configured LDAP server can be queried for CRL, and the SSL handshake fails if the backend server is not reachable.

SSLClientAuthGroup directive

The SSLClientAuthGroup directive defines a named expression group that contains a set of specific client certificate attribute and value pairs. This named group can be used by the SSLClientAuthRequire directives. A certificate must be provided by the client, which passes this expression, before the server allows access to the protected resource.

Name	Description
Syntax	SSLClientAuthGroup <i>group name attribute expression</i>
Scope	Server config, virtual host.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	Permitted.
Override	None.
Values	Logical expression consisting of attribute checks linked with AND, OR, NOT, and parentheses. For example: SSLClientAuthGroup IBMUSpeople (Org = IBM) AND (Country = US)

The following section provides a description of examples with valid logical expressions. For example: SSLClientAuthGroup ((CommonName = "Fred Smith") OR (CommonName = "John Deere")) AND (Org = IBM) means that the object is not served, unless the client certificate contains a common name of either Fred Smith or John Deere and the organization is IBM. The only valid comparisons for the attribute checks, are equal and not equal (= and !=). You can link each attribute check with AND, OR, or NOT (also &&, ||, and !). Any comparisons that you link with AND, OR, or NOT must be contained within parentheses. If the value of the attribute contains a non-alphanumeric character, you must delimit the value with quotation marks.

This list contains attribute values that you can specify for this directive:

Table 9. Attribute values for the SSLClientAuthGroup directive. The table lists each attribute value as a long name and short name.

Long name	Short name
CommonName	CN
Country	C
Email	E

Table 9. Attribute values for the `SSLClientAuthGroup` directive (continued). The table lists each attribute value as a long name and short name.

Long name	Short name
IssuerCommonName	ICN
IssuerEmail	IE
IssuerLocality	IL
IssuerOrg	IO
IssuerOrgUnit	IOU
IssuerPostalCode	IPC
IssuerStateOrProvince	IST
Locality	L
Org	O
OrgUnit	OU
PostalCode	PC
StateOrProvince	ST

The long name or the short name can be used in this directive.

The user specifies a logical expression of specific client certificate attributes. You can logically use AND, OR, or NOT for multiple expressions if you must specify groupings of client certificate attribute values. Any comparisons that are linked with AND, OR, or NOT must be contained within parentheses. Valid operators include '=' and '!='. For example:

```
SSLClientAuthGroup IBMpeople Org = IBM)
```

or

```
SSLClientAuthGroup  
NotMNIBM (ST != MN) && (Org = IBM)
```

A group name cannot include spaces. See “`SSLClientAuthRequire` directive” for more information.

SSLClientAuthRequire directive

The `SSLClientAuthRequire` directive specifies attribute values, or groups of attribute values, that must be validated against a client certificate before the server allows access to the protected resource.

Name	Description
Syntax	<code>SSLClientAuthRequire attribute expression</code>
Scope	server config, virtual host
Default	None.
Module	<code>mod_ibm_ssl</code>
Multiple instances in the configuration file	Permitted. The function joins these directives by "AND".
Override	<code>AuthConfig</code>
Values	Logical expression consisting of attribute checks linked with AND, OR, NOT, and parentheses. For example: <code>SSLClientAuthRequire (group != IBMpeople) && (ST = M)</code>

If the certificate you received does not have a particular attribute, then there is no verification for an attribute match. Even if the specified matching value is " ", this might still not be the same as not having the attribute there at all. Any attribute specified on the SSLClientAuthRequire directive that is not available on the certificate causes the request to be rejected.

The list contains attribute values that you can specify for this directive:

Table 10. Attribute values for the SSLClientAuthRequire directive. The table lists each attribute value as a long name and short name.

Long name	Short name
CommonName	CN
Country	C
Email	E
IssuerCommonName	ICN
IssuerEmail	IE
IssuerLocality	IL
IssuerOrg	IO
IssuerOrgUnit	IOU
IssuerPostalCode	IPC
IssuerStateOrProvince	IST
Locality	L
Org	O
OrgUnit	OU
PostalCode	PC
StateOrProvince	ST

The long name or the short name can be used in this directive.

The user specifies a logical expression of specific client certificate attributes. You can logically use AND , OR, or NOT for multiple expressions if you must specify groupings of client certificate attribute values. Any comparisons that are linked with AND, OR, or NOT must be contained within parentheses. Valid operators include '=' and '!='. The user can also specify a group name, that is configured using the “SSLClientAuthGroup directive” on page 145, to configure a group of attributes.

You can specify multiple SSLClientAuthRequire directives within the same scope. The logical expressions for each directive are used to evaluate access rights for each certificate, and the results of the individual evaluations are logically ANDed together. For example:

```
SSLClientAuthRequire
((CommonName="John Doe") || (StateOrProvince=MN)) && (Org
!=IBM)
```

or

```
SSLClientAuthRequire
(group!=IBMpeople) && (ST=MN)
```

You can put quotes around the short and long names. For example:

```
SSLClientAuthRequire (group  
!= IBMpeople) && ("ST= MN")
```

See “SSLClientAuthGroup directive” on page 145 for more information.

SSLClientAuthVerify directive

The SSLClientAuthVerify directive controls whether IBM HTTP Server fails requests when a client certificate is received, but it fails validation (for example, it is expired or revoked).

Name	Description
Syntax	SSLClientAuthVerify statuscode OFF
Scope	Global server or virtual host.
Default	500
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per directory stanza.
Values	HTTP response status code, or OFF

Use this directive with SSLClientAuth Optional Noverify to provide a user friendly web page, instead of the default browser error message.

If you configure a virtual host with SSLClientAuth Optional Noverify, an SSL connection can be established when a client certificate is received, but it fails validation (for example, it is expired or revoked).

Use this directive in a context such as Location or Directory to fail requests that are received on that connection with a specific error code, or handled normally by setting OFF.

By providing a custom error document for that status, the administrator can control the page that is presented to the user, for example, to tell the user their certificate is invalid and provide further instructions.

If the error document is an internal redirect to another URL in the same virtual host, you must ensure that URL has SSLClientAuthVerify OFF in its context so it does not immediately fail, as well. An example of this scenario follows.

The specified status code must be a response status that is valid in HTTP and known to IBM HTTP Server. The values are between 100 and 599, and are typically defined in an RFC or standards proposal. If you are unsure, try a status code in a test configuration and use `apachectl -t` to see if it is valid. Other unused codes that are valid and would be good choices include: 418, 419, 420, and 421.

Because the client certificate was invalid, the error document will not have any of the environment variables available that would contain information about the client certificate. The cause of the client certificate validation failure is available in the `SSL_LAST_VALIDATION_ERROR` environment variable. The variable could be `GSKVAL_ERROR_REVOKED_CERT` or `GSKVAL_ERROR_CERT_EXPIRED`. If the certificate has multiple validation problems, the reported cause is often `GSKVAL_ERROR_CA_MISSING_CRITICAL_BASIC_CONSTRAINT`.

Each time a client certificate validation fails, two messages are logged in the error log at `loglevel Error`. The second message includes the cause, for example:

```
[Tue Jun 08 08:54:25 2010] [error] [client 9.37.243.128] [9e44c28] [731] SSL0208E: SSL Handshake Failed,
Certificate validation error. [9.37.243.128:60347 -> 9.37.243.67:443] [08:54:25.000223331]
[Tue Jun 08 08:54:25 2010] [error] [client 9.37.243.128] [9e44c28] [731] Certificate validation error
during handshake, last PKIX/RFC3280 certificate validation error was
GSKVAL_ERROR_CA_MISSING_CRITICAL_BASIC_CONSTRAINT
[9.37.243.128:60347 -> 9.37.243.67:443] [08:54:25.000223331]
```

Example configuration:

```
<VirtualHost *:443
SSLClientAuth Optional Noverify
<Location />
SSLClientAuthVerify 419
</Location>
ErrorDocument 419 /error419.html
<Location /error419.html>
SSLClientAuthVerify OFF
</Location>
</VirtualHost>
```

SSLCRLHostname directive

The SSLCRLHostname directive specifies the TCP/IP name or address of LDAP server where the Certificate Revocation List (CRL) database resides.

Name	Description
Syntax	<SSLCRLHostName <TCP/IP name or address>
Scope	Global server or virtual host.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	TCP/IP name or address of the LDAP Server

Use the SSLCRLHostname directive, along with SSLCRLPort, SSLCRLUserID, and SSLStashfile directives, for static configuration of an LDAP-based CRL repository. It is only necessary to use these directives to query the LDAP-based CRL repository if an explicit CRLDistributionPoint X.509v3 certificate extension is absent or the server specified in the extension is unresponsive (unavailable).

If a CRLDistributionPoint extension is present in the certificate and the server specified in the extension is responsive (available), then the LDAP server specified in the CRLDistributionPoint is queried anonymously, without using these directives.

SSLCRLPort directive

The SSLCRLPort directive specifies the port of the LDAP server where the Certificate Revocation List (CRL) database resides.

Name	Description
Syntax	SSLCRL<port>
Scope	Global server or virtual host.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	Port of LDAP server; default = 389.

Use the SSLCRLPort directive, along with SSLCRLUserID, SSLCRLHostname, and SSLStashfile directives, for static configuration of an LDAP-based CRL repository. It is only necessary to use these directives to query the LDAP-based CRL repository if an explicit CRLDistributionPoint X.509v3 certificate extension is absent or the server specified in the extension is unresponsive (unavailable).

If a CRLDistributionPoint extension is present in the certificate and the server specified in the extension is responsive (available), then the LDAP server specified in the CRLDistributionPoint is queried anonymously, without using these directives.

SSLCRLUserID directive

The SSLCRLUserID directive specifies the user ID to send to the LDAP server, where the Certificate Revocation List (CRL) database resides.

Name	Description
Syntax	SSLCRLUserID <[prompt] <userid>
Scope	Global server or virtual host.
Default	Defaults to anonymous if you do not specify a user ID.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	User ID of LDAP server. Use the prompt option to enable the HTTP server to prompt you for the password to access the LDAP server during start up.

Use the SSLCRLUserID directive, along with SSLCRLPort, SSLCRLHostname, and SSLStashfile directives, for static configuration of an LDAP-based CRL repository. It is only necessary to use these directives to query the LDAP-based CRL repository if an explicit CRLDistributionPoint X.509v3 certificate extension is absent or the server specified in the extension is unresponsive (unavailable).

If a CRLDistributionPoint extension is present in the certificate and the server specified in the extension is responsive (available), then the LDAP server specified in the CRLDistributionPoint is queried anonymously, without using these directives.

SSLDisable directive

The SSLDisable directive disables SSL for the virtual host.

Name	Description
Syntax	SSLDisable
Scope	Global server or virtual host.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	None.

SSLEnable directive

The SSLEnable directive enables SSL for the virtual host.

Attention: This directive should not be specified in the base server configuration if you do not want the directive automatically copied to a given virtual host configuration.

Name	Description
Syntax	SSLEnable
Scope	Global server or virtual host.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	None.

SSLFakeBasicAuth directive

The SSLFakeBasicAuth directive enables the fake basic authentication support.

This support enables the client certificate distinguished name to become the user portion of the user and password basic authentication pair. Use **password** for the password.

Attention: This directive might be overridden by the base server configuration.

Name	Description
Syntax	SSLFakeBasicAuth
Scope	Within a directory stanza, used along with AuthName, AuthType, and require directives.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per directory stanza.
Values	None.

Distributed operating systems

SSLFIPSDisable directive

The SSLFIPSDisable directive disables Federal Information Processing Standards (FIPS).

Name	Description
Syntax	SSLFIPSDisable
Scope	Virtual and global.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	None.

SSLFIPSEnable directive

The SSLFIPSEnable directive enables Federal Information Processing Standards (FIPS).

This directive is applicable to distributed platforms.

Note: **z/OS** This directive is supported on the z/OS platform with the following limitations.

- The directive is valid in the global scope only.
- If you change the value of the directive, you must stop and then start the IBM HTTP Server for the new value to take effect. The new value will not take effect if you do a restart.

Name	Description
Syntax	SSLFIPSEnable
Scope	Virtual and global.
Default	Disabled by default.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	None.

Attention: See the SSL cipher specification topic for values.

SSLInsecureRenegotiation directive

The SSLInsecureRenegotiation directive determines whether insecure (pre RFC5746) SSL renegotiation is permitted. SSL Renegotiation of any kind is not common, and this directive should not be changed from its default value of off.

Distributed operating systems

Attention: Prior to V8.0.0.1, the server has RFC5746 support and accepts secure renegotiation requests. In V8.0.0.1 and later, secure renegotiation requests are not accepted without first enabling the SSLRenegotiation directive.

When on is specified, insecure SSL renegotiation is permitted. When off is specified (the default), insecure SSL renegotiation is not permitted.

Name	Description
Syntax	SSLInsecureRenegotiation directive <i>on off</i>
Scope	Virtual hosts
Default	off
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server
Values	<i>on off</i>

Distributed operating systems

SSLPKCSDriver directive

The SSLPKCSDriver directive identifies the fully qualified name to the module, or driver used to access the PKCS11 device.

Name	Description
Syntax	<i>Fully qualified name to module used to access PKCS11 device</i> >. If the module exists in the user path, then specify just the name of the module.
Scope	Global server or virtual host.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	Path and name of PKCS11 module or driver.



The default locations of the modules for each PKCS11 device follow:

- nCipher
 - AIX: /opt/nfast/toolkits/pkcs11/libcknfast.so
 - HP: /opt/nfast/toolkits/pkcs11/libcknfast.sl
 - Solaris: /opt/nfast/toolkits/pkcs11/libcknfast.so
 - Windows: c:\nfast\toolkits\pkcs11\cknfast.dll
- IBM 4758
 - AIX: /usr/lib/pkcs11/PKCS11_API.so
 - Windows: \$PKCS11_HOME\bin\nt\cryptoki.dll
- IBM e-business Cryptographic Accelerator
 - AIX: /usr/lib/pkcs11/PKCS11_API.so

SSLProtocolDisable directive

The SSLProtocolDisable directive enables you to specify one or more SSL protocols which cannot be used by the client for a specific virtual host. This directive must be located in a <VirtualHost> container.

Supported protocols for a virtual host are supported separately. If all supported protocols are disabled, clients cannot complete an SSL handshake.

Name	Description
Syntax	SSLProtocolDisable <protocolname>
Scope	Virtual host
Default	Disabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	Multiple instances permitted per virtual host.
Values	The following possible values are available for this directive.
	SSLv2
	SSLv3
	TLS
	TLSv1
	 TLSv1.1
	 TLSv1.2

A value of TLS disables all TLS versions.

A value of TLSv1 disables TLS Version 1.0.

Distributed operating systems A value of TLSv1.1 disables TLS version 1.1.

Distributed operating systems A value of TLSv1.2 disables TLS version 1.2.

The following example disables support for multiple protocols on a virtual host.

```
<VirtualHost *:443> SSLEnable SSLProtocolDisable SSLv2  
SSLv3 (any other directives) </VirtualHost>
```

Attention: SSL0230I is logged for each SSL connection attempt if the client and server do not share at least one protocol and cipher combination.

SSLProtocolEnable directive

The SSLProtocolEnable directive can be used to enable individual SSL protocols.

Distributed operating systems On distributed platforms, this directive has limited usefulness because all useful protocols are automatically enabled by default.

z/OS On z/OS, this directive can be used after z/OS service adds support for TLSv1.1 and TLSv1.2. The TLSv1.1 and TLSv1.2 protocols are not enabled by default in an IBM HTTP Server running on z/OS.

Syntax	SSLSuiteBMode
Scope	Virtual host
Default	Unset
Module	mod_ibm_ssl
Multiple instances in the configuration file	Multiple instances permitted per virtual host.

SSLProxyEngine directive

The SSLProxyEngine toggles whether the server uses SSL for proxied connections. SSLProxyEngine *on* is required if your server is acting as a reverse proxy for an SSL resource.

Name	Description
Syntax	SSLProxyEngine <i>on off</i>
Scope	IP-based virtual hosts
Default	Off
Module	mod_ibm_ssl
Multiple instances in the configuration file	One per virtual host and global server
Values	<i>on off</i>

SSLRenegotiation directive

The SSLRenegotiation directive controls IBM HTTP Server support of Transport Layer Security (TLS) renegotiation. The directive controls the types of TLS renegotiation permitted by IBM HTTP Server. TLS renegotiation is how clients can initiate a new SSL handshake on an existing secure connection, which is rarely used by normal browser-based clients.

Name	Description
Syntax	SSLRenegotiation <i>on off LEGACY_AND_RFC5746</i>
Default	Off
Module	mod_ibm_ssl
Context	virtual host
Status	extension
Values	<i>on off LEGACY_AND_RFC5746</i>

OFF (default)

No renegotiation is permitted.

ON Secure renegotiation, as currently defined by RFC5746 is permitted.

LEGACY_AND_RFC5746

Both secure renegotiation and legacy insecure renegotiation are permitted.

Compatibility

- This directive supersedes the SSLInsecureRenegotiation directive in IBM HTTP Server 8.0 and later.
- IBM HTTP Server 8.0.0.0 defaulted to ON (accepting RFC5746 renegotiations).
- Prior to 7.0.0.21, the bundled GSKit security library was not aware of RFC5746, and "ON" referred to legacy insecure renegotiation.
- Support for the LEGACY_AND_RFC5746 option depends on IBM HTTP Server 7.0.0.21 and later.

SSLServerCert directive

The SSLServerCert directive sets the server certificate to use for this virtual host.

Name	Description
Syntax	SSLServerCert [token-name:]label [, [token-name]:label]
Scope	IP-based virtual hosts.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	Specify the label of the certificate to be used. If two labels are specified, associate one of the labels with an ECDSA signed certificate. Associate the other label with an RSA signed certificate.

Examples of the SSLServerCert directive are as follows:

```
SSLServerCert example.com
SSLServerCert myRSA, myECDSA
SSLServerCert swtoken:cert1
```

Important: If you use a PKCS11 accelerator, prefix the label with the name of the PKCS11 token and a colon ":" separator.

SSLSNIMap

The SSLSNIMap directive maps TLS Server Name Indication (SNI) hostnames to certificate labels.

Syntax	SSLSNIMap hostname cert-label
Scope	Virtual host
Default	Disabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	Multiple per virtual host
Values	Hostnames used by the client and certificate labels present in the configured KeyFile

The SSLSNIMap directive allows the server to respond with a different TLS certificate, based on the hostname the client requested. If name-based virtual hosts are used, SSLSNIMap should be present only in the first-listed virtual host for an ip:port combination (the default virtual host for a set of name-based virtual hosts).

SSLStashfile directive

The SSLStashfile directive indicates path to file with file name containing the encrypted password for opening the PKCS11 device.

Name	Description
Syntax	SSLStashFile /usr/HTTPServer/mystashfile.sth
Scope	Virtual host and global server.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	File name of an LDAP and/or PKCS11 stash file that is created with the sslstash command.

The SSLStashFile does not point to a stash file for the KeyFile in use, as that is calculated automatically based on the name of the KeyFile, and is a different type of stashfile.

Use the **sslstash** command, located in the bin directory of IBM HTTP Server, to create your CRL or cryptographic device stash file. The password you specify using the **sslstash** command should be the same as the password you use to log into your LDAP server or cryptographic hardware.

The stash file that the sslstash command creates is completely independent of the stash file that often accompanies a CMS KeyFile (*.kdb). Therefore, make sure that you:

- Do not overwrite an existing *.sth file when you issue the sslstash command.
- Never choose a filename for the output of the sslstash command that corresponds to the filename of a CMS KeyFile (*.kdb).

Usage: sslstash [-c] <directory_to_password_file_and_file_name>
<function_name> <password>

where:

- **-c**: Creates a new stash file. If not specified, an existing file updates.
- **File**: Represents the fully qualified name of the file to create, or update.

- **Function:** Indicates the function for which to use the password. Valid values include `crl`, or `crypto`.
- **Password:** Represents the password to stash.

Attention: See also “SSL certificate revocation list” on page 124.

Use the `SSLStashFile` directive, along with `SSLCRLPort`, `SSLCRLHostname`, and `SSLCRLUserID` directives, for static configuration of an LDAP-based CRL repository. It is only necessary to use these directives to query the LDAP-based CRL repository if an explicit `CRLDistributionPoint X.509v3` certificate extension is absent or the server specified in the extension is unresponsive (unavailable).

If a `CRLDistributionPoint` extension is present in the certificate and the server specified in the extension is responsive (available), then the LDAP server specified in the `CRLDistributionPoint` is queried anonymously, without using these directives.

SSLSuiteBMode

The `SSLSuiteBMode` directive can be used to configure the enclosing virtual host to use the Suite B profile for TLS.

The Suite B profile drastically reduces the available signature algorithm and cipher specifications that the server uses. The set of acceptable algorithms and ciphers is subject to change over time as relevant standards change. The 128 and 192 arguments refer to the two levels of security discussed in RFC 6460.

Specifying this directive overrides most previously specified SSL directives. The `SSLAttributeSet` setting is not overridden by this directive because it has a higher priority. All Suite B profiles require the certificate chain for the server to use strong ECC signatures. The RFC 6460 documents the restrictions of the Suite B profile.

Syntax	<code>SSLSuiteBMode</code>
Scope	Virtual host
Default	Unset
Module	<code>mod_ibm_ssl</code>
Multiple instances in the configuration file	Once per virtual host

SSLTrace directive

The `SSLTrace` directive enables debug logging in `mod_ibm_ssl`. It is used in conjunction with the `LogLevel` directive. To enable debug logging in `mod_ibm_ssl`, set `LogLevel` to `debug` and add the `SSLTrace` directive to global scope in the IBM HTTP Server configuration file, after the `LoadModule` directive for `mod_ibm_ssl`. This directive is typically used at the request of IBM support while investigating a suspected problem with `mod_ibm_ssl`. It is not recommended to enable this directive under normal working conditions.

Name	Description
Syntax	<code>SSLTrace</code>
Scope	Global
Default	<code>mod_ibm_ssl</code> debug logging in not enabled
Module	<code>mod_ibm_ssl</code>
Multiple instances in the configuration file	Ignored
Values	None

Attention: See also LogLevel Directive.

Distributed operating systems

SSLUnknownRevocationStatus

The SSLUnknownRevocationStatus directive specifies how IBM HTTP Server reacts when IBM HTTP Server cannot readily determine the revocation status, which is coming through CRL or OCSP.

Name	Description
Syntax	SSLUnknownRevocationStatus ignore log log_always deny
Scope	Virtual host
Default	ignore
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance permitted for each virtual host
Values	<p>ignore Specifies that a debug level message is issued when a handshake completes and the revocation status is not known. This message is not re-issued when the SSL session is resumed.</p> <p>log Specifies that a notice-level message is issued when a handshake completes and the revocation status is not known. This message is not re-issued when the SSL session is resumed.</p> <p>log_always Specifies that a notice-level message is issued when a handshake completes and the revocation status is not known. IBM HTTP Server issues the same message for subsequent handshakes.</p> <p>deny Specifies that a notice-level message is issued when a handshake completes, the revocation status is not known, the session is not resumable, and the HTTPS connection is immediately closed. IBM HTTP Server reports the same message for subsequent handshakes.</p>

config: Whenever a message is logged for UnknownRevocationStatus, the SSL_UNKNOWNREVOCAION_SUBJECT variable, an internal SSL environment variable, is set. You can log this variable with the following syntax:

```
%{SSL_UNKNOWNREVOCAION_SUBJECT}e
```

You could also use the variable in mod_rewrite expressions when the SSLUnknownRevocationStatus directive has any value other than deny. Use the following variable name:

```
%{ENV:SSL_UNKNOWNREVOCAION_SUBJECT}
```

SSLV2Timeout directive

The SSLV2Timeout directive sets the timeout for SSL Version 2 session IDs.

Name	Description
Syntax	SSLV2Timeout 60
Scope	Global base and virtual host.
Default	40
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	0 to 100 seconds.

SSLV3Timeout directive

The SSLV3Timeout directive sets the timeout for SSL Version 3 and TLS session IDs.

Name	Description
Syntax	SSLV3Timeout 1000
Scope	Global base and virtual host.
	Windows The virtual host scope or global scope are applicable.
	AIX HP-UX Linux
	Solaris The virtual host scope is applicable if the SSLCacheDisable directive is also being used. Otherwise, only the global scope is allowed.
Default	120
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per virtual host and global server.
Values	0 to 86400 seconds.

SSLVersion directive

The SSLVersion directive causes object access rejection with a 403 response if the client has connected with an SSL protocol version other than the one specified.

In most cases, the SSLProtocolDisable directive is a better choice than the SSLVersion directive for ensuring use of particular SSL protocol versions. The SSLProtocolDisable directive enables the client browser to negotiate another protocol version if possible whereas the SSLVersion directive causes IBM HTTP Server to send a 403 response, which might confuse the user.

Name	Description
Syntax	SSLVersion ALL
Scope	One per directory stanza.
Default	None.
Module	mod_ibm_ssl
Multiple instances in the configuration file	One instance per <Directory> or <Location> stanza.

Name	Description
Values	Distributed operating systems z/OS SSLV2 SSLV3 TLS TLSV1 TLSV11 TLSV12 SSLV2 SSLV3 TLS TLSV1 ALL

Setting advanced SSL options

Distributed operating systems z/OS

You can enable advanced security options such as: client authentication, setting and viewing cipher specifications, defining SSL for multiple-IP virtual hosts, and setting up a reverse proxy configuration with SSL.

About this task

After setting up secure connections, follow these instructions to enable advanced security options:

Procedure

1. Enable client authentication. If you enable client authentication, the server validates clients by checking for trusted certificate authority (CA) root certificates in the local key database.
2. Set and view cipher specifications.

Important: If you specify V3 or TLS ciphers and no SSL V2 ciphers, SSL V2 support is disabled. Also, if you specify SSL V2 ciphers and no SSL V3 or TLS ciphers, SSL V3 and TLS support is disabled.

3. Define Secure Sockets Layer (SSL) for multiple-IP virtual hosts.

Choosing the level of client authentication

Distributed operating systems z/OS

If you enable client authentication, the server validates clients by requesting a certificate from the client and verifying that is signed by a trusted certificate authority (CA) root certificate in the server key database.

About this task

For each virtual host, choose the level of client authentication:

Procedure

1. Specify one of the following values in the configuration file on the SSLClientAuth directive, for each virtual host stanza . A virtual host stanza represents a section of the configuration file that applies to one virtual host.

Table 11. Client authentication level. The table lists the value for the client authentication level and a description of the value

Value	Description
None	The server requests no client certificate from the client.
Optional	The server requests, but does not require, a client certificate. If presented, the client certificate must prove valid.

Table 11. Client authentication level (continued). The table lists the value for the client authentication level and a description of the value

Value	Description
Required	The server requires a valid certificate from all clients, returning a 403 status code if no certificate is present.
Required_reset	The server requires a valid certificate from all clients, and if no certificate is available, the server sends an SSL alert to the client. This enables the client to understand that the SSL failure is client-certificate related, and will cause browsers to re-prompt for client certificate information on subsequent access.

For example, `SSLClientAuth required`.

If you want to use a certificate revocation list (CRL), add `cr1`, as a second argument for `SSLClientAuth`. For example: `SSLClientAuth required cr1`.

2. Save the configuration file and restart the server.

Server Name Indication

Server Name Indication (SNI) support for IBM HTTP Server allows you to use certificate selection, based on the SNI extension that is sent by TLS clients. It does allow you to use other handshake-related settings from a name-based virtual host.

Definitions for SNI

- Each virtual host with a matching address-spec, such as `"*:443"`, forms a name-based virtual host group.
- The first listed virtual host in a name-based virtual host group is the default virtual host.

Requirements for SNI

- The default virtual must specify the SNI argument to the `SSLServerCert` directive.
- Only virtual hosts with a single address-spec (such as `"*:443"`) can participate in SNI.
- Non-default virtual hosts for a name-based virtual host must not contain directives from this module other than `SSLServerCert`.
- `"invalid"` is a reserved server name. Virtual hosts must not specify `"ServerName invalid"`.

Forms of SNI

There are two forms of SNI:

1. In the first form of SNI, only a single virtual host is used, and the `SSLSNIMap` directive is used to map between host names and certificate labels.

```
<virtualhost *:443>
  ServerName example.com
  SSLEnable SNI
  SSLServerCert default
  SSLSNIMap a.example.com sni1-rsa
  SSLSNIMap a.example.com sni1-ecc
  SSLSNIMap b.example.com sni2
</virtualhost>
```

2. In the second form of SNI, a series of virtual hosts are created, and the mapping from hostnames to certificate labels is via `ServerName`, `non-wildcard ServerAlias`, and `SSLServerCert`.

```
<virtualhost *:443>
  ServerName example.com
  SSLEnable SNI
</virtualhost>
<virtualhost *:443>
  ServerName a.example.com
  SSLEnable
  SSLServerCert sni1
</virtualhost>
<virtualhost *:443>
  ServerName b.example.com
  ServerAlias other.example.com
  SSLEnable
  SSLServerCert sni2
</virtualhost>
```

Related tasks:

“Securing with SSL communications” on page 118

This section provides information to help you set up Secure Sockets Layer (SSL), using the default `httpd.conf` configuration file.

Choosing the type of client authentication protection

Distributed operating systems z/OS

If you enable client authentication, the server validates clients by checking for trusted certificate authority (CA) root certificates in the local key database.

About this task

For each virtual host, choose the type of client authentication:

Procedure

1. Specify one of the following directives in the configuration file, for each virtual host stanza:
 - a. `SSLClientAuthRequire`. For example, **`SSLClientAuthRequire CommonName=Richard`**
 - b. `SSLFakeBasicAuth`. If you specify `SSLFakeBasicAuth`, verify that the `mod_ibm_ssl` module is displayed last in the module list.
2. Save the configuration file and restart the server.

Viewing cipher specifications

Distributed operating systems z/OS

This section describes viewing cipher specifications for secure transactions and for a specific HTTP request.

About this task

To see which cipher specifications the server uses for secure transactions or for a specific HTTP request, complete one of the following steps.

Procedure

1. **To see which cipher specifications the server uses for secure transactions.**
Specify `LogLevel info` in the configuration file to include informational messages in the error log using the `LogLevel` directive. The error log is specified by the `ErrorLog` directive in the `http` configuration file. The location is set by the `ErrorLog` directive, which can be configured. Review the error log for messages in this format: `TimeStamp info_message mod_ibm_ssl: Using Version 2/3 Cipher:longname|shortname`. The order that the cipher specifications are displayed in the error log from top to bottom represents the attempted order of the cipher specifications.
2. **To see which cipher specification was negotiated with a specific client for a specific request.** Change the `LogFormat` directive to include the cipher specification as part of the information logged for each request. The format string `%{HTTPS_CIPHER}e` will log the name of the cipher (for example, "TLS_RSA_WITH_AES_256_CBC_SHA"). Be sure that the `LogFormat` directive you change is for the format used on the `CustomLog` directive. Here is an example:

```
LogFormat "%h %l %u %t \"%r\" %>s %b %{HTTPS_CIPHER}e" common
CustomLog logs/access_log common
```

Check the access log to find the cipher used. The position of the cipher will depend on where the `%{HTTPS_CIPHER}e` format string was placed in the `LogFormat` directive. Following are some example `access_log` entries, using the previous example for the `LogFormat` directive:

```
9.48.108.152 - - [17/Feb/2005:15:37:39 -0500]
"GET / HTTP/1.1" 200 1507 SSL_RSA_WITH_RC4_128_SHA

9.48.108.152 - - [17/Feb/2005:15:37:40 -0500]
"GET /httpTech.view1.gif HTTP/1.1" 200 1814 SSL_RSA_WITH_RC4_128_SHA

9.48.108.152 - - [17/Feb/2005:15:37:40 -0500]
"GET /httpTech.masthead.gif HTTP/1.1" 200 11844 SSL_RSA_WITH_RC4_128_SHA

9.48.108.152 - - [17/Feb/2005:15:37:41 -0500]
"GET /httpTech.visit1.gif HTTP/1.1" 200 1457 SSL_RSA_WITH_RC4_128_SHA
```

For non-secure requests, "-" will be logged for the cipher specification. You can log other SSL environment variables in the same manner as `HTTPS_CIPHER`.

SSL cipher specifications

Distributed operating systems

z/OS

When an SSL connection is established, the client (web browser) and the web server negotiate the cipher to use for the connection. The web server has an ordered list of ciphers, and the first cipher in the list that is supported by the client is selected.

Introduction

View the list of current of SSL ciphers.

Attention: This list of ciphers could change as a result of updates to industry standards. You can determine the list of ciphers supported in a particular version of IBM HTTP Server by configuring it to load `mod_ibm_ssl` and running `bin/apachectl -t -f path/to/httpd.conf -DDUMP_SSL_CIPHERS`.

The SSLFIPSEnable directive enables Federal Information Processing Standards (FIPS). When the SSLFIPSEnable directive is enabled, the set of ciphers available is restricted as shown, and SSLv2 and SSLv3 are disabled.

gotcha:

- Ciphers containing "ECDHE" in their name are only available in 8.5.0.2/8.0.0.6 and later.
- Ciphers containing "ECDHE" in their name must be explicitly enabled and should be enabled via their "long name".
- Ciphers containing "ECDHE_RSA" in their name use a standard RSA certificate and can coexist with older RSA ciphers and clients.
- Ciphers containing "ECDHE_ECDSA" in their name requires an ECC (Elliptic Curve Cryptography) certificate/key to be created (with gskcapicmd if you are running on a distributed platform, or gskkyman if you are running on z/OS).
- On z/OS, several criteria must be met to use "ECDHE" ciphers:
 - TLSv1.2 must be explicitly enabled using the SSLProtocolEnable directive.
 - z/OS V1R13 with OA39422, or later, is required to use TLSv1.2 on z/OS.
 - ICSF must be available to use ECC or AES-GCM ciphers. See “RACF CSFSERV Resource Requirements” in the *z/OS Cryptographic Services System SSL Programming* for more information.

SSL and TLS ciphers

Attention: Note the following SSL and TLS cipher values:

- - = cipher that is not valid for the protocol
- d = cipher is enabled by default
- y = cipher is valid but not enabled by default

Attention: TLS v1.1 and v1.2 are not available on the z/OS operating system unless two conditions are met:

- z/OS V1R13 with OA39422, or later is required.
- You must update the IBM HTTP Server configuration to specify SSLProtocolEnable TLSv1.1 TLSv1.2.

Note: To improve security, IBM HTTP Server Version 8.0 disables weak SSL ciphers, export SSL ciphers, and the SSL Version 2 protocol by default. SSL Version 2, weak ciphers, and export ciphers are generally unsuitable for production SSL workloads on the internet and are flagged by security scanners. To enable ciphers, use the SSLCipherSpec directive.

Table 12. Medium and high strength TLS ciphers

Short name	Long name	Key size (bits)	FIPS	SSLV2	SSLV3	TLSv10	TLSv11	TLSv12
35	SSL_RSA_WITH_RC4_128_SHA	128	-	-	Y	Y	Y	Y
34	SSL_RSA_WITH_RC4_128_MD5	128	-	-	Y	Y	Y	-
9C	TLS_RSA_WITH_AES_128_GCM_SHA256	128	Y	-	-	-	-	d
9D	TLS_RSA_WITH_AES_256_GCM_SHA384	256	Y	-	-	-	-	d
3C	TLS_RSA_WITH_AES_128_CBC_SHA256	128	Y	-	-	-	-	d
3D	TLS_RSA_WITH_AES_256_CBC_SHA256	256	Y	-	-	-	-	d

Table 12. Medium and high strength TLS ciphers (continued)

Short name	Long name	Key size (bits)	FIPS	SSLV2	SSLV3	TLSv10	TLSv11	TLSv12
2F	TLS_RSA_WITH_AES_128_CBC_SHA	128	Y	-	d	d	d	d
35b	TLS_RSA_WITH_AES_256_CBC_SHA	256	Y	-	d	d	d	d
3A	SSL_RSA_WITH_3DES_EDE_CBC_SHA	168	Y	-	d	d	d	d
C007	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	128	Y	-	-	-	-	d*
C008	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	168	Y	-	-	-	-	d*
C009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	128	Y	-	-	-	-	d*
C00A	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	256	Y	-	-	-	-	d*
C010	TLS_ECDHE_RSA_WITH_NULL_SHA	0	Y	-	-	-	-	d*
C011	TLS_ECDHE_RSA_WITH_RC4_128_SHA	128	Y	-	-	-	-	d*
C012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	168	Y	-	-	-	-	d*
C013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	128	Y	-	-	-	-	d*
C014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	256	Y	-	-	-	-	d*
C023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	128	Y	-	-	-	-	d*
C024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	256	Y	-	-	-	-	d*
C027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	128	Y	-	-	-	-	d*
C028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	256	Y	-	-	-	-	d*
C02B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	128	Y	-	-	-	-	d*
C02C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	256	Y	-	-	-	-	d*
C02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	128	Y	-	-	-	-	d*
C030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	256	Y	-	-	-	-	d*

Note: ECDHE ciphers are enabled by default for TLSv1.2, except on z/OS platforms (denoted with d*).

Weaker ciphers, not enabled by default:

Table 13. Other TLS ciphers

Short name	Long name	Key size (bits)	FIPS	SSLV2	SSLV3	TLSv10	TLSv11	TLSv12
39	SSL_RSA_WITH_DES_CBC_SHA	56	-	-	y	y	y	-
33	SSL_RSA_EXPORT_WITH_RC4_40_MD5	40	-	-	y	y	-	-
36	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	40	-	-	y	y	-	-
62	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	56	-	-	y	y	-	-
64	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	56	-	-	y	y	-	-
32	SSL_RSA_WITH_NULL_SHA	0	-	-	y	y	y	y
31	SSL_RSA_WITH_NULL_MD5	0	-	-	y	y	y	-
3B	TLS_RSA_WITH_NULL_SHA256	0	Y	-	-	-	-	y
30	SSL_NULL_WITH_NULL_NULL	0	-	-	y	y	y	y
27	SSL_DES_192_EDE3_CBC_WITH_MD5	168	-	y	-	-	-	-

Table 13. Other TLS ciphers (continued)

Short name	Long name	Key size (bits)	FIPS	SSLV2	SSLV3	TLSv10	TLSv11	TLSv12
21	SSL_RC4_128_WITH_MD5	128	-	y	-	-	-	-
23	SSL_RC2_CBC_128_CBC_WITH_MD5	128	-	y	-	-	-	-
26	SSL_DES_64_CBC_WITH_MD5	56	-	y	-	-	-	-
24	SSL_RC2_CBC_128_CBC_EXPORT40_WITH_MD5	40	-	y	-	-	-	-
22	SSL_RC4_128_EXPORT40_WITH_MD5	40	-	y	-	-	-	-
FE	SSL_RSA_FIPS_WITH_DES_CBC_SHA	56	-	-	-	-	-	-
FF	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	168	-	-	-	-	-	-

Related tasks:

“Securing with SSL communications” on page 118

This section provides information to help you set up Secure Sockets Layer (SSL), using the default `httpd.conf` configuration file.

Defining SSL for multiple-IP virtual hosts

Distributed operating systems z/OS

You can define different Secure Sockets Layer (SSL) options for various virtual hosts, or multiple servers running on one machine. In the configuration file, define each SSL directive in the stanza for the virtual host to which the directive applies. When you do not define an SSL directive on a virtual host, the server uses the directive default.

About this task

The default disables SSL for each virtual host. To enable SSL:

Procedure

1. Specify the `SSLEnable` directive on the virtual host stanza in the configuration file, to enable SSL for a virtual host.
2. Specify a `Keyfile` directive and any SSL directives you want enabled for that particular virtual host. You can specify any directive, except the cache directives inside a virtual host.
3. Restart the server.

Setting up a reverse proxy configuration with SSL

Distributed operating systems z/OS

This topic describes how to set up a site to act as a reverse proxy for a resource that is hosted on a secure site.

About this task

The following steps describe how to set up a reverse proxy configuration for a company (for example, `www.example.com`) which wants to act as a reverse proxy for a resource that is hosted on a secure site (for example, `internal.example.com`).

Procedure

1. Configure `www.example.com` similar to the following example:

```
<VirtualHost *:80>
ServerName host1
SSLProxyEngine On
KeyFile "c:/program files/ibm http server/clientkey.kdb"
ProxyPass /ssl/password.html https://examplehost/password.html
</VirtualHost>
```

2. Configure `internal.example.com` similar to the following example:

```
<VirtualHost *:443>
SSLEnable
KeyFile "c:/program files/ibm http server/serverkey.kdb"
</VirtualHost>
```

Results

When a browser requests `http://www.example.com/ssl/password.html`, IBM HTTP Server makes a connection to `internal.ibm.com` using SSL. If `internal.example.com` requires a client certificate, IBM HTTP Server uses the default certificate of the `KeyFile` for which it is configured.

IBM HTTP Server certificate management

Distributed operating systems

z/OS

Before you can configure IBM HTTP Server to accept TLS (also known as SSL) connections, you must create a certificate for your web server. An SSL certificate authenticates your web servers identity to clients.

Background information and tools

The primary tool for creating certificates with IBM HTTP Server is **iKeyman**, a graphical pure Java key management tool.

z/OS

On z/OS operating systems, all certificate management is done with the native **gskkyman** certificate management tool.

Distributed operating systems

On Microsoft Windows, you can start iKeyman using the Start Menu. On other platforms, start the tool from the IBM HTTP Server `bin/` directory, like all IBM HTTP Server executable files.

Native and Java supplemental command-line certificate management tools are also provided in the IBM HTTP Server `bin/` directory as **gskcmd** (also known as **iKeycmd**) and **gskcapicmd** (also known as **gsk8capicmd**). Both share similar syntax and contain extensive embedded usage information.

Certificate limitations in IBM HTTP Server

- Only RSA certificates (keys) are supported with IBM HTTP Server. DSA and ECC certificates are not supported.
- Certificates with a key length of up to 4096 bits are supported at run time with IBM HTTP Server.
- Ikeyman and **gskcmd** (**ikeycmd**) support creating certificates of lengths up to 4096 bits. The **gskcapicmd** command supports creating certificates of lengths up to 4096 bits.

- Multiple key database files can be used with each instance of IBM HTTP Server, but only one, which can still contain multiple personal certificates, can be used per TLS-enabled virtual host.

Complete documentation for certificate management tools

- Complete documentation of **gskkyman** is available in the “Cryptographic Services PKI Services Guide and Reference” document (SA22-7693) in the z/OS Internet Library.
- Complete documentation for **iKeyman** and **gskcmd** (Ikeycmd) are available in the iKeyman v8 Users Guide.
- Complete documentation for **gskcapicmd** (gsk8capicmd), the native command-line certificate management tool, is available on the IBM HTTP Server library page.

System setup

- Unlike prior releases of IBM HTTP Server, do not move or modify the `java/jre/lib/ext/gskikm.jar` file.
- Optionally install the Unrestricted JCE policy files from DeveloperWorks to use unlimited strength cryptography in iKeyman and gskcmd (ikeycmd). This step is often required to manipulate PKCS12 keystores.

z/OS

Certificate management tasks

Detailed example scenarios for certificate management are documented in the complete documentation for iKeyman (distributed operating systems) and gskkyman (z/OS operating systems).

Distributed operating systems

Certificate management tasks

Detailed example scenarios for certificate management are documented in the complete documentation for iKeyman (distributed operating systems) and gskkyman (z/OS operating systems).

Distributed operating systems

See the following command-line examples of common tasks. You can view full usage syntax by entering the following commands with only the first two parameters, or you can refer to the comprehensive documentation for the command. The following table lists the operations that you can perform on CA certificates, the AdminTask object that you can use to perform that operation, and how to navigate to the certificate on the console:

Create a CMS keystore

When creating a keystore to be used with IBM HTTP Server, specify the option to stash the password to a file regardless of the tool used.

```
# Syntax: <ihsroot>/bin/gskcapicmd -keydb -create -db <database> -pw <password> -stash
<ihsroot>/bin/gskcapicmd -keydb -create -db /opt/IBM/HTTPServer/key.kdb -pw password -stash
```

Populate a keystore with a set of default trusted CA certificates

By default, new keystores contain no trusted CA certificates.

```
# The populate operation is supported with iKeyman and gskcmd (ikeycmd) only, not with gskcapicmd.
# Syntax: <ihsroot>/bin/gskcmd -cert -populate -db <database> -pw <password>
<ihsroot>/bin/gskcmd -cert -populate -db /opt/IBM/HTTPServer/key.kdb -pw password
```


Add additional CA certificates, if wanted (optional)

```
# Syntax: <ihsroot>/bin/gskcapiCmd -cert -add -db <database> -pw <password> -file <inputcert> -label <labelname>
<ihsroot>/bin/gskcapiCmd -cert -add -db /opt/IBM/HTTPServer/key.kdb -pw password -file cacert.cer -label "CA certificate from example.com"
```

Create a self-signed certificate for test purposes (optional)

```
#Syntax: <ihsroot>/bin/gskcapiCmd -cert -create -db <database> -pw <password> \
        -dn <distinguished name> -label <labelname> -size <size>
<ihsroot>/bin/gskcapiCmd -cert -create -db /opt/IBM/HTTPServer/key.kdb -pw password \
        -dn "cn=www.example.com" -label "example.com" -size 2048
```

Create a certificate request

Most of the fields and options are optional, including selecting a Signature Algorithm (this signature is used only by your certificate authority, not at runtime). You can also specify other host names for your web server.

```
# Syntax: <ihsroot>/bin/gskcapiCmd -certreq -create -db <database> -pw <password> \
        -dn <distinguished name> -label <labelname> -size <size> -file <outputfilename>
<ihsroot>/bin/gskcapiCmd -certreq -create -db /opt/IBM/HTTPServer/key.kdb -pw password \
        -dn "cn=www.example.com" -label www.example.com -size 2048 -file example.csr
```

Submit the certificate request to a trusted certificate authority

This task does not include using any local tools. Typically, the certificate request (example.csr) is sent in an email or uploaded to a trusted certificate authority.

Receive the issued certificate

Receiving a certificate associates a signed certificate from your CA with the private key (personal certificate) in your KDB file. A certificate can only be received into the KDB that generated the certificate request.

```
# Syntax: <ihsroot>/bin/gskcapiCmd -cert -receive -db <database> -pw <password> -file <inputcertificate>
<ihsroot>/bin/gskcapiCmd -cert -receive -db /opt/IBM/HTTPServer/key.kdb -pw password -file certificate.arm
```

List certificates in a keystore.

```
# Syntax <ihsroot>/bin/gskcapiCmd -cert -list -db <database> -pw <password>
<ihsroot>/bin/gskcapiCmd -cert -list -db /opt/IBM/HTTPServer/key.kdb -pw password
```

Import certificates from JKS or PKCS12 into a key file usable by IBM HTTP Server (optional)

Instead of creating a new private key (personal certificate), you can import an existing private key and certificate created by another tool into an existing key file.

```
# Syntax: <ihsroot>/bin/gskcapiCmd -cert -import -db <inputpk12file> -pw <pkcs12password> \
        -target <existingkdbfile> -target_pw <existingkdbpassword>
<ihsroot>/bin/gskcapiCmd -cert -import -db other.pk12 -pw pkcs12password \
        -target key.kdb -target_pw password
```

View certificate expiration data (optional)

The `-expiry` flag causes certificates that will be considered expired “numdays” in the future to be displayed. Use 0 to display already expired certificates, or large numbers to display all certificate expiration dates.

```
# Syntax:<ihsroot>/bin/gskcapiCmd -cert -list -db <database> -pw <password> -expiry <numdays>
<ihsroot>/bin/gskcapiCmd -cert -list -db key.kdb -password -expiry 365
```

Related information:



iKeyman v8 User Guide



GSKCapiCmd User Guide

IBM HTTP Server library page

Unrestricted JCE policy files

Managing keys with the IKEYMAN graphical interface (Distributed systems)

Distributed operating systems

This section describes topics on how to set up and use the key management utility (IKEYMAN) with IBM HTTP Server. Using the graphical user interface, rather than the command line interface, is recommended.

Linux

Before you begin

Ensure that the required `compat-libstdc++` package exists for your operating system architecture. For more information, see the installation and verification information for Linux packages.

About this task

Global Security Kit (GSKit) certificate management tools are installed in the `<ihsinst>/bin/` directory. These tools should only be run from the installation directory. Examples for the following commands should include the full directory path, such as `<ihsinst>/bin/gskcmd`.

- `gskver`
- `ikeyman`
- `gskcapicmd`
- `gskcmd`

For IKEYMAN, you can run the following command in the installation directory to generate debug information.

```
<ihsinst>/bin/ikeyman -x
```

To have a secure network connection, create a key for secure network communications and receive a certificate from a certificate authority (CA) that is designated as a trusted CA on your server.

Procedure

- Start the Key Management utility user interface. Use IKEYMAN to create key databases, public and private key pairs, and certificate requests.
- Work with key databases. You can use one key database for all your key pairs and certificates, or create multiple databases.
- Change the database password. When you create a new key database, you specify a key database password, which protects the private key. The private key is the only key that can sign documents or decrypt messages that are encrypted with the public key. Changing the key database password frequently is a good practice.
- Create a new key pair and certificate request. You find key pairs and certificate requests stored in a key database.
- Import and export your key into another database or to a PKCS12 file. PKCS12 is a standard for securely storing private keys and certificates.
- List certificate authorities within a key database.
- Display certificate expiration date your key database by viewing the certificate information with the IKEYMAN Key Management utility GUI or using the `gskcmd` command.

- If you act as your own CA, you can use IKEYMAN to create self-signed certificates.
- Receive a signed certificate from a certificate authority. If you act as your own CA for a private Web network, you have the option to use the server CA utility to generate and issue signed certificates to clients and servers in your private network.
- Display default keys and certificate authorities within a key database.
- Store a certificate from a certificate authority (CA) that is not a trusted CA.
- Store the encrypted database password in a stash file.
- Use IKEYMAN to create key databases, public and private key pairs, and certificate requests.
- If you act as your own CA, you can use IKEYMAN to create self-signed certificates.
- If you act as your own CA for a private Web network, you have the option to use the server CA utility to generate and issue signed certificates to clients and servers in your private network.

What to do next

You may experience a certificate problem when you open a certificate that has a key with a higher level of cryptography than your policy files permit. You can optionally install unlimited strength JCE policy files.

- Download and install the files from the following Web site.
<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk>

For more information about the IKEYMAN utility, see the IKEYMAN User's Guide on the IHS Library page.

Starting the Key Management utility user interface

Distributed operating systems

This section describes how to start the Key Management (IKEYMAN) utility.

Procedure

- From a command line:

```
<install_root>/bin/ikeyman
```

or change to the <install_root>/bin directory and type `ikeyman`
- On Windows operating systems: Click **Start > Programs > IBM HTTP Server > Start Key Management Utility**. If you start IKEYMAN to create a new key database file, the utility stores the file in the directory where you start IKEYMAN.

Working with key databases

Distributed operating systems

This article describes how to create a new key database and open an existing key database.

About this task

A *key database* is a file that the server uses to store one or more key pairs and certificates. You can use one key database for all your key pairs and certificates, or create multiple databases.

Procedure

- Create a new key database as follows:
 1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
 2. Click **key database file** from the main user interface, then click **New**. Select **CMS** for the Key database type. IBM HTTP Server does not support database types other than CMS.
 3. Enter your password in the Password Prompt dialog box, and confirm the password. Select **Stash the password to a file**. Click **OK**. The new key database should display in the IKEYMAN utility with default signer certificates. Ensure that there is a functional, non-expiring signer certificate for each of your personal certificates. .
- Open an existing key database as follows:
 1. Start the IKEYMAN user interface.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. In the Open dialog box, enter your key database name, or click the `key.kdb` file, if you use the default. Click **OK**.
 4. Enter your correct password in the Password Prompt dialog box, and click **OK**.
 5. The key database name is displayed in the File Name text box.

What to do next

You can add a default list of signer certificates to your new database using the following instructions. The version of iKeyman that is provided by the bundled Java Runtime Environment (JRE) does not add a default list of signer certificates to newly-created key databases. Add default signer certificates in iKeyman, as follows:

1. Select **Signer Certificates** from the drop-down menu in the iKeyman window.
2. Click **Populate**.
3. Click the grey boxes next to the certificate authority names (Entrust, RSA Data Security, Thawte, Verisign) so they display as checked.
4. Click **OK**.

Changing the database password

Distributed operating systems

When you create a new key database, you specify a key database password, which protects the private key. The private key is the only key that can sign documents or decrypt messages that are encrypted with the public key. Changing the key database password frequently is a good practice.

About this task

Complete the following steps to change the database password:

Procedure

1. Start the IKEYMAN user interface.
2. Click **Key Database File** from the main UI, then click **Open**.
3. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if you use the default. Click **OK**.
4. Enter your password in the Password Prompt dialog box, and click **OK**.
5. Click **Key Database File** from the main UI, then click **Change Password**.
6. Enter a new password in the Password Prompt dialog box, and a new confirming password. Click **OK**.

Use the following guidelines when specifying the password:

- The password must come from the U.S. English character set.
- The password must contain at least six characters and contain at least two nonconsecutive numbers. Make sure that the password does not consist of publicly obtainable information about you, such as the initials and birth date for you, your spouse, or children.
- Stash the password or enable secure sockets layer (SSL) password prompting.

Keep track of expiration dates for the password. If the password expires, a message writes to the error log. The server starts, but a secure network connection does not exist, if the password has expired.

Creating a new key pair and certificate request

Distributed operating systems

You find key pairs and certificate requests stored in a key database. This section provides information on how to create a key pair and certificate request.

Before you begin

There are GSKit certificate support limitations that you should remember as you create a new key pair and certificate request:

- You cannot use IKEYMAN to create certificates with key sizes that are larger than 4096 bits.
- You can import certificates with key sizes up to 4096 bits into the key database.

About this task

To create a public and private key pair and certificate request, complete the following steps:

Procedure

1. If you have not created the key database, see [Creating a new key database](#) for instructions.
2. Start the IKEYMAN user interface.
3. Click **Key Database File** from the main user interface, then click **Open**.
4. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if you use the default. Click **OK**.
5. In the Password Prompt dialog box, enter your correct password and click **OK**.
6. Click **Create** from the main user interface, then click **New Certificate Request**.

7. In the New Key and Certificate Request dialog box, complete the following information:
 - Key label: Enter a descriptive comment to identify the key and certificate in the database.
 - Key size: Choose your level of encryptions from the drop-down menu.
 - Organization Name: Enter your organization name.
 - Organization Unit
 - Locality
 - State/Province
 - Zip code
 - Country: Enter a country code. Specify at least two characters. Example: US Certificate request file name, or use the default name.

A checksum of the certificate request is cryptographically signed with the new private key, and contains a copy of the new public key. The public key can then be used by a certificate authority to validate that the certificate signing request (CSR) has not been tampered with. Some certificate authorities might require that the checksum that is signed by the public key be calculated with a stronger algorithm such as SHA-1 or SHA-2 (SHA-256, SHA-384, SHA-512).

This checksum is the "Signature Algorithm" of the CSR

Subject Alternate Name (SAN) extensions are fields in a certificate request that inform SSL Clients of alternate hostnames that correspond to the signed certificate. Normal certificates (issued without a wildcard string in their Distinguished Name) are only valid for a single hostname. For example, a certificate created for example.com is not valid on www.example.com unless a Subject Alternate Name of "www.example.com" is added to the certificate. A certificate authority may charge an additional fee if your certificate contains 1 or more SAN extensions.

8. Click **OK**.
9. Click **OK** in the Information dialog box. A reminder to send the file to a certificate authority displays.
10. Optional: On UNIX-based platforms, remove the end of line characters (^M) from the certificate request. To remove the end of line characters, type the following command:

```
cat certreq.arm |tr -d "\r" > new_certreq.arm
```
11. Send the file to the certificate authority (CA) following the instructions from the CA Web site for requesting a new certificate.

Importing and exporting keys

Distributed operating systems

This article describes how to import and export your key into another database or to a PKCS12 file. PKCS12 is a standard for securely storing private keys and certificates.

About this task

To import and export keys from another database, complete the following steps:

Procedure

- Import keys from another database by completing the following steps:

1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. Enter your key database name in the Password prompt dialog box, or click **key.kdb** if you are using the default.
 4. Enter your correct password in the Password prompt dialog box, and click **OK**.
 5. Click **Personal Certificates** in the Key Database content frame, then click **Export/Import** on the label.
 6. In the Export/Import Key window:
 - a. Click **Import Key**.
 - b. Click the target database type.
 - c. Enter the file name, or use the Browse option.
 - d. Enter the current location.
 7. Click **OK**.
 8. Click **OK** in the Password prompt dialog box, to import the selected key to another key database.
- Import keys to a PKCS12 file by completing the following steps:
 1. Enter `ikeyman` on a command line on the Linux or UNIX platforms, or start the Key Management utility in the IBM HTTP Server folder on the Windows operating system.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click **key.kdb**, if you use the default. Click **OK**.
 4. Enter your password in the Password prompt dialog box, and click **OK**.
 5. Click **Personal Certificates** in the Key Database content frame, then click **Export/Import** on the label.
 6. In the Export/Import Key window:
 - a. Click **Import Key**.
 - b. Click the PKCS12 database file type.
 - c. Enter the file name, or use the Browse option.
 - d. Enter the correct location.
 7. Click **OK**.
 8. Enter the correct password in the Password prompt dialog box, then click **OK**.
 - Export keys from another database by completing the following steps:
 1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
 2. Click **key database file** from the main user interface, then click **Open**.
 3. Enter your key database name in the Password Prompt dialog box, or click **key.kdb** if you are using the default.
 4. Enter your correct password in the Password Prompt dialog box, and click **OK**.
 5. Click **Personal Certificates** in the Key database content frame, then click **Export/Import** on the label.
 6. In the Export/Import Key window:
 - a. Click **Export Key**.
 - b. Click the target database type.

- c. Enter the file name, or use the Browse option.
- d. Enter the current location.
- Export keys to a PKCS12 file by completing the following steps:
 1. Enter `ikeyman` on a command line on the Linux or UNIX platforms, or start the Key Management utility in the IBM HTTP Server folder on the Windows operating system.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click `key.kdb` if you use the default. Click **OK**.
 4. Enter your password in the Password Prompt dialog box, and click **OK**.
 5. Click **Personal Certificates** in the Key Database content frame, then click **Export/Import** on the label.
 6. In the Export/Import Key window:
 - a. Click **ExportKeyM**.
 - b. Click the PKCS12 database file type.
 - c. Enter the file name, or use the Browse option.
 - d. Enter the correct location.
 7. Click **OK**.
 8. Enter the correct password in the Password prompt dialog box, and enter the password again to confirm. Click **OK** to export the selected key to a PKCS12 file.

Listing certificate authorities

Distributed operating systems

You can display a list of trusted certificate authorities within a key database.

About this task

A trusted certificate authority issues and manages public keys for data encryption. A key database is used to share public keys that are used for secure connections.

To display a list of trusted certificate authorities (CAs) in a key database, complete the following steps:

Procedure

1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
2. Click **Key Database File** from the main UI, then click **Open**.
3. Enter your key database name in the Open dialog box, or click `key.kdb` if you are using the default.
4. Enter your correct password in the Password prompt dialog box, and click **OK**.
5. Click **Signer Certificates** in the Key database content frame.
6. Click **Signer Certificates**, **Personal Certificates**, or **Certificate Requests**, to view the list of CAs in the Key Information window.

What to do next

When the `<ihsinst>/java/jre/lib/ext/gskikm.jar` file has not been removed, the version of iKeyman that is provided by the bundled Java Runtime Environment

(JRE) does not add a default list of signer certificates to newly-created key databases. Add default signer certificates in iKeyman, as follows:

1. Select **Signer Certificates** from the drop-down menu in the iKeyman window.
2. Click **Populate**.
3. Click the grey boxes next to the certificate authority names (Entrust, RSA Data Security, Thawte, Verisign) so they display as checked.
4. Click **OK**.

Certificate expiration dates

Distributed operating systems

You can display expiration dates of certificates in your key database by viewing the certificate information with the IKEYMAN Key Management utility GUI or using the `gskcmd` command.

The following is an example of how to use the `gskcmd` command to display the validity dates on all certificates in the `key.kdb` certificate key file that will expire within 1825 days (5 years):

```
<ihsinst>/bin/gskcmd -cert -list all -expiry 1825 -db key.kdb -pw <password>
Certificates in database: key.kdb
VeriSign Class 1 CA Individual Subscriber-Persona Not Validated
Validity
Not Before: Mon May 11 20:00:00 EDT 1998
Not After: Mon May 12 19:59:59 EDT 2008
```

where `<password>` is the password you specified when creating the `key.kdb` key database file.

Creating a self-signed certificate

Distributed operating systems

It usually takes two to three weeks to get a certificate from a well known certificate authority (CA). While waiting for a certificate to be issued, use IKEYMAN to create a self-signed server certificate to enable SSL sessions between clients and the server. Use this procedure if you act as your own CA for a private Web network.

About this task

Complete the following steps to create a self-signed certificate:

Procedure

1. If you have not created the key database, see [Creating a new key database for instructions](#).
2. Start the IKEYMAN user interface.
3. Click **Key Database File** from the main UI, and then click **Open**.
4. Enter your key database name in the Open dialog box, or click the `key.kdb` file, if you use the default. Click **OK**.
5. In the Password Prompt dialog box, enter your correct password and click **OK**.
6. Click **Personal Certificates** in the Key Database content frame, and click the **New Self-Signed** radio button.
7. Enter the following information in the Password Prompt dialog box:

- **Key label:** Enter a descriptive comment to identify the key and certificate in the database.
- **Key size:** Choose your level of encryptions from the drop-down menu.
- **Common Name:** Enter the fully qualified host name of the Web server as the common name. Example: `www.myserver.com`.
- **Organization Name:** Enter your organization name.
- **Optional:** Organization Unit
- **Optional:** Locality
- **Optional:** State/Province
- **Optional:** Zip code
- **Country:** Enter a country code. Specify at least two characters. Example: US
- **Validity Period**

A checksum of the certificate request is cryptographically signed with the new private key, and contains a copy of the new public key. The public key can then be used by a certificate authority to validate that the certificate signing request (CSR) has not been tampered with. Some certificate authorities might require that the checksum that is signed by the public key be calculated with a stronger algorithm such as SHA-1 or SHA-2 (SHA-256, SHA-384, SHA-512).

This checksum is the "Signature Algorithm" of the CSR.

Note: IBM HTTP Server 8.0 ships IKEYMAN version 8.x. When using IKEYMAN version 8.x to create a certificate request, the user is asked to select a signature algorithm from a pull-down list.

Subject Alternate Name (SAN) extensions are fields in a certificate request that inform SSL Clients of alternate hostnames that correspond to the signed certificate. Normal certificates (issued without a wildcard string in their Distinguished Name) are only valid for a single hostname. For example, a certificate created for `example.com` is not valid on `www.example.com` unless a Subject Alternate Name of "`www.example.com`" is added to the certificate. A certificate authority may charge an additional fee if your certificate contains 1 or more SAN extensions.

8. Click OK.

Receiving a signed certificate from a certificate authority

Distributed operating systems

This topic describes how to receive an electronically mailed certificate from a certificate authority (CA), that is designated as a trusted CA on your server. A certificate authority is a trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs.

About this task

The certificate authority can send more than one certificate. In addition to the certificate for your server, the CA can also send additional signing certificates or intermediate CA certificates. For example, Verisign includes an intermediate CA certificate when sending a Global Server ID certificate. Before receiving the server certificate, receive any additional intermediate CA certificates. Follow the instructions in the Storing a CA certificate topic to receive intermediate CA certificates.

Receive the CA-signed certificate into a key database as follows:

Procedure

1. Start the IKEYMAN user interface.
2. Click **Key Database File** from the main UI, then click **Open**.
3. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if you use the default. Click **OK**.
4. Enter your correct password in the Password Prompt dialog box, then click **OK**.
5. Click **Personal Certificates** in the Key database content frame, then click **Receive**.
6. Enter the name of a valid Base64-encoded file in the Certificate file name text field in the Receive certificate from a file dialog box. Click **OK**.

Displaying default keys and certificate authorities

Distributed operating systems

This section describes how to view trusted certificate authorities and display default keys within a key database.

About this task

A trusted certificate authority (CA) issues and manages public keys for data encryption. A key database is used to share public keys that are used for secure connections. The tasks that follow show how to view the certificate authorities that are in your database, along with their expiration dates.

Procedure

- Display the default key entry as follows:
 1. Start the IKEYMAN user interface.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if using the default. Click **OK**.
 4. Enter your password in the Password Prompt dialog box, then click **OK**.
 5. Click **Personal Certificates** in the Key Database content frame, and click the **CA certificate** label name.
 6. Click **View/Edit** and view the certificate default key information in the Key Information window.
- Display a list of trusted certificate authorities (CAs) in a key database as follows:
 1. Start the IKEYMAN user interface.
 2. Click **Key Database File** from the main UI, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click the **key.kdb** file if you are using the default.
 4. Enter your correct password in the Password prompt dialog box, and click **OK**.
 5. Click **Signer Certificates** in the Key database content frame.
 6. Click **Signer Certificates**, **Personal Certificates**, or **Certificate Requests**, to view the list of CAs in the Key Information window.

What to do next

The version of iKeyman that is provided by the bundled Java Runtime Environment (JRE) does not add a default list of signer certificates to newly-created key databases. Add default signer certificates in iKeyman, as follows:

1. Select **Signer Certificates** from the drop-down menu in the iKeyman window.
2. Click **Populate**.
3. Click the grey boxes next to the certificate authority names (Entrust, RSA Data Security, Thawte, Verisign) so they display as checked.
4. Click **OK**.

Storing a certificate authority certificate

Distributed operating systems

This topic describes how to store a certificate from a certificate authority (CA) that is not a trusted CA.

About this task

Store a certificate from a certificate authority (CA) who is not a trusted CA as follows:

Procedure

1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
2. Click **Key Database File** from the main user interface, then click **Open**.
3. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if using the default. Click **OK**.
4. Enter your password in the Password Prompt dialog box, then click **OK**.
5. Click **Signer Certificates** in the Key Database content frame, then click **Add**.
6. In the Add CA Certificate from a File dialog box, click the **Base64-encoded ASCII data certificate file name**, or use the Browse option. Click **OK**.
7. In the Label dialog box, enter a label name and click **OK**.

Storing the encrypted database password in a stash file

Distributed operating systems

This section describes how you would store your database password in a stash file.

About this task

For a secure network connection, you can store the encrypted database password in a stash file.

Important: These stash files should be treated as highly sensitive information.

Procedure

- Store the password while a database creates as follows:
 1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.

2. Click **Key Database File** from the main user interface, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click the **key.kdbfile**, if using the default. Click **OK**.
 4. Enter your password in the Password Prompt dialog box, then enter again to confirm your password.
 5. Select the stash box and click **OK**.
 6. Click **Key Database File > Stash Password**.
 7. Click **OK** in the information dialog box.
- Store the password after creating a database as follows:
 1. Start the IKEYMAN user interface. Refer to Starting the Key Management utility for platform-specific instructions.
 2. Click **Key Database File** from the main user interface, then click **Open**.
 3. Enter your key database name in the Open dialog box, or click the **key.kdb** file, if you use the default. Click **OK**.
 4. Enter your correct password in the Password Prompt dialog box, and click **OK**.
 5. Click **Key Database File**, then click **Stash Password**.
 6. Click **OK** in the Information dialog box.

Managing keys with the command line (Distributed systems)

Distributed operating systems

The Java command line interface to IKEYMAN, `gskcapicmd`, provides the necessary options to create and manage keys, certificates and certificate requests. The native utility `/bin/gskcapicmd` is always preferred over `/bin/gskcmd`. `gskcapicmd` is faster and some features are added to `gskcapicmd` before `gskcmd`.

About this task

Global Security Kit (GSKit) certificate management tools are installed in the `<ihst>/bin/` directory. These tools should only be run from the installation directory. Examples for the following commands should include the full directory path, such as `<ihst>/bin/gskcapicmd`.

- **Windows** `gskver.bat`, `ikeyman.bat`, `gskcmd.bat`, `gskcmd`, and `gskcapicmd`.
- **AIX** **Linux** **Solaris** **HP-UX** `gskver`, `ikeyman`, and `gskcmd`.

To have a secure network connection, create a key for secure network communications and receive a certificate from a certificate authority (CA) that is designated as a trusted CA on your server. Use `gskcapicmd`, the utility command line interface, for configuration tasks that are related to public and private key creation and management.

The `gskcapicmd` user interface uses Java and native command line invocation, enabling IKEYMAN task scripting.

You cannot use `gskcapicmd` for configuration options that update the server configuration file, `httpd.conf`. For options that update the server configuration file, use the IBM HTTP Server administration server.

Procedure

- Use `gskcapicmd` to create key databases, public and private key pairs, and certificate requests using the command-line interface.
- If you act as your own certificate authority (CA), you can use `gskcapicmd` to create self-signed certificates.
- If you act as your own CA for a private Web network, you have the option to use the server CA utility to generate and issue signed certificates to clients and servers in your private network.
- Manage the database password using the command line.
- Create a public and private key pair and certificate request using the `gskcapicmd` command-line interface or `GSKCapiCmd`.
- Import and export keys using the command line. If you want to reuse an existing key from another database, you can import that key. Conversely, you can export your key into another database or to a PKCS12 file. PKCS12 is a standard for securely storing private keys and certificates. You can use the `gskcapicmd` command-line interface or `GSKCapiCmd` tool.
- Display default keys and certificate authorities within a key database.
- Store a certificate authority certificate from a certificate authority (CA) that is not a trusted CA.
- Store the encrypted database password in a stash file.
- Use `gskcapicmd` to create key databases, public and private key pairs, and certificate requests.
- If you act as your own certificate authority (CA), you can use `gskcapicmd` to create self-signed certificates.
- If you act as your own CA for a private Web network, you have the option to use the server CA utility to generate and issue signed certificates to clients and servers in your private network.

What to do next

For more information about the `gskcapicmd` command line interface, see the `GSKCapiCmd` User's Guide on the WebSphere Application Server Library page. For more information about the `gskcmd` (`ikeycmd`) command, see the IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 6.0 iKeyman 8.0 User's Guide .

Using the `gskcapicmd` command

Distributed operating systems

The `gskcapicmd` command provides a command line interface for certificate management tasks that might otherwise be provided by the `ikeyman` command. (The `gskcmd` command is a Java-based alternative.)

Procedure

1. You can invoke the `gskcapicmd` from the `<ihst>/bin/` directory.

- `Windows` `gskcapicmd.bat`
- `AIX` `Linux` `HP-UX` `Solaris` `gskcapicmd`

2. Perform the certificate management tasks that you want to complete.

Key Management Utility command-line interface (gskcmd) syntax

Distributed operating systems

This topic contains a description of the syntax that you can use with the gskcmd command.

Syntax

For more information, see “Using the gskcapicmd command” on page 182.

The syntax follows.

```
gskcmd <object> <action> [options]
```

Where:

- The object includes one of the following:
 - -keydb: Actions taken on the key database (either a CMS key database file, a WebDB key ring file, or SSLight class)
 - -cert: Actions taken on a certificate
 - -certreq: Actions taken on a certificate request
 - -help: Displays help for the gskcmd invocations
 - -version: Displays version information for gskcmd

The action represents the specific action to take on the object, and options represents the options, both required and optional, specified for the object and action pair.

The object and action keywords are positional and you must specify them in the selected order. However, options are not positional and you can specify them in any order, as an option and operand pair.

Table 14. Actions for gskcmd command objects. The table describes each action possible on a specified object that you can use with the gskcmd command.

Object	Actions	Description
-keydb	-changepw	Change the password for a key database
	-convert	Convert a key database from one format to another
	-create	Create a key database
	-delete	Delete the key database
	-stashpw	Stash the password of a key database into a file
-cert	-add	Add a CA certificate from a file into a key database
	-create	Create a self-signed certificate
	-delete	Delete a CA certificate

Table 14. Actions for gskcmd command objects (continued). The table describes each action possible on a specified object that you can use with the gskcmd command.

Object	Actions	Description
	-export	Export a personal certificate and its associated private key from a key database into a PKCS#12 file, or to another key database
	-extract	Extract a certificate from a key database
	-getdefault	Get the default personal certificate
	-import	Import a certificate from a key database or PKCS#12 file
	-list	List all certificates
	-modify	Modify a certificate. (Currently the only field you can modify is the Certificate trust field)
	-receive	Receive a certificate from a file into a key database
	-setdefault	Set the default personal certificate
	-sign	Sign a certificate stored in a file with a certificate stored in a key database and store the resulting signed certificate in a file
-certreq	-create	Create a certificate request
	-delete	Delete a certificate request from a certificate request database
	-details	List the detailed information of a specific certificate request
	-extract	Extract a certificate request from a certificate request database into a file
	-list	List all certificate requests in the certificate request database
	-recreate	Recreate a certificate request
-help		Display help information for the gskcmd command
-version		Display gskcmd version information

The following table describes the options that you can use with the gskcmd command.

Option	Description
dB	Fully qualified path name of a key database
-default_cert	Sets a certificate to use as the default certificate for client authentication (yes or no). Default is no.
-dn	X.500 distinguished name. Input as a quoted string of the following format (only CN, O, and C are required): "CN=Jane Doe,O=IBM,OU=Java Development,L=Endicott,ST=NY,ZIP=13760,C=country"
encryption	Strength of encryption used in certificate export command (strong or weak). Default is strong.
-expire	Expiration time of either a certificate or a database password (in days).
-file	File name of a certificate or certificate request (depending on specified object).
-format	Format of a certificate (either ASCII for Base64_encoded ASCII or binary for Binary DER data). Default is ASCII.
-label	Label attached to a certificate or certificate request
-new_format	New format of key database
-new_pw	New database password
-old_format	Old format of key database
-pw	Password for the key database or PKCS#12 file. See Creating a new key database.
-size	Key size (512, 1024, or 2048). Default is 1024. The 2048 key size is available if you are using Global Security Kit (GSKit) Version 7.0.4.14 and later.
-stash	Indicator to stash the key database password to a file. If specified, the password will be stashed in a file.
-target	Destination file or database
-target_pw	Password for the key database if -target specifies a key database. See Creating a new key database.
-target_type	Type of database specified by -target operand (see -type)
-trust	Trust status of a CA certificate (enable or disable). Default is enable.
-type	Type of database. Allowable values are CMS (indicates a CMS key database), webdb (indicates a keyring), sslight (indicates an SSLight .class), or pkcs12 (indicates a PKCS#12 file).
-x509version	Version of X.509 certificate to create (1, 2 or 3). Default is 3.

Related tasks:

“Managing keys with the command line (Distributed systems)” on page 181
The Java command line interface to IKEYMAN, `gskcapiCmd`, provides the necessary options to create and manage keys, certificates and certificate requests. The native utility `/bin/gskcapiCmd` is always preferred over `/bin/gskcmd`. `gskcapiCmd` is faster and some features are added to `gskcapiCmd` before `gskcmd`

Creating a new key database using the command-line interface

Distributed operating systems

A *key database* is a file that the server uses to store one or more key pairs and certificates. You can use one key database for all your key pairs and certificates, or create multiple databases.

About this task

You can create multiple databases if you prefer to keep certificates in separate databases.

Procedure

- Create a new key database using the `gskcmd` command-line interface by entering the following command (as one line):

```
<ihsinst>/bin/gskcmd -keydb -create -db <filename> -pw <password> -type  
<cms | jks | jceks | pks12> -expire <days> -stash
```

where:

- `-db <filename>` is the name of the database.
- `-expire <days>` is the number of days before password expires. This parameter is only valid for CMS key databases.
- `-keydb` Specifies the command is for the key database.
- `-pw <password>` is the password to access the key database.
- `-type <cms | jks | jceks | pkcsk>` is the database type. Note: IBM HTTP Server only handles a CMS key database.
- `-stash` stashes the password for the key database. When the `-stash` option is specified during the key database creation, the password is stashed in a file with a filename built as follows:
`<filename_of_key_database>.sth`

This parameter is only valid for CMS key databases. For example, if the database being created is named `keydb.kdb`, the stash filename is `keydb.sth`.

Note: Stashing the password is required for IBM HTTP Server.

- Create a new key database using the `GSKCapiCmd` tool. `GSKCapiCmd` is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing `GSKit` Java command line tool has, except `GSKCapiCmd` supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use `GSKCapiCmd` to manage all aspects of a CMS key database. `GSKCapiCmd` does not require Java to be installed on the system.

```
<ihsinst>/bin/gskcapiCmd -keydb -create -db <name> [-pw <passwd>] [-type <cms>] [-expire <days>] [-stash]  
[-fips] [-strong]
```

Managing the database password using the command line

Distributed operating systems

This topic describes passwords for key databases. A key database is used to store public keys that are used for secure connections.

About this task

When you create a new key database, you specify a key database password. This password protects the private key. The private key is the only key that can sign documents or decrypt messages that are encrypted with the public key. Changing the key database password frequently is a good practice.

Use the following guidelines when specifying the password:

- The password must come from the U.S. English character set.
- The password must contain at least six characters and contain at least two nonconsecutive numbers. Make sure that the password does not consist of publicly obtainable information about you, such as the initials and birth date for you, your spouse, or children.
- Stash the password.

Procedure

- Change the password for a key database using the `gskcmd` command-line interface. Enter the following command as one line:

```
<ihsinst>/bin/gskcmd -keydb -changepw -db <filename>.kdb -pw <password> -new_pw <new_password> -expire <days> -stash
```

where:

- `-db <filename>` is the name of the database.
 - `-changepw` changes the password.
 - `-keydb` specifies the command is for the key database.
 - `-new_pw <new_password>` is the new key database password. This password must be different than the old password and cannot be a NULL string.
 - `-pw <password>` is the password to access the key database.
 - `-expire <days>` is the number of days before password expires. This parameter is only valid for CMS key databases.
 - `-stash` stashes the password for the key database. This parameter is only valid for CMS key databases. Stashing the password is required for IBM HTTP Server.
- Change the password using the `GSKCapiCmd` tool. `GSKCapiCmd` is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing `GSKit` Java command line tool has, except `GSKCapiCmd` supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use `GSKCapiCmd` to manage all aspects of a CMS key database. `GSKCapiCmd` does not require Java to be installed on the system.

```
<ihsinst>/bin/gskcapiCmd -keydb -changepw -db <name> [-crypto <module name> -tokenlabel <token label>][-pw <passwd>]-new_pw <new passwd> [-expire <days>] [-stash] [-fips] [-strong]
```

Results

The key database now accepts the new password.

Creating a new key pair and certificate request

Distributed operating systems

You find key pairs and certificate requests stored in a key database. This topic provides information on how to create a key pair and certificate request.

About this task

Create a public and private key pair and certificate request using the gskcapicmd command-line interface or GSKCapiCmd tool, as follows:

Procedure

1. Use the gskcapicmd command-line interface. Enter the following command (as one line):

```
<ihsinst>/bin/gskcapicmd -certreq -create -db <name> [-crypto <module name> [-tokenlabel <token label>]]  
[-pw <passwd>] -label <label> -dn <dist name> [-size <2048 | 1024 | 512>] -file <name> [-secondaryDB  
<filename> -secondaryDBpw <password>] [-fips] [-sigalg <md5 | sha1|sha224|sha256|sha384|sha512>]
```

where:

- -certreq specifies a certificate request.
- -create specifies a create action.
- -db <filename> specifies the name of the database.
- -pw is the password to access the key database.
- label indicates the label attached to the certificate or certificate request.
- dn <distinguished_name> indicates an X.500 distinguished name. Input as a quoted string of the following format (only CN, O, and C are required): CN=common_name, O=organization, OU=organization_unit, L=location, ST=state, province, C=country

Note: For example, CN=weblinux.raleigh.ibm.com,O=IBM,OU=IBM HTTP Server,L=RTP,ST=NC,C=US

- -size <2048 | 1024 | 512> indicates a key size of 2048, 1024, or 512. The default key size is 1024. The 2048 key size is available if you are using Global Security Kit (GSKit) Version 7.0.4.14 and later.
- -file <filename> is the name of the file where the certificate request will be stored.
- -san * <subject alternate name attribute value> | <subject alternate name attribute value> specifies the subject alternate name extensions in the certificate request that inform SSL clients of alternate hostnames that correspond to the signed certificate.

These options are only valid if the following line is entered in the ikminit.properties file.

DEFAULT_SUBJECT_ALTERNATE_NAME_SUPPORT=true. The * (asterisk) can have the following values:

dnsname

The value must be formatted using the preferred name syntax, according to RFC 1034. For example, zebra.tek.ibm.com.

emailaddr

The value must be formatted as an addr-spec according to RFC 822. For example, myname@zebra.tek.ibm.com

ipaddr

The value is a string representing an IP address formatted according to RFC 1338 and RFC 1519. For example, 193.168.100.115

The values of these options are accumulated into the subject alternate name extended attribute of the generated certificate. If the options are not used then this extended attribute is not added to the certificate.

- `-ca <true | false>` specifies the basic constraint extension to the self-signed certificate. The extension is added with a `CA:true` and `PathLen:<max int>` if the value passed is true or not added if the value passed is false.

gotcha: On Unix type operating systems it is recommended to always encapsulate string values associated with all tags in double quotes (`""`). You will also need to escape, using a `\` character, the following characters if they appear in the string values: `!`, `\`, `"`, `'`. This will prevent some command line shells from interpreting specific characters within these values. (e.g. `gskcapiCmd -keydb -create -db "/tmp/key.kdb" -pw "j\jj"`). Note however when prompted by `gskcapiCmd` for a value (for example a password) quoting the string and adding the escape characters should not be done. This is because the shell is no longer influencing this input.

Use the `GSKCapiCmd` tool. `GSKCapiCmd` is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing `GSKit` Java command line tool has, except `GSKCapiCmd` supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use `GSKCapiCmd` to manage all aspects of a CMS key database. `GSKCapiCmd` does not require Java to be installed on the system.

2. Verify that the certificate was successfully created:

- a. View the contents of the certificate request file you created.
- b. Ensure that the key database recorded the certificate request:

```
<ihsinst>/bin/gskcapiCmd -certreq -list -db <filename> -pw <password>
```

You should see the label listed that you just created.

3. Send the newly-created file to a certificate authority.

Importing and exporting keys using the command line

Distributed operating systems

This topic describes how to import and export keys.

About this task

If you want to reuse an existing key from another database, you can import that key. Conversely, you can export your key into another database or to a PKCS12 file. PKCS12 is a standard for securely storing private keys and certificates. You can use the `gskcmd` command-line interface or `GSKCapiCmd` tool.

Procedure

- Use the `gskcmd` command-line interface to import certificates from another key database, as follows:

```
<ihsinst>/bin/gskcmd -cert -import -db <filename> -pw <password>  
-label <label> -type <cms | JKS | JCEKS | pkcs12>  
-new_label <label> -target <filename> -target_pw <password>  
-target_type <cms | JKS | JCEKS | pkcs12>
```

where:

- `-cert` - specifies a certificate.
- `-import` - specifies an import action.
- `-db <filename>` - indicates the name of the database.
- `-pw <password>` - indicates the password to access the key database.
- `-label <label>` - indicates the label that is attached to the certificate.
- `-new_label <label>` - re-labels the certificate in the target key database.
- `-type <cms | JKS | JCEKS | pkcs12>` - specifies the type of database.
- `-target <filename>` - indicates the destination database.
- `-target_pw <password>` - indicates the password for the key database if `-target` specifies a key database
- `-target_type <cms | JKS | JCEKS | pkcs12>` - indicates the type of database that is specified by the `-target` operand.
- `pfX` - imported file in Microsoft `.pfX` file format.

Use the GSKCapiCmd tool to import certificates from another key database. GSKCapiCmd is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing GSKit Java command line tool has, except GSKCapiCmd supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use GSKCapiCmd to manage all aspects of a CMS key database. GSKCapiCmd does not require Java to be installed on the system.

```
<ihsinst>/bin/gskcapiCmd -cert
-import -db <name> [-crypto <module name> [-tokenlabel <token label>]] [-pw <passwd>]
[-secondaryDB <filename> -secondaryDBpw <password>]
-label <label> [-type < cms>] -target <name>
[-target_pw<passwd>] [-target_type <cms|pkcs11>] [-new_label < label>] [-fips]
```

- Use the `gskcmd` command-line interface to export certificates from another key database, as follows:

```
gskcmd -cert -export -db <filename> -pw <password> -label <label>
-type <cms | jks | jceks | pkcs12> -target <filename>
-target_pw <password> -target_type <cms | jks | jceks | pkcs12>
```

where:

- `-cert` specifies a personal certificate.
- `-export` specifies an export action.
- `-db <filename>` is the name of the database.
- `-pw <password>` is the password to access the key database.
- `-label <label>` is the label attached to the certificate.
- `-target <filename>` is the destination file or database. If the **target_type** is JKS, CMS, or JCEKS, the database specified here must exist.
- `-target_pw` is the password for the target key database.
- `-target_type <cms | jks | jceks | pkcs12>` is the type of database specified by the `-target` operand.
- `-type <cms | jks | jceks | pkcs12>` is the type of database key.

Use the GSKCapiCmd tool to export certificates from another key database. GSKCapiCmd is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing GSKit Java command line tool has, except GSKCapiCmd supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or

PKCS11, use the existing Java tool. You can use GSKCapiCmd to manage all aspects of a CMS key database. GSKCapiCmd does not require Java to be installed on the system.

```
<ihsinst>/bin/gskcapiCmd -cert extract -db <name> |  
-crypto <module name> [-tokenlabel <token label>] -pw <passwd>  
-label <label> -target <name> [-format <ascii | binary>] [-secondaryDB <filename>  
-secondaryDBpw <password> ][-fips]
```

Creating a self-signed certificate

Distributed operating systems

A self-signed certificate provides a certificate to enable SSL sessions between clients and the server, while waiting for the officially-signed certificate to be returned from the certificate authority (CA). A private and public key are created during this process. Creating a self-signed certificate generates a self-signed X509 certificate in the identified key database. A self-signed certificate has the same issuer name as its subject name.

About this task

Use this procedure if you are acting as your own CA for a private Web network. Use the IKEYCMD command-line interface or the GSKCapiCmd tool to create a self-signed certificate.

Procedure

- Create a self-signed certificate using the IKEYCMD command-line interface, as follows:

```
gskcmd -cert -create -db <filename> -pw <password> -size <2048 | 1024 | 512> -dn <distinguished_name>  
-label <label> -default_cert <yes | no> - expire <days> -san dnsname <DNS name value>[,<DNS name value>]  
-san emailaddr <email address value>[,<email address value>]  
-san ipaddr <IP address value>[,<IP address value>][ -ca <true | false>]
```

where:

- -cert specifies a self-signed certificate.
- -create specifies a create action.
- -db <filename> is the name of the database.
- -pw <password> is the password to access the key database.
- -dn <distinguished_name> - indicates an X.500 distinguished name. Input as a quoted string of the following format (Only CN, O, and C are required):
CN=common_name, O=organization, OU=organization_unit, L=location, ST=state, province, C=country
For example, CN=weblinux.raleigh.ibm.com,O=IBM,OU=IBM HTTP Server,L=RTP,ST=NC,C=US
- -label <label> is a descriptive comment used to identify the key and certificate in the database.
- -size <2048 | 1024 | 512> indicates a key size of 2048, 1024, or 512. The default key size is 1024. The 2048 key size is available if you are using Global Security Kit (GSKit) Version 7.0.4.14 and later.
- -default_cert<yes | no> specifies whether this is the default certificate in the key database.
- -expire <days> indicates the default validity period for new self-signed digital certificates is 365 days. The minimum is 1 day. The maximum is 7300 days (twenty years).

- `-san * <subject alternate name attribute value> | <subject alternate name attribute value>` specifies the subject alternate name extensions in the certificate request that inform SSL clients of alternate hostnames that correspond to the signed certificate.

These options are only valid if the following line is entered in the `ikmunit.properties` file.

`DEFAULT_SUBJECT_ALTERNATE_NAME_SUPPORT=true`. The * (asterisk) can have the following values:

dnsname

The value must be formatted using the preferred name syntax according to RFC 1034. For example, `zebra,tek.ibm.com`.

emailaddr

The value must be formatted as an `addr-spec` according to RFC 822. For example, `myname@zebra.tek.ibm.com`

ipaddr

The value is a string representing an IP address formatted according to RFC 1338 and RFC 1519. For example, `193.168.100.115`

The values of these options are accumulated into the subject alternate name extended attribute of the generated certificate. If the options are not used then this extended attribute is not added to the certificate.

- `-ca <true | false>` specifies the basic constraint extension to the self-signed certificate. The extension is added with a `CA:true` and `PathLen:<max int>` if the value passed is true or not added if the value passed is false.
- Create a self-signed certificate using the `GSKCapiCmd` tool. `GSKCapiCmd` is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing `GSKit` Java command line tool has, except `GSKCapiCmd` supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use `GSKCapiCmd` to manage all aspects of a CMS key database. `GSKCapiCmd` does not require Java to be installed on the system.

```
gskcapiCmd -cert -create [-db <name>][[-crypto <module name> -tokenlabel <token label>]][-pw <passwd>]
-label <label> -dn <dist name> [-size <2048|1024|512>][[-x509version <1|2|3>]][-default_cert <yes|no>]
[-expire <days>][[-secondaryDB <filename> -secondaryDBpw <password>] [-ca <true|false>]][-fips]
[-sigalg<md5|sha1|sha224|sha256|sha384|sha512>]
```

Note: On Unix type operating systems it is recommended to always encapsulate string values associated with all tags in double quotes (“”). You will also need to escape, using a ‘\’ character, the following characters if they appear in the string values: ‘|’, ‘\’, ‘”’, ‘\’’. This will prevent some command line shells from interpreting specific characters within these values. (e.g. `gsk7capiCmd -keydb -create -db “/tmp/key.kdb” -pw “j\!jj”`). Note however when prompted by `gsk7capiCmd` for a value (for example a password) quoting the string and adding the escape characters should not be done. This is because the shell is no longer influencing this input.

Receiving a signed certificate from a certificate authority

Distributed operating systems

This topic describes how to receive an electronically mailed certificate from a certificate authority (CA) that is designated as a trusted CA on your server. A certificate authority is a trusted third-party organization or company that issues digital certificates that are used to create digital signatures and public-private key pairs.

About this task

The certificate authority can send more than one certificate. In addition to the certificate for your server, the CA can also send additional signing certificates or intermediate CA certificates. For example, Verisign includes an intermediate CA certificate when sending a Global Server ID certificate. Before receiving the server certificate, receive any additional intermediate CA certificates. Follow the instructions in the Storing a CA certificate topic to receive intermediate CA certificates.

If the CA that issuing your CA-signed certificate is not a trusted CA in the key database, store the CA certificate first and designate the CA as a trusted CA. Then you can receive your CA-signed certificate into the database. You cannot receive a CA-signed certificate from a CA that is not a trusted CA. For instructions, see Storing a certificate authority certificate.

Procedure

- Receive the CA-signed certificate into a key database using the `gskcmd` command-line interface, as follows:

```
<ihsinst>/bin/gskcmd -cert -receive -file <filename> -db <filename> -pw <password>  
-format <ascii | binary> -label <label> -default_cert <yes | no>
```

where:

- `-cert` specifies a self-signed certificate.
 - `-receive` specifies a receive action.
 - `-file <filename>` is a file containing the CA certificate.
 - `-db <filename>` is the name of the database.
 - `-pw <password>` is the password to access the key database.
 - `-format <ascii | binary>` specifies that the certificate authority might provide the CA certificate in either ASCII or binary format.
 - `-default_cert <yes | no>` indicates whether this is the default certificate in the key database.
 - `-label` specifies the label that is attached to a CA certificate.
 - `-trust` indicates whether this CA can be trusted. Use enable options when receiving a CA certificate.
- Receive the CA-signed certificate into a key database using the `GSKCapiCmd` tool. `GSKCapiCmd` is a tool that manages keys, certificates, and certificate requests within a CMS key database. The tool has all of the functionality that the existing `GSKit` Java command line tool has, except `GSKCapiCmd` supports CMS and PKCS11 key databases. If you plan to manage key databases other than CMS or PKCS11, use the existing Java tool. You can use `GSKCapiCmd` to manage all aspects of a CMS key database. `GSKCapiCmd` does not require Java to be installed on the system.

```
<ihsinst>/bin/gskcapiCmd -cert -receive -file <name>  
-db <name> [-crypto <module name> [-tokenlabel <token label>]]  
[-pw <passwd>][-default_cert <yes|no>][-fips>
```

Displaying default keys and certificate authorities

Distributed operating systems

This section describes how to view trusted certificate authorities and display default keys within a key database.

About this task

A trusted certificate authority (CA) issues and manages public keys for data encryption. A key database is used to share public keys that are used for secure connections. The tasks that follow show how to view the certificate authorities that are in your database, along with their expiration dates.

Procedure

- Display a list of trusted CAs in a key database by entering the following command as one line:

```
<ihsinst>/bin/gskcmd -cert -list CA -db < dbname > -pw <password> -type <cms | jks | jceks | pkcs12>
```

- Display a list of certificates in a key database and their expiration dates by enter the following command:

```
<ihsinst>/bin/gskcmd -cert -list -expiry < days > -db < filename > -pw < paswword > - type < type >
```

where:

- -cert indicates the operation applies to a certificate.
- -list <all | personal | CA | site> specifies a list action. The default is to list all certificates.
- -expiry <days> indicates that validity dates should be displayed. Specifying the number of days is optional, though when used will result in displaying all certificates that expire within that amount of days. To list certificates that have already expired, enter the value 0.
- -db <filename> is the name of the key database. It is used when you want to list a certificate for a specific key database.
- -pw <password> specifies the password to access the key database.
- -type <cms | JKS | JCEKS | pkcs12> specifies the type of database.

Storing a certificate authority certificate

Distributed operating systems

This topic describes how to store a certificate from a certificate authority (CA) that is not a trusted CA.

Procedure

To store a certificate from a CA that is not a trusted CA, use the following command:

```
<ihsinst>/bin/gskcmd -cert -add -db <filename>.kdb -pw <password> -label <label>  
-format <ascii | binary> -trust <enable | disable> -file <filename>
```

where:

- -add specifies an add action.
- -cert indicates the operation applies to a certificate.
- -db <filename> is the name of the database.
- -file <filename> specifies the file containing the CA certificate.
- -format <ascii | binary> indicates the certificate authorities might supply a binary or an ASCII file.
- -label <label> is the label attached to a certificate or certificate request.
- -pw <password> is the password to access the key database.

- `-trust <enable | disable>` indicates whether this CA can be trusted. The default is `enable` and indicates that the CA can be trusted.

Storing the encrypted database password in a stash file

Distributed operating systems

For a secure network connection, you can store the CMS encrypted database password in a stash file.

About this task

Complete one of the following steps to store the encrypted database password in a stash file.

Procedure

- To store the password using the `gskcmd` command-line interface, enter the following command on one line.

```
<ihsinst>/bin/gskcmd -keydb -create -db <path_to_db>/<db_name> -pw <password> -type cms -expire <days> -stash
```
- If the CMS database has been created, enter the following command on one line to store the password using the `gskcmd` command line interface.

```
<ihsinst>/bin/gskcmd -keydb -stashpw -db <db_name> -pw <password>
```

Managing keys with the native key database `gskkyman` (z/OS systems)

z/OS

Use the native z/OS key management (`gskkyman` key database) support for key management tasks.

About this task

To have a secure network connection, create a key for secure network communications and receive a certificate from a certificate authority (CA) that is designated as a trusted CA on your server.

IBM HTTP Server on z/OS does not support `IKEYMAN` or `gskcmd`.

Use `gskkyman` to create key databases, public and private key pairs, and certificate requests. If you act as your own CA, you can use `gskkyman` to create self-signed certificates. If you act as your own CA for a private Web network, you have the option to use the server CA utility to generate and issue signed certificates to clients and servers in your private network.

Procedure

- To use native z/OS key management (`gskkyman`) tasks, refer to *Cryptographic Services PKI Services Guide and Reference* document (SA22-7693). Link to this document from the *z/OS Internet Library*.
- A typical task that this document contains is using a `gskkyman` key database for your certificate store. See section “Appendix B. Using a `gskkyman` key database” for a description of how to use `gskkyman`.

Important: The certificate requests that `gskkyman` generates for use with IBM HTTP Server should use RSA keys and not DSA keys.

Getting started with the cryptographic hardware for SSL (Distributed systems)

Distributed operating systems

Cryptographic devices require the PKCS11 support software for the host machine and internal firmware. For more information, contact the vendor of the device.

Cryptographic hardware for Secure Sockets Layer

Distributed operating systems

IBM HTTP Server supports many types of cryptographic hardware devices.

The following table contains hardware cryptographic devices that have been tested with IBM HTTP Server. However, since device drivers for these devices are frequently upgraded by the hardware vendors to correct customer-reported problems or to provide support for new operating system platforms, check with the hardware vendors for specific applications of these devices.

A list of cryptographic devices tested with GSKit is available at this IBM Web site:

IBM Global Security Kit, Version 8 - PKCS#11 device integration

If your device is not listed, contact the device vendor to ensure that the device functions correctly when used with IBM HTTP Server.

You can find a list of supported Hardware Cryptographic Cards for Java Version 7 at the IBM Web site:

Supported Hardware Cryptographic Cards for Java Version 7.

Device	Key Storage	Acceleration Support	Notes
Rainbow Cryptoswift PCI with BSAFE Interface Model	No	Yes	Use with SSLAcceleratorDisable directive only. Supported on HP, Solaris, and the Windows operating systems.
nCipher nFast Accelerator with BHAPI plug-in under BSAFE 4.0	No	Pure accelerator	Requires either a SCSI or PCI-based nForce unit; use with SSLAcceleratorDisable directive only. Supported on Solaris and Windows operating systems.
nCipher nForce Accelerator, <i>accelerator mode</i>	No	Yes	Uses the BHAPI and BSAFE interface. Supported on Solaris and Windows operating systems.

Device	Key Storage	Acceleration Support	Notes
nCipher nForce Accelerator, Key stored accelerator mode	Yes	Yes	Uses the PKCS#11 interface. Requires either a SCSI, or PCI-based nForce unit. Move to nCipher nForce Accelerator V4.0 or later for better performance. Supported on AIX, HP, Linux, Solaris, and Windows operating systems.
IBM 4758 Model 002/023 PCI Cryptographic Coprocessors	Yes	No	Supported on AIX and Windows operating systems.

AIX operating systems. Support for the following adapters has been tested with WebSphere Application Server V4.0.2 or later:

Device	Key Storage	Acceleration Support	Notes
Rainbow Cryptoswift PCI with BSAFE Interface Model CS/200 and CS/600	No	Yes	Supported on the AIX operating system.
IBM e-business Cryptographic Accelerator	No	Yes	Uses the PKCS11 interface. Because this device uses the PKCS11 interface, the SSLAcceleratorDisable directive does not apply to this device. Supported on the AIX operating system.

Use the Rainbow Cryptoswift, IBM e-business Cryptographic Accelerator, nCipher nFast Accelerator and nCipher nForce Accelerator, for public key operations, and RSA key decryption. These devices store keys on your hard drive. Accelerator devices speed up the public key cryptographic functions of SSL, freeing up your server processor, which increases server throughput and shortens wait time. The Rainbow Cryptoswift, IBM e-business Cryptographic Accelerator, and nCipher accelerators incorporate faster performance and more concurrent secure transactions.

The PKCS#11 protocol either stores RSA keys on cryptographic hardware, or encrypts keys using cryptographic hardware to ensure protection. The nCipher nForce Accelerator can either perform acceleration, or it can perform both acceleration and key storage with PKCS#11 support. The IBM 4758 and nCipher nForce Accelerator with PKCS#11 support ensures inaccessible keys to the outside world. This support never reveals keys in an unencrypted form because the key is either encrypted by the hardware, or stored on the hardware.

nCipher nForce Accelerator V4.0 and later using PKCS11 key storage, has a nonremovable option which can noticeably improve performance. Contact nCipher Technical Support for instructions to turn on this feature.

Using IKEYMAN to store keys on a PKCS11 device

Distributed operating systems

For IBM HTTP Server, you can use IKEYMAN for storing keys on a PKCS11 device.

Before you begin

Read about the IBMPKCS11Impl provider at <http://www.ibm.com/developerworks/java/jdk/security/60/secguides/pkcs11implDocs/IBMJavaPKCS11ImplementationProvider.html#ConfigFile>.

Procedure

- Obtain the file name and the path location of the cryptographic driver in order to store the keys on the PKCS11 device. The following are examples of path locations for PKCS11 devices:

– nCipher:

- **AIX** /opt/nfast/toolkits/pkcs11/libcknfast.so
- **HP-UX** /opt/nfast/toolkits/pkcs11/libcknfast.sl
- **Linux** /opt/nfast/toolkits/pkcs11/libcknfast.so
- **Solaris** /opt/nfast/toolkits/pkcs11/libcknfast.so
- **Windows** C:\nfast\toolkits\pkcs11\cknfast.dll

– IBM 4758

- **AIX** /usr/lib/pkcs11/PKCS11_API.so
- **Windows** \$PKCS11_HOME\bin\NT\cryptoki.dll

– IBM e-business Cryptographic Accelerator

- **AIX** /usr/lib/pkcs11/PKCS11_API.so

- If your Web server and Java Development Kit (JDK) are 64-bit, select a 64-bit vendor PKCS11 library.

On some platforms, the 64-bit PKCS11 library filename has 64 appended to it.

AIX **HP-UX** **Linux** **Solaris** You can display the architecture of the Web server by running `apachectl -V`.

Windows You can display the architecture of the Web server by running `httpd.exe -V`.

- Determine the token label of the PKCS11 token that you use.
Native vendor tools, such as `pkcsconf -its`, often display the token label.
- Create a PKCS11 configuration file describing your PKCS11 device using the following fields:
 - *library*: Full path to the proper architecture PKCS11 driver
 - *name*: Same as the token label from the previous step
 - *description*: Text field with your description
 - *attributes*: A set of attributes that you copy verbatim from the certificates example that the Web server uses

Note: With some cryptographic accelerators, an alternate syntax is required to avoid SSL0227E errors.

/opt/HTTPServer/conf/pkcs11.cfg example:

```
library = /usr/lib/pkcs11/PKCS11_API.so
name = PCI
description = description
attributes(*,CKO_PRIVATE_KEY,*) =
    {CKA_PRIVATE=true CKA_TOKEN=true}
```

- Update the java/jre/lib/security/java.security file under the installation root to add a new security provider.
 - Have the new security provider reference the GSKit PKCS11 classes and the location of your PKCS11 configuration file.
 - Append the provider to the end of the provider list as the new highest-numbered provider.
 - Modify the following examples to specify the location of your configuration file.

Some of the lines are split on multiple lines for display purposes. Enter a line as a single line even if it displays in this task as multiple lines.

AIX

HP-UX

Linux

Solaris

```
# The following line is the last pre-existing security provider.
security.provider.12=com...
# Add the following line.
security.provider.13=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/HTTPServer/conf/pkcs11.cfg
```

Windows

```
# The following line is the last pre-existing security provider.
security.provider.12=com...
# Add the following line.
security.provider.13=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl C:\opt\HTTPServer\conf\pkcs11.cfg
```

- Run IKEYMAN to store the keys on the PKCS11 device.

After launching IKEYMAN:

1. Select **Key Database File** from the menu, then **Open** to navigate to the Key database information dialog
2. From the drop-down for **Key Database Type**, select **PKCS11Config**.
If **PKCS11Config** is not an option, but **PKCS11Direct** is, you have an error that you must fix. Check your java.security work in prior steps. The **PKCS11Direct** option is not visible to the Web server.
All other fields become locked, as the parameters are provided from the pkcs11.cfg file.
3. Click **OK** to navigate to the **Open Cryptographic Token** dialog.
The **Cryptographic Token Label** label of the PKCS11 device is displayed in the panel. This label comes from the name field of the pkcs11.cfg file, and might be different from the native token label.
4. Complete the following actions on the **Open Cryptographic Token** dialog.
 - a. Enter the cryptographic token password for the PKCS11 device in the **Cryptographic Token Password** field. The password was previously set and is hardware-specific. This password is often called a user PIN in vendor documentation and tools.
 - b. Select the **Create new secondary key database file** option and complete prompts for creating a new secondary key database.
 - c. Click **OK**.

Results

After opening a cryptographic token successfully, IKEYMAN will display the certificates stored in the cryptographic token.

What to do next

You can create, import, or receive a personal certificate as you normally would. The private key is stored on your PKCS11 device.

Configuring IBM HTTP Server to use nCipher and Rainbow accelerator devices and PKCS11 devices

Distributed operating systems

The IBM HTTP Server enables nCipher and Rainbow accelerator devices by default. To disable your accelerator device, add the **SSLAcceleratorDisable** directive to your configuration file.

Before you begin

When using the IBM e-business Cryptographic Accelerator, or the IBM 4758, the user ID under which the Web server runs must be a member of the PKCS11 group. You can create the PKCS11 group by installing the `bos.pkcs11` package or its updates. Change the **Group** directive in the configuration file to `group pkcs11`.

About this task

If you want the IBM HTTP Server to use the PKCS11 interface, configure the following:

Procedure

1. Stash your password to the PKCS11 device, or optionally enable password prompting.

The stash file that the `sslstash` command creates is completely independent of the stash file that often accompanies a CMS KeyFile (*.kdb). Therefore, make sure that you:

- Do not overwrite an existing *.sth file when you issue the `sslstash` command.
- Never choose a filename for the output of the `sslstash` command that corresponds to the filename of a CMS KeyFile (*.kdb).

Syntax: `sslstash [-c] <file> <function> <password>` where:

- `-c`: Creates a new stash file. If not specified, an existing stash file is updated.
 - `file`: Represents a fully-qualified name of the file to create or update.
 - `function`: Represents the function for which the server uses the password. Valid values include `cr1` or `crypto`.
 - `password`: Indicates the password to stash.
2. Place the following directives in your configuration file.
 - `SSLPKCSDriver <fully qualified name of the PKCS11 driver used to access PKCS11 device>`
See `SSLPKCSDriver` directive for the default locations of the PKCS11 module, for each PKCS11 device.
 - `SSLServerCert <token label: key label of certificate on PKCS11 device>`

- SSLStashfile<fully qualified path to the file containing the password for the PKCS11 device>
- Keyfile<fully qualified path to key file with signer certificates>

Authenticating with LDAP on IBM HTTP Server using mod_ldap

You can configure Lightweight Directory Access Protocol (LDAP) to authenticate and protect files on IBM HTTP Server.

Before you begin

Best Practice: Distributed operating systems If you are using the mod_ibm_ldap module for your LDAP configuration, consider migrating your mod_ibm_ldap directives to use the mod_ldap module. The mod_ibm_ldap module is provided with this release of IBM HTTP Server for compatibility with previous releases, however, you must migrate existing configurations to use the mod_authnz_ldap and mod_ldap modules to ensure future support for your LDAP configuration.

The LoadModule directive for LDAP does not load into IBM HTTP Server by default. Without the LoadModule directive, the LDAP features are not available for use.

In order to enable the LDAP function, add a LoadModule directive to the IBM HTTP Server httpd.conf file as follows:

```
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
```

About this task

LDAP authentication is provided by the mod_ldap and mod_authnz_ldap Apache modules.

- The mod_ldap module provides LDAP connection pooling and caching.
- The mod_authnz_ldap makes use of the LDAP connection pooling and caching services to provide Web client authentication.

See the following Web sites to obtain detailed descriptions of the LDAP (ldap_module and authnz_ldap_module) directives:

- http://publib.boulder.ibm.com/httserv/manual70/mod/mod_ldap.html
- http://publib.boulder.ibm.com/httserv/manual70/mod/mod_authnz_ldap.html

Procedure

1. Edit the httpd.conf IBM HTTP Server configuration file.
2. Determine the resource for which you want to limit access. For example:
<Directory "/secure_info">
3. Add the LDAPTrustedGlobalCert directive to httpd.conf if the IBM HTTP Server connection to the LDAP server is an SSL connection.

The LDAPTrustedGlobalCert directive specifies the directory path and file name of the trusted certificate authority (CA) that mod_ldap uses when establishing an SSL connection to an LDAP server.

Certificates can be stored in a .kdb file or a SAF key ring. If a .kdb file is used, a .sth file must be located in the same directory path and have the same filename, but the extension must be .sth instead of .kdb.

Distributed operating systems The LDAPTrustedGlobalCert directive must be a CMS_KEYFILE value type. Use this value if the certificates indicated by the LDAPTrustedGlobalCert directive are stored in a .kdb file.

z/OS The LDAPTrustedGlobalCert directive must be a SAF_KEYRING value type. Use this value if the certificates indicated by the LDAPTrustedGlobalCert directive are stored in a SAF key ring. Example when the certificate is stored in a .kdb file: **Distributed operating systems**

```
LDAPTrustedGlobalCert CMS_KEYFILE /path/to/keyfile.kdb myKDBpassword
```

Example when the certificate is stored in a SAF key ring. **z/OS**

```
LDAPTrustedGlobalCert SAF saf_keyring
```

Important: The user ID that you use to start IBM HTTP Server must have access to the SAF key ring that you name in this directive. If the user ID does not have access to the SAF key ring, SSL initialization fails. See “Performing required z/OS system configurations” on page 65 for information on accessing SAF key rings defined in RACF.

4. Add the AuthLDAPUrl directive, which specifies the LDAP search parameters to use.

The syntax of the URL is:

```
ldap://host:port/basedn?attribute?scope?filter
```

5. Add directives in httpd.conf to the directory location (container) to be protected with values specific to your environment, such as:

- Order deny, allow
- Allow from all
- AuthName *Title of your protected Realm*
- AuthType Basic
- AuthBasicProvider ldap
- AuthLDAPURL *your_ldap_url*
- Require valid-user
- AuthLDAPBindDN "cn=Directory Manager"
- AuthLDAPBindPassword *auth_password*

In addition to allowing any user present in the LDAP repository, mod_authnz_ldap provides Require ldap-user, Require ldap-group, and Require ldap-filter. When you use multiple Require directives, authorization succeeds if any, but not all, Require directives match the current user.

- Require ldap-user user1
Looks up "user1" based on AuthLDAPURL and makes sure their DN matches the DN of the authenticated user
- Require ldap-group cn=group1,o=example,c=US
Searches for the currently authenticated user in the listed LDAP group
- Require ldap-filter "(someAttr=val1)(someVal=val2)"
Searches for the authenticated user under the provided LDAP filter. If the filter returns 1 result, authorization passes.

For each combination of LDAP server, protection setup, and protect directive, code a Location container similar to the following example:

```
<Location /ldapdir>  
  AuthName "whatever_LDAP"  
  AuthType Basic  
  AuthBasicProvider ldap
```

```

AuthLDAPURL ldap://9.27.163.182:389/o=abc.xyz.com?cn?sub?
Require valid-user
AuthLDAPBindDN "cn=Directory Manager"
AuthLDAPBindPassword d44radar
</Location>

```

http://publib.boulder.ibm.com/httserv/manual70/mod/mod_authnz_ldap.html

Converting your directives from mod_ibm_ldap to mod_ldap

Distributed operating systems

Convert directives that use the mod_ibm_ldap module to use the mod_ldap Apache module to ensure continued IBM HTTP Server support for your LDAP configuration.

Before you begin

Determine which directives to convert.

Complete these steps to convert your directives.

Procedure

1. Edit the LoadModule directive in the httpd.conf or ldap.prop configuration file to remove mod_ibm_ldap.
LoadModule ibm_ldap_module modules/mod_ibm_ldap.so
2. Add the mod_ldap LoadModule directive to the httpd.conf configuration file.
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule ldap_module modules/mod_ldap.so
3. Convert one or more of the following directives. For more information about converting your directives, see the topic about mod_ibm_ldap migration.

Note: A one to one correlation might not exist for some directives.

Table 15. LDAP configuration directives conversion

mod_ibm_ldap	mod_ldap
ldapCodePageDir	None. The codepages directory cannot be moved from its installed location.
LdapConfigFile	include
LdapRequire	require
ldap.application.authType	None. If the mod_ldap directive, AuthLDAPBindDN, is specified, then you will get Basic auth. If no AuthLDAPBindDN is specified, then you get what would have been the None auth type (anonymous). If the mod_ldap configuration specifies an LDAPTrustedClientCert value then you will get the Cert auth type.
ldap.application.DN	AuthLDAPBindDN
ldap.application.password	AuthLDAPBindPassword
ldap.application.password.stashFile	None. The mod_ldap module does not provide a directive for using stashed passwords.
ldap.cache.timeout	LDAPCacheTTL
ldap.group.dnattributes	AuthLDAPSubGroupClass
ldap.group.memberattribute	AuthLDAPSubGroupAttribute
ldap.group.memberattributes	AuthLDAPGroupAttribute
ldap.group.name.filter	None. The mod_ldap module uses the filter provided at the end of the AuthLDAPURL directive.
ldap.group.search.depth	AuthLDAPMaxSubGroupDepth

Table 15. LDAP configuration directives conversion (continued)

mod_ibm_ldap	mod_ldap
ldap.group.URL	AuthLDAPURL
ldap.idleConnection.timeout	None. The mod_ldap module does not provide a directive for connection timeouts.
ldap.key.file.password.stashfile	None. The mod_ldap module does not provide a directive for using stashed passwords. Specify the keyfile password, in clear text, at the end of the LDAPTrustedGlobalCert directive. Alternatively, omit the password on the LDAPTrustedGlobalCert directive and the mod_ldap module automatically looks for a /path/to/keyfile.sth file, assuming /path/to/keyfile.kdb was the specified value of the LDAPTrustedGlobalCert directive.
ldap.key.fileName	LDAPTrustedGlobalCert
ldap.key.label	LDAPTrustedClientCert
ldap.ReferralHopLimit	LDAPReferralHopLimit
ldapReferrals	LDAPReferrals
ldap.realm	None. The mod_ibm_ldap value of this directive was only used for logging purposes. No equivalent directive is required in mod_ldap.
ldap.search.timeout	LDAPSearchTimeout
ldap.transport	LDAPTrustedMode
ldap.URL	AuthLDAPURL
ldap.user.authType	None. The mod_ldap module authenticates users based on the user ID and password credentials provided.
ldap.user.cert.filter	None. The mod_ldap module does not work directly with client certificates. Authorization directives use the environment values set by the SSL module.
ldap.user.name.fieldSep	None. The mod_ldap module does not provide support for parsing the provided credentials into subcomponents.
ldap.user.name.filter	None. The mod_ldap module specifies the user name filter as part of the AuthLDAPURL directive.
ldap.version	None. The mod_ldap module uses only LDAP version 3.
ldap.waitToRetryConnection.interval	None. The mod_ldap module does not have a timed delay between connection retries when a connection attempt fails. The connection attempt is retried for a maximum of 10 times before request fails.

4. Run the Apache control with the verify flag to verify the configuration.

```
<ihst>bin/apachectl -t
```

Attention: This configuration check confirms that the syntax is correct, but you must verify any configuration changes for a directive using the documentation for that directive to ensure an optimal configuration.

Attention: All `mod_ibm_ldap` directives that use the form `ldap.*` used to optionally display in the `LDAPConfigFile` configuration file without the `ldap` prefix.

A mod_ldap SSL configuration

The following configuration directives show a sample SSL-enabled LDAP configuration. Some of the directives specify default values and would not typically need to be specified, but are retained to provide context. Those directives are included, but are commented out with `'##'` symbols.

```
##LDAPReferrals On
##LDAPReferralHopLimit 5
```

```
LDAPTrustedGlobalCert CMS_KEYFILE /full/path/to/
ldap_client.kdb clientkdbPassword #default cert in this kdb is my_cert1 #
Alternatively, you can specify a SAF-based keyring, on systems that support it, as
```

```

follows: #LDAPTrustedGlobalCert SAF saf_keyring <VirtualHost *> ServerAdmin
admin@my.address.com DocumentRoot /path/to/htdocs # Ignored because LDAP
URLs use ldaps:, where needed ##LDAPTrustedMode SSL <Directory
/minimal_ldap_config> AuthBasicProvider ldap AuthLDAPURL
ldap://our_ldap.server.org/o=OurOrg,c=US AuthName "Private root access"
require valid-user </Directory> <Directory /path/to/htdocs>
##AuthzLDAPAuthoritative on AuthBasicProvider ldap # This
LDAPTrustedClientCert is required to use a different certificate # than the default
LDAPTrustedClientCert CMS_LABEL my_cert2 AuthLDAPURL
ldaps://our_ldap.server.org:636/o=OurOrg,c=US?cn?sub? (objectclass=person)
AuthLDAPBindDN "cn=ldapadm,ou=OurDirectory,o=OurCompany,c=US"
AuthLDAPBindPassword mypassword AuthName "Private root access" require
ldap-group cn=OurDepartment,o=OurOrg,c=us </Directory> <Directory
"/path/to/htdocs/employee_of_the_month"> ##AuthzLDAPAuthoritative on
AuthBasicProvider ldap #Uses default cert (my_cert1) ##LDAPTrustedClientCert
CMS_LABEL my_cert1 AuthLDAPURL ldaps://our_ldap.server.org:636/
o=OurOrg,c=US?cn?sub?(objectclass=person) AuthLDAPBindDN
"cn=ldapadm,ou=OurDirectory,o=OurCompany,c=US" AuthLDAPBindPassword
mypassword AuthName "Employee of the month login" require ldap-attribute
description="Employee of the Month." </Directory> <Directory
"/path/to/htdocs/development_groups"> #These are the default values for the
subgroup-related directives and only need to be #specified when the LDAP
structure differs. ##AuthzLDAPAuthoritative on AuthBasicProvider ldap # This
LDAPTrustedClientCert is required to use a different certificate # than the default
LDAPTrustedClientCert CMS_LABEL my_cert3 AuthLDAPURL
ldaps://groups_ldap.server.org:636/o=OurOrg,c=US?cn?sub?
(l(objectclass=groupofnames)(object class=groupo1 funiquenames))
AuthLDAPBindDN "cn=ldapadm,ou=OurDirectory,o=OurCompany,c=US"
AuthLDAPBindPassword mypassword AuthName "Developer Access"
AuthLDAPGroupAttribute member AuthLDAPMaxSubGroupDepth 2
AuthLDAPSubGroupClass groupOfUniqueNames ##AuthLDAPSubGroupClass
groupOfNames ##AuthLDAPSubGroupAttribute uniqueMember
##AuthLDAPSubGroupAttribute member require ldap-group
cn=Developers_group,o=OurOrg,c=us </Directory> </VirtualHost>
LDAPTrustedMode None

```

mod_IBM_ldap directives migration

Distributed operating systems

z/OS

This article contains information to help with migration from existing directives that use the mod_IBM_ldap module to the use of the open source LDAP modules (mod_authnz_ldap and mod_ldap). Migration will ensure future support for your LDAP configuration.

Attention: Although many of the mod_IBM_ldap directives are located in the ldap.prop file, the open source LDAP directives are all located in the httpd.conf file.

The open source LDAP features are provided by two modules. The AuthLDAP directives are provided by the mod_authnz_ldap module and the LDAP directives are provided by the mod_ldap module. Both modules need to be loaded for the LDAP features to be available. Throughout the following section the generic name, mod_ldap, is used to reference the open source LDAP modules.

ldapCodePageDir:

The mod_ldap module does not provide a directive for specifying a codepages directory. The codepages directory is automatically installed in the correct directory, and the codepages directory cannot be moved from its installed location.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldapCodePageDir /location/of/codepages
```

LDAPConfigfile:

The mod_ldap module does not provide a directive for specifying an LDAP configuration file. Although there is no mod_ldap directive for specifying the LDAP configuration file, if you want to put your LDAP configuration in a separate file, you might use the Apache include directive.

Convert this:

```
ldapConfigFile ldap.prop
```

to this:

```
Include /location/of/ldap_conf/apache_ldap.conf
```

Another alternative for migrating the mod_ibm_ldap LDAPConfigfile directive is to use the mod_authn_alias module AuthnProviderAlias container to create one or more groupings of ldap directives, and then use them by referencing the alias labels where required

LdapRequire:

The mod_ldap module provides the require directive, with LDAP extensions, for LDAP authentication security.

If you used require valid-user previously for IBM HTTP Server, you may leave this require directive in place without modification. For the highest level of LDAP authentication security, you should migrate require valid-user to a more specific form. For additional information, see the Apache documentation for these require directives: ldap-user, ldap-dn, ldap-attribute, ldap-group, ldap-filter, and valid-user.

Convert this:

```
LdapRequire filter "&(objectclass=person)(cn=*)(ou=OurUnit)(o=OurOrg)"  
LdapRequire group MyDepartment
```

to this:

```
require ldap-filter &(objectclass=person)(cn=*)(ou=OurUnit)(o=OurOrg)  
require ldap-group cn=MyDepartment,o=OurOrg,c=US
```

ldap.application.authType:

The mod_ldap module does not provide a directive specifying an authentication type. If a value is specified for the AuthLDAPBindDN directive, then basic authentication is enabled. If a value is not specified for the AuthLDAPBindDN directive, then what was previously the None authentication type for the mod_ibm_ldap module, or anonymous, is enabled.

If a value is specified for the LDAPTrustedClientCert directive, then the certificate authentication type is used automatically.

```
ldap.application.authType=[None | Basic | Cert]
```

ldap.application.DN:

The `mod_ldap` module provides the `AuthLDAPBindDN` directive to determine the application authentication type.

If a value is specified for the `AuthLDAPBindDN` directive, then the value of the `authType` directive is `Basic`. If the `AuthLDAPBindDN` directive is not enabled, then the value for the `authType` directive is `None`. If a value is specified for the `LDAPTrustedClientCert` directive, then the value for the `authType` directive is `Cert`.

Important: `AuthLDAPBindDN` also takes the place of `ldap.application.authType`.

Convert this:

```
ldap.application.DN=cn=ldapadm,ou=OurDirectory,o=OurCompany,c=US
```

to this:

```
AuthLDAPBindDN "cn=ldapadm,ou=OurDirectory,o=OurCompany,c=US"
```

ldap.application.password:

The `mod_ldap` module provides the `AuthLDAPBindPassword` directive to specify a bind password. The value is stored in the configuration file in plain text. Therefore, you should restrict access to the configuration file

Convert this:

```
ldap.application.password=mypassword
```

to this:

```
AuthLDAPBindPassword mypassword
```

ldap.application.password.stashFile:

The `mod_ldap` module does not provide a directive for stashing the password. The directive `AuthLDAPBindPassword` is the only means to specify a password, and the value is stored in the configuration file in plain text. Therefore, you should restrict access to the configuration file.

This `mod_ibm_ldap` directive has no `mod_ldap` equivalent:

```
ldap.application.password.stashfile=/path/to/stashfile.sth
```

ldap.cache.timeout:

The `mod_ldap` module provides the `LDAPCacheTTL` directive to specify a timeout for the LDAP cache. The `LDAPCacheTTL` directive is globally scoped and must be located at the highest level of the configuration file. This is different from the `mod_ibm_ldap` module, because the `ldap.cache.timeout` directive could be located anywhere in the configuration file.

Convert this:

```
ldap.cache.timeout=60
```

to this:

```
LDAPCacheTTL 60
```

The default value is 600 seconds.

ldap.group.dnattributes:

The `mod_ldap` module provides the `AuthLDAPSubGroupClass` directive to specify the object classes which identify groups. For the `mod_ibm_ldap` module all values were specified on a single directive line; but for the `mod_ldap` module, the values can either be specified all on one line or on multiple lines, with the directive and one value on each line.

Convert this:

```
ldap.group.dnattributes=groupOfNames GroupOfUniqueNames
```

to this:

```
AuthLDAPSubGroupClass groupOfNames  
AuthLDAPSubGroupClass groupOfUniqueNames
```

These are the default values.

ldap.group.memberattribute:

The `mod_ldap` module provides the `AuthLDAPSubGroupAttribute` directive to specify the labels which identify the subgroup members of the current group. For the `mod_ibm_ldap` module, you could only specify one label; but for the `mod_ldap` module, you can specify multiple labels either by listing all of the labels in one directive line or by providing multiple directive lines, with each label on a separate directive line.

Convert this:

```
ldap.group.memberattribute=member
```

to this:

```
AuthLDAPSubGroupAttribute member  
AuthLDAPSubGroupAttribute uniqueMember
```

ldap.group.memberattributes:

The `mod_ldap` module provides the `AuthLDAPGroupAttribute` directive to specify the labels which identify any member of the current group, such as a user or subgroup. For the `mod_ibm_ldap` module, you specified all labels on one directive line; but for the `mod_ldap` module, you may either specify them all on one directive line or specify each label on a separate directive line.

Convert this:

```
ldap.group.membreattributes=member uniqueMember
```

to this:

```
AuthLDAPGroupAttribute member  
AuthLDAPGroupAttribute uniqueMember
```

ldap.group.name.filter:

The `mod_ldap` module does not provide a directive to specify separate user and group filters. The `mod_ldap` module uses the filter that is provided at the end of the `AuthLDAPURL` directive. You can use the `AuthnProviderAlias` container directive, which is provided by the `mod_authn_alias` module, to create separate

my_ldap_user_alias and my_ldap_group_alias aliases containing the required ldap directives. You can then use your group alias in locations where authorization is controlled by way of group membership.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.group.name.filter=(&(cn=%v1)(|(objectclass=groupofnames)(objectclass=groupofuniquenames)))
```

ldap.group.search.depth:

The mod_ldap module provides the AuthLDAPMaxSubGroupDepth directive to limit the recursive depth pursued before stopping attempts to locate a user within nested groups.

Convert this:

```
ldap.group.search.depth=5
```

to this:

```
AuthLDAPMaxSubGroupDepth 5
```

The default value is 10.

ldap.group.URL:

The mod_ldap module does not provide a directive for specifying an LDAP server for authorizing a group membership that is different from the LDAP server that is used to authenticate users.

You must also specify the LDAP group server in the AuthLDAPURL directive for the container. Ensure that you specify the correct filter for each group.

```
ldap.group.URL=ldap://groups_ldap.server.org:389/o=0ur0rg,c=US  
ldap.group.URL=ldaps://groups_ldap.server.org:636/o=0ur0rg,c=US
```

ldap.idleConnection.timeout:

The mod_ldap module does not provide a directive for specifying when established connections to the LDAP server, that have gone idle, should timeout. The mod_ldap module automatically detects when the LDAP server expires connections, but does not cause connections to expire.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.idleConnection.timeout=60
```

ldap.key.file.password.stashfile:

If no password is specified in the LDAPTrustedGlobalCert directive, the mod_ldap module automatically uses a /path/to/keyfile.sth file (assuming that /path/to/keyfile.kdb is the keyfile that is specified in the LDAPTrustedGlobalCert directive).

For information about how to specify the keyfile password, see the Apache information for the LDAPTrustedGlobalCert directive. The value is stored in the configuration file in plain text. Therefore, you should restrict access to the configuration file.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.key.file.password.stashfile=/path/to/ldap.sth
```

ldap.key.fileName:

The mod_ldap module provides the LDAPTrustedGlobalCert directive to specify the keyfile to be used when loading certificates. The mod_ldap module also uses these directives to specify the password in plain text in the configuration file. Therefore, you should restrict access to the configuration file.

Convert this:

```
ldap.key.filename=/path/to/keyfile.kdb
```

to this: **Distributed operating systems**

```
LDAPTrustedGlobalCert CMS_KEYFILE /path/to/keyfile.kdb myKDBpassword
```

z/OS

```
LDAPTrustedGlobalCert SAF saf_keyring
```

ldap.key.label:

The mod_ldap module provides the LDAPTrustedClientCert directive to specify which certificate to use from the KDB keyfile. If the default certificate is used, then you do not need to specify a value for these directives.

Convert this:

```
ldap.key.label=certname_from_kdb
```

to this:

```
LDAPTrustedClientCert CMS_LABEL certname_from_kdb
```

ldap.ReferralHopLimit:

The mod_ldap module provides the LDAPReferralHopLimit directive to limit the number of referrals to chase before stopping attempts to locate a user in a distributed directory tree.

Convert this:

```
ldapReferralHopLimit 5
```

to this:

```
LDAPReferralHopLimit 5
```

The default value is 5.

ldapReferrals:

The mod_ldap module provides the LDAPReferrals directive to enable or disable referral chasing when locating users in a distributed directory tree.

Convert this:

```
ldapReferrals On
```

to this:

```
LDAPReferrals On
```

The default value is On.

ldap.realm:

The mod_ldap module provides the AuthName directive to specify the authorization realm.

Convert this:

```
ldap.realm=Some identifying text
```

to this:

```
AuthName "Some identifying text"
```

ldap.search.timeout:

The mod_ldap module provides the LDAPSearchTimeout directive to specify when a search request should be abandoned.

Convert this:

```
ldap.search.timeout=10
```

to

```
LDAPSearchTimeout 10
```

The default value is 10 seconds.

ldap.transport:

The mod_ldap module provides the LDAPTrustedMode directive to specify the type of network transport to use when communicating with the LDAP server.

If no port is specified on the AuthLDAPURL directive, then the mod_ldap module ignores the LDAPTrustedMode directive, and specifies a network transport value of SSL. For more information, see the Apache documentation for the LDAPTrustedMode and AuthLDAPURL directives.

You can specify a value for the following network transport types.

- None or TCP, which indicates no encryption. If no port is specified on the AuthLDAPURL directive, then port 389 is used.
- SSL. If a value of None is specified, then port 636 is used.
- TLS or STARTTLS. These open source types are not supported by IBM HTTP Server.

Convert this:

```
ldap.transport=TCP (or SSL)
```

to this:

```
LDAPTrustedMode NONE (or SSL)
```

If an ldaps://URL is specified, the mode becomes SSL and the setting of LDAPTrustedMode is ignored.

ldap.URL:

The mod_ldap module provides the AuthLDAPURL directive for specifying the LDAP server hostname and port as well as the base DN to use when connecting to the server. The mod_ldap module also provides a means for specifying the user attribute, scope, user filter, and transport mode. For more information, see the Apache documentation for the AuthLDAPURL directives.

Convert this:

```
ldap.URL=ldap://our_ldap.server.org:389/o=OurOrg,c=US
ldap.URL=ldaps://our_ldap.server.org:636/o=OurOrg,c=US
```

to this:

```
AuthLDAPURL ldap://our_ldap.server.org:389/o=OurOrg,c=US?cn?sub?(objectclass=person)
AuthLDAPURL ldaps://our_ldap.server.org:636/o=OurOrg,c=US?cn?sub?(objectclass=person)
```

ldap.user.authType:

The mod_ldap module does not provide a directive for specifying a user authentication type. The mod_ldap module authenticates users based on the user ID and password credentials provided.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.user.authType=Basic [Basic | Cert | BasicIfNoCert]
```

ldap.user.cert.filter:

The mod_ldap module does not provide a directive for filtering client certificates. The mod_ldap module does not work directly with client certificates.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.user.cert.filter=(&(objectclass=person)(cn=%v1)(ou=%v2)(o=%v3)(c=%v4))
```

ldap.user.name.fieldSep:

The mod_ldap module does not provide a directive for parsing provided credentials into subcomponents. The mod_ibm_ldap module uses the ldap.user.name.fieldSep directive to specify the separator characters used to parse the credentials into the %v1, %v2, ...%vN tokens.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.user.name.fieldSep=/ ,
```

ldap.user.name.filter:

The mod_ldap module does not provide a directive for specifying the user name filter. The mod_ldap module specifies the user name filter as part of the AuthLDAPURL directive.

The AuthLDAPURL directive combines the user attribute specified in the directive with the provided filter to create the search filter. The provided filter follows the standard search filter specification. The mod_ldap module also does not provide the %vx token parsing function available for the mod_ibm_ldap module.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.user.name.filter=(&(objectclass=person)(cn=%v1 %v2))
```

ldap.version:

The mod_ldap module does not provide a directive for specifying the LDAP version. The mod_ldap module uses only LDAP version 3.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.version=2 (or 3)
```

ldap.waitToRetryConnection.interval:

The mod_ldap module does not provide a directive for specifying an amount of time before retrying a failed connection attempt. The mod_ldap module does not have a timed delay between connection retries when a connection attempt fails. The connection attempt is automatically retried for a maximum of 10 times before a request fails.

When a new request needs to access the same LDAP server, the connection is retried for a maximum of 10 times again. The retry throttle is based on the volume of new requests sent to the LDAP server.

This mod_ibm_ldap directive has no mod_ldap equivalent:

```
ldap.waitToRetryConnection.interval=300
```

Related tasks:

Authenticating with LDAP on IBM HTTP Server using **mod_ibm_ldap** (Distributed systems)

This section describes how to configure LDAP to protect files on IBM HTTP Server.

Related reference:

“Apache modules (containing directives) supported by IBM HTTP Server” on page 78

This section provides information on Apache modules that are supported by IBM HTTP Server. The directives defined within the supported Apache modules can be used to configure IBM HTTP Server.

Authenticating with SAF on IBM HTTP Server (z/OS systems)

z/OS

You can authenticate to the IBM HTTP Server on z/OS using HTTP basic authentication or client certificates with the System Authorization Facility (SAF) security product. Use SAF authentication for verification of user IDs and passwords or certificates.

Before you begin

The mod_authz_default and mod_auth_basic directives provide basic authentication and authorization support which is needed in mod_authnz_saf configurations. In addition, the mod_ibm_ssl directive provides support for SSL client certificates. If you use SAF authentication, ensure that the first three LoadModule directives from the following example are activated. If you use SSL client certificates, ensure that the mod_ibm_ssl.so LoadModule directive is activated as well.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
# mod_authz_core will typically already load by default
```

```
LoadModule authz_core_module modules/mod_authz_core.so
# Uncomment mod_IBM_ssl if any type of SSL support is required,
# such as client certificate authentication
#LoadModule IBM_ssl_module modules/mod_IBM_ssl.so
```

If the `mod_authz_default` module is not loaded by your Web server, the server returns a response code 500 instead of 401 if the user is not authorized.

About this task

SAF authentication is provided by the `mod_authnz_saf` module. The `mod_authnz_saf` module allows the use of HTTP basic authentication or client certificates to restrict access by looking up users, groups, and SSL client certificates in SAF. This module also allows you to switch the thread from the server ID to another ID prior to responding to the request by using the `SAFRunAS` directive. For additional information, see the information center topic about SAF directives. Also, see the topic about migrating and installing IBM HTTP Server on z/OS systems for information about migrating your SAF directives.

Procedure

1. If you are using `SAFRunAs`, permit the IBM HTTP Server `userid` to the `BPX.SERVER FACILITY` class profile in RACF, and provide the target `userid` with an OMVS segment.
2. Determine the directory location you want to limit access to. For example:
`<Location "/admin-bin">`
3. Add directives in the `httpd.conf` file to the directory or location to be protected with values specific to your environment. If you want to restrict access to files under the `/secure` directory to only users who provide a valid SAF user ID and password, consider this example.

```
<Directory /secure>
  AuthName protectedrealm_title
  AuthType Basic
  AuthBasicProvider saf
  Require valid-user
</Directory>
```

You can also restrict access based on user ID or SAF group membership by replacing the `Require` directive in the previous example, as follows:

```
require saf-user USERID
require saf-group GROUPNAME
```

4. Optional: Specify `Require saf-user` or `Require saf-group` to restrict access to a specific SAF user or group.

SAF directives

z/OS

These configuration parameters control the System Authorization Facility (SAF) feature for IBM HTTP Server. Use the SAF directives to provide IBM HTTP Server with user authentication.

- “AuthSAFAuthoritative directive” on page 215
- “AuthSAFExpiration directive” on page 215
- “AuthSAFExpiredRedirect directive” on page 216
- “AuthSAFReEnter directive” on page 216
- “SAFRunAs directive” on page 217

AuthSAFAuthoritative directive

The AuthSAFAuthoritative directive sets whether authorization is passed to lower-level modules.

Directive	Description
Syntax	AuthSAFAuthoritative on off
Default	on
Context	directory, .htaccess
Module	mod_authnz_saf
Values	on or off

Setting the AuthSAFAuthoritative directive `off` allows for authorization to be passed to lower-level modules (as defined in the `modules.c` files), if there is no user ID or rule matching the supplied user ID. If there is a user ID or rule specified, then the usual password and access checks will be applied and a failure will result in an Authentication Required reply.

If a user ID appears in the database of more than one module, or if a valid Require directive applies to more than one module, then the first module will verify the credentials, and no access is passed on, regardless of the AuthSAFAuthoritative setting.

By default, control is not passed on and an unknown user ID or rule will result in an Authentication Required reply. Not setting it thus keeps the system secure and forces an NCSA compliant behavior.

AuthSAFExpiration directive

The AuthSAFExpiration directive sets the value displayed in the browser prompt. The server sends the value specified for the AuthName directive and this short phrase in an HTTP response header, and then the browser displays them to the user in a password prompt window. The short phrase is subject to the same character limitations as the specified value for the AuthName directive. Therefore, to display a special character in the password prompt window, the server must translate the special character from the EBCDIC CharsetSourceEnc codepage to the ASCII CharsetDefault codepage. For example, if you want to display a lowercase 'a' with umlaut, and the `httpd.conf` file contains the German language EBCDIC codepage "CharsetSourceEnc IBM-1141" and the ASCII codepage "CharsetDefault ISO08859-1", then you must code the phrase using the hex value '43', which translates to the correct ASCII character.

Directive	Description
Syntax	AuthSAFExpiration <i>short_phrase</i>
Default	off
Context	directory, .htaccess
Module	mod_authnz_saf
Values	off or <i>short_phrase</i>

Setting the AuthSAFExpiration directive to a phrase allows IBM HTTP Server to prompt the user to update his SAF password if it expires. When the user enters a valid ID and SAF password but the password has expired, the server will return an Authentication Required reply with a special prompt to allow the user to

update the expired password. The prompt consists of the realm (the value from the AuthName directive) followed by the *short_phrase* value from the AuthSAFExpiration directive.

For example, consider the following configuration:

```
<Location /js>
AuthType basic
AuthName "zwas051_SAF"
AuthBasicProvider saf
Require valid-user
Require saf-group SYS1 WASUSER
AuthSAFExpiration "EXPIRED! oldpw/newpw/newpw"
</Location>
```

If the user attempts to access a file whose URL starts with /js, then the server prompts for a SAF ID and password. The browser will display a prompt containing the realm. The realm is the value from the AuthName directive, which is *zwas051_SAF* in this example.

When the user supplies a valid ID and password, if the password has expired, the server will repeat the prompt, but this time with the value *zwas051_SAF EXPIRED! oldpw/newpw/newpw*. Whatever the prompt, the user must then re-enter the expired password, followed by a slash, the new password, another slash, and the new password again.

If the password update is successful, the server will send another Authentication Required reply with a distinct special prompt. This last interaction is necessary in order to force the browser to understand which password it should cache. The prompt this time will consist of the realm followed by the prompt *Re-enter new password*. In this example, it would be *zwas051_SAF Re-enter new password*.

AuthSAFExpiredRedirect directive

The AuthSAFExpiredRedirect directive specifies a URL that a request should be redirected to if your password is expired when you are using *mod_authnz_saf* for authentication on z/OS.

This is an alternative to using AuthSAFExpiration.

Directive	Description
Syntax	AuthSAFExpiredRedirect <i>url</i>
Default	off
Context	directory, .htaccess
Module	mod_authnz_saf
Values	off or <i>url</i>

AuthSAFReEnter directive

The AuthSAFReEnter directive sets the value appended to realm after a successful password change. For information about coding special characters, see the BAuthSAFExpiration directive.

Directive	Description
Syntax	AuthSAFReEnter <i>short_phrase</i>
Default	Re-enter new password
Context	directory, .htaccess

Directive	Description
Module	mod_authnz_saf
Values	off or <i>short_phrase</i>

Setting the AuthSAFReEnter directive explicitly to a phrase other than “Re-enter new password” allows the administrator to display an alternative message after an expired password has been updated successfully. If AuthSAFEExpiration has been set to off, this directive has no effect.

For example, consider the following configuration:

```
<Location /js>
AuthType basic
AuthName "zwasa051_SAF"
AuthBasicProvider saf
Require saf-user SYSADM USER152 BABAR
AuthSAFEExpiration "EXPIRED! oldpw/newpw/newpw"
AuthSAFReEnter "Enter new password one more time"
</Location>
```

In this example, after the expired password is updated successfully, the server will send another Authentication Required reply with the value from the AuthSAFReEnter directive. This last interaction is necessary in order to force the browser to understand which password it should cache. The prompt this time will consist of the realm followed by a special phrase. In this example, it would be zwasa051_SAF Enter new password one more time.

SAFRunAs directive

The SAFRunAs directive sets the SAF user ID under which a request will be served.

Directive	Description
Syntax	SAFRunAs <i>value</i>
Default	off
Context	directory, .htaccess
Module	mod_authnz_saf

Directive Values

Description

off | %%CLIENT%% | %%CERTIF%% | %%CERTIF_REQ%% | <surrogate ID>

Off: The server will run the request under the Web server user ID.

%%CLIENT%%: The server will run the request under the ID supplied in the Authorization request header. Generally, the user supplies the ID and password in a pop-up window on the browser, and the browser creates the header. Requires that SAF is configured to authenticate the URL.

%%CERTIF%%: The server will run the request under the ID associated with the SSL client certificate in SAF. If there is no SSL certificate or if the SSL certificate has not been associated with an ID in SAF, the processing will continue as if %%CLIENT%% had been coded. Does not require SAF authn or authz to be configured.

%%CERTIF_REQ%%: The server will run the request under the ID associated with the SSL client certificate in SAF. If there is no SSL certificate, or if the SSL certificate has not been associated with an ID in SAF, the server will not allow access. Does not require SAF authn or authz to be configured.

%%CERTIF%% /prefix: The server changes the threads identity to the SSL client authentication provided identity for URLs under /prefix.

gotcha:

- This syntax is valid only in global and <virtualHost> context.
- No other values are permitted as the second argument.
- The server will not switch identities twice during a request if SAFRunAS is also configured using the one-argument version inside of <Location> or <Directory> context.
- This feature can be used in conjunction with "AuthBasicProvider saf".

%%CERTIF%% / ? ?: The server changes early to user's home directory (assumes mod_userdir is configured).

gotcha:

- This syntax is valid only in global and <virtualHost> context.
- No other values are permitted as the second argument.
- The server will not switch identities twice during a request if SAFRunAS is also configured using the one-argument version inside of <Location> or <Directory> context.
- This feature can be used in conjunction with "AuthBasicProvider saf".

IBM HTTP Server can communicate with FastCGI applications using either TCP sockets or UNIX sockets. However, when using SAFRunAs for FastCGI requests, you must use TCP sockets for communication with the application. UNIX sockets that are created for FastCGI applications are accessible by the Web server user ID only. The alternate user ID controlled with the SAFRunAs directive does not have permission to access the UNIX sockets, so requests will fail.

To configure FastCGI to use TCP sockets, define the FastCGI application to the `mod_fastcgi` module using the `FastCGIServer` directive with the `-port` option or using the `FastCGIExternalServer` directive. Dynamic FastCGI servers that you do not configure with the `FastCGIServer` or `FastCGIExternalServer` are not usable with SAFRunAs.

If you do not enable SAFRunAs for FastCGI requests, TCP sockets are not required.

If you want to use SAF for authentication and authorization, consider the following example. This is the most common scenario for SAF users and groups and meets the requirements for web access.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_default_module modules/mod_authz_default.so
...
<Location /saf_protected>
AuthType basic
AuthName x1
AuthBasicProvider saf
# Code "Require valid-user" if you want any valid
# SAF user to be able to access the resource.
Require valid-user
#
# Alternately, you can provide a list of specific SAF users
# who may access the resource.
# Require saf-user USER84 USER85
#
# Alternatively, you can provide a list of specific SAF groups
# whose members may access the resource.
# Require saf-group WASGRP1 WASGRP2
</Location>
```

If you want to use a SAF file for authentication, but use a non-SAF group file for authorization, consider the following example. In this example, users are authenticated using SAF, but authorized using a different mechanism.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_default_module modules/mod_authz_default.so
...
<Location /saf_password>
AuthType basic
AuthName "SAF auth with hfs groupfile"
AuthBasicProvider saf
AuthGroupFile /www/config/foo.grp
# Code "Require file-group" and a list of groups if you want
# a user in any of the groups in the specified group file to be able
# to access the resource.
# Note: Any authorization module, with its standard configuration, can be used here.
Require group admin1 admin2
</Location>
```

If you want to allow access to a user if the user is authorized by SAF or by a group file, consider the following example.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_default_module modules/mod_authz_default.so
...
<Location /either_group>
AuthType basic
AuthName "SAF auth with SAF groups and hfs groupfile"
AuthBasicProvider saf
AuthGroupFile /www/groupfiles/foo.grp
Require saf-group WASGRP
Require saf-group ADMINIS
AuthzGroupFileAuthoritative Off
AuthSAFAuthoritative Off
</Location>
```

If you want to require a request to run using the SAF privileges associated with the authenticated username, consider the following example.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_default_module modules/mod_authz_default.so
...
<Location /runas_admin_bin>
AuthName "SAF RunAs client"
AuthType basic
Require valid-user
AuthBasicProvider saf
SAFRunAs %%CLIENT%%
</Location>
```

If you want to support the changing of expired SAF passwords, consider the following example.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_default_module modules/mod_authz_default.so
...
<Location /custom_password_change>
AuthType basic
AuthName "Support expired PW"
Require valid-user
AuthBasicProvider saf
AuthSAFExpiration "EXPIRED PW: oldpw/newpw/newpw"
AuthSAFReEnter "New PW again:"
</Location>
```

If you want to require a client certificate before a user can access a resource, use the `mod_ibm_ssl` directive. The `mod_authnz_saf` directive is not needed for this configuration. For additional information, see the documentation for the `SSLClientAuth` and `SSLClientAuthRequire` directives.

If you want to use a client certificate to determine the user for whom request processing is performed, consider the following example. If the user does not have a valid certificate, access is denied.

```
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
...
<Location /certificate_required>
SAFRunAs %%CERTIF_REQ%%
</Location>
```

If you want to require a request to run using the SAF privileges associated with a client certificate, but require username and password authentication if the client certificate is not mapped to a SAF user, consider the following example. If the user provides a certificate that SAF can map to a user ID, then the user ID must also pass any Require directives.

```
<Location /certificate_or_basic>
AuthName "SAF RunAs certif"
AuthType basic
Require saf-user USER84 USER103
AuthBasicProvider saf
SAFRunAs %%CERTIF%%
</Location>
```

If you want to require a request to run using the SAF privileges associated with a surrogate ID, consider the following example.

```
<Location /runas_public>
SAFRunAs PUBLIC
# This can be combined with SAF or non-SAF authentication/authorization
</Location>
```

Related reference:

“FastCGI directives” on page 89

These configuration parameters control the FastCGI feature in IBM HTTP Server.

“SSL directives” on page 134

Secure Sockets Layer (SSL) directives are the configuration parameters that control SSL features in IBM HTTP Server.

Related information:

Using the AuthBasicProvider directive for SAF password authentication

Chapter 6. Troubleshooting and support: IBM HTTP Server

This section provides information about how to troubleshoot a problem with IBM HTTP Server.

Troubleshoot problems with IBM HTTP Server, using the problem determination tools provided with the product. For example, you can perform problem determination with IBM HTTP Server, including platform-specific problems and error messages.

Troubleshooting IBM HTTP Server

This section describes how to start troubleshooting IBM HTTP Server.

Procedure

1. Check the error log to help you determine the type of problem. You can find the error logs in the directory specified by the ErrorLog directive in the configuration file. Depending on the operating system, the default directories are:
 - **AIX** /usr/IBM/HTTPServer/logs/error_log
 - **HP-UX** **Linux** **Solaris** /opt/IBM/HTTPServer/logs/error_log
 - **Windows** <server_root>/logs/error.log
 - **z/OS** <server_root>/logs/error_log
2. Check the IBM HTTP Server Diagnostic Tools and Information package at <http://www.ibm.com/support/docview.wss?uid=swg24008409> for additional diagnostic information, as well as MustGather steps for some problems.
3. Check the IBM HTTP Server support page at <http://www.ibm.com/software/webservers/httpservers/support/> for technotes on a variety of topics.
4. Ensure that you are running with the current level of fixes for your release of IBM HTTP Server. The problem may already be resolved. Find the IBM HTTP Server recommended updates page is at <http://www.ibm.com/support/docview.wss?rs=177&context=SSEQTJ&uid=swg27005198>.

Known problems on Windows platforms

Windows

This topic contains troubleshooting information about known problems on Windows platforms.

Problems when the IBM HTTP Server runs on the same system as a Virtual Private Networking Client

A problem occurs when the IBM HTTP Server runs on a system, along with a Virtual Private Networking client, for example, Aventail Connect. You can experience the following problem, or see the following error message:

- The IBM HTTP Server does not start - see Apache HTTP Server - Frequently asked questions.
- The IBM HTTP Server does not start. The error log contains the following message:

"[crit] (10045) The attempted operation is not supported for the type of object referenced: Parent: WSADuplicateSocket failed for socket ###"

Aventail Connect is a Layered Service Provider (LSP) that inserts itself, as a shim, between the Winsock 2 API and the Windows native Winsock 2 implementation. The Aventail Connect shim does not implement WSADuplicateSocket, the cause of the failure. The shim is not unloaded when Aventail Connect is shut down.

Fix the problem by doing one of the following:

- Explicitly unloading the shim
- Rebooting the machine
- Temporarily removing the Aventail Connect V3.x shim

An application start-up error occurs for some IBM HTTP Server and web server plug-in components on Windows operating systems

A message occurs for some IBM HTTP Server and web server plug-in components on Windows operating systems. The message indicates that an application has failed to start because its side-by-side configuration is incorrect. When Secure Sockets Layer (SSL) is configured in either IBM HTTP Server or the web server plug-in, the web server fails to load.

Additionally, the *<ihsinst>\bin\gskver* and *<ihsinst>\bin\gskcapicmd* programs fail with the same error. These two programs are part of the Global Security Kit (GSKit) certificate management tools.

These programs fail with the following error message: The application has failed to start because its side-by-side configuration is incorrect. Please see the application event log for more detail.

In the application event log, the following event is logged:

Activation context generation failed for "_IHS_install_path_\gsk8\bin\gsk8ver.exe". Dependent Assembly Microsoft.VC90.CRT,processorArchitecture="x86",publicKeyToken="1fc8b3b9a1e18e3b",type="win32", version="9.0.21022.8" could not be found. Please use sxstrace.exe for detailed diagnosis.

Fix the problem by installing the Microsoft Visual C++ 2008 Redistributable Package (x86), available from <http://www.microsoft.com/downloads/details.aspx?familyid=9b2da534-3e03-4391-8a4d-074b9f2bc1bf>. You can also search for the *vc redistrib_x86.exe* file on the Microsoft website. If you are using a 64-bit web server plug-in, also install the 64-bit redistributable package.

Note:

- The installation process checks to see if the Microsoft package is installed. If it is not, you receive the following message. The installation package IBM HTTP Server V8.5 requires components supplied by other packages. To fix the issue, either install the required components or deselect the installation package. The required components may be supplied by the following installation packages: Package: Microsoft Visual C++ 2008 Redistributable Package.
- Microsoft Visual C++ 2010 Redistributable Package cannot be used with this version of the product.

The 32 bit version of the Microsoft Visual C++ 2008 Redistributable Package is required to install the IBM HTTP Server.

Known problems on z/OS platforms

This topic contains troubleshooting information for known problems on z/OS platforms.

MEMLIMIT parameter must be set for the IBM HTTP Server address spaces

The MEMLIMIT parameter can be set on a system-wide basis (in the SMFPRMxx parmlib member) or in the OMVS segment of the server ID for each IBM HTTP Server instance. See the z/OS V1R8.0 MVS Extended Addressability Guide for more information. For recommended MEMLIMIT values, see “Performing required z/OS system configurations” on page 65.

If you do not set the MEMLIMIT parameter, the Web server will not start, and one of the following console messages might result:

- ABEND=S000 U4093 REASON=00000224
- no output from bin/apachectl -v
- bin/ab returns "Killed"

To determine if any 64-bit programs run on this system, run the following command from a shell prompt: /bin/localedef64.

Expected output:

```
# /bin/localedef64
EDC4175 40 Missing output locale name.
```

Example of a failure:

```
# /bin/localedef64
Killed
```

To resolve this problem for IBM HTTP Server, which is an AMODE64 application, the MEMLIMIT must be changed from the system default of 0.

Integrated Cryptographic Services Facility (ICSF) is not enabled for AMODE64

z/OS V1R6 might need ICSF 64-bit Virtual Support to use ICSF cryptographic hardware. To issue messages on ICSF status, GSK_SSL_HW_DETECT_MESSAGE=1 is set in bin/envvars.

If ICSF is not enabled for AMODE64, the GSK_SSL_HW_DETECT_MESSAGE will result in the following message logged to the error log at startup:

```
System SSL: ICSF services are not available
```

Known problems with hardware cryptographic support on AIX

Distributed operating systems

This topic contains troubleshooting information for known problems with the cryptographic hardware on AIX.

AIX

You must install the bos.pkcs11 package to get the PKCS11 module, and to initialize the device on AIX.

An added update to the bos.pkcs11 package fixed a forking problem. Obtain the most recent copy of the bos.pkcs11 package from the IBM PSeries Support Site, to ensure you have this fix.

If you are having problems using the IBM eBusiness Cryptographic Accelerator Device with IBM HTTP Server, do the following:

1. Reboot the machine.
2. Kill **pkcsslotd** and the shared memory that it created. To determine the shared memory that was created, type `ipcs -a`. Find the segment with size 270760 to determine the memory segment that was created by **pkcsslotd**.
3. Export `EXPSHM=ON`.
4. Start the pkcs11 process: `/etc/rc.pkcs11`
5. Restart IBM HTTP Server: `./apachectl start`

Symptoms of poor server response time

Distributed operating systems z/OS

If you notice that server CPU utilization appears low, but client requests for static pages take a long time to service, your server may be running out of server threads to handle requests.

This situation results when you have more inbound requests than you have Apache threads to handle those requests. New connections queue in the TCP/IP stack listen queue and wait for acceptance from an available thread. As a thread becomes available, it accepts and handles a connection off of the listen queue. Connections can take a long time to reach the beginning of the listen queue. When this condition occurs, the following message will appear in the error log:

- **AIX** **HP-UX** **Linux** **Solaris** **z/OS**
Server reached MaxClients setting, consider raising the MaxClients setting
- **Windows**
Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting

Hints and tips for managing IBM HTTP Server using the administrative console

Distributed operating systems z/OS

This topic contains helpful tips on using the WebSphere Application Server administrative console for managing the following operations for IBM HTTP Server: Starting, stopping, viewing log files, editing configuration files, and propagating the plug-in configuration file.

Administering IBM HTTP Server with the administrative console using the node agent and deployment manager:

- The following list describes hints and tips on starting, stopping, and obtaining status for IBM HTTP Server using the administrative console.
 - **Windows** The IBM HTTP Server you are managing must be installed as a service. You must install IBM HTTP Server with log on as system rights.
 - **Windows** When defining a Web server using the administrative console, use the actual service name, instead of the display name. The actual service name will not contain spaces. If you do not do this, you will have problems starting and stopping the service.
 - Status is obtained using the Web server host name and port that you have defined. You do not use the remote administration port.

- If you have problems starting and stopping IBM HTTP Server, check the WebSphere console logs (trace).
- If you have problems starting and stopping IBM HTTP Server using nodeagent, you can try to start and stop the server by setting up the managed profile and issuing the **startserver** <IBM HTTP Server> **-nowait -trace** command and check the startServer.log file for the IBM HTTP Server specified.
- If communication between the administrative console and the Web server is through a firewall, then you must define the Web Server port to the firewall program.
- The following list describes hints and tips for viewing log files, editing configuration files and propagating the plug-in configuration file:
 - Access to files is controlled by AdminAllowDirective in the admin.conf file. Access is granted to the conf and logs directory from the IBM HTTP Server installation directory. If you are reading or writing plug-in configuration or trace files, you must add an entry to the admin.conf file to allow access there.
 - Always back up the configuration file. It is possible on the upload of the configuration file, information will be lost.

Distributed operating systems **Administering IBM HTTP Server with the administrative console using the IBM HTTP Server administration server:**

- The following list describes hints and tips on starting, stopping, and obtaining status for IBM HTTP Server using the administrative console.
 - **Windows** The IBM HTTP Server you are managing must be installed as a service.
 - **Windows** When defining a Web server using the administrative console, use the actual service name, instead of the display name. The actual service name will not contain spaces. If you do not do this, you will have problems starting and stopping the service on the Windows 2003 operating system.
 - Status is obtained using the Web server host name and port that you have defined. You do not use the remote administration port.
 - If you have problems starting and stopping IBM HTTP Server, check the WebSphere console logs (trace) and check the admin_error.log file.
 - The administration server should be started as root.
 - If communication between the administrative console and the administration server is through a firewall, you must enable the administration server port (default 8008).
 - If communication between the administrative console and the Web server is through a firewall, then you must define the Web Server port to the firewall program.
- The following list describes hints and tips for viewing log files, editing configuration files and propagating the plug-in configuration file:
 - **AIX** **HP-UX** **Linux** **Solaris** File permissions must be correct in order to transfer a file. The **setupadm** script is provided to set appropriate file permissions.
The setupadm script should be run prior to starting the administration server. This script will setup file permission and update the User ID and Group ID directives in the admin.conf file. The User ID and Group ID created through the setupadm script are UNIX IDs that must correspond to the admin.conf directives: User and Group.

- Access to files is controlled by AdminAllowDirective in the admin.conf file. Access is granted to the conf and logs directory from the IBM HTTP Server installation directory. If you are reading or writing plug-in configuration or trace files, you must add an entry to the admin.conf file to allow access there.
- Always back up the configuration file. It is possible on the upload of the configuration file, information will be lost.

Could not connect to IBM HTTP Server administration server error

Distributed operating systems

This topic contains troubleshooting information if you receive an error when attempting to connect to the administration server.

If you get the following error:

"Could not connect to IHS Administration server error"

when you are managing an IBM HTTP Server using the WebSphere administrative console, try one of the following:

- Verify that the IBM HTTP Server administration server is running.
- Verify that the Web server hostname and port that is defined in the WebSphere administrative console matches the IBM HTTP Server administration host name and port.
- Verify that the firewall is not preventing you from accessing the IBM HTTP Server administration server from the WebSphere administrative console.
- Verify that the user ID and password that is specified in the WebSphere administrative console, under remote managed, is created in the admin.passwd file, using the **htpasswd** command.
- If trying to connect securely, verify that you export the IBM HTTP Server administration server keydb personal certificate into the WebSphere key database as a signer certificate. This key database will be specified by the `com.ibm.ssl.trustStore` in the `sas.client.props` file in the profile your console is running in. This is mainly for self-signed certificates.
- If you still have problems, check the IBM HTTP Server `admin_error.log` file and the WebSphere Application Server logs (`trace.log`) to see if problem can be determined.

Experiencing an IBM HTTP Server Service logon failure on Windows operating systems

Windows

When installing the IBM HTTP Server, prompts appear for a login ID and password. The ID you select must have the capability to log on as a service.

About this task

If you get an error when you try to start the IBM HTTP Server Service, indicating a failure to start as a service, try one of the following:

Procedure

1. Click **Start > Programs > Administrative Tools > User Manager**.
2. Select the user from the User Manager list.
3. Click **Policies > User Rights**.
4. Select the **Show Advanced User Rights** check box.
5. Click **Log on as a Service**, from the drop-down menu.
or
 - a. Click **Start > Settings > Control Panel**.
 - b. Open Administrative Tools.
 - c. Open Services. The local user you select is created in Local Users and Groups, under Computer Management.
 - d. Click **Service > Actions > Properties**.
 - e. Choose the Log on tab.
 - f. Select this account option and click **Browse**, to select the user to associate with the service.

What to do next

If you get the following error when you try to start the IBM HTTP Server Service: Windows could not start the IBM HTTP Server on Local Computer. For more information, review the Event Log. If this is a non-Microsoft service, contact the service vendor, and refer to service-specific error code 1.

complete the following steps:

1. Check the IBM HTTP Server `<install_root>/logs/error.log` file for a specific error.
2. If there is no error in the `<install_root>/logs/error.log` file, try starting IBM HTTP Server from a command prompt by running the `<install_root>/bin/httpd.exe` command.
3. If the `<install_root>/logs/error.log` file indicates that there was a problem loading the WebSphere Application Server plugin module, check the `http_plugin.log` file for the error.

Viewing error messages for a target server that fails to start

Distributed operating systems

z/OS

If you encounter an error starting a target server, you can view the error message in the server logs.

About this task

If the target Web server fails to start, a message might appear on the WebSphere Application Server administrative console that indicates that the Web server cannot be started and to view the error messages in the server logs for further details. The types of errors that can result are:

- errors due to caching problems
- errors due to configuration problems
- errors due to SSL handshake failures
- errors due to SSL initialization problems
- errors due to I/O failures

- errors due to Secure Sockets Layer (SSL) stash utility problems

Cache messages

Distributed operating systems z/OS

This topic contains error messages that might result due to caching problems and provides a solution to help you troubleshoot the problem.

The following messages are displayed due to caching problems:

- Message: **SSL0600E: Unable to connect to session ID cache**
 - Reason: The server cannot connect to the Session ID caching daemon.
 - Solution: Verify that the daemon successfully started.
- Message: **SSL0601E: Session ID cache daemon process <process-id> exited with exit code <exit-code>; restarting**
 - Reason: If the value of <exit-code> is 0, the session ID cache daemon (sidd) received the SIGTERM signal. Other exit codes are not expected. Sidd automatically restarted.
 - Solution: If the value of <exit-code> is 0 and IBM HTTP Server did not stop or restart, verify that locally installed CGI scripts, scheduled operating system tasks, or other monitoring software cannot send SIGTERM to sidd.
- Message: **SSL0602E: Session ID cache daemon process <process-id> exited with terminating signal <signal-number>; restarting**
 - Reason: The session ID cache daemon (sidd) received a signal other than SIGTERM was received by the session ID cache daemon (sidd), which caused it to exit. Sidd automatically restarted.
 - Solution: If the value of <exit-code> is 0 and IBM HTTP Server did not stop or restart, verify that locally installed CGI scripts, scheduled operating system tasks, or other monitoring software cannot send the signal to sidd.
- Message: **SSL0603E: Session ID cache daemon process <process-id> exited with exit code<exit-code>; not restarting; check sidd configuration or enable sidd error log with SSLCacheErrorLog**
 - Reason: The session ID cache daemon (sidd) did not initialize. The following possible exit code values might be displayed:

Value	Reason
2	Log files could not be opened. The SSLCacheTraceLog or the SSLCacheErrorLog directive is not valid.
3	The AF_UNIX socket cannot be initialized. Use the SSLCachePortFilename directive to specify a different socket for the session ID cache daemon.
4	Sidd cannot switch to the configured user and group. Verify the values for the user and group directives.

- Solution: Provide a valid value for the directives and restart IBM HTTP Server.

Configuration messages

Distributed operating systems z/OS

This topic contains error messages that might result due to configuration problems and provides solutions to help you troubleshoot these problems.

The following messages appear due to configuration problems:

- Message: **SSL0300E: Unable to allocate terminal node.**
- Message: **SSL0301E: Unable to allocate string value in node.**
- Message: **SSL0302E: Unable to allocate non terminal node.**
- Message: **SSL0303E: Syntax Error in SSLClientAuthGroup directive.**
- Message: **SSL0304E: Syntax Error in SSLClientAuthRequire directive.**
- Message: **SSL0307E: Invalid token preceding NOT or !**
- Message: **SSL0308E: A group is specified in SSLClientAuthRequire but no groups are specified.**
- Message: **SSL0309E: The group <group> is specified in SSLClientAuthRequire is not defined.**
- Message: **SSL0310I: Access denied to object due to invalid SSL version <version>, expected <version>.**
- Message: **SSL0311E: Unable to get cipher in checkBanCipher.**
- Message: **SSL0312I: Cipher <cipher> is in ban list and client is forbidden to access object.**
- Message: **SSL0313E: Fell through to default return in checkCipherBan.**
- Message: **SSL0314E: Cipher is NULL in checkRequireCipher.**
- Message: **SSL0315E: Cipher <cipher> used is not in the list of required ciphers to access this object.**
- Message: **SSL0316E: Fell through to default return in checkCipherRequire.**
- Message: **SSL0317E: Unable to allocate memory for fake basic authentication username.**
- Message: **SSL0318E: Limit exceeded for specified cipher specs, only 64 total allowed.**
 - Reason: The number of ciphers configured using the SSLCipherSpec directive exceeds the maximum allowed of 64.
 - Solution: Check for duplicate SSLCipherSpec directives.
- Message: **SSL0319E: Cipher Spec <cipher> is not supported by this GSK library.**
 - Reason: The cipher is not a valid cipher for use with the installed SSL libraries.
 - Solution: Check that a valid cipher value was entered with the SSLCipherSpec directive.
- Message: **SSL0320I: Using Version 2|3 Cipher: <cipher>.**
 - Reason: This is an informational message listing the ciphers used for connections to this virtual host.
 - Solution: None.
- Message: **SSL0321E: Invalid cipher spec <cipher>.**
 - Reason: The cipher is not a valid cipher.
 - Solution: Check the documentation for a list of valid cipher specs.
- Message: **SSL0322E: Cipher Spec <cipher> is not valid.**
 - Reason: The cipher is not a valid cipher.
 - Solution: Check the documentation for a list of valid cipher specs.
- Message: **SSL0323E: Cipher Spec <cipher> has already been added.**

- Reason: A duplicate SSLCipherSpec directive has been encountered.
- Solution: This instance of the directive is ignored and should be removed from the configuration file.
- Message: **SSL0324E: Unable to allocate storage for cipher specs.**
 - Reason: The server could not allocate memory needed to complete the operation.
 - Solution: Take action to free up some additional memory. Try reducing the number of threads or processes running, or increasing virtual memory.
- Message: **SSL0325E: Cipher Spec <cipher> has already been added to the v2 | v3 ban | require list.**
 - Reason: A duplicate cipher was specified on the SSLCipherBan directive.
 - Solution: This instance of the directive is ignored and should be removed from the configuration file.
- Message: **SSL0326E: Invalid cipher spec <cipher> set for SSLCipherBan | SSLCipherRequire.**
 - Reason: The cipher is not a valid cipher.
 - Solution: Check the documentation for a list of valid cipher specs.
- Message: **SSL0327E: Invalid value for sslv2timeout | sslv3timeout, using default value of nn seconds.**
 - Reason: The timeout value specified is not in the valid range.
 - Solution: Check the documentation for the proper range of values.
- Message: **SSL0328W: Invalid argument for SSLClientAuth: <args>. CRL can not be turned on unless Client Authentication is on.**
- Message: **SSL0329W: Invalid argument for SSLClientAuth: <args>. If a second argument is entered it must be CRL. CRL cannot be turned on unless client authentication is on.**
- Message: **SSL0330W: Invalid argument for SSLClientAuth: <args>. If a second value is entered it must be crl.**
- Message: **SSL0331W: Invalid argument for SSLClientAuth: <args>. The first value must be 0, 1, 2 none, optional, or required.**
- Message: **SSL0332E: Not enough arguments specified for SSLClientAuthGroup.**
- Message: **SSL0333E: No parse tree created for <parm>.**
 - Reason: An error occurred processing the SSLClientAuthRequire directive.
 - Solution: Check for other error messages. Enable tracing of Client Authentication by adding the directive SSLClientAuthRequireTraceOn to the configuration file.
- Message: **SSL0334E: Function ap_make_table failed processing label <certificate>.**
- Message: **SSL0337E: OCSP is not supported with this level of GSKit**
 - Reason: OCSP support requires GSKit 7.0.4.14 or higher
 - Solution: Upgrade the level of GSKit on the system to 7.0.4.14 or higher

Handshake messages

Distributed operating systems z/OS

This topic contains error messages that might result due to SSL handshake failures and provides solutions to help you troubleshoot these problems.

The following messages display due to handshake failures:

- Message: **SSL0192W: IBM HTTP Server is configured to permit client renegotiation which is vulnerable to man-in-the-middle attacks**
<servername:port>
 - Reason: IBM HTTP Server is configured to allow client handshake renegotiation using the SSLInsecureRenegotiation directive. This configuration is vulnerable to man-in-the middle attacks. Use this configuration only if it is necessary for your client and be aware of the risk. For more information about the exposure, refer to the public documentation about CVE-2009-3555.
 - Solution: Remove the SSLInsecureRenegotiation directive or set the directive to OFF to avoid the vulnerability. If proprietary clients require SSL renegotiation to function, update these clients to establish new connections.
- Message: **SSL0193W: Error setting GSK_NO_RENEGOTIATION to <GSK_TRUE | GSK_FALSE> <errorcode>**
 - Reason: An error occurred when the server attempted to disable client renegotiation. This setting is the default value. However, this value is also set if you specify the SSLInsecureRenegotiation directive with an OFF value.
 - Solution: Report this problem to IBM Support.
- Message: **SSL0196I: Security library does not support GSK_SESSION_RESET_CALLBACK, rejecting insecure SSL client renegotiation by monitoring SIDs**
 - Reason: When the server attempted to disable client renegotiation, it was determined that the security library on this system does not support GSK_SESSION_RESET_CALLBACK. It will be configured to reject insecure SSL client renegotiation using an alternate mechanism of monitoring SIDs.
 - Solution: This informational message does not indicate a failure, but it reports a configuration condition. An action is not necessary. You can upgrade to a newer z/OS security library that includes support for GSK_SESSION_RESET_CALLBACK or for disabling SSL client renegotiation.
- Message: **SSL0197I: Configured security library to reject insecure SSL client renegotiation.**
 - Reason: The security library has been successfully configured to reject client renegotiation.
 - Solution: This informational message does not indicate a failure, but it reports a particular configuration setting. An action is not necessary.
- Message: **SSL0198I: System is running without a security library capable of directly rejecting insecure SSL client renegotiation. Aborting HTTPS requests that span SSL sessions**
 - Reason: While the server attempted to disable client renegotiation, it was determined that the security library on this system does not support directly rejecting SSL client renegotiation. It will be configured to use an alternate callback mechanism.
 - Solution: This informational message does not indicate a failure, but it reports a configuration condition. An action is not necessary. For z/OS systems, upgrade to a newer security library that includes support for GSK_SESSION_RESET_CALLBACK or for disabling SSL client renegotiation. For distributed systems, upgrade to GSKit Version 7.0.4.27 or later.
- Message: **SSL0200E: Handshake Failed, <code>.**
 - Reason: The handshake failed when the SSL library returned an unknown error.
 - Solution: Report this problem to IBM Support.

- Message: **SSL0201E: Handshake Failed, Internal error - Bad handle.**
 - Reason: An internal error has occurred.
 - Solution: Report this problem to IBM Support.
- Message: **SSL0202E: Handshake Failed, The GSK library unloaded.**
 - Reason: A call to the GSKit function failed because the dynamic link library unloaded (Windows operating systems only).
 - Solution: Shut down the server and restart.
- Message: **SSL0203E: Handshake Failed, GSK internal error.**
 - Reason: The communication between client and the server failed due to an error in the GSKit library.
 - Solution: Retry connection from the client. If the error continues, report the problem to IBM Support.
- Message: **SSL0204E: Handshake Failed, Internal memory allocation failure.**
 - Reason: The server could not allocate memory needed to complete the operation.
 - Solution: Take action to free up some additional memory. Try reducing the number of threads or processes running, or increasing virtual memory.
- Message: **SSL0205E: Handshake Failed, GSK handle is in an invalid state for operation.**
 - Reason: The SSL state for the connection is invalid.
 - Solution: Retry connection from the client. If the error continues, report the problem to IBM Support.
- Message: **SSL0206E: Handshake Failed, Key-file label not found**
 - Reason: The label specified for the SSLServerCert directive was not found in the key database (KDB) file specified for the KeyFile directive.
 - Solution: Specify a value for the SSLServerCert directive that corresponds to a personal certificate available in the KDB file specified for the KeyFile directive
- Message: **SSL0207E: Handshake Failed, Certificate is not available.**
 - Reason: The client did not send a certificate.
 - Solution: Set client authentication to optional if a client certificate is not required. Contact the client to determine why it is not sending an acceptable certificate.
- Message: **SSL0208E: Handshake Failed, Certificate validation error.**
 - Reason: The received certificate failed one of the validation checks.
 - Solution: Use another certificate. Contact IBM Support to determine why the certificate failed validation.
- Message: **SSL0209E: Handshake Failed, ERROR processing cryptography.**
 - Reason: A cryptography error occurred.
 - Solution: None. If the problem continues, report it to IBM Support.
- Message: **SSL0210E: Handshake Failed, ERROR validating ASN fields in certificate.**
 - Reason: The server was not able to validate one of the ASN fields in the certificate.
 - Solution: Try another certificate.
- Message: **SSL0211E: Handshake Failed, ERROR connecting to LDAP server.**
 - Reason: The Web server failed to connect to the CRL LDAP server.
 - Solution: Verify that the values entered for the SSLCRLHostname and SSLCRLPort directives are correct. If access to the CRL LDAP server requires

authentication, is the SSLCRLUserID directive coded and was the password added to the stash file pointed to by the SSLStashfile directive.

- Message: **SSL0212E: Handshake Failed, Internal unknown error.**
 - Reason: An unknown error has occurred in the SSL library.
 - Solution: Report the problem to IBM Support.
- Message: **SSL0213E: Handshake Failed, Open failed due to cipher error.**
 - Reason: An unknown error has occurred in the SSL library.
 - Solution: Report the problem to IBM Support.
- Message: **SSL0214E: Handshake Failed, I/O error reading key file.**
 - Reason: The server could not read the key database file.
 - Solution: Check file access permissions and verify the Web server user ID is allowed access.
- Message: **SSL0215E: Handshake Failed, Key file has an invalid internal format. Recreate key file.**
 - Reason: Key file has an invalid format.
 - Solution: Recreate key file.
- Message: **SSL0216E: Handshake Failed, Key file has two entries with the same key. Use IKEYMAN to remove the duplicate key.**
 - Reason: Two identical keys exist in key file.
 - Solution: Use IKEYMAN to remove duplicate key.
- Message: **SSL0217E: Handshake Failed, Key file has two entries with the same label. Use IKEYMAN to remove the duplicate label.**
 - Reason: A second certificate with the same label was placed in the key database file.
 - Solution: Use IKEYMAN to remove duplicate label.
- Message: **SSL0218E: Handshake failed, Either the key file has become corrupted or the password is incorrect.**
 - Reason: The key file password is used as an integrity check and the test failed. Either the key database file is corrupted, or the password is incorrect.
 - Solution: Use IKEYMAN to stash the key database file password again. If that fails, recreate the key database.
- Message: **SSL0219E: SSL Handshake Failed, Either the default key in the keyfile has an expired certificate or the keyfile password expired. Use iKeyman to renew or remove certificates that are expired or to set a new keyfile password.**
 - Reason: Either the default key in the keyfile has an expired certificate or the keyfile password expired.
 - Solution: Use iKeyman to renew or remove certificates that are expired or to set a new keyfile password.
- Message: **SSL0220E: Handshake Failed, There was an error loading one of the GSKdynamic link libraries. Be sure GSK was installed correctly.**
 - Reason: Opening the SSL environment resulted in an error because one of the GSKdynamic link libraries could not load.
 - Solution: Contact Support to make sure the GSKit is installed correctly.
- Message: **SSL0221E: Handshake Failed, Either the certificate has expired or the system clock is incorrect.**
 - Reason: Either the certificate expired or the system clock is incorrect.

- Solution: Use the key management utility (iKeyman) to recreate or renew your server certificate or change the system date to a valid date.
- Message: **SSL0222W: Handshake failed, no ciphers specified.**
 - Reason: SSLV2 and SSLV3 are disabled.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0223E: Handshake Failed, No certificate.**
 - Reason: The client did not send a certificate.
You can also see this message when your keyfile does not have a default certificate specified and you have not specified an SSLServerCert directive. It will pass initialization but fail at connection (handshake) time.
 - Solution: Set client authentication to optional if a client certificate is not required. Contact the client to determine why it is not sending a certificate.
- Message: **SSL0224E: Handshake failed, Invalid or improperly formatted certificate.**
 - Reason: The client did not specify a valid certificate.
 - Solution: Client problem.
- Message: **SSL0225E: Handshake Failed, Unsupported certificate type.**
 - Reason: The certificate type received from the client is not supported by this version of IBM HTTP Server SSL.
 - Solution: The client must use a different certificate type.
- Message: **SSL0226I: Handshake Failed, I/O error during handshake.**
 - Reason: The communication between the client and the server failed. This is a common error when the client closes the connection before the handshake has completed.
 - Solution: Retry the connection from the client.
- Message: **SSL0227E: Handshake Failed, Specified label could not be found in the key file.**
 - Reason: Specified key label is not present in key file.
 - Solution: Check that the SSLServerCert directive is correct, if coded, and that the label is valid for one of the keys in the key database.
- Message: **SSL0228E: Handshake Failed, Invalid password for key file.**
 - Reason: The password retrieved from the stash file could not open the key database file.
 - Solution: Use IKEYMAN to open the key database file and recreate the password stash file. This problem can also result from a corrupted key database file. Creating a new key database file may resolve the problem.
- Message: **SSL0229E: Handshake Failed, Invalid key length for export.**
 - Reason: In a restricted cryptography environment, the key size is too long to be supported.
 - Solution: Select a certificate with a shorter key.
- Message: **SSL0230I: Handshake Failed, An incorrectly formatted SSL message was received.**
- Message: **SSL0231W: Handshake Failed, Could not verify MAC.**
 - Reason: The communication between the client and the server failed.
 - Solution: Retry the connection from the client.
- Message: **SSL0232W: Handshake Failed, Unsupported SSL protocol or unsupported certificate type.**

- Reason: The communication between the client and the server failed because the client is trying to use a protocol or certificate which the IBM HTTP Server does not support.
- Solution: Retry the connection from the client using an SSL Version 2 or 3, or TLS 1 protocol. Try another certificate.
- Message: **SSL0233W: Handshake Failed, Invalid certificate signature.**
- Message: **SSL0234W: Handshake Failed, The certificate sent by the peer expired or is invalid.**
 - Reason: The partner did not specify a valid certificate. The server is acting as a reverse proxy to an SSL URL and the `_server_` cert could not be validated.
 - Solution: Partner problem. If this occurs during an SSL Proxy connection, the remote SSL server sent a bad certificate to IBM HTTP Server. Check the certificate and certificate authority chain at the other end of the SSL connection. For more information, see “Securing with SSL communications” on page 118.
- Message: **SSL0235W: Handshake Failed, Invalid peer.**
- Message: **SSL0236W: Handshake Failed, Permission denied.**
- Message: **SSL0237W: Handshake Failed, The self-signed certificate is not valid.**
- Message: **SSL0238E: Handshake Failed, Internal error - read failed.**
 - Reason: The read failed.
 - Solution: None. Report this error to IBM Support.
- Message: **SSL0239E: Handshake Failed, Internal error - write failed.**
 - Reason: The write failed.
 - Solution: None. Report this error to IBM Support.
- Message: **SSL0240I: Handshake Failed, Socket has been closed.**
 - Reason: The client closed the socket before the protocol completed.
 - Solution: Retry connection between client and server.
- Message: **SSL0241E: Handshake Failed, Invalid SSLV2 Cipher Spec.**
 - Reason: The SSL Version 2 cipher specifications passed into the handshake were invalid.
 - Solution: Change the specified Version 2 cipher specs.
- Message: **SSL0242E: Handshake Failed, Invalid SSLV3 Cipher Spec.**
 - Reason: The SSL Version 3 cipher specifications passed into the handshake were invalid.
 - Solution: Change the specified Version 3 cipher specs.
- Message: **SSL0243E: Handshake Failed, Invalid security type.**
 - Reason: There was an internal error in the SSL library.
 - Solution: Retry the connection from the client. If the error continues, report the problem to IBM Support.
- Message: **SSL0245E: Handshake Failed, Internal error - SSL Handle creation failure.**
 - Reason: There was an internal error in the security libraries.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0246E: Handshake Failed, Internal error - GSK initialization has failed.**
 - Reason: An error in the security library has caused SSL initialization to fail.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0247E: Handshake Failed, LDAP server not available.**

- Reason: Unable to access the specified LDAP directory when validating a certificate.
- Solution: Check that the SSLCRLHostname and SSLCRLPort directives are correct. Make sure the LDAP server is available.
- Message: **SSL0248E: Handshake Failed, The specified key did not contain a private key.**
 - Reason: The key does not contain a private key.
 - Solution: Create a new key. If this was an imported key, include the private key when doing the export.
- Message: **SSL0249E: Handshake Failed, A failed attempt was made to load the specified PKCS#11 shared library.**
 - Reason: An error occurred while loading the PKCS#11 shared library.
 - Solution: Verify that the PKCS#11 shared library specified in the SSLPKCSDriver directive is valid.
- Message: **SSL0250E: Handshake Failed, The PKCS#11 driver failed to find the token label specified by the caller.**
 - Reason: The specified token was not found on the PKCS#11 device.
 - Solution: Check that the token label specified on the SSLServerCert directive is valid for your device.
- Message: **SSL0251E: Handshake Failed, A PKCS#11 token is not present for the slot.**
 - Reason: The PKCS#11 device has not been initialized correctly.
 - Solution: Specify a valid slot for the PKCS#11 token or initialize the device.
- Message: **SSL0252E: Handshake Failed, The password/pin to access the PKCS#11 token is either not present, or invalid.**
 - Reason: Specified user password and pin for PKCS#11 token is not present or invalid.
 - Solution: Check that the correct password was stashed using the SSLStash utility and that the SSLStashfile directive is correct.
- Message: **SSL0253E: Handshake Failed, The SSL header received was not a properly SSLV2 formatted header.**
 - Reason: The data received during the handshake does not conform to the SSLV2 protocol.
 - Solution: Retry connection between client and server. Verify that the client is using HTTPS.
- Message: **SSL0254E: Internal error - I/O failed, buffer size invalid.**
 - Reason: The buffer size in the call to the I/O function is zero or negative.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0255E: Handshake Failed, Operation would block.**
 - Reason: The I/O failed because the socket is in non-blocking mode.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0256E: Internal error - SSLV3 is required for reset_cipher, and the connection uses SSLV2.**
 - Reason: A reset_cipher function was attempted on an SSLV2 connection.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0257E: Internal error - An invalid ID was specified for the gsk_secure_soc_misc function call.**
 - Reason: An invalid value was passed to the gsk_secure_soc_misc function.

- Solution: None. Report this problem to IBM Support.
- Message: **SSL0258E: Handshake Failed, The function call, <function>, has an invalid ID.**
 - Reason: An invalid function ID was passed to the specified function.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0259E: Handshake Failed, Internal error - The attribute has a negative length in: <function>.**
 - Reason: The length value passed to the function is negative, which is invalid.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0260E: Handshake Failed, The enumeration value is invalid for the specified enumeration type in: <function>.**
 - Reason: The function call contains an invalid function ID.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0261E: Handshake Failed, The SID cache is invalid: <function>.**
 - Reason: The function call contains an invalid parameter list for replacing the SID cache routines.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0262E: Handshake Failed, The attribute has an invalid numeric value: <function>.**
 - Reason: The function call contains an invalid value for the attribute being set.
 - Solution: None. Report this problem to IBM Support.
- Message: **SSL0263W: SSL Connection attempted when SSL did not initialize.**
 - Reason: A connection was received on an SSL-enabled virtual host but it could not be completed because there was an error during SSL initialization.
 - Solution: Check for an error message during startup and correct that problem.
- Message: **SSL0264E: Failure obtaining Cert data for label <certificate>.**
 - Reason: A GSKit error prevented the server certificate information from being retrieved.
 - Solution: Check for a previous error message with additional information.
- Message: **SSL0265W: Client did not supply a certificate.**
 - Reason: A client who connected failed to send a client certificate and the server is configured to require a certificate.
 - Solution: Nothing on the server side.
- Message: **SSL0266E: Handshake failed.**
 - Reason: Could not establish SSL proxy connection.
 - Solution: IBM HTTP Server could not establish a proxy connection to a remote server using SSL.
- Message: **SSL0267E: SSL Handshake failed.**
 - Reason: Timeout on network operation during handshake.
 - Solution: Check client connectivity, adjust TimeOuts.
- Message: **SSL0270I: SSL Handshake Failed, Timeout (dd seconds) occurred before any data received.**
 - Reason: A connection was received on an SSL port, but no data was received from the client before the timeout expired.
 - Solution: If the timeout (set by the Timeout directive) has been reduced from the default value, verify that it is reasonable. If the message occurs intermittently, it is probably normal, due to things like users cancelling page

loads and browser or system crashes. If the message occurs in bursts, it might indicate a denial of service attack in progress.

- Message: **SSL0271I: SSL Handshake Failed, client closed connection without sending any data.**
 - Reason: A connection was received on an SSL port, but the client closed the connection without beginning the handshake.
 - Solution: If the timeout (set by the Timeout directive) has been reduced from the default value, verify that it is reasonable. If the message occurs intermittently, it is probably normal, due to things like users cancelling page loads and browser or system crashes. If the message occurs in bursts, it might indicate a denial of service attack in progress.
- Message: **SSL0272I: SSL Handshake Failed, I/O error before any data received.**
 - Reason: A connection was received on an SSL port, but a network error broke the connection before any data was received from the client.
 - Solution: If the message occurs intermittently, it is probably normal, due to things like users cancelling page loads and browser or system crashes. If the message occurs in bursts, it might indicate a denial of service attack in progress.
- Message: **SSL0273I: Non-SSL request received on connection configured for SSL**
 - Reason: A connection was received on an SSL port, but the data received was not SSL, and looked like a normal non-SSL request.
 - Solution: Verify that the port in question is intended to be configured for SSL. Look for bad links to the page in question that should use https:, but instead use http:.
- Message: **SSL0273I: Non-SSL request received on connection configured for SSL**
 - Reason: A connection was received on an SSL port, but the data received was not SSL, and looked like a normal non-SSL request.
 - Solution: Verify that the port in question is intended to be configured for SSL. Look for bad links to the page in question that should use https:, but instead use http:.
- Message: **SSL0276E: SSL: Unexpected SSL client renegotiation detected, aborting SSL connection.**
 - Reason: SSL client renegotiation was attempted, but the configuration does not allow SSL renegotiation. Thus, the SSL connection was stopped.
 - Solution: Retry the connection between the client and the server. Configure the connection to allow SSL renegotiation only if necessary. Be aware of the risk. If proprietary clients require SSL renegotiation to function, update them to establish new connections.

SSL initialization messages

Distributed operating systems

z/OS

This topic contains error messages that might result due to SSL initialization problems and provides solutions to help you troubleshoot these problems.

The following messages display as a result of initialization problems:

- Message: **SSL0100E: GSK could not initialize, <errorCode>**
 - Reason: Initialization failed when the SSL library returned an unknown error.

- Solution: None. Report this problem to Service.
- Message: **SSL0101E: GSK could not initialize, Neither the password nor the stash file name was specified. Could not open key file.**
 - Reason: The stash file for the key database could not be found or is corrupted.
 - Solution: Use IKEYMAN to open the key database file and recreate the password stash file.
- Message: **SSL0102E: GSK could not initialize, Could not open key file.**
 - Reason: The server could not open the key database file.
 - Solution: Check that the Keyfile directive is correct and that the file permissions allow the Web server user ID to access the file.
- Message: **SSL0103E: Internal error - GSK could not initialize, Unable to generate a temporary key pair.**
 - Reason: GSK could not initialize; Unable to generate a temporary key pair.
 - Solution: Report this problem to Service.
- Message: **SSL0104E: GSK could not initialize, Invalid password for key file.**
 - Reason: The password retrieved from the stash file could not open the key database file.
 - Solution: Use IKEYMAN to open the key database file and recreate the password stash file. This problem could also result from a corrupted key database file. Creating a new key database file may resolve the problem.
- Message: **SSL0105E: GSK could not initialize, Invalid label.**
 - Reason: Specified key label is not present in key file.
 - Solution: Check that the SSLServerCert directive is correct, if coded, and that the label is valid for one of the keys in the key database.
- Message: **SSL0106E: Initialization error, Internal error - Bad handle**
 - Reason: An internal error has occurred.
 - Solution: Report this problem to Service.
- Message: **SSL0107E: Initialization error, The GSK library unloaded.**
 - Reason: A call to the GSKit function failed because the dynamic link library unloaded (Windows only).
 - Solution: Shut down the server and restart.
- Message: **SSL0108E: Initialization error, GSK internal error.**
 - Reason: The communication between client and the server failed due to an error in the GSKit library.
 - Solution: Retry connection from the client. If the error continues, report the problem to Service.
- Message: **SSL0109E: GSK could not initialize, Internal memory allocation failure.**
 - Reason: The server could not allocate memory needed to complete the operation.
 - Solution: Take action to free up some additional memory. Try reducing the number of threads or processes running, or increasing virtual memory.
- Message: **SSL0110E: Initialization error, GSK handle is in an invalid state for operation.**
 - Reason: The SSL state for the connection is invalid.
 - Solution: Retry connection from the client. If the error continues, report the problem to Service.

- Message: **SSL0111E: Initialization error, Key file label not found.**
 - Reason: Certificate or key label specified was not valid.
 - Solution: Verify that the certificate name specified with the SSLServerCert directive is correct or, if no SSLServerCert directive was coded, that a default certificate exists in the key database.
- Message: **SSL0112E: Initialization error, Certificate is not available.**
 - Reason: The client did not send a certificate.
 - Solution: Set Client Authentication to optional if a client certificate is not required. Contact the client to determine why it is not sending an acceptable certificate.
- Message: **SSL0113E: Initialization error, Certificate validation error.**
 - Reason: The received certificate failed one of the validation checks.
 - Solution: Use another certificate. Contact Service to determine why the certificate failed validation.
- Message: **SSL0114E: Initialization error, Error processing cryptography.**
 - Reason: A cryptography error occurred.
 - Solution: None. If the problem continues, report it to Service.
- Message: **SSL0115E: Initialization error, Error validating ASN fields in certificate.**
 - Reason: The server was not able to validate one of the ASN fields in the certificate.
 - Solution: Try another certificate.
- Message: **SSL0116E: Initialization error, Error connecting to LDAP server.**
 - Reason: The Web server failed to connect to the CRL LDAP server.
 - Solution: Verify that the values entered for the SSLCRLHostname and SSLCRLPort directives are correct. If access to the CRL LDAP server requires authentication, is the SSLCRLUserID directive coded and was the password added to the stash file pointed to by the SSLStashfile directive.
- Message: **SSL0117E: Initialization error, Internal unknown error. Report problem to service.**
 - Reason: Initialization error, Internal unknown error. Report problem to service.
 - Solution: Initialization error, Internal unknown error. Report problem to service.
- Message: **SSL0118E: Initialization error, Open failed due to cipher error.**
 - Reason: Report problem to service.
 - Solution: Report problem to service.
- Message: **SSL0119E: Initialization error, I/O error reading keyfile.**
 - Reason: I/O error trying to read SSL keyfile.
 - Solution: Check the file permissions for keyfile.
- Message: **SSL0120E: Initialization error, Keyfile has and invalid internal format. Recreate keyfile.**
 - Reason: Initialization error, the keyfile has an invalid internal format. Recreate the keyfile.
 - Solution: Verify the keyfile is not corrupted.
- Message: **SSL0121E: Initialization error, Keyfile has two entries with the same key. Use Ikeyman to remove the duplicate key.**

- Reason: The keyfile has two entries with the same key. Use Ikeyman to remove the duplicate key.
- Solution: Use Ikeyman to remove the duplicate key.
- Message: **SSL0122E: Initialization error, Keyfile has two entries with the same label. Use Ikeyman to remove the duplicate label.**
 - Reason: The keyfile has two entries with the same label. Use Ikeyman to remove the duplicate label.
 - Solution: Use Ikeyman to remove the duplicate label.
- Message: **SSL0123E: Initialization error, The keyfile password is used as an integrity check. Either the keyfile has become corrupted or the password is incorrect.**
 - Reason: The keyfile password is used as an integrity check. Either the keyfile has become corrupted or the password is incorrect.
 - Solution: Use Ikeyman to verify that the keyfile is valid, check permissions on the stash file, verify passwords.
- Message: **SSL0124E: SSL Handshake Failed, Either the default key in the keyfile has an expired certificate or the keyfile password expired. Use iKeyman to renew or remove certificates that are expired or to set a new keyfile password.**
 - Reason: Either the default key in the keyfile has an expired certificate or the keyfile password expired.
 - Solution: Use iKeyman to renew or remove certificates that are expired or to set a new keyfile password.
- Message: **SSL0125E: Initialization error, There was an error loading one of the GSK dynamic link libraries. Be sure GSK is installed correctly.**
 - Reason: There was an error loading one of the GSK dynamic link libraries. Be sure GSK is installed correctly.
 - Solution: Verify GSK is installed and appropriate level for release of IBM HTTP Server.
- Message: **SSL0126E: Handshake Failed, Either the certificate has expired or the system clock is incorrect.**
 - Reason: Either the certificate expired or the system clock is incorrect.
 - Solution: Use the key management utility (iKeyman) to recreate or renew your server certificate or change the system date to a valid date.
- Message: **SSL0127E: Initialization error, No ciphers specified.**
 - Reason: Initialization error, no ciphers specified.
 - Solution: Report problem to service.
- Message: **SSL0128E: Initialization error, Either the certificate expired or the system clock is incorrect.**
 - Reason: Initialization error, no certificate.
 - Solution: Report problem to service.
- Message: **SSL0129E: Initialization error, The received certificate was formatted incorrectly.**
 - Reason: The received certificate is formatted incorrectly.
 - Solution: Use Ikeyman to validate certificates used for connection.
- Message: **SSL0130E: Initialization error, Unsupported certificate type.**
 - Reason: Unsupported certificate type.
 - Solution: Check certificates that are used for this connection in Ikeyman.
- Message: **SSL0131I: Initialization error, I/O error during handshake.**

- Reason: I/O error during handshake.
- Solution: Check network connectivity.
- Message: **SSL0132E: Initialization error, Invalid key length for export.**
 - Reason: Invalid key length for export.
 - Solution: Report problem to service.
- Message: **SSL0133W: Initialization error, An incorrectly formatted SSL message was received.**
 - Reason: An incorrectly formatted SSL message was received.
 - Solution: Check client settings.
- Message: **SSL0134W: Initialization error, Could not verify MAC.**
 - Reason: Could not verify MAC.
 - Solution: Report problem to service.
- Message: **SSL0135W: Initialization error, Unsupported SSL protocol or unsupported certificate type.**
 - Reason: Unsupported SSL protocol or unsupported certificate type.
 - Solution: Check server ciphers and certificate settings.
- Message: **SSL0136W: Initialization error, Invalid certificate signature.**
 - Reason: Invalid certificate signature.
 - Solution: Check certificate in Ikeyman.
- Message: **SSL0137W: Initialization error, Invalid certificate sent by partner.**
 - Reason: Invalid certificate sent by partner.
 - Solution: If this occurs during an SSL Proxy connection, the remote SSL server sent a bad certificate to IBM HTTP Server. Check the certificate and certificate authority chain at the other end of the SSL connection.
- Message: **SSL0138W: Initialization error, Invalid peer.**
 - Reason: Invalid peer.
 - Solution: Report problem to service.
- Message: **SSL0139W: Initialization error, Permission denied.** Distributed operating systems
 - Reason: Permission denied.
 - Solution: Report problem to service.
- z/OS
 - Reason: If a System Authorization Facility (SAF) SSL keyring is in use, the current user ID is not authorized to read the keyring.
 - Solution: See the information about access to SAF keyrings in “Performing required z/OS system configurations” on page 65
- Message: **SSL0140W: Initialization error, The self-signed certificate is not valid.**
 - Reason: The self-signed certificate is not valid.
 - Solution: Check the certificate in Ikeyman.
- Message: **SSL0141E: Initialization error, Internal error - read failed.**
 - Reason: Internal error - read failed.
 - Solution: Report to service.
- Message: **SSL0142E: Initialization error, Internal error - write failed.**
 - Reason: Internal error - write failed.
 - Solution: Report to service.
- Message: **SSL0143I: Initialization error, Socket has been closed.**

- Reason: Socket has been closed unexpectedly.
- Solution: Check the client and network. Report problem to service.
- Message: **SSL0144E: Initialization error, Invalid SSLV2 Cipher Spec.**
 - Reason: Invalid SSLV2 cipher spec.
 - Solution: Check the SSLCipherSpec directive.
- Message: **SSL0145E: Initialization error, Invalid SSLV3 Cipher Spec.**
 - Reason: Invalid SSLV3 Cipher Spec.
 - Solution: Check the SSLCipherSpec directive.
- Message: **SSL0146E: Initialization error, Invalid security type.**
 - Reason: Invalid security type.
 - Solution: Report to service.
- Message: **SSL0147E: Initialization error, Invalid security type combination.**
 - Reason: Invalid security type combination.
 - Solution: Report to service.
- Message: **SSL0148E: Initialization error, Internal error - SSL Handle creation failure.**
 - Reason: Internal error - SSL handle creation failure.
 - Solution: Report to service.
- Message: **SSL0149E: Initialization error, Internal error - GSK initialization has failed.**
 - Reason: Internal error - GSK initialization has failed.
 - Solution: Report to service.
- Message: **SSL0150E: Initialization error, LDAP server not available.**
 - Reason: LDAP server not available.
 - Solution: Check CRL directives.
- Message: **SSL0151E: Initialization error, The specified key did not contain a private key.**
 - Reason: The specified key did not contain a private key.
 - Solution: Check the certificate in use in Ikeyman.
- Message: **SSL0152E: Initialization error, A failed attempt was made to load the specified PKCS#11 shared library.**
 - Reason: A failed attempt was made to load the specified PKCS#11 shared library.
 - Solution: Check SSLPKCSDriver directive and file system.
- Message: **SSL0153E: Initialization error, The PKCS#11 driver failed to find the token specified by the caller.**
 - Reason: The PKCS#11 driver failed to find the token specified by the caller.
- Message: **SSL0154E: Initialization error, A PKCS#11 token is not present for the slot.**
 - Reason: A PKCS#11 token is not present for the slot.
 - Solution: Verify PKCS#11 directives.
- Message: **SSL0155E: Initialization error, The password/pin to access the PKCS#11 token is invalid.**
 - Reason: The password and pin to access the PKCS#11 token is invalid.
- Message: **SSL0156E: Initialization error, The SSL header received was not a properly SSLV2 formatted header.**
 - Reason: The SSL header received was not a properly SSLV2 formatted header.

- Message: **SSL0157E: Initialization error, The function call, %s, has an invalid ID.**
 - Reason: The function call, %s, has an invalid ID.
 - Solution: Report problem to service.
- Message: **SSL0158E: Initialization error, Internal error - The attribute has a negative length: %s.**
 - Reason: Internal error - The attribute has a negative length.
 - Solution: Report problem to service.
- Message: **SSL0159E: Initialization error, The enumeration value is invalid for the specified enumeration type: %s.**
 - Reason: The enumeration value is invalid for the specified enumeration type: %s.
 - Solution: Report problem to service.
- Message: **SSL0160E: Initialization error, The SID cache is invalid: %s.**
 - Reason: The SID cache is invalid.
 - Solution: Report problem to service.
- Message: **SSL0161E: Initialization error, The attribute has an invalid numeric value: %s.**
 - Reason: The attribute has an invalid numeric value: %s.
 - Solution: Check SSL directives.
- Message: **SSL0162W: Setting the LD_LIBRARY_PATH or LIBPATH for GSK failed.**
 - Reason: Could not update the environment for GSK libraries.
 - Solution: Report problem to service.
- Message: **SSL0163W: Setting the LIBPATH for GSK failed, could not append /usr/opt/ibm/gskkm/lib.**
 - Reason: Could not append to LD_LIBRARY_PATH or LIBPATH for GSK failed.
 - Solution: Report problem to service.
- Message: **SSL0164W: Error accessing Registry, RegOpenKeyEx/RegQueryValueEx returned [%d].**
 - Reason: Error accessing registry.
 - Solution: Check GSK installation and windows registry.
- Message: **SSL0165W: Storage allocation failed.**
 - Reason: Storage allocation failed.
 - Solution: Check memory usage, report problem to service.
- Message: **SSL0166E: Failure attempting to load GSK library.**
 - Reason: Failure while attempting to load GSK library.
 - Solution: Check the GSK installation.
- Message: **SSL0167E: GSK function address undefined.**
 - Reason: GSK function address is undefined.
 - Solution: Check the GSK installation and level.
- Message: **SSL0168E: SSL initialization for server: %s, port: %u failed due to a configuration error.**
 - Reason: Initialization for server: %s, port: %u failed due to a configuration error.
 - Solution: Check the SSL configuration.

- Message: **SSL0169E: Keyfile does not exist: %s.**
 - Reason: Keyfile does not exist.
 - Solution: Check to ensure the path that is provided to the KeyFile directive exists, and is readable by the user that IBM HTTP Server is running as.
- Message: **SSL0170E: GSK could not initialize, no keyfile specified.**
 - Reason: Keyfile is not specified.
 - Solution: Specify Keyfile directive.
- Message: **SSL0171E: CRL cannot be specified as an option for the SSLClientAuth directive on HPUX because the IBM HTTP Server does not support CRL on HPUX.**
 - Reason: CRL cannot be specified as an option for the SSLClientAuth directive on HPUX because IBM HTTP Server does not support CRL on HPUX.
 - Solution: Remove CRL directives.
- Message: **SSL0172E: If CRL is turned on, you must specify an LDAP hostname for the SSLCRLHostname directive.**
 - Reason: If CRL is turned on, you must specify an LDAP hostname for the SSLCRLHostname directive.
 - Solution: Specify SSLCRLHostname.
- Message: **SSL0173E: Failure obtaining supported cipher specs from the GSK library.**
 - Reason: Failure obtaining supported cipher specs from the GSK library.
 - Solution: Check the GSK installation, report problem to service.
- Message: **SSL0174I: No CRL password found in the stash file: %s.**
 - Reason: No CRL password is found in the stash file: %s.
 - Solution: Check the stash file permissions, regenerate stash file.
- Message: **SSL0174I: No CRYPTO password found in the stash file: %s.**
 - Reason: No CRYPTO password is found in the stash file: %s.
 - Solution: Check stash file permissions, regenerate stash file.
- Message: **SSL0175E: fopen failed for stash file: %s.**
 - Reason: fopen failed for stash file.
 - Solution: Check stash file permissions, regenerate stash file.
- Message: **SSL0176E: fread failed for the stash file: %s.**
 - Reason: fread failed for the stash file.
 - Solution: Make sure the stash file is readable by user IBM HTTP Server is running as.
- Message: **SSL0179E: Unknown return code from stash_recover(), %d.**
 - Reason: Unknown return code from stash_recover(), %d.
 - Solution: Check the stash file.
- Message: **SSL0181E: Unable to fork for startup of session ID cache.**
 - Reason: Unable to fork for startup of session ID cache.
 - Solution: Check the location of sidd daemon, file permissions.
- Message: **SSL0182E: Error creating file mapped memory for SSL passwords.**
 - Reason: Error creating file mapped memory for SSL passwords.
 - Solution: Report problem to service.
- Message: **SSL0183E: Exceeded map memory limits.**
 - Reason: Exceeded map memory limits.

- Solution: Report problem to service.
- Message: **SSL0184E: Could not find a password for the resource: %s.**
 - Reason: SSL0184E: Could not find a password for the resource: %s.
 - Solution: Report problem to service, disable password prompting.
- Message: **SSL0185E: ssl_getpwd() failed, unable to obtain memory.**
 - Reason: ssl_getpwd() failed, unable to obtain memory.
 - Solution: Report problem to service, disable password prompting.
- Message: **SSL0186E: Linked list mismatch.**
 - Reason: SSL0186E: Linked list mismatch.
 - Solution: Report problem to service, disable password prompting.
- Message: **SSL0186E: ssl_getpwd() failed, password exceeded maximum size of 4095.**
 - Reason: ssl_getpwd() failed, password exceeded the maximum size of 4095.
 - Solution: The password must be smaller than 4K.
- Message: **SSL0187E: It is invalid to enable password prompting for the SSLServerCert directive without specifying a Crypto Card Token.**
 - Reason: It is invalid to enable password prompting for the SSLServerCert directive without specifying a crypto card token.
 - Solution: Specify a crypto card token or disable password prompting for the SSLServerCert directive.
- Message: **SSL0188E: SSL initialization for server: %s, port: %u failed. SSL timeouts cannot be set in a virtualhost when the SSLCacheDisable directive has not been specified globally.**
 - Reason: When the SSL session cache is being used, only the global timeout settings apply because they are managed by the external session cache daemon. See information about the SSLCacheDisable and SSLCacheEnable directives in the information center topic entitled *SSL directives*.
 - Solution: If separate SSL timeouts are required, disable use of the session ID cache (SSLCacheDisable), otherwise make sure the SSLV3Timeout and SSLV2Timeout directives are only set in the global scope.

I/O error messages

Distributed operating systems z/OS

This topic contains error messages that might result due to I/O failures and provides solutions to help you troubleshoot these problems.

The following messages appear due to read failures:

- Message: **SSL0400I: I/O failed, RC <code>.**
 - Reason: The server received an error trying to read on the socket.
 - Solution: Some errors are expected during normal processing, especially a '406' error, which you can ignore. If you are unable to access the server and receive these errors, report this problem to Service.
- Message: **SSL0401E: I/O failed with invalid handle <handle>.**
 - Reason: An internal error has occurred.
 - Solution: Report this problem to Service.
- Message: **SSL0402E: I/O failed, the GSKit library is not available.**
 - Reason: A call to the GSKit function failed because the dynamic link library unloaded (Windows operating systems only).

- Solution: Shut down the server and restart.
- Message: **SSL0403E: I/O failed, internal error.**
 - Reason: The communication between client and the server failed due to an error in the GSKit library.
 - Solution: Retry connection from the client. If the error continues, report the problem to Service.
- Message: **SSL0404E: I/O failed, insufficient storage.**
 - Reason: The server could not allocate memory needed to complete the operation.
 - Solution: Take action to free up some additional memory. Try reducing the number of threads or processes running, or increasing virtual memory.
- Message: **SSL0405E: I/O failed, SSL handle <handle> is in an invalid state.**
 - Reason: The SSL state for the connection is invalid.
 - Solution: Retry connection from the client. If the error continues, report the problem to Service.
- Message: **SSL0406E: I/O failed, cryptography error.**
 - Reason: A cryptography error occurred.
 - Solution: None. If the problem continues, report it to Service.
- Message: **SSL0407I: I/O failed, Error validating ASN fields in certificate.**
 - Reason: The server was not able to validate one of the ASN fields in the certificate.
 - Solution: Try another certificate.
- Message: **SSL0408E: I/O failed with invalid buffer size. Buffer <address>, size <length>.**
 - Reason: The buffer size in the call to the read function is zero or negative.
 - Solution: None. Report this problem to Service.
- Message: **SSL0409I: I/O error occurred**
 - Reason: An unexpected network error occurred while reading or writing data over an SSL connection, likely a client disconnecting.
 - Solution: This is an informational message that does not indicate any failure in delivering a response, therefore no solution is provided.
- Message: **SSL0410I: Socket was closed**
 - Reason: An SSL client connection was closed by the client.
 - Solution: This is an informational message that does not indicate any failure in delivering a response, therefore a solution is not provided.
- Message: **SSL0411E: Connection aborted due to unexpected client renegotiation or other malformed SSL record <errorcode>**
 - Reason: An unexpected client renegotiation has been detected or an incorrectly formatted SSL message has been received. Thus, the SSL connection has been stopped.
 - Solution: Check the client settings and retry connection between the client and the server.

SSL stash utility messages

This topic contains error messages that might result due to Secure Sockets Layer (SSL) stash utility problems and provides solutions to help you troubleshoot these problems.

The following messages appear due to SSL Stash utility errors:

- Message: **SSL0700S: Invalid function <function>**
 - Reason: An invalid parameter was entered. The valid values are `crl` or `crypto`.
 - Solution: Rerun the command with the proper function.
- Message: **SSL0701S: The password was not entered.**
 - Reason: The password was not entered on the command line.
 - Solution: Rerun the command with the password added.
- Message: **SSL0702S: Password exceeds the allowed length of 512.**
 - Reason: The password that was entered is longer than the allowed maximum of 512 characters.
 - Solution: Use a shorter password.

Chapter 7. Glossary

Distributed operating systems

z/OS

authentication

In computer security, verification of the identity of a user or the user's eligibility to access an object.

cache To place, hide, or store frequently used information locally for quick retrieval.

cache accelerator

Provides support for caching on multiple Web servers and on servers with multiple IP addresses.

certificate authority (CA)

In computer security, an organization that issues certificates. The certificate authority authenticates the certificate owner's identity and the services that the owner is authorized to use. It also manages the issuance of new certificates and revokes certificates from unauthorized users who are no longer authorized to use them. A certificate authority is considered to be trusted when a user accepts any certificate issued by that certificate authority as proof of the certificate owner's identity.

certificate revocation list (CRL)

A list of certificates that need to be revoked before their expiration date.

cipher In Cryptographic Support, data that is unintelligible to all except those who have the key to decode it to plaintext.

cipher specifications

Indicate the data encryption algorithm and key size to use for secure connections.

cryptographic support

The IBM licensed program that provides support for the encryption and decryption of data, according to the Data Encryption Algorithm, and for the management of cryptographic keys and personal identification numbers (PINs).

Data Encryption Standard (DES)

In computer security, the National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm.

digital certificate

A form of personal identification that can be verified electronically. Only the certificate owner who holds the corresponding private key can present a certificate for authentication through a Web browser session. Anyone can verify that the certificate is valid by using a readily available public key.

digital signature

Information that is encrypted with an entity private key and is appended to a message to assure the recipient of the authenticity and integrity of the message. The digital signature proves that the message was signed by the entity that owns, or has access to, the private key or shared secret symmetric key.

directive

A statement that is used in the configuration file for a Web server to define a particular setting for the server.

distinguished name (DN)

In computer security, information that uniquely identifies the owner of a certificate.

dynamic shared object (DSO)

A mechanism which provides a way to build a piece of program code in a special format for loading at run time into the address space of an executable program. The DSO gets knowledge of the executable program symbol set as if it had been statically linked with it in the first place

encrypt

In Cryptographic Support, to systematically scramble information so that it cannot be read without knowing the coding key.

environment variable

A variable that specifies how an operating system or another program runs, or the devices that the operating system recognizes.

Fast Common Gateway Interface Protocol (FastCGI)

The Fast Common Gateway Interface (FastCGI) is an enhancement to the existing Common Gateway Interface (CGI), which is a standard for interfacing external applications with Web servers.

handshake

A Secure Sockets Layer (SSL) session always begins with an exchange of messages called the SSL handshake. The handshake allows the server to authenticate itself to the client by using public key techniques, and then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

Java An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

Java Development Kit (JDK)

A software package that can be used to write, compile, debug, and run Java applets and applications.

Java Runtime Environment (JRE)

A subset of the Java Development Kit (JDK) that contains the core executables and files that constitute the standard Java platform. The JRE includes the Java Virtual Machine (JVM), core classes, and supporting files.

Java Virtual Machine (JVM)

A software implementation of a central processing unit (CPU) that runs compiled Java code (applets and applications).

key In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data.

key database

Exists as a file that the server uses to store one or more key pairs and certificates. You can use one key database for all your key pairs and certificates, or create multiple databases.

key file

In the Distributed Computing Environment (DCE), a file that contains encryption keys for noninteractive principals.

key pair

Contains a public, distributed key and a private key. A key pair is issued by a public key cryptography system and is used in combination with each other to validate and authenticate a connection between a client and server for secure connections.

Lightweight Directory Access Protocol (LDAP)

In TCP/IP, a protocol that enables users to locate people, organizations, and other resources in an Internet directory or intranet directory.

module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

password stashing

The password is encrypted in a file or on a hard drive. Your keydb password needs to reside in a file in order to use secure sockets layer (SSL).

PKCS12

Sometimes referred to as PFX files; PKCS#12 files are used by several programs including Netscape, MSIE and MS Outlook.

plug-in

A self-contained software component that modifies (adds or changes) function in a particular software system. When a user adds a plug-in to a software system, the foundation of the original software system remains intact. The development of plug-ins requires well defined application programming interfaces (APIs).

port

(1) A system or network access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. One or more ports are controlled by a single data link control (DLC) process. (4) In the Internet suite of protocols, a specific logical connector between the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) and a higher level protocol or application. (5) To modify a computer program to enable it to run on a different platform.

port number

In the Internet suite of protocols, the identifier for a logical connector between an application entity and the transport service.

private key

In secure communication, an algorithmic pattern used to encrypt messages that only the corresponding public key can decrypt. The private key is also used to decrypt messages that were encrypted by the corresponding public key. The private key is kept on the user's system and is protected by a password.

public key

In secure communication, an algorithmic pattern used to decrypt messages that were encrypted by the corresponding private key. A public key is also

used to encrypt messages that only the corresponding private key can decrypt. Users broadcast their public keys to everyone with whom they must exchange encrypted messages.

public key infrastructure (PKI)

An infrastructure that supports digital signatures and other public key-enabled security services.

Secure Sockets Layer (SSL)

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corporation and RSA Data Security, Inc.

stash file

A file that hides other data files within.

symmetric keys

In computer security, the two keys in a key pair. The keys are called symmetric because each key holds as much of the encryption pattern as the other does.

trust policy

Contains a trusted list of certificates that are used to control the trust and validity period of certificates. It enables one to limit the trust of certificates issued by a certificate authority.

trusted root

A certificate signed by a certificate authority (CA), designated as a trusted CA on your server.

virtual host

Refers to the practice of maintaining more than one server on one machine, differentiated by their apparent host name.

X.500 The directory services standard of International Telecommunication Union (ITU), International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC).

Appendix. Accessibility

Accessible publications for this product are offered through .

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

-
-
-

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually

exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element

The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element

The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A

default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

*** indicates an optional syntax element that is repeatable**

The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

Distributed operating systems

z/OS

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

APACHE INFORMATION. This information may include all or portions of information which IBM obtained under the terms and conditions of the Apache License Version 2.0, January 2004. The information may also consist of voluntary contributions made by many individuals to the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org>. You may obtain a copy of the Apache License at <http://www.apache.org/licenses/LICENSE-2.0>.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Intellectual Property & Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Requests

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

© Copyright IBM Corp. 1997-2015.

Index

A

- accessibility 255
 - contact IBM 255
 - features 255
- assistive technologies 255

C

- certificate authorities 193
- commands
 - apachectl 71
 - gskcmd 183
 - setupadm 111
- contact
 - z/OS 255
- cryptographic hardware
 - SSL 196
 - troubleshooting 225

F

- FRCA
 - starting
 - Windows services 72

H

- HTTP Server
 - Apache 85

I

- IBM HTTP Server 109, 110
 - administering 226
 - Apache 2, 78
 - changing database passwords 172
 - error messages 229
 - IBM 2
 - installation
 - z/OS 39, 43
 - login failure 228
 - mounting CD-ROMS 13
 - new functions 1
 - starting
 - administrative console 69, 70
 - system configurations
 - z/OS 65
 - third-party plug-ins 101
 - troubleshooting 223
 - Windows 223
 - z/OS 225
 - uninstallation
 - GUI 33
 - updating 21
 - Windows 15
- IKEYMAN
 - PKCS11 devices 198
 - starting 171

- IPv4 support
 - HTTP Server 87
- IPv6 support
 - HTTP Server 87

K

- keyboard
 - navigation 255
 - PF keys 255
 - shortcut keys 255

L

- LDAP
 - configuration 201

M

- messages
 - cache errors 230
 - configuration errors 231
 - handshake errors 233
 - I/O errors 248
 - SSL initialization 240
 - SSL stach utility errors 250

N

- navigation
 - keyboard 255

S

- shortcut keys 255

U

- user interface
 - ISPF 255
 - TSO/E 255
- utilities
 - htpasswd 110



Printed in USA

SC27-8417-00

