IBM HTTP Server, Version 7.0

IBM HTTP Server

User's Guide

Note

Before using this information, be sure to read the general information under “Notices” on page 189.

Compilation date: September 20, 2008

Contents

How to send your comments vii

Changes to serve you more quickly ix

Chapter 1. Migrating and installing IBM HTTP Server > Distributed operating systems 1
Installing IBM HTTP Server 1
Mounting CD-ROMS on AIX, HP-UX, Linux and Solaris systems 3
Installing IBM HTTP Server with a non-administrator user ID 5
Creating multiple instances of IBM HTTP Server on Windows operating systems 6
Migrating from previous releases of IBM HTTP Server 6
Installing IBM HTTP Server silently 7
Uninstalling IBM HTTP Server 9
Manually uninstalling IBM HTTP Server 9

Chapter 2. Migrating and installing IBM HTTP Server on z/OS systems 15
Installing IBM HTTP Server 16
Uninstalling IBM HTTP Server 19

Chapter 3. Performing required z/OS system configurations 21

Chapter 4. Starting and stopping IBM HTTP Server 25
Using the administrative console to start IBM HTTP Server 25
Using apachectl commands to start IBM HTTP Server 26
Using Windows services to start IBM HTTP Server 27
Using JCL procedures to start IBM HTTP Server on z/OS 28

Chapter 5. Configuring IBM HTTP Server 31
Apache modules (containing directives) supported by IBM HTTP Server 31
Apache programs supported by IBM HTTP Server 35
Apache APR and APR-util libraries supported by IBM HTTP Server 36
Apache MPM and addressing modes supported by IBM HTTP Server 37
IPv4 and IPv6 configuration for Windows operating systems 37

Chapter 6. Serving static content faster with Fast Response Cache Accelerator 39
Customizing Fast Response Cache Accelerator logging 39
Restrictions on cached content 40
Fast Response Cache Accelerator operational restrictions 40
Servlets and JavaServer Pages files caching 41
AIX considerations for Fast Response Cache Accelerator (FRCA) 41
AFPA directives 41

Chapter 7. Enabling IBM HTTP Server for FastCGI applications 45
Learn about FastCGI 45
FastCGI directives 46

Chapter 8. Managing remotely with the WebSphere Application Server administrative console 57

Chapter 9. Extending IBM HTTP Server functionality with third-party plug-in modules 59
Viable compilers for Apache and third-party plug-in modules 59
Build method options for dynamic modules 60
Considerations for building dynamic modules on Windows platforms 60

Chapter 10. Administering and configuring the administration server 61

Starting and stopping the IBM HTTP Server administration server 61
Protecting access to the IBM HTTP Server administration server 61
Enabling access to the administration server using the htpasswd utility 62
Running the setupadm script for the administration server 62
Setting permissions manually for the administration server 63

Chapter 11. Task overview: Securing IBM HTTP Server 65

Chapter 12. Securing with SSL communications 67
Secure Sockets Layer (SSL) protocol 69
Certificates 70
Public Key Infrastructure 72
Session ID cache 73
SSL directive considerations 73
Authentication 74
Encryption 75
Secure Sockets Layer environment variables 75
SSL handshake environment variables. 76
Server certificate environment variables 77
Client certificate environment variables 78
SSL directives. 79

Chapter 13. Setting advanced SSL options 95
Choosing the level of client authentication 95
Choosing the type of client authentication protection 95
Setting cipher specifications 96
Viewing cipher specifications 97
SSL Version 2 cipher specifications 97
SSL Version 3 and TLS Version 1 cipher specifications 98
Defining SSL for multiple-IP virtual hosts 99
Setting up a reverse proxy configuration with SSL 99

Chapter 14. Managing keys with the IKEYMAN graphical interface (Distributed systems) . . . 101
Setting your system environment for using IKEYMAN 101
Starting the Key Management utility user interface 102
Working with key databases 102
Changing the database password 103
Creating a new key pair and certificate request 103
Importing and exporting keys. 104
Listing certificate authorities 105
Certificate expiration dates 106
Creating a self-signed certificate 106
Receiving a signed certificate from a certificate authority 107
Displaying default keys and certificate authorities 108
Storing a certificate authority certificate 108
Storing the encrypted database password in a stash file. 109

Chapter 15. Managing keys with the gsk7cmd command line interface (Distributed systems) 111
Using the gsk7cmd command 111
Key Management Utility command-line interface (gsk7cmd) syntax 111
Creating a new key database using the command-line interface 114
Managing the database password using the command line 115
Creating a new key pair and certificate request 116
Importing and exporting keys using the command line 116
Creating a self-signed certificate 118
Receiving a signed certificate from a certificate authority. 119

Displaying default keys and certificate authorities 120
Storing a certificate authority certificate 120
Storing the encrypted database password in a stash file 121

Chapter 16. Managing keys with the native key database gskkyman (z/OS systems) 123

Chapter 17. Getting started with the cryptographic hardware for SSL (Distributed systems) 125
Cryptographic hardware for Secure Sockets Layer 125
Initializing IBM 4758 and IBM e-business Cryptographic Accelerator on AIX systems 127
Initializing IBM 4758 Cryptographic Accelerator on Windows systems 127
Using IKEYMAN to store keys on a PKCS11 device 127
Configuring IBM HTTP Server to use nCipher and Rainbow accelerator devices and PKCS11 devices 128

Chapter 18. Authenticating with LDAP on IBM HTTP Server using mod_ibm_ldap (Distributed systems) 131
Lightweight Directory Access Protocol 133
Querying the Lightweight Directory Access Protocol server 134
Secure Sockets Layer and the Lightweight Directory Access Protocol module 135
SSL certificate revocation list 135
LDAP directives 136
Converting your directives from mod_ibm_ldap to mod_ldap 146
mod_ibm_ldap directives migration 149

Chapter 19. Authenticating with LDAP on IBM HTTP Server using mod_ldap 157

Chapter 20. Authenticating with SAF on IBM HTTP Server (z/OS systems) 159
SAF directives 159

Chapter 21. Troubleshooting IBM HTTP Server 165
Known problems on Windows platforms 165
Known problems on z/OS platforms 165
Known problems with hardware cryptographic support 166
Symptoms of poor server response time 166
Hints and tips for managing IBM HTTP Server using the administrative console 167
Could not connect to IBM HTTP Server administration server error 168
Experiencing an IBM HTTP Server Service logon failure on Windows operating systems 169

Chapter 22. Viewing error messages for a target server that fails to start 171
Cache messages 171
Configuration messages 172
Handshake messages 173
SSL initialization messages 179
I/O error messages 186
SSL stash utility messages 187

Notices 189

Trademarks and service marks 191

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

- To send comments on articles in the WebSphere Application Server Information Center
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the **Feedback** link at the bottom of the article, and a separate window containing an e-mail form appears.
  3. Fill out the e-mail form as instructed, and click on **Submit feedback** .
- To send comments on PDF books, you can e-mail your comments to: **wasdoc@us.ibm.com** or fax them to 919-254-5250.

  Be sure to include the document name and number, the WebSphere Application Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Changes to serve you more quickly

Print sections directly from the information center navigation

PDF books are provided as a convenience format for easy printing, reading, and offline use. The information center is the official delivery format for IBM WebSphere Application Server documentation. If you use the PDF books primarily for convenient printing, it is now easier to print various parts of the information center as needed, quickly and directly from the information center navigation tree.

To print a section of the information center navigation:

1. Hover your cursor over an entry in the information center navigation until the **Open Quick Menu** icon is displayed beside the entry.
2. Right-click the icon to display a menu for printing or searching your selected section of the navigation tree.
3. If you select **Print this topic and subtopics** from the menu, the selected section is launched in a separate browser window as one HTML file. The HTML file includes each of the topics in the section, with a table of contents at the top.
4. Print the HTML file.

For performance reasons, the number of topics you can print at one time is limited. You are notified if your selection contains too many topics. If the current limit is too restrictive, use the feedback link to suggest a preferable limit. The feedback link is available at the end of most information center pages.

Under construction!

The Information Development Team for IBM WebSphere Application Server is changing its PDF book delivery strategy to respond better to user needs. The intention is to deliver the content to you in PDF format more frequently. During a temporary transition phase, you might experience broken links. During the transition phase, expect the following link behavior:

- 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

Thanks for your patience, in the short term, to facilitate the transition to more frequent PDF book updates.

Chapter 1. Migrating and installing IBM HTTP Server > Distributed operating systems

You can install the IBM® HTTP Server product on distributed operating systems in two ways: from the discs in the product package, or by downloading installation images from the Passport Advantage® site, if you are licensed to do so. Distributed operating systems include AIX®, HP-UX, Linux®, Solaris, and Windows® systems.

The installation procedure on distributed platforms uses the InstallShield for Multiplatforms (ISMP) program to perform the installation. You can use the Installation wizard in graphical interface mode or in silent mode. In silent mode, the Installation wizard does not display a graphical interface, but instead, reads your responses from a flat file that you prepare beforehand.

Use the links provided in this topic to learn about the installation features.

Installing IBM HTTP Server

This topic describes how to install IBM HTTP Server using the graphical interface.

Migrating from previous releases of IBM HTTP Server

This topic provides information on what to look for when you are migrating IBM HTTP Server from a previous release.

Installing IBM HTTP Server silently

A silent installation uses the installation wizard to install the product in silent mode, without the graphical user interface. Instead of displaying a wizard interface, the silent installation enables the installation program to read all of your responses from a file that you provide.

Uninstalling IBM HTTP Server

This topic contains procedures for uninstalling the IBM HTTP Server. The uninstaller program is customized for each product installation, with specific disk locations and routines for removing installed features. The uninstaller program does not remove configuration and log files.

Installing IBM HTTP Server

This article describes installing IBM HTTP Server using the launchpad.

Before you begin

Set the umask value to `022`. To verify that the umask value is set to `022`, run the umask command.

1. Prepare your operating platform for installing IBM HTTP Server as you would for installing any of the installable components on the product disc. Refer to the Information center topic ″Preparing the operating system for product installation″ that is specific for your WebSphere® Application Server product.
2. Insert the product disc and mount the disc if necessary.

   AIX HP-UX Linux Solaris See “Mounting CD-ROMS on AIX, HP-UX, Linux and Solaris systems” on page 3 for information about mounting the product disc.
3. Start the installation with the launchpad command:
   - AIX HP-UX Linux Solaris **launchpad.sh**
   - Windows **launchpad.bat**

   You can also start the installation from the `IHS` directory, where `IHS` is the installable component directory on the product disc. Launch the following command from the product disc:

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

When using the launchpad, launch the Installation wizard for IBM HTTP Server. Refer to the Information center topic ″Using the launchpad to start the installation″ that is specific for your WebSphere Application Server product.

After launching the Installation wizard from the launchpad or from the command line, the ISMP wizard initializes and then presents the Welcome panel.

4. Click **Next** to display the License agreement panel.
5. Accept the license agreement and click **Next** to display the operating system prerequisites check panel.
6. Click **Next** to display the installation root directory panel.
7. Specify the root directory information and click **Next** to display the port specification panel. The port specification panel enables you to modify the ports to use for IBM HTTP Server and the IBM HTTP Server administration server. The default port values are `80` for IBM HTTP Server and `8008` for the IBM HTTP administration module. Specify unique port values if the default ports are already in use by another application.

   Note: Issue `netstat -an` from the command prompt to display a list of active ports.
8. Windows Click **Next** to display the Windows Service Definition panel. You have the option to create a Windows service for IBM HTTP Server and the IBM HTTP Server administration 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 and Log on as a service.
   - If you are planning to administer IBM HTTP Server using the WebSphere Application Server administrative console, select Run IBM HTTP Server Administration as a Windows Service with Log on as Local System account. A user name and password is not required for this selection.
   - If you will not administer IBM HTTP Server using the WebSphere Application Server administrative console, select Run IBM HTTP Server Administration Server as a Windows Service with Log on as a specified user account. Specify your user ID and password information.
9. Distributed platforms Click **Next** to display the IBM HTTP Server Administration Server Authentication panel. If selected, this panel creates a user ID and password to authenticate to the IBM HTTP Server administration server using the Websphere Application Server administrative console. This user ID and password can optionally be created after installation using the htpasswd utility.
10. AIX HP-UX Linux Solaris Click **Next** to display the IBM HTTP Server Administration Server setup panel. This panel collects information to enable the installation to grant a user ID write access to the necessary IBM HTTP Server and plug-in configuration files. The IBM HTTP Server administration server runs as the specified user ID.
11. Click **Next** to display the IBM HTTP Server Plug-in for Websphere Application Server panel. This panel collects information to install the Websphere Application Server Plug-in into a directory that is relative to the IBM HTTP Server installation location, using the remote setup type.

    Note: If the `plugin` directory does not exist at the same level as the IBM HTTP Server directory, the prompt panel for selecting the plug-ins installer does not display. In that case, launch the plug-ins installer from the launchpad or launch the following command from the `plugin` directory:
    - AIX HP-UX Linux Solaris **./install**
    - Windows **install.exe**
12. Click **Next** to review the confirmation panel to verify your selections. Click **Back** to change any of your specifications.

13. Click **Next** to begin installing IBM HTTP Server.

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 Chapter 12, “Securing with SSL communications,” on page 67.

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

You can also make many other configuration changes with Apache directives.

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

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.

- 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.

Installing IBM HTTP Server with a non-administrator user ID

The common way to install IBM HTTP Server is to run the installation program using an administrator user ID. However, it is sometimes necessary to install IBM HTTP Server using a non-administrator (non-root) user ID.

Before you begin

You must run the setupadm command if you are installing IHS as a non-root user. The setupadm command is run in the <IHS_HOME>/bin directory so that you can properly use the administrative server with the WebSphere Application Server. The format for the command is as follows (on one line):

```
setupadm -usr <userName> -grp <groupName> -cfg <IHS Web server configuration file> -adm
<IHS admistrative server configuration file> -plg <plug-in configuration file>
```

About this task

Launching the IBM HTTP Server installation program is done the same way for a non-root installation as it is for a root installation, but there are several installation steps that require root privileges that cannot be completed or must be completed separately. Complete the installation steps as follows:

- **Register the installed program with the operating system.** This cannot be done for a non-root installation. The non-root IBM HTTP Server installation is not listed when using operating system facilities to display installed programs.
- Windows **Create the Windows service entries for IBM HTTP Server and IBM HTTP Administration Server.** This cannot be done for a non-root installation. Neither of these service entries are created and IBM HTTP Server cannot start as a service.

  Start IBM HTTP Server as follows:

  `<ihs_install_directory>/bin/httpd.exe`

  Start the IBM HTTP administration server as follows:

  `<ihs_install_directory>/bin/httpd.exe -f <ihs_install_directory>/conf/admin.conf`

  Stop IBM HTTP server as follows:

  – Press Control+C in the IBM HTTP Server window, or

  – End the `httpd.exe` processes using the Windows Task Manager
- Windows **Create an entry in Start > Programs.** This cannot be done for a non-root installation. No entries are created.
- Windows **Create an entry in Add/Remove programs.** This cannot be done for a non-root installation. No entry is created.
- Windows **Install AFPA.** This cannot be done for a non-root installation. Do not enable AFPA for the non-administrator IBM HTTP Server installation, even if AFPA is already installed from a previous administrator installation. Only enable AFPA for one instance of IBM HTTP Server.
- **Silent installations.** To enable a non-root installation, add the following option to the silent installation response file:

  `-OPT allowNonRootSilentInstall="true"`

What to do next

Uninstall a non-root installation of IBM HTTP as follows: AIX HP-UX Linux Solaris

`<ihs_install_directory>/uninstall/uninstall`

Windows

`<ihs_install_directory>\uninstall\uninstall.exe`

Creating multiple instances of IBM HTTP Server on Windows operating systems

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.

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 releases of IBM HTTP Server

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

Before you begin

Consider the following information before you upgrade the Apache HTTP Server or IBM HTTP Server.

Note: 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.

If you are upgrading from a previous version of IBM HTTP Server and you want to install the new version in the same directory location, you must first uninstall the previous version of IBM HTTP Server.

Distributed platforms Issue these commands from the <*ihsinst*>/bin directory instead of the system directory: gsk7cmd, gsk7ver, and gsk7capicmd.

Some modules and commands might have changed for this release. See the topics about supported Apache modules and programs for information about updates or changes to modules or programs before you upgrade IBM HTTP Server.

WebSphere Application Server provides a new plug-in for the Apache HTTP Server and the IBM HTTP Server. See the list item below about upgrading Apache plug-in modules.

About this task

Consider the following information when upgrading your version of IBM HTTP Server.

- **Upgrading IBM HTTP Server from your previous installation**

  When you upgrade IBM HTTP Server from a previous version, complete these 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 these steps.

  1. Uninstall the previous IBM HTTP Server version. This will leave behind any customized configuration files added after the initial installation.
  2. Rename the directory containing the files of the uninstalled IBM HTTP Server.
  3. Install the new IBM HTTP Server.
  4. Overwrite the new IBM HTTP Server configuration files with the saved files from the renamed directory.

- **Migrating the mod_ibm_ldap module directives to use the mod_ldap module directives.**

  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 directives. 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_ldap module directives to ensure future support for your LDAP configuration.

- **Upgrading Apache plug-in modules**

  Apache plug-in modules from sources other then the IBM HTTP Server 7.0 installation must be built to support Apache 2.2. The distributors of modules used with older versions of IBM HTTP Server might need to recompile the modules to support Apache 2.2.

  – WebSphere Application Server provides a new plug-in for Apache 2.2 and IBM HTTP Server 7.0.
  – If you use modules from third party vendors, contact your vendor for a version of the module that works with the Apache 2.2 API (application programming interface).
  – If you use modules developed in-house, you must rebuild your modules to support Apache 2.2. The modules might also require some modifications.

Installing IBM HTTP Server silently

A silent installation uses the installation wizard to install the product in silent mode, without the graphical user interface. Instead of displaying a wizard interface, the silent installation enables the installation program to read all of your responses from a file that you provide.

Before you begin

Installing IBM HTTP Server using silent installation refers to using a file to supply installation options without user interaction. To configure the installation, change the options in the response file before you issue the installation command. Silent installation mode does not accept interactive installation options. To specify non-default options during a silent installation, you must use the response file. To install silently, you must accept the license agreement in the agreement option.

Verify that the required disk space is available.

See Preparing the operating system for product installation for more information. Do not use the default response file that is shipped on the product disc to install the product, because the value of the silentInstallLicenseAcceptance bean is ″false″. Copy the file to change the value to ″true″.

1. AIX HP-UX Linux Solaris Log on as root.
2. Windows Log on as a member of the administrator group. Considerations for Windows operating systems follow:
   - Some steps for installing silently require the administrator group user to have the following advanced user rights:
     – *Act as part of the operating system*
     – *Log on as a service*
   - The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform the silent installation, you must restart the system to activate the proper authorizations for the user ID, before you can perform a successful silent installation.
   - When installing IBM HTTP Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with the service configured to run as LocalSystem, and then modify the user ID after install.
3. Copy the `responsefile.txt` file to your disk drive and rename it, for example *`myoptionsfile.txt`*. You can now customize it. Accept the IBM HTTP Server license by setting `-OPT silentInstallLicenseAcceptance="true"` in your response file.
4. Issue the proper command to use your custom response file. For example, issue one of the following commands:
   - AIX HP-UX Linux Solaris *`mnt_cdrom`*`/IHS/install -options` *`myoptionsfile.txt`* `-silent`
   - Windows *`"CD-ROM drive`*`:\IHS\install" -options` *`"myoptionsfile.txt"`* `-silent`

   AIX HP-UX Linux Solaris To silently install IBM HTTP Server, the X Windows software must be installed on the system.

   You can find the sample options response file in the IBM HTTP Server directory on the product CD.

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.

Linux The IBM HTTP Server installation may hang during the silent install of GSKit. The following message displays in the installConfig.log file:

```
<message>error: failed to stat /mnt/xxx: Stale NFS file handle</message>
```

This problem might occur if the system has an unresponsive Network File System (NFS) mount, then the Linux rpm command, which is used to install the GSKit, attempts to query the unresponsive file system mount until it times out. To work around this problem, unmount the stale NFS mount, and then mount it again.

Uninstalling IBM HTTP Server

This section contains procedures for uninstalling the IBM HTTP Server. The uninstaller program is customized for each product installation, with specific disk locations and routines for removing installed features. The uninstaller program does not remove configuration and log files.

1. Stop IBM HTTP Server.
2. Go to the directory where you installed the IBM HTTP Server. Change to the `uninstall` directory located in the root directory.
3. Double-click **uninstall** to launch the uninstallation program. You can also choose to do a silent uninstall by running the `uninstall -silent` command.
4. Click **Next** to begin uninstalling the product.The Uninstaller wizard displays a Confirmation panel that lists the product and features that you are uninstalling.
5. Click **Next** to continue uninstalling the product.The Uninstaller wizard deletes existing profiles first. After deleting profiles, the Uninstaller wizard deletes core product files by component.
6. Click **Finish** to close the wizard after the wizard removes the product.

Results

The IBM HTTP Server uninstallation is now complete. The uninstallation is logged in the `log.txt` file in the *<ihs_install_directory>*`/logs/uninstall` directory.

Manually uninstalling IBM HTTP Server

Uninstall IBM HTTP Server by running the operating system's uninstaller program and performing some manual steps to remove log files and registry entries. Such registry entries can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, you do not have to uninstall manually.

Before you begin

Determine the installation root directory for the product so that you remove the correct product and produce a clean system. Before you start the uninstall procedure, save any files you have modified under *HTTPServer_root*, *HTTPServer_root*`/conf`, or *HTTPServer_root*`/bin` directories, including: httpd.conf, admin.conf, any other configuration files, script files, password files, or Web documents that you might have created.

About this task

Use the installRegistryUtils command to examine the installation locations for all installed IBM HTTP Server products. Perform the following procedure to produce a clean system.

1. Log on as the same user ID that installed the product.
2. Make sure that the instances of IBM HTTP Server and IBM HTTP Server administration server being removed are stopped.
3. Issue the uninstall command. If you have already run the uninstaller program or if you cannot run the uninstaller program, skip this step. AIX HP-UX Linux Solaris

   HTTPServer_root`/uninstall/uninstall`

   Windows

HTTPServer_root/uninstall/uninstall.exe

The Uninstaller wizard begins and displays the Welcome panel. Continue with the uninstall process until it is finished.

4. IMPORTANT: If you installed IBM HTTP Server as root or administrator, the following steps clean up the registries when an uninstall request fails. If the uninstall process completes successfully, you do not need to manually perform these steps. AIX
   a. List IBM HTTP Server components that are installed. enter the following command to search for related packages:

      ```
      lslpp -l| grep -i IHS
      ```

      Package names for IBM HTTP Server are: WSIHS70 and WSIHS70LicensingComponent.

      To remove a package, issue the following command:

      ```
      geninstall -u packagename
      ```

      Do not remove packages for IBM HTTP Server products that you are not uninstalling.
   b. Change directories to the /usr/IBM directory, or the equivalent top directory of your install.
   c. Enter the following command to delete the IBM HTTP Server directory:

      ```
      rm -rf HTTPServer
      ```
   d. Use the installRegistryUtils command to examine the installation locations for all installed IBM HTTP Server products and remove the products from the install registry that you want to uninstall.
   e. Edit the vpd.properties file. This file is located in the root directory or in the /usr/lib/objrepos directory. Do not delete or rename the vpd.properties file because the InstallShield MultiPlatform (ISMP) program uses it for other products that it installs. If the IBM HTTP Server product that you are uninstalling is the only product with entries in the vpd.properties file, you can delete this file.

   HP-UX
   a. Use HP-UX System Administration Manager (SAM) to remove packages.
      1) Start the SAM utility with the /usr/sbin/sam command
      2) Verify that your DISPLAY and TERM environment variables are set properly
      3) Click **Software management**
      4) Click **View installed software**
      5) Search for IBM HTTP Server entries in the SD list
      6) Close the SD list
      7) Click **Remove local host software**
      8) Click any of the following instances that display in the SD Remove List: - IBM HTTP Server
      9) Click **Actions** > **Mark for remove**
      10) Click **Actions** > **Remove**
      11) Click **OK** in the Remove analysis dialog box
      12) Click **Logs** to display real-time removal of selected packages
      13) Click Done when all packages are removed
      14) Exit SAM
      15) Search for the packages to verify their removal
   b. Enter the following command to display the IBM HTTP Server package:

      ```
      swlist | grep IHS
      ```

      The package name for IBM HTTP Server is: WSIHS70.
   c. Enter the following command to remove IBM HTTP Server directories in the *HTTPServer_root*directory:

```
rm -rf HTTPServer_root
```

d. Use the installRegistryUtils command to examine the installation locations for all the installed IBM HTTP Server products and remove the products from the install registry that you want to uninstall.

Linux

a. Search for IBM HTTP Server related packages. Do not remove packages for IBM HTTP Server products that you are not uninstalling.

b. If there are packages to delete, enter the following command to remove any packages for the product that you are uninstalling.

```
rpm -e packagename
```

Alternatively, you can enter the following command that will list all the IBM HTTP Server packages and then verify that everything in the list is something you want to delete:

```
rpm -qa | grep IHS
```

If there is a problem with package dependencies, you can use the following command to remove the packages:

```
rpm -e packagename --nodeps --justdb
```

The nodeps option skips the dependency check. The justdb option updates only the package database, and not the file system. Using only the nodeps option can cause a failure in package removal if there is any mismatch in the dependent file system (files and directories).

c. enter the following command to remove IBM HTTP Server directories in the *HTTPServer_root* directory:

```
rm -rf HTTPServer_root
```

d. Edit the vpd.properties file. Do not delete or rename the vpd.properties file because the InstallShield MultiPlatform (ISMP) program uses it for other products that it installs. If the IBM HTTP Server product that you are uninstalling is the only product with entries in the vpd.properties file, you can delete this file.

e. Use the installRegistryUtils command to examine the installation locations for all installed IBM HTTP Server products and remove the products from the install registry that you want to uninstall.

Solaris

a. Search for IBM HTTP Server related packages. enter the following command to list all the packages for the IBM HTTP Server products:

```
pkginfo | grep IHS
```

If no packages appear when using these commands, skip the next step. The resulting list of packages has the following format:

- application WSIHS70
- IBM HTTP Server

b. Change directories to the directory where package information is registered:

```
cd /var/sadm/pkg
```

c. Issue the following command to remove any IBM HTTP Server related packages, for example:

```
pkgrm packagename1 packagename2 ...
```

Do not remove packages for IBM HTTP Server products that you are not uninstalling. Issue the following commands from the /var/sadm/pkg directory to search for and remove any IBM HTTP Server product related packages that are registered in the /var/sadm/pkg directory.

Change directories to the correct directory for IBM HTTP Server products:

```
cd /var/sadm/pkg
ls |grep IHS|xargs -i pkgrm -n {}
```

The package names for IBM HTTP Server are: WSIHS70 and WSIHS70LI. If there is a problem removing the packages, remove the related package directories in the /var/sadm/pkg directory, including the preremove files. For example, remove the following file before issuing the pkgrm -n WSIHS70 command:

```
/var/sadm/pkg/WSIHS70/install/preremove
```

d. To remove IBM HTTP Server directories in the *HTTPServer_root* directory, enter the following command:

```
rm -rf HTTPServer_root
```

e. Use the installRegistryUtils command to examine the installation locations for all installed IBM HTTP Server products and remove the products from the install registry that you want to uninstall.

Windows

a. Log on with Adminstrator privilege to complete the steps for updating the registry.

b. Verify that you have an Emergency Recovery Disk. Instructions for creating this disk are in the Windows help documentation. This step is a safeguard. This procedure does not require the recovery disk.

c. Use the regback.exe program from the Windows Resource Kit to back up the registry. This step is a safeguard. This procedure does not require the backup copy of the registry.

d. Delete product registry entries for the IBM HTTP Server product that you are uninstalling. Edit the Windows system registry by entering the theregedit.exe command from a command prompt.

Note: Edit the Registry carefully. You can easily make a mistake while using the registry editor to view and edit registry contents. The editor does not warn you of editing errors, which can be extremely dangerous. A corrupt registry can disrupt your system to the point where your only option is to reinstall the Windows operating system.

1) Press **Ctrl-F** to search for all instances of HTTP Server to determine whether you should delete each entry. You might not be able to remove all of the entries related to IBM HTTP Server, which is not a problem.

2) Expand and select keys related to IBM HTTP Server products and IBM HTTP Server Window Services listed below:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\HTTP Server\7.0.0.0
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IBMHTTPServer7.0
HKEY_LOCAL_MACHINE\SYSTEM\CurrrentControlSet\Services\IBMHTTPAdministration7.0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IBM HTTP Server 7.0.0.0
```

Note: Depending on how you installed the product, the following registry keys that are previously listed are only optionally included: `HKLM\...IBMHTTPServer7.0` and `HKLM\...IBMHTTPAdministration7.0`

3) Click **Edit** > **Delete** from the menu bar for each related key.

4) Click **Yes** when asked to confirm deletion of the key.

5) Click **Registry** > **Exit** from the menu bar when you are finished.

8.

e. Delete the installation root directory for the product that you are uninstalling.

f. Open a Windows Explorer window and browse to the C:\Documents and Settings\All Users\Start Menu\Programs directory. If you have an installation of a IBM HTTP Server product, delete the following directory folder: IBM HTTP Server V7.0.

g. Delete the %WINDIR%\IsUninst.exe file.

h. Edit the vpd.properties file. The file is located in the installation directory of the operating system, such as the C:\WINNT directory or the C:\windows directory. Do not delete or rename the vpd.properties file because the InstallShield MultiPlatform (ISMP) program uses it for other products that it installs. If the IBM HTTP Server product that you are uninstalling is the only product with entries in the vpd.properties file, you can delete this file.

i. Use the installRegistryUtils command to examine the installation locations for all installed IBM HTTP Server products and remove the products from the install registry that you want to uninstall.
j. Restart your machine.

Results

This procedure results in removing IBM HTTP Server from your system. There will be no trace of the previously deleted installation. You can reinstall IBM HTTP Server into the same directories after manually uninstalling the product.

Chapter 2. Migrating and installing IBM HTTP Server on z/OS systems

You can install the IBM HTTP Server for WebSphere Application Server product on z/OS® systems. Ensure that you order the current version of WebSphere Application Server that contains the IBM HTTP Server for z/OS.

Install IBM HTTP Server using SMP/E and then run an installation script to create a configuration in the installation directory:

Note: IBM HTTP Server for WebSphere Application Server is different from the HTTP Server for z/OS. The information contained within the IBM HTTP Server product documentation pertains to IBM HTTP Server for WebSphere Application Server, not the HTTP Server for z/OS.

- The product code for the IBM HTTP Server is installed into the WebSphere Application Server for z/OS Optional Materials tree. The default location is: /usr/lpp/zWebSphere_OM/V7R0/HTTP/Server.
- After you install the product code, choose an installation directory for each IBM HTTP Server instance that you want to run under z/OS. Each server instance requires its own installation directory.
- You can use an install script provided with IBM HTTP Server to: Copy files into this directory, perform initial customization, and create symbolic links to the product code directory.

You can use the Web server jobname or other identifier in the installation directory name. For example:

```
/opt/wwww/webserver1
/var/webservers/AAST1
```

In the instruction examples in the following topic for installing IBM HTTP Server, an installation directory of /etc/websrv1 is used.

Use these links to learn more about the installation features for **IBM HTTP Server**:

Installing IBM HTTP Server

This topic describes how to install IBM HTTP Server using the installer program.

Uninstalling IBM HTTP Server

This topic contains the procedure for uninstalling IBM HTTP Server.

Migrating the mod_auth_saf module directives to use the mod_authnz_saf module directives.

Use the mod_authnz_saf directive for your SAF configuration instead of the mod_auth_saf directive. In addition, SAF password authentication is now enabled by specifying the SAF basic authentication provider directive.

```
AuthBasicProvider saf
```

Also, the SAFRequire and AuthSAF directives are not supported in this release of IBM HTTP Server. For information about SAF directives, see the information center topic about SAF directives.

Migrating the mod_ibm_ldap module directives to use the mod_ldap module directives.

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 directives. 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_ldap module directives to ensure future support for your LDAP configuration.

Installing IBM HTTP Server

This topic describes how to install a running version (server instance) of the IBM HTTP Server for WebSphere Application Server on z/OS using the installer program.

Before you begin

Prior to using the installer program:

- Obtain and apply the current version of WebSphere Application Server that provides support of IBM HTTP Server for z/OS systems using SMP/E. The IBM HTTP Server code is installed into the WebSphere Application Servers Optional Materials tree. The default directory name is: /usr/lpp/IHSA/V7R0/server.
- 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 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
  ```
- The user running the installer must have write access to the selected installation directory. The user ID must have a valid OMVS segment, and the user ID must be the same as the user ID that the Web server runs as. (See the z/OS system configuration section referenced in the previous steps.)
- Set the umask value to `022`. To verify that the umask value is set to `022`, run the umask command.
- 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.

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

   ```
   /usr/lpp/IHSA/V7R0/server
   ```
2. 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.

   The installer program is: bin/install_ihs.

   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 thehttpd.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, you should point the symbolic links to the following default product directory: /usr/lpp/IHSA/V7R0/server. If you do not use the default product directory, you must invoke the installation script using its absolute path, such as /WebSphere/7.0/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.

3. 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.
4. 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/7.0.0.0 (Unix)
   Server built:   Jan  9 2008 11:20:34
   # bin/apachectl configtest
   Syntax OK
   ```

   The actual version string and build date will vary.

3. Start IBM HTTP Server.

   ```
   bin/apachectl start
   ```

4. Load the default Web page from the browser to confirm that IBM HTTP Server is operational: 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.
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_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 filename 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 Chapter 12, “Securing with SSL communications,” on page 67 and “SSL directives” on page 79.
  - Enable mod_status by removing the comment delimiters in the default configuration file which are highlighted below:

    ```
    <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.

Uninstalling IBM HTTP Server

This topic contains the procedure for uninstalling the IBM HTTP Server.

1. Stop IBM HTTP Server.
2. Go to the installation directory for the IBM HTTP Server instance.
3. Save any modified configuration files, such as `httpd.conf`.
4. Delete the IBM HTTP Server installation directory.

Results

The IBM HTTP Server uninstallation is now complete.

Chapter 3. 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.** The `MEMLIMIT` parameter controls the amount of virtual memory above 2 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 * (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 `WWWSERV` 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%%C CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
PERMIT CSFPK% CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
PERMIT CSFCK% CLASS(CSFSERV) ID(WWWSERV PUBLIC) ACCESS(READ)
SETROPTS CLASSACT(CSFSERV)
SETROPTS 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%%C UACC(READ)
RDEFINE CSFSERV CSFPK% UACC(READ)
RDEFINE CSFSERV CSFCK% UACC(READ)
SETROPTS CLASSACT(CSFSERV)
SETROPTS 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*.

Chapter 4. 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 platforms Migrating and installing IBM HTTP Server > Distributed operating systems
- z/OS Migrating and installing IBM HTTP Server on z/OS systems

Note: 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:

- Using the WebSphere Application Server administrative console
- AIX HP-UX Linux Solaris z/OS Using the command line interface
- Windows Using the Windows service
- 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

Note: You must start the IBM HTTP Server administrative server as a root user. This also, includes non-root installations of IBM HTTP Server.

Distributed platforms 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 IBM HTTP Server” on page 16.

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.

- **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)
- AIX 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 on getting started with IBM HTTP Server on Windows operating systems.

About this task

Start IBM HTTP Server as a Windows service as follows:

1. Click **Start > Programs > IBM HTTP Server > Start Server**. A message box is displayed that indicates the server has started.
2. 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 hlq.SIWOJCL(IWOAPROC) 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
```

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. In the examples below, 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=HTTPD
  ```

- To stop the server, enter:

  ```
  S WEBSRV1,ACTION='stop'
  ```

  Note: When using SDSF.LOG, 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 lower case.

- To issue other apachectl commands, enter:

```
S WEBSRV1,ACTION='<command>'
```

The output files for the start and stop commands are:

- *install_directory*/logs/proc.output
- *install_directory*/logs/proc.errors

Note: 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*.

Note: 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.

Note: 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.

Chapter 5. Configuring IBM HTTP Server

To configure the IBM HTTP Server, edit the `httpd.conf` configuration file.

- **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.

  IBM HTTP Server also provides the following configuration files:

  - Distributed platforms `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)
    - Distributed platforms `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 (\).

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 or are not supported.

Note: 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 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 procedure below about migrating from the mod_ldap and mod_auth_ldap module configurations.
- The Apache mod_example source is installed in the <ihsinst>/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.

The following table contains a list of Apache modules supported for IBM HTTP Server.

Module	Description	URL
core	Core Apache HTTP Server features	http://publib.boulder.ibm.com/httpserv/manual70/mod/core.html
Windows Windows mpm_winnt	Multi-processing module (MPM)	http://publib.boulder.ibm.com/httpserv/manual70/mod/mpm_winnt.html
AIX HP-UX Linux Solaris z/OS AIX HP-UX Linux Solaris z/OS worker	MPM	http://publib.boulder.ibm.com/httpserv/manual70/mod/worker.html
mod_actions	Provides for executing CGI scripts, based on media type or request method.	http://publib.boulder.ibm.com/httpserv/manual70/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/manual70/mod/mod_actions.html
mod_asis	Sends files that contain their own HTTP headers.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_asis.html
mod_auth_basic	Basic authentication	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_auth_basic.html
mod_authn_anon	Allows anonymous user access to authenticated areas.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authn_anon.html
mod_authn_dbm	User authentication using DBM files.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authn_dbm.html
mod_authn_default	Authentication fallback module	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authn_default.html
mod_authn_file	User authentication using text files	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authn_file.html
z/OS mod_authnz_ldap	Allows an LDAP directory to be used to store the database for HTTP basic authentication.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authnz_ldap.html
mod_authz_dbm	Group authorization using DBM files.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authz_dbm.html

mod_authz_default	Authorization fallback module	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authz_default.html
mod_authz_groupfile	Group authorization using text files	http://publib.boulder.ibm.com/httpserv/manual70/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/manual70/mod/mod_authz_host.html
mod_authz_user	User authorization	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authz_user.html
Windows z/OS 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/manual70/mod/mod_autoindex.html
AIX HP-UX Linux Solaris mod_cache	Content cache keyed to URIs	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_cache.html
AIX HP-UX Linux Solaris Windows z/OS mod_cgi	Execution of CGI scripts	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_cgi.html
AIX HP-UX Linux Solaris AIX HP-UX Linux Solaris mod_cgid	Execution of CGI scripts using an external CGI daemon.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_cgid.html
z/OS z/OS mod_charset_lite	Specifies character set translation or recoding.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_charset_lite.html
Distributed platforms Distributed platforms mod_dav	Distributed Authoring and Versioning (WebDAV) functionality. **Note:** 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/manual70/mod/mod_dav.html
Distributed platforms Distributed platforms mod_dav_fs	File system provider for mod_dav.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_dav_fs.html
mod_deflate	Compress content before it is delivered to the client.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_deflate.html
mod_dir	Provides for ″trailing slash″ redirects and serving directory index files.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_dir.html
mod_env	Modifies the environment which is passed to CGI scripts and SSI pages.	http://publib.boulder.ibm.com/httpserv/manual70/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/manual70/mod/mod_expires.html

Distributed platforms mod_ext_filter	Pass the response body through an external program before delivery to the client.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_ext_filter.html
Distributed platforms Distributed platforms 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. **Note:** The recommended caching mechanism for file handling is the CacheEnable feature of the mod_cache module.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_file_cache.html
Distributed platforms mod_filter	Specifies the context-sensitive smart filter configuration module.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_filter.html
mod_headers	Customization of HTTP request and response headers.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_headers.html
mod_imagemap	Server-side image map processing.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_imagemap.html
mod_include	Server-parsed HTML documents (Server Side Includes).	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_include.html
mod_info	Provides a comprehensive overview of the server configuration.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_info.html
z/OS mod_ldap	Provides LDAP connection pooling and result caching services for use by other LDAP modules.	http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_ldap.html
mod_log_config	Logging of the requests made to the server.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_log_config.html
mod_logio	Logging of input and output bytes per request.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_logio.html
mod_mem_cache	Content cache keyed to URIs.	http://publib.boulder.ibm.com/httpserv/manual70//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/httpserv/manual70//mod/mod_mime.html
Distributed platforms Distributed platforms 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. **Note:** 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/httpserv/manual70//mod/mod_mime_magic.html

mod_negotiation	Provides for content negotiation.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_negotiation.html
mod_proxy	HTTP, 1.1 proxy, and gateway server	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_proxy.html
mod_proxy_connect	Specifies the mod_proxy module extension for CONNECT request handling.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_proxy_connect.html
Distributed platforms Distributed platforms 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/httpserv/manual70//mod/mod_proxy_ftp.html
mod_proxy_http	Provides HTTP support for the mod_proxy module.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_proxy_http.html
mod_rewrite	Provides a rule-based rewriting engine to rewrite requested URLs.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_rewrite.html
mod_setenvif	Enables the setting of environment variables based on characteristics of the request.	http://publib.boulder.ibm.com/httpserv/manual70//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/httpserv/manual70//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/httpserv/manual70//mod/mod_speling.html
mod_status	Provides information on server activity and performance.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_status.html
mod_suexec	Allows CGI scripts to run as the specified user or group.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_suexec.html
mod_unique_id	Provides an environment variable with a unique identifier for each request.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_unique_id.html
mod_userdir	User-specific directories.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_userdir.html
mod_usertrack	Clickstream logging of user activity on a site.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_usertrack.html
mod_vhost_alias	Provides for dynamically configured mass virtual hosting.	http://publib.boulder.ibm.com/httpserv/manual70//mod/mod_vhost_alias.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: 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/httpserv/manual70/programs/ab.html
Windows Linux UNIX z/OS apachectl	Provides start, stop, and restart functionality for the Web server.	http://publib.boulder.ibm.com/httpserv/manual70/programs/apachectl.html
AIX HP-UX Linux Solaris z/OS Windows httpd.exe	Provides start, stop, and restart functionality for the Web server.	http://publib.boulder.ibm.com/httpserv/manual70/programs/httpd.html
Linux UNIX z/OS apxs	Builds plug-in modules.	http://publib.boulder.ibm.com/httpserv/manual70/programs/apxs.html
dbmmanage	Creates and updates user authentication files in DBM format for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual70/programs/dbmmanage.html
htdbm	Creates and updates user authentication files in DBM format for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual70/programs/htdbm.html
htpasswd	Creates and updates user authentication files for basic authentication.	http://publib.boulder.ibm.com/httpserv/manual70/programs/htpasswd.html
httxt2dbm	Creates DMB files for use with RewriteMap.	http://publib.boulder.ibm.com/httpserv/manual70/programs/httxt2dbm.html
logresolve	Resolves host names for IP addresses in Apache log files.	http://publib.boulder.ibm.com/httpserv/manual70/programs/logresolve.html
rotatelogs	Rotates log files without having to stop the server.	http://publib.boulder.ibm.com/httpserv/manual70/programs/rotatelogs.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.

MPM and addressing modes

The following table contains a list of platforms and the MPM and addressing modes supported on those platforms by IBM HTTP Server.

Platform	Addressing mode	MPM
AIX	32-bit	worker MPM
Windows HP-UX/PA-RISC	32-bit	worker MPM
AIX HP-UX Linux Solaris z/OS HP-UX/ia64	64-bit	worker MPM
Linux/x86	32-bit	worker MPM
Linux/PPC	32-bit	worker MPM
Linux/zSeries	32-bit	worker MPM
Solaris/SPARC	32-bit	worker MPM
Solaris/x64	64-bit	worker MPM
Windows	32-bit	WintNT MPM
z/OS	64-bit	worker 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 `AfpaPort 80`. Advanced fast path architecture (AFPA) is only supported for IPv4.

Configure Windows IPv6 networking before enabling the Listen directive for IPv6.

Chapter 6. Serving static content faster with Fast Response Cache Accelerator

The fast response cache accelerator (FRCA) can improve the performance of the IBM HTTP Server when serving static content, such as text and image files. Support for FRCA is on AIX and Windows systems only.

About this task

When FRCA is enabled, the default configuration setting allows all static files to be cached. The cache automatically loads during server operation so that individual files do not need to be listed. Use the AfpaCache directive to turn caching on or off for specific directories.

FRCA will remove files from the cache when they change to avoid serving stale content.

- To enable FRCA, edit the `httpd.conf` configuration file and delete the comment character (#) from the beginning of the LoadModule directive as follows:

  ```
  #LoadModule ibm_afpa_module modules/mod_afpa_cache.so
  ```

  becomes

  ```
  LoadModule ibm_afpa_module modules/mod_afpa_cache.so
  ```

- Enable IBM HTTP Server for dynamic page caching. You can use FRCA with WebSphere Application Server to cache certain dynamically-generated servlet and JavaServer Pages (JSP) files. This feature is only available on Windows versions of IBM HTTP Server.

  The `afpaplugin_22.dll` component that is compatible with IBM HTTP Server V7.0 must be configured by WebSphere Application Server.

  For details on how to enable this capability, see the external caching description in the WebSphere Application Server documentation.

Customizing Fast Response Cache Accelerator logging

The fast response cache accelerator (FRCA) can improve the performance of the IBM HTTP Server when serving static content, such as text and image files. By default, FRCA generates an access log of all requests that are served out of the cache. In order to minimize the effect of logging on performance, this is a separate file from the normal Apache access log.

About this task

When FRCA is enabled, the default configuration setting allows all static files to be cached. The cache automatically loads during server operation so that individual files do not need to be listed. Use the AfpaCache directive to turn caching on or off for specific directories.

FRCA will remove files from the cache when they change to avoid serving stale content.

Enable the FRCA access log if you want to maintain a record of requests served by FRCA. Requests that are not served out of the cache will be logged in the FRCA access log file. The FRCA access log file provides a useful way to verify that caching is enabled and to identify cached files.

Note: Even though a particular file might be cached, it might not always be served from the cache. Therefore, not every request for a cached file will result in an FRCA access log entry.

If you do not need access logging, turn the logging off for better performance.

- To turn FRCA logging off, edit the `httpd.conf` configuration file.

AIX Configure AfpaLogging `off`.

Windows Insert a comment character (#) at the beginning of the AfpaLogFile line. For example:

```
#AfpaLogFile "_path_to_server_/logs/afpalog" V-ECLF
```

- For each request that is served by the fast response cache accelerator, a log entry in the access log displays the following:
  - Source host address
  - Windows Date and time of the request
  - HTTP method of the request and what is requested
  - HTTP return code, which indicates whether the request is honored
  - Size of the returned data

  A log entry can also optionally display the following:
  - Target virtual host (use the formatting option V-CLF or V-ECLF)
  - HTTP referer (use the formatting option V-CLF or V-ECLF)
  - HTTP user agent (use the formatting option V-CLF or V-ECLF)

  Note: The log file has a date stamp that automatically appends to its name. Everyday at midnight the server closes the current access log and creates a new one. This action enables the log file to process without having to stop and restart the server. Under heavy load conditions the log file can grow rapidly. Provide sufficient space on the hard drive for storage.

Restrictions on cached content

This section discusses the caching restrictions for the fast response cache accelerator (FRCA).

Caching does not occur on the following page types:

- Default welcome pages
- Requests ending in ″/″
- Access-protected documents and pages requested over Secure Sockets Layer (SSL)

Caching limitations exist for the following situations:

- FRCA supports only limited multi-language content negotiation. Caching occurs for only a single language version, where a given URL maps to multiple translated versions.
- FRCA must not be used with locally-mounted network file systems, such as Network File System (NFS) or Windows shared drives.
- FRCA does not cache proxied content.

Fast Response Cache Accelerator operational restrictions

This section discusses the operational restrictions for the fast response cache accelerator (FRCA).

The following operational restrictions apply:

- When FRCA is enabled, the default value of 0 for the MaxRequestsPerChild directive should be used, because graceful server restart is not supported with the cache accelerator.
- FRCA does not support Windows 64-bit operating systems.
- FRCA cannot be used when certain antivirus software is enabled. Currently Norton Antivirus has been identified as one such program.
- FRCA access log entries are not integrated with the Apache access log.
- Only access logging facilities exist for monitoring FRCA.
- On a given machine, only one instance of the IBM HTTP Server can have FRCA enabled.

- Do not install the IBM HTTP Server on a machine running the IBM Netfinity® Web Server Accelerator.

Servlets and JavaServer Pages files caching

You can use the fast response cache accelerator (FRCA) with WebSphere Application Server to cache certain dynamically-generated servlet and JavaServer Pages (JSP) files. This feature is only available on Windows versions of IBM HTTP Server.

Enable IBM HTTP Server for dynamic page caching by enabling the cache accelerator. In addition, the `afpaplugin_22.dll` component that is compatible with IBM HTTP Server V7.0 must be configured by WebSphere Application Server.

For details on how to enable this capability, see Configuring high-speed external caching through the Web server in the WebSphere Application Server product documentation.

AIX considerations for Fast Response Cache Accelerator (FRCA)

There are special considerations when using FRCA on AIX platforms. The FRCA kernel extension must load before starting IBM HTTP Server with FRCA enabled. Also, increasing the upper-bound limit of the percentage of CPU time that the FRCA kernel extension can spend in the interrupt (high priority) context is not recommended.

The following items must be considered when you use fast response cache accelerator (FRCA) on AIX platforms:

- The FRCA kernel extension must load before starting IBM HTTP server with FRCA enabled. To do this, issue the **frcactrl load** command. This is normally configured to run whenever the system boots and before IBM HTTP Server starts. See the AIX man pages for more details about the **frcactrl** command.
- In order to place an upper bound on the percentage of CPU time that the FRCA kernel extension can spend in its interrupt (high priority) context, use the **frcactrl pctonintr** command. Increasing this above the default value of 80% is not recommended in order to allow other applications a reasonable amount of time to execute. Decrease this value if more time needs to be allocated to other applications, but note that reducing the value will result in more cache misses, even if a file is in the cache.

AFPA directives

These configuration parameters control the Advanced Fast Path Architecture (AFPA) feature in IBM HTTP Server.

The fast response cache accelerator (FRCA) utilizes a special high-performance component, based on the IBM Advanced Fast Path Architecture, from which the AFPA prefix is derived. You can configure FRCA for IPV4. IPV6 is not supported.

- “AfpaBindLogger directive” on page 42
- “AfpaCache directive” on page 42
- “AfpaDynacacheMax directive” on page 42
- “AfpaEnable directive” on page 42
- “AfpaLogFile directive” on page 43
- “AfpaLogging directive” on page 43
- “AfpaMaxCache directive” on page 43
- “AfpaMinCache directive” on page 43
- “AfpaPort directive” on page 44
- “AfpaRevalidationTimeout directive” on page 44
- “AfpaSendServerHeader” on page 44

AIX

AfpaBindLogger directive

Use the AfpaBindLogger directive to bind the fast response cache accelerator (FRCA) logging thread in the kernel to a specific processor.

The format of the command is AfpaBindLogger [-1, 0, 1, ..., n], where -1 leaves the logging thread unbound and a number from 0 to total number of processors on the system, binds the logging thread to that processor.

Syntax	AfpaBindLogger [-1,0,1,..,n]
Scope	One per physical Apache server
Default	(-1)
Notes	Valid on AIX operating systems only.

AIX Windows

AfpaCache directive

The AfpaCache directive turns the fast response cache accelerator (FRCA) on or off for a particular scope (such as a directory). The AfpaCache directive applies to all descendants in a scope, unless otherwise modified by another directive.

Scope	Server configuration, virtual host, directory
Syntax	On or off
Usage	AfpaCache on
Override	Options
Multiple instances in the configuration file	Allowed
Notes	Valid on Windows 32-bit and AIX operating systems.

Windows

AfpaDynacacheMax directive

The AfpaDynacacheMax directive is used on Windows operating systems to control the total amount of memory utilized for caching servlets and JavaServer Pages files.

When static files are cached, there is very little overhead for each entry since the file itself does not take up space in the cache, just the file handle. However, for servlets and JavaServer Pages files, the body of the response is stored in physical memory, so care must be taken to avoid consuming all available memory. Without this directive, the fast response cache accelerator will automatically set the upper bound to be approximately one eighth of physical memory. Use the directive to override that default.

Syntax	AfpaDynacacheMax size (Megabytes)
Scope	One per physical Apache server
Notes	Valid on Windows 32-bit operating systems

AIX Windows

AfpaEnable directive

The AfpaEnable directive enables the fast response cache accelerator (FRCA). If the AfpaEnable directive is present and mod_afpa_cache.so is loaded, FRCA listens on the port specified by the AfpaPort directive.

Syntax	AfpaEnable
Scope	One per physical Apache server
Notes	Valid on AIX and Windows operating systems.

AIX Windows

AfpaLogFile directive

The AfpaLogFile directive defines the fast response cache accelerator (FRCA) log file name, location, and logging format.

Scope	One per physical Apache server
Syntax	AfpaLogFile *log_file_name* [CLF | ECLF | V-CLF | V-ECLF| BINARY]
Notes	Valid on AIX and Windows 32-bit operating systems. On Windows 32-bit operating systems, the current date is used as the file type for the log file, and the log file is automatically rolled over at midnight each day.

The log formats are as follows:

- CLF = Common Log Format
- ECLF = Extended Common Log Format
- V-CLF = Common Log Format with virtual host information
- V-ECLF = Extended Common Log Format with virtual host information
- AIX BINARY = Binary log with virtual host information

AIX

AfpaLogging directive

The AfpaLogging directive turns the fast response cache accelerator (FRCA) logging on or off.

Scope	One per physical Apache server
Syntax	AfpaLogging On | Off
Notes	Valid only on AIX operating systems.

AIX

AfpaMaxCache directive

The AfpaMaxCache directive specifies the maximum file size inserted into the fast response cache accelerator (FRCA) cache.

Syntax	AfpaMaxCache [size (bytes)]
Scope	One per physical Apache server
Default	none
Notes	Valid only on AIX operating systems.

AIX

AfpaMinCache directive

The AfpaMinCache directive specifies the minimum file size inserted into the fast response cache accelerator (FRCA) cache.

Syntax	AfpaMinCache [size]
Scope	One per physical Apache server
Default	none
Notes	Valid only on AIX operating systems.

AIX Windows

AfpaPort directive

The AfpaPort directive tells the FRCA on which TCP port to listen. The AfpaPort directive issues a listen command for all TCP network adapters that are active on the server machine. The listen command is effective for all TCP addresses.

Syntax	AfpaPort *port number*
Scope	One directive per server
Notes	Valid only on AIX and Windows 32-bit operating systems

AIX

AfpaRevalidationTimeout directive

The AfpaRevalidationTimeout directive sets the time interval for revalidation of a cached object. When the RevalidationTimeout is exceeded for a cached object, a fresh copy is cached.

Syntax	AfpaRevalidationTimeout [value]
Scope	Global
Default	60 seconds
Notes	Valid on AIX operating systems only.

AIX

AfpaSendServerHeader

The AfpaSendServerHeader directive specifies whether or not the fast response cache accelerator (FRCA) sends the HTTP Server header in the response.

Syntax	AfpaSendServerHeader true or false
Scope	One per physical Apache server
Default	true
Notes	Valid only on AIX operating systems.

Chapter 7. 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.

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"
- "FastCGIAccessCheckerAuthoritatve directive" on page 47
- "FastCGIAuthenticator directive" on page 47
- "FastCGIAuthenticatorAuthoritative directive" on page 48
- "FastCGIAuthorizer directive" on page 48
- "FastCGIAuthorizerAuthoritative directive" on page 49
- "FastCGIConfig directive" on page 50
- "FastCGIExternalServer directive" on page 51
- "FastCGIIpcDir directive" on page 52
- "FastCGIServer directive" on page 53
- "FastCGIsuEXEC directive" on page 54

FastCGIAccessChecker directive

The FastCGIAccessChecker directive defines a FastCGI application as a per-directory access validator.

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.

FastCGIAccessCheckerAuthoritatve directive

The FastCGIAccessCheckerAuthoritatve directive enables access checking passing to lower level modules.

Syntax	`FastCGIAccessCheckerAuthoritative On | Off`
Scope	directory, location
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.

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 `FCGI_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.

Syntax	`FastCGIAuthenticatorAuthoritative` *`On`* | *`Off`*
Scope	directory
Default	`FastCgiAuthenticatorAuthoritative On`
Module	mod_fastcgi
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.

Syntax	`FastCgiAuthorizer file name` *`[-compat]`*
Scope	directory
Default	None
Module	mod_fastcgi
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.

Syntax	`FastCgiAuthorizerAuthoritative file name` *`On`* | *`Off`*
Scope	directory
Default	`FastCgiAuthorizerAuthoritative file name On`
Module	mod_fastcgi
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.

Syntax	`FastCgiConfig option option...` The FastCgiConfig directive does not affect static or external applications.
Scope	directory
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes
Values	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.
- **multiThreshhold 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, singleThreshhold 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.
- **singleThreshhold 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.

Syntax	`FastCgiExternalServer` *`file name -host hostnameport [-appConnTimeout n]`* `FastCgiExternalServer` *`file name -socket file name [-appConnTimeout n]`*
Scope	Server configuration
Default	None
Module	mod_fastcgi
Multiple instances in the configuration file	yes

Values

- **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.

FastCGIIpcDir directive

The FastCGIIpcDir directive specifies directory as the place to store the UNIX socket files used for communication between the applications and the Web server.

Syntax

- On UNIX platforms - FastCgiIpcDir *directory*
- On Windows operating systems - FastCgiIpcDir *name*

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.

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 platforms

FastCGIsuEXEC directive

The FastCGIsuEXEC directive supports the suEXEC-wrapper.

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.

Chapter 8. Managing 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: 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.

- **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 platforms **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 on the unmanaged node. The administration server must be started and defined to an unmanaged WebSphere Application Server node. (Remote administration of IBM HTTP Server is available without the administration server if the IBM HTTP Server is installed on a machine with a managed node.)

- Distributed platforms **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 platforms "Starting and stopping the IBM HTTP Server administration server" on page 61
This topic describes how to start and stop the IBM HTTP Server administration server on distributed platforms.

Chapter 9. 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 platforms

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.

Note: 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.

- 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 AIX V5.0.2.3: C or VisualAge® C++ Professional
- 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 V2.2.4 and build the module for Apache V2.2.4, 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.

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.

Chapter 10. Administering and configuring 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” on page 1.

About this task

Start the IBM HTTP Server administration server as follows:

- 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
     ```

     Note: 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.

- Enable access to the administration server using the htpasswd utility
- Run the setupadm script for the administration server
- 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.

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 script for the administration server

When using the IBM HTTP Server administration server, the setupadm script establishes permissions for configuration file updates.

Before you begin

You must run the setupadm command if you are installing IHS as a non-root user. The setupadm command is run in the <IHS_HOME>/bin directory so that you can properly use the administrative server with the WebSphere Application Server. The format for the command is as follows (on one line):

```
setupadm -usr <userName> -grp <groupName> -cfg <IHS Web server configuration file> -adm
<IHS admistrative server configuration file> -plg <plug-in configuration file>
```

About this task

When using the administration server, you cannot update the configuration files after a default server installation, unless you run the setupadm script, or you set permissions manually.

The setupadm script prompts you for the following input:

- **User ID** - The user ID that you use to log on to the administration server. The script creates this user ID.

- **Group name** - The administration server accesses the configuration files and authentication files through group file permissions. The script creates the specified group through this script.
- **Directory** - The directory where you can find configuration files and authentication files.
- **File name** - The following file groups and file permissions change:
  - Single file name
  - File name with wildcard
  - All (default) - All of the files in the specific directory
  - Processing - The setupadm script changes the group and file permissions of the configuration files and authentication files.

What to do next

The administration server requires read and write access to configuration files and authentication files to perform Web server configuration data administration. In addition to the Web server files, you must change the permissions to the targeted plug-in configuration files. See Setting permissions manually for instructions.

The administration server has to invoke **apachectl restart** as root to perform successful restarts of the IBM HTTP Server.

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.

- 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 11. Task overview: Securing IBM HTTP Server

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.

- Chapter 12, “Securing with SSL communications,” on page 67. For secure communication, you can set up the Secure Sockets Layer (SSL) directives in the default `httpd.conf` configuration file.
- Chapter 13, “Setting advanced SSL options,” on page 95. 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 platforms Chapter 14, “Managing keys with the IKEYMAN graphical interface (Distributed systems),” on page 101. 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 platforms Chapter 15, “Managing keys with the gsk7cmd command line interface (Distributed systems),” on page 111. You can use IKEYCMD, which is the Java command line interface to IKEYMAN. Use the command line only if you are unable to implement the graphical user interface.
- z/OS Chapter 16, “Managing keys with the native key database gskkyman (z/OS systems),” on page 123 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.
- Chapter 17, “Getting started with the cryptographic hardware for SSL (Distributed systems),” on page 125. You can use cryptographic hardware for SSL. The IBM 4758 requires the PKCS11 software for the host machine and internal firmware.
- Distributed platforms Chapter 18, “Authenticating with LDAP on IBM HTTP Server using **mod_ibm_ldap** (Distributed systems),” on page 131. You can configure LDAP to protect files on IBM HTTP Server.
- z/OS Chapter 19, “Authenticating with LDAP on IBM HTTP Server using **mod_ldap**,” on page 157 You can configure LDAP to protect files on IBM HTTP Server.
- z/OS Chapter 20, “Authenticating with SAF on IBM HTTP Server (z/OS systems),” on page 159. You can provide IBM HTTP Server with user authentication using the System Authorization Facility security product.

Results

Your IBM HTTP Server is secured.

Chapter 12. Securing with SSL communications

This section provides information to help you set up Secure Sockets Layer (SSL), using the default `httpd.conf` configuration file.

1. Distributed platforms Use the IBM HTTP Server IKEYMAN utility (graphical user interface) or IKEYMAN utility (command line) to create a CMS key database file and self-signed 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 self-signed 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 Chapter 20, “Authenticating with SAF on IBM HTTP Server (z/OS systems),” on page 159 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
<IfModule mod_ibm_ssl.c>
  Listen 443
  <VirtualHost *:443>
    SSLEnable
  </VirtualHost>
</IfModule>
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

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. SSL Version 3, requires authentication of the client identity.

When your server has a digital certificate, SSL-enabled browsers like Netscape Navigator and Microsoft Internet Explorer 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

This topic provides information on Secure Sockets Layer certificates.

Distributed platforms Use the IBM HTTP Server IKEYMAN utility to create a CMS key database file and self-signed 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 self-signed 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 platforms 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 self-signed certificates.

z/OS You can use the native z/OS key management (gskkyman key database) to create self-signed 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

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

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 gsk7cmd command.

The following is an example of how to use the gsk7cmd 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/gsk7cmd  -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

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.

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.
- 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] <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.

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

Obtaining certificates

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.

- **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 70.

  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 platforms IKEYMAN graphical user interface. If you are unable to use the IKEYMAN interface, use the command line interface gsk7cmd 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

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

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 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 platforms In most cases, you will not need to take an additional configuration steps to effectively use SSL session ID caching in IBM HTTP Server.

z/OS It is recommended that you disable IBM HTTP Server session ID caching (**sidd**). The z/OS System SSL provides an equivalent function that can perform better with some additional configuration.

- Disable the IBM HTTP Server **sidd** with the SSLCacheDisable directive and remove any existing SSLCacheEnable directives in httpd.conf.
- Enable ″SSL Started Task″ for z/OS System SSL. For more information on the following setup instructions, refer to the section ″SSL Started Task″ in z/OS *Cryptographic Services System Secure Sockets Layer (SSL) Programming* (SC24-5901), which you can link to from the *z/OS Internet Library*:
  - Set the following environment variables in bin/envars:
    - GSK_V3_SIDCACHE_SIZE=2048
    - GSK_V2_SIDCACHE_SIZE=2048
    - GSK_SYSPLEX_SIDCACHE=ON
    - export GSK_V3_SIDCACHE_SIZE GSK_V2_SIDCACHE_SIZE GSK_SYSPLEX_SIDCACHE
  - Configure the limits in the started task by editing /etc/gskssl/server/envar.
    - GSK_LOCAL_THREADS
    - GSK_SIDCACHE_SIZE

SSL directive considerations

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 71.

Authentication

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 platforms You can add or remove trusted CAs using the IBM Key Management utility (ikeyman) that is included with your server.

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

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

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 1. 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 `getenv ("HTTPS")` in C or `$ENV{'HTTPS'}` in Perl. For a SSL environment variable to be used in CGI or FastCGI, there must be a corresponding PassEnv directive.
access from a plug-in module	The information is available in the subprocess_env table after the quick handler has run. Access it with a call such as `apr_table_lookup (r->subprocess_env,"HTTPS")`
logging into the access log with other information about the request	Use the following `%{`*varname*`}e` example. `LogFormat "%h %l %u %t \ "%r\ " %>s %b %{HTTPS}e" ssl-custom` If the information is not available, mod_log_config logs a dash (-) for the field.
use with the setenvif variable	# Silly example, don't compress SSL connections `SetEnvIf HTTPS no-gzip`

use as part of a mod_rewrite rule variable	`RewriteEngine  On`
	`RewriteCond  %{ENV:HTTPS} ^OFF$`
	`RewriteRule  .*  /no-sssl.html`

access in an SSI document	In order for an SSL environment variable to be used in an SSI document, there must be a corresponding PassEnv directive. `SSL is <!--#echo var="HTTPS" -->`
access control	`Allow from env=HTTPS`

SSL handshake environment variables

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

The following 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.
HTTPS_CIPHER	Contains the cipher used in the SSL handshake.	See the table below.
HTTPS_KEYSIZE	Indicates the size of the key.	See the table below.
HTTPS_SECRETKEYSIZE	Indicates the strength of the key.	See the table below.
SSL_PROTOCOL_VERSION	Contains the protocol version.	String contains either `SSLV2`, `SSLV3`, or `TLSV1`.

The following table provides a list of variables for HTTPS_KEYSIZE and HTTPS_SECRETKEYSIZE in Secure Sockets Layer V3 and Transport Layer Security V1

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

The following table provides a list of variables for HTTPS_CIPHER in Secure Sockets Layer V2.

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

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 certifiate	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

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_SESSIONID	Contains the session ID	String
SSL_CLIENT_ST	Contains the client certificate state or province	String

SSL directives

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.

Note: 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 80
- “SSLOCSPEnable” on page 80
- “Keyfile directive” on page 81
- “SSLAcceleratorDisable directive” on page 81
- “SSLAllowNonCriticalBasicConstraints directive” on page 82
- “SSLCacheDisable directive” on page 82
- “SSLCacheEnable directive” on page 82
- “SSLCacheErrorLog directive” on page 83
- “SSLCachePath directive” on page 83

- “SSLCachePortFilename directive” on page 83
- “SSLCacheTraceLog directive” on page 83
- “SSLCipherBan directive” on page 84
- “SSLCipherRequire directive” on page 84
- “SSLCipherSpec directive” on page 84
- “SSLClientAuth directive” on page 85
- “SSLClientAuthGroup directive” on page 85
- “SSLClientAuthRequire directive” on page 86
- “SSLCRLHostname directive” on page 87
- “SSLCRLPort directive” on page 88
- “SSLCRLUserID directive” on page 88
- “SSLDisable directive” on page 89
- “SSLEnable directive” on page 89
- “SSLFakeBasicAuth directive” on page 89
- “SSLFIPSDisable directive” on page 89
- “SSLFIPSEnable directive” on page 90
- “SSLPKCSDriver directive” on page 90
- “SSLProtocolDisable directive” on page 90
- “SSLProxyEngine directive” on page 91
- “SSLServerCert directive” on page 91
- “SSLStashfile directive” on page 92
- “SSLTrace directive” on page 92
- “SSLV2Timeout directive” on page 93
- “SSLV3Timeout directive” on page 93
- “SSLVersion directive” on page 93

SSLOCSPResponderURL

Enables checking of client certificates through a statically configured online certificate status protocol (OCSP) responder.

Syntax	 SSLOCSPResponderURL<*URL*>
Scope	Virtual host
Default	Disabled
Module	mod_ibm_ssl
Multiple instances in the configuration file	Multiple instances permitted for each virtual host
Values	A fully qualified URL that points to an OCSP responder, for example, http://hostname:2560/. The path portion of the URL is not used when submitting OCSP requests.

If SSLOCSPResponderURL is set, IHS uses the supplied URL to check for certificate revocation status when an SSL client certificate is provided.

If CRL checking is configured, CRL checking is performed before any OCSP checking. OCSP checking only occurs if the result of the CRL 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, IHS checks OCSP responders as described above for SSLOCSPEnable.

SSLOCSPEnable

Enables checking of client certificates through OCSP responders defined in the Authority Information Access (AIA) extension of their certificate.

Syntax	Distributed platforms `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, IHS contacts the OCSP responder indicated by the AIA extension to check revocation status of the client certificate. The path portion of the URL is ignored.

If CRL checking is configured, CRL checking is performed before any OCSP checking. OCSP checking only occurs if the result of the CRL 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, IHS checks OCSP responders as described above for SSLOCSPEnable.

Keyfile directive

The keyfile directive sets the key file to use.

Note: This directive might be overridden by the base server configuration.

Syntax	AIX Solaris Linux Windows `Keyfile [/prompt] /fully qualified path to key file/keyfile.kdb` **Note:** 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.
Scope	Global base and virtual host
Default	None
Module	`mod_ibm_ssl`
Multiple instances in the configuration file	One instance per virtual host and global server
Values	File name of the key file. Distributed platforms Use the prompt option to enable the HTTP server to prompt you for the Key file password during start up.

SSLAcceleratorDisable directive

The SSLAcceleratorDisable directive disables the accelerator device.

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 platforms

SSLAllowNonCriticalBasicConstraints directive

The SSLAllowNonCriticalBasicConstraints directive allows compatibility with one aspect of the GPKI specification from the government of Japan that conflicts with RFC3280.

Syntax	`SSLAllowNonCriticalBasicConstraints` *on*|*off*
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 will not cause a validation failure. This allows compatibility with one aspect of the GPKI specification from the government of Japan that conflicts with RFC3280. **Note:** 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.

Syntax	`SSLCacheDisable`
Scope	One per physical Apache server instance, allowed 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.

Syntax	`SSLCacheEnable`
Scope	One per physical Apache server instance, allowed 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.

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.

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.

Syntax	`SSLCachePath /usr/HTTPServer/logs/sidd`
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.
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 name.

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.

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.
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 will fail with a 403 status code.

Note: 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.

Syntax	`SSLCipherBan <cipher_specification>`
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 “SSL Version 2 cipher specifications” on page 97 and “SSL Version 3 and TLS Version 1 cipher specifications” on page 98.

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 will fail with a ’403’ status code.

Note: 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.

Syntax	`SSLCipherRequire <cipher_specification>`
Scope	Multiple instances per directory stanza.
Default	None.
Module	`mod_ibm_ssl`
Multiple instances in the configuration file	Permitted per directory stanza.
Values	See “SSL Version 2 cipher specifications” on page 97 and “SSL Version 3 and TLS Version 1 cipher specifications” on page 98.

SSLCipherSpec directive

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.

Syntax	`SSLCipherSpec` *short name* or `SSLCipherSpec` *long name*
Scope	Virtual host.
Default	If nothing is specified, the server uses all of the cipher specifications available from the installed GSK library.
Module	`mod_ibm_ssl`
Multiple instances in the configuration file	Permitted. Order of preference is top to bottom, first to last. If the client does not support the cipher specifications, the connection closes.

Values

See “SSL Version 2 cipher specifications” on page 97 and “SSL Version 3 and TLS Version 1 cipher specifications” on page 98.

SSLClientAuth directive

The SSLClientAuth directive sets the mode of client authentication to use (none (0), optional (1), or required (2)).

Syntax `SSLClientAuth <level required> [crl]`

Scope Virtual host.

Default `SSLClientAuth none`

Module `mod_ibm_ssl`

Multiple instances in the configuration file One instance per virtual host.

Values

- 0/None: No client certificate requested.
- 1/Optional: Client certificate requested, but not required.
- 2/Required: Valid client certificate required.
- CRL: Turns crl on and off inside an SSL virtual host. If you use certificate revocation list (CRL), you need to 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.

If you specify the value 0/None, you cannot use the CRL option.

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 will allow access to the protected resource.

Syntax `SSLClientAuthGroup group name attribute expression`

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 IBMpeople Org = IBM
```

Description of valid logical expressions. 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 !). Use parentheses to group comparisons. If the value of the attribute contains a nonalphanumeric character, you must delimit the value with quotes.

The following is a list of the attribute values that you can specify for this directive:

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 to specify the desired grouping of client certificate attribute values. 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 will allow access to the protected resource.

Syntax	`SSLClientAuthRequire attribute expression`
Scope	server config, virtual host
Default	None.
Module	`mod_ibm_ssl`
Multiple instances in the configuration file	Permitted. The function joins these directives by ″AND″.
Override	AuthConfig
Values	Logical expression consisting of attribute checks linked with AND, OR, NOT, and parentheses. For example: `SSLClientAuthRequire group != IBMpeople && ST = M`

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 may 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 following is a list of the attribute values that you can specify for this directive:

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 to specify the desired grouping of client certificate attribute values. Valid operators include '=' and '!='. The user can also specify a group name, that is configured using the "SSLClientAuthGroup directive" on page 85, 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 85 for more information.

SSLCRLHostname directive

The SSLCRLHostname directive specifies the TCP/IP name or address of LDAP server where the Certificate Revocation List (CRL) database resides.

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.

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.

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 needed 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.

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.

Note: 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.

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.

Note: This directive might be overridden by the base server configuration.

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 platforms

SSLFIPSDisable directive

The SSLFIPSDisable directive disables Federal Information Processing Standards (FIPS).

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.

Distributed platforms

SSLFIPSEnable directive

The SSLFIPSEnable directive enables Federal Information Processing Standards (FIPS).

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.

Note: See also "SSL Version 3 and TLS Version 1 cipher specifications" on page 98.

Distributed platforms

SSLPKCSDriver directive

The SSLPKCSDriver directive identifies the fully qualified name to the module, or driver used to access the PKCS11 device.

Syntax	*Fully qualified name to module used to access PKCS11 device*>. If the module exists in the user's 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, platform:

- 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 allows 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.

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 TLSv1

The following example disables support for multiple protocols on a virtual host.

```
<VirtualHost *:443>
SSLEnable
SSLProtocolDisable SSLv2 SSLv3
(any other directives)
</VirtualHost>
```

Note: SSL0230I is logged for each SSL connection attempt if the client and server do not share at least one protocol and cipher combination.

SSLProxyEngine directive

The SSLProxyEngine toggles whether the server will use SSL for proxied connections. `SSLProxyEngine` *on* is required if your server is acting as a reverse proxy for an SSL resource.

Syntax	`SSLProxyEngine` *on*|*off*
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	*on*|*off*

SSLServerCert directive

The SSLServerCert directive sets the server certificate to use for this virtual host.

Syntax	`SSLServerCert [prompt]` *my_certificate_label;* `on PKCS11 device - SSLServerCert` *mytokenlabel:mykeylabel*
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	Certificate label. Use the /prompt option to enable the HTTP server to prompt you for the Crypto token password during start up. Use no delimiters around the certificate label. Ensure that the label is contained on one line; leading and trailing white space is ignored.

SSLStashfile directive

The SSLStashfile directive indicates path to file with file name containing the encrypted password for opening the PKCS11 device.

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 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] <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.

Note: See also “SSL certificate revocation list” on page 71.

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.

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. We do not recommend enabling this directive under normal working conditions.

Syntax	`SSLTrace`
Scope	Global
Default	mod_ibm_ssl debug logging in not enabled
Module	`mod_ibm_ssl`
Multiple instances in the configuration file	Ignored
Values	None

Note: See also LogLevel Directive.

SSLV2Timeout directive

The SSLV2Timeout directive sets the timeout for SSL Version 2 session IDs.

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.

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 enables object access rejection, if the client attempts to connect with an SSL protocol version other than the one specified.

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.
Values	SSLV2|SSLV3|TLSV1|ALL

Chapter 13. Setting advanced SSL options

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:

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.

   Note: 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

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 level of client authentication:

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.

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.
Required	The server requires a valid certificate from all clients.

   For example, `SSLClientAuth required`.

   If you want to use a certificate revocation list (CRL), add `crl`, as a second argument for SSLClientAuth. For example: `SSLClientAuth required crl`.
2. Save the configuration file and restart the server.

Choosing the type of client authentication protection

If you enable client authentication, the server validates clients by checking for trusted certificate authority (CA) root certificates in the local key database.

Before you begin

By default, the IBM HTTP Server enables the cache accelerator.

About this task

For each virtual host, choose the type of client authentication:

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 displayedlast in the module list.
2. Save the configuration file and restart the server.

Setting cipher specifications

This topic describes setting cipher specifications for secure transactions.

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 which are supported by both IBM HTTP Server and the client.

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.

1. Specify a value for each virtual host stanza in the configuration file that are on the SSLCipherSpec directive, as in the following examples: `SSLCipherSpec` *short_name* or `SSLCipherSpec` *long_name*, where *short_name* and *long_name* represent the name of “SSL Version 2 cipher specifications” on page 97 or “SSL Version 3 and TLS Version 1 cipher specifications” on page 98.
2. Save the configuration file and restart the server.

What to do next

If IBM HTTP Server uses a Verisign Global Server ID for SSL transactions, a 40-bit encryption browser can get a connection to a server at 128-bit encryption. This connection does not work for someone using Internet Explorer 5.01x. You can fix this situation by adding the following directives to the IBM HTTP Server configuration file (add the directives in the order shown):

- SSLCipherSpec 34
- SSLCipherSpec 35
- SSLCipherSpec 3A
- SSLCipherSpec 33
- SSLCipherSpec 36
- SSLCipherSpec 39
- SSLCipherSpec 32
- SSLCipherSpec 31
- SSLCipherSpec 30

Viewing cipher specifications

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.

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. Look in 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
   ```

   Look in 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 example LogFormat directive above:

   ```
   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 Version 2 cipher specifications

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.

SSL V2

If you specify V3 or TLS ciphers, and you do not specify SSL V2 ciphers, then SSL V2 support is disabled. If you specify SSL V2 ciphers, and do not specify SSL V3 or TLS ciphers, then SSL V3 and TLS support is disabled.

Short name	Long name	Description

27	SSL_DES_192_EDE3_CBC_WITH_ MD5	Triple-DES (168-bit)
21	SSL_RC4_128_WITH_MD5	RC4 (128-bit)
23	SSL_RC2_CBC_128_CBC_WITH_ MD5	RC2 (128-bit)
26	SSL_DES_64_CBC_WITH_MD5	DES (56-bit)
22	SSL_RC4_128_EXPORT40_WITH_ MD5	RC4 (40-bit)
24	SSL_RC2_CBC_128_CBC_ EXPORT40_WITH_MD5	RC2 (40-bit)

SSL Version 3 and TLS Version 1 cipher specifications

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.

SSL V3 and TLSV1

If you specify V3 or TLS ciphers, and you do not specify SSL V2 ciphers, then SSL V2 support is disabled. If you specify SSL V2 ciphers, and you do not specify SSL V3 or TLS ciphers, then SSL V3 and TLS support is disabled.

Note: In accordance with the NIST change for 19 May 2007, IBM HTTP Server does not support single-DES ciphers in FIPS mode for versions 6.0.2.1 or later and 6.0.1.11 or later.

Short name	Long name	Description
3A	SSL_RSA_WITH_3DES_EDE_CBC_ SHA	Triple-DES SHA (168-bit)
33	SSL_RSA_EXPORT_WITH_RC4_40_ MD5	RC4 SHA (40-bit)
34	SSL_RSA_WITH_RC4_128_MD5	RC4 MD5 (128-bit)
39	SSL_RSA_WITH_DES_CBC_SHA	DES SHA (56-bit)
35	SSL_RSA_WITH_RC4_128_SHA	RC4 SHA (128-bit)
35b	TLS_RSA_WITH_AES_256_CBC_ SHA	AES SHA (256 bit)
2F	TLS_RSA_WITH_AES_128_CBC_ SHA	AES SHA (128 bit)
36	SSL_RSA_EXPORT_WITH_RC2_ CBC_40_MD5 Cipher specification 36 requires Netscape Navigator V4.07; it does not work on earlier versions of Netscape browsers.	RC2 MD5 (40-bit)
32	SSL_RSA_WITH_NULL_SHA	
31	SSL_RSA_WITH_NULL_MD5	
30	SSL_NULL_WITH_NULL_NULL	

Distributed platforms Distributed platforms 62	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	DES SHA Export 1024 (56-bit)
Distributed platforms Distributed platforms 64	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	RC4 SHA Export 1024 (56-bit)

FIPS Approved NIST SSLV3 and TLSV1 ciphers

The SSLFIPSEnable directive enables Federal Information Processing Standards (FIPS). When the SSLFIPSEnable directive is enabled, the set of ciphers available is restricted to the ciphers listed in the following table.

Short name	Long name	Description
3A	SSL_RSA_WITH_3DES_EDE_CBC_SHA	Triple-DES SHA (168-bit)
FF	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	Triple-DES SHA (168 bit)
35b	TLS_RSA_WITH_AES_256_CBC_SHA	AES SHA (256 bit)
2F	TLS_RSA_WITH_AES_128_CBC_SHA	AES SHA (128 bit)

Defining SSL for multiple-IP virtual hosts

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:

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

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.mycompany.com`) which wants to act as a reverse proxy for a resource that is hosted on a secure site (for example, `internal.mycompany.com`).

1. Configure `www.mycompany.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.mycompany.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.mycompany.com/ssl/password.html`, IBM HTTP Server makes a connection to `internal.ibm.com` using SSL. If `internal.mycompany.com` requires a client certificate, IBM HTTP Server uses the default certificate of the KeyFile for which it is configured.

Chapter 14. Managing keys with the IKEYMAN graphical interface (Distributed 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.

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/gsk7cmd.

- gsk7ver,
- ikeyman,
- gsk7capicmd
- gsk7cmd.

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.

Use IKEYMAN for configuration tasks that are related to public and private key creation and management. You cannot use IKEYMAN for configuration options that update the httpd.conf configuration 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

For more information about the IKEYMAN utility, see the IKEYMAN User's Guide on the IHS Library page.

Setting your system environment for using IKEYMAN

This topic provides detailed information on tasks that you can perform using the IBM Key Management utility (IKEYMAN). This information does not explain how to configure security options that require updates to the server configuration file.

About this task

The IKEYMAN user interface is Java-based and uses the Java support that is installed with IBM HTTP Server.

IBM HTTP Server installs a java virtual machine (JVM) for IKEYMAN. Using IKEYMAN with a JVM, other than the one installed by IBM HTTP Server is not supported.

If you are unable to open IKEYMAN, take the following actions:

Rename and move the <*ihsinst*>/java/jre/lib/ext/gskikm.jar file to a directory that is not visible to the JDK class path, extdirs, or bootclasspath. For example, on Linux platforms: `mv <ihsinst>/java/jre/lib/ext/gskikm.jar to /gskfiles/gskikm.jar.org`

What to do next

Installing unlimited strength JCE policy files (optional). 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.

- Download and install the files from the following Web site. https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk.
- Find the maximum key sizes permitted by key type with the default policy files at the following Web site.http://java.sun.com/j2se/1.4.2/docs/guide/security/jce/JCERefGuide.html#AppE.

Starting the Key Management utility user interface

This section describes how to start the Key Management (IKEYMAN) utility.

- 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

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.

You can create multiple databases if you prefer to keep certificates in separate databases.

- 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

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 the ″Populate″ on the right-hand side of the iKeyman window.
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

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:

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

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.

About this task

To create a public and private key pair and certificate request, complete the following steps:

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.

8. Click **OK**.
9. Click **OK** in the Information dialog box. A reminder to send the file to a certificate authority displays.

What to do next

GSKit certificate support limitation:

There is a limitation on the certificate key size for IKEYMAN. You cannot use IKEYMAN to create certificates with key sizes larger than 1024 bits. However, you can import certificates with key sizes up to 4096 into the key database.

Importing and exporting keys

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:

- 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

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:

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 the ″Populate″ on the right-hand side of the iKeyman window.
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

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 gsk7cmd command.

The following is an example of how to use the gsk7cmd 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/gsk7cmd  -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

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:

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 Certificate request file name, or use the default name.
   - Validity Period
8. Click **OK**.

Receiving a signed certificate from a certificate authority

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

By default, the following CA certificates are stored in the key database and marked as trusted CA certificates:

- RSA Secure Server Certification Authority (from VeriSign)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Personal Server CA
- Thawte Server CA
- Verisign Class 1 CA Individual-Persona Not Validated
- Verisign Class 2 CA Individual-Persona Not Validated
- Verisign Class 3 CA Individual-Persona Not Validated
- Verisign Class 1 CA Public Primary Certification Authority
- Verisign Test CA Root Certificate

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:

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

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.

- 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 **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 the ″Populate″ on the right-hand side of the iKeyman window.
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

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:

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

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.

Note: These stash files should be treated as highly sensitive information.

- 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.kdb**file, 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.

Chapter 15. Managing keys with the gsk7cmd command line interface (Distributed systems)

The Java command line interface to IKEYMAN, gsk7cmd, provides the necessary options to create and manage keys, certificates and certificate requests.

About this task

Note: Only use gsk7cmd, the command line interface, if you are unable to implement IKEYMAN, the graphical user interface.

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/gsk7cmd.

- Windows gsk7ver.bat, ikeyman.bat, gsk7cmd.bat, gsk7cmd, and gsk7capicmd.
- AIX Linux Solaris gsk7ver, ikeyman, and gsk7cmd.

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 gsk7cmd, the utility command line interface, for configuration tasks that are related to public and private key creation and management.

The gsk7cmd user interface uses Java and native command line invocation, enabling IKEYMAN task scripting.

You cannot use gsk7cmd 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.

- Use gsk7cmd to create key databases, public and private key pairs, and certificate requests.
- If you act as your own certificate authority (CA), you can use gsk7cmd 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 gsk7cmd command line interface, see the GSKCapicmd User's Guide on the IHS Library page.

Using the gsk7cmd command

The gsk7cmd command provides a command line interface for certificate management tasks that might otherwise be provided by the ikeyman command.

1. You can invoke the gsk7cmd from the <*ihsinst*>/bin/ directory.
   - Windows gsk7cmd.bat
   - AIX Linux Solaris gsk7cmd
2. Perform the certificate management tasks that you want to complete.

Key Management Utility command-line interface (gsk7cmd) syntax

This topic contains a description of the syntax that you can use with the gsk7cmd command.

Syntax

To run the gsk7cmd command from the <*ihsinst*>/bin/ directory, you must set the PATH environmental variable to the location of your Java or JRE executable. For more information, see “Using the gsk7cmd command” on page 111.

The syntax of the Java command line interface follows.

```
java [-Dgsk7cmd.properties=<properties_file>] com.ibm.gsk.ikeyman.gsk7cmd <object> <action>
[options]
```

Where:

- `-Dgsk7cmd.properties` specifies the name of an optional properties file to use for this Java invocation. A default properties file, `gsk7cmd.properties`, exists as a sample file that you can modify and use with any Java application.
- `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 gsk7cmd invocations
  - -version: Displays version information for gsk7cmd

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.

The following table describes each `action` possible on a specified `object` that you can use with the gsk7cmd 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
	-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 gsk7cmd command
-version		Display gsk7cmd version information

The following table describes the `options` that you can use with the gsk7cmd 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). Defaults are: 365 days for a certificate and 60 days for a database password.
-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 or 1024). Default is 1024.
-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

Chapter 15, “Managing keys with the gsk7cmd command line interface (Distributed systems),” on page 111
The Java command line interface to IKEYMAN, gsk7cmd, provides the necessary options to create and manage keys, certificates and certificate requests.

Creating a new key database using the command-line interface

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.

- Create a new key database using the gsk7cmd command-line interface by entering the following command (as one line):

  ```
  <ihsinst>/bin/gsk7cmd -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/gsk7capicmd -keydb -create -db <name> [-pw <passwd>] [-type <cms>] [-expire <days>] [-stash] [-fips]
  [-strong]
  ```

Managing the database password using the command line

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.
- Change the password for a key database using the gsk7cmd command-line interface. Enter the following command as one line:

  ```
  <ihsinst>/bin/gsk7cmd -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/gsk7capicmd -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

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 gsk7cmd command-line interface or GSKCapiCmd tool, as follows:

1. Use the gsk7cmd command-line interface. Enter the following command (as one line):

   ```
   <ihsinst>/bin/gsk7cmd -certreq -create -db <filename> -pw <password> -label <label> -dn <distinguished_name>
   -size <1024 | 512> -file <filename>
   ```

   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 <1024 | 512>` indicates a key size of 512 or 1024.
   - `-file <filename>` is the name of the file where the certificate request will be stored.

   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.

   ```
   <ihsinst>/bin/gsk7capicmd -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]
   ```

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/gsk7cmd -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

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 gsk7cmd command-line interface or GSKCapiCmd tool.

- Use the gsk7cmd command-line interface to import certificates from another key database, as follows:

  ```
  <ihsinst>/bin/gsk7cmd -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` opearnd.
  - `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/gsk7capicmd -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 gsk7cmd command-line interface to export certificates from another key database, as follows:

  ```
  gsk7cmd -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/gsk7capicmd -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

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.

- Create a self-signed certificate using the IKEYCMD command-line interface, as follows:

  ```
  gsk7cmd -cert -create -db <filename> -pw <password> -size <1024 | 512> -dn <distinguished_name>
  -label label> -default_cert <yes | no> - expire <days>
  ```

  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` specifies the key size 512 or 1024.
  - `-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).
- 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.

  ```
  gsk7capicmd -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>]
  ```

Receiving a signed certificate from a certificate authority

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

By default, the following CA certificates are stored in the key database and marked as trusted CA certificates:

- Verisign Class 2 OnSite Individual CA
- Verisign International Server CA -- Class 3
- VeriSign Class 1 Public Primary CA -- G2
- VeriSign Class 2 Public Primary CA -- G2
- VeriSign Class 3 Public Primary CA -- G2
- VeriSign Class 1 CA Individual Subscriber-Persona Not Validated
- VeriSign Class 2 CA Individual Subscriber-Persona Not Validated
- VeriSign Class 3 CA Individual Subscriber-Persona Not Validated
- RSA Secure Server CA (from RSA)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

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.

- Receive the CA-signed certificate into a key database using the gsk7cmd command-line interface, as follows:

  ```
  <ihsinst>/bin/gsk7cmd -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/gsk7capicmd -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

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.

- Display a list of trusted CAs in a key database by entering the following command as one line:

  <*ihsinst*>/bin/gsk7cmd -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/gsk7cmd -cert -list -expiry < days > -db < filename > -pw < paswsword > - 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

This topic describes how to store a certificate from a certificate authority (CA) that is not a trusted CA.

To store a certificate from a CA that is not a trusted CA, use the following command:

<*ihsinst*>/bin/gsk7cmd -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. Should be *yes*.

Storing the encrypted database password in a stash file

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.

- To store the password using the gsk7cmd command-line interface, enter the following command on one line.

  ```
  <ihsinst>/bin/gsk7cmd -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 gsk7cmd command line interface.

  ```
  <ihsinst>/bin/gsk7cmd -keydb -stashpw -db <db_name> -pw <password>
  ```

Chapter 16. Managing keys with the native key database gskkyman (z/OS systems)

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 gsk7cmd.

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.

You cannot use gskkyman for configuration options that update the httpd.conf configuration file.

- 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.

Chapter 17. Getting started with the cryptographic hardware for SSL (Distributed systems)

The IBM 4758 and other cryptographic devices require the PKCS11 support software for the host machine and internal firmware.

About this task

You will need the manual that explains software installation and cryptographic coprocessor microcode loading.

Note: The support software and manual do not come with the IBM 4758 card, but you can download them from http://www.ibm.com/security/cryptocards/index.shtml. From the download site, obtain the PKCS#11 Model 002/023 software and the PKCS#11 installation manual.

1. After installing the support software on your machine and loading the microcode on the cryptographic device, initialize the card.
2. Configure IBM HTTP Server to pass the module for the PKCS11 device, the token label, the key label of the key created by the PKCS11 device, and the user PIN password of the token to the GSKit for access to the key for the PKCS11 device by modifying the configuration file. The PKCS11 module differs for each platform and PKCS11 device.

   AIX For the IBM hardware cryptographic devices (for example, IBM 4758 card and IBM e-business Cryptographic Accelerator) the PKCS11 module ships with the `bos.pkcs11` package.
3. Install the `devices.pci.14109f00` device for the IBM 4758 and the `devices.pci.1410e601` device for the IBM e-business Cryptographic Accelerator.

   For the IBM 4758 on Windows, the PKCS11 module comes with the PKCS11 software available for download from: http://www.ibm.com/security/cryptocards/ordersoftware.shtml. For nCipher, the PKCS11 module ships with nCipher software and is located in the `$NFAST_HOME/toolkits/pkcs11` directory.

   The default locations of the PKCS11 modules for each PKCS11 device follow:

   - nCipher:
     - AIX Linux Solaris `/opt/nfast/toolkits/pkcs11/libcknfast.so`
     - HP-UX `/opt/nfast/toolkits/pkcs11/libcknfast.sl`
     - 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`

Cryptographic hardware for Secure Sockets Layer

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 7 - 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.

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, *accelerator mode*	No	Yes	Uses the BHAPI and BSAFE interface. Supported on Solaris and Windows operating systems.
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.

Initializing IBM 4758 and IBM e-business Cryptographic Accelerator on AIX systems

To initialize IBM cryptographic hardware (IBM 4758 and IBM e-business Cryptographic Accelerator), you must obtain and install the most recent PKCS11 module.

1. To initialize the IBM cryptographic hardware (IBM 4758 and IBM e-business Cryptographic Accelerator) on AIX, obtain and install the `bos.pkcs11` software. Obtain the most recent `bos.pkcs11` package from http://www.ibm.com/servers/eserver/support/pseries/aixfixes.html. Select your AIX version under **Specific fixes**. The `bos.pkcs11` package installs the PKCS11 module needed for the SSLPKCSDriver directive discussed below. You also need the devices.pci.1410e601 device for the IBM e-business Cryptographic Accelerator and the devices.pci.14109f00 and devices.pci.14109f00 for the IBM 4758.
2. Initialize your token. After you install the PKCS11 software, initialize your device. You can access the Manage the PKCS11 subsystem panel from Smitty to initialize your PKCS11 device.
   a. Select Initialize your token.
   b. Set a security officer and User PIN, if not already set.
   c. Initialize your user PIN. See Chapter 5: Token Initialization from the PKCS11 manual for more detailed information.

Initializing IBM 4758 Cryptographic Accelerator on Windows systems

To initialize the IBM 4758 card on Windows operating systems, you will need the PKCS11 software.

1. Obtain the PKCS11 software from the following site: http://www.ibm.com/security/cryptocards/.
2. Use the `TOKUTIL.EXE` utility that installs with the PKCS11 software to initialize your IBM 4758 card on Windows operating systems.
3. Ensure you have the `cryptoki.dll` module in your path.
4. Refer to *Chapter 5: Token Initialization* from the PKCS11 documentation for more details.

Using IKEYMAN to store keys on a PKCS11 device

For IBM HTTP Server, you can use IKEYMAN for storing keys on a PKCS11 device.

- You will need to 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

• Run IKEYMAN to store the keys on the PKCS11 device.
  After launching IKEYMAN:
  – Select **Key Database File** from the menu, then **Open** to navigate to the Key database information dialog
  – From the drop down for **Key Database Type**, select **CMS Cryptographic Token**
  – Enter the **File Name** and **Location** for the PKCS11 driver name and path location
  – Click **OK** to navigate to the **Open Cryptographic Token** dialog
  – Choose the **Cryptographic Token Label** of the PKCS11 device
  – Provide the Cryptographic Token Password for the PKCS11 device (which is a previously set password that is hardware-specific)
  – Select the **Create new secondary key database file** option and fill in prompts for creating a new secondary key database

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 and the private key will be stored on your PKCS11 device.

Configuring IBM HTTP Server to use nCipher and Rainbow accelerator devices and PKCS11 devices

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:

1. Stash your password to the PKCS11 device, or optionally enable password prompting: 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 `crl` 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>`

Chapter 18. 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.

Before you begin

Note: 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 is not loaded 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:

- Windows

  ```
  LoadModule ibm_ldap_module modules/IBMModuleLDAP.dll
  ```

- AIX HP-UX Linux Solaris

  ```
  LoadModule ibm_ldap_module modules/mod_ibm_ldap.so
  ```

If you have the LDAP client installed on your computer, you can use ldapsearch as a tool to test the values you intend to use for the various settings.

About this task

See “LDAP directives” on page 136 to obtain detailed descriptions of the LDAP (mod_ibm_ldap) directives.

1. Edit the httpd.conf IBM HTTP Server configuration file.
2. Determine the resource you want to limit access to. For example: <Directory ″/secure_info″>.
3. Add directives in httpd.conf to the directory location (container) to be protected with values specific to your environment. For example:
   - `LdapConfigFile path_to_ldap.prop`
   - `AuthType Basic`
   - `AuthName ″Title of your protected Realm″`
   - `Require valid-user`
4. There are three options for how to use IBM HTTP Server to authenticate with your existing LDAP installation.
   - **Authorization based on LDAP group membership.**

     Use LDAP to check user passwords and verify that the user exists in a group defined in LDAP.

     Note: The membership that identifies the user as being able to access the resource is a part of the group, not part of the user’s own LDAP entry.

     For example, to restrict access to a group, add the following directive:

     ```
     LDAPRequire group grp1
     ```

     For this form of LDAPRequire, you must have groups configured in your LDAP repository that conform to the following rules (using the example group name *grp1*):

     – There is an entry in your LDAP repository that matches the following search filter, where the values *groupofnames* and *groupofuniquenames* are example values specified in ldap.group.dnattributes.

Note: The proper value of ldap.group.dnattributes is a list of what objectclasses signify is a group in your LDAP schema.

```
ldapsearch ... "(&(cn=grp1)(|(objectclass=groupofnames)
(objectclass=groupofuniquenames)))"
```

– As part of the LDAP entry for *"grp1,"* there are a series of attributes that match the following, where the values *member* and *uniquemember* are example values of ldap.group.memberAttributes.

Note: The proper value of ldap.group.memberAttributes is a list of what objectclasses signify is a membership in a group. The values of these entries are the Distinguished Names (DN) of your users.

```
ldapsearch ... "(&(cn=grp1)(|(objectclass=groupofnames)
(objectclass=groupofuniquenames)))" member uniquemember
```

Example:

```
ldapsearch -x -h myldapserver -D cn=root -w rootpw
"(&(cn=grp1)(|(objectclass=groupofnames)(objectclass=groupofuniquenames)))"
member uniquemember
```

```
dn: cn=group1,ou=myunit,o=myorg,c=US
member: cn=user1, ou=otherunit, o=myorg, c=US
member: cn=user12, ou=otherunit, o=myorg, c=US
```

If an object of the type listed in ldap.group.dnattributes is a member of the group being searched, then it will be recursively searched in the same fashion, up to a depth of ldap.group.search.depth

– First IBM HTTP Server uses the ldap.group.name.filter and ldap.user.cert.filter to translate the CN provided for the user and the group into distinguished names (DN). Next, IBM HTTP Server searches using the group DN as a base for entries whose value is the user DN.

Example:

```
ldapsearch ... -b "cn=grp1,ou=myunit,o=myorg,c=US"
"|((member=cn=user1,ou=otherunit,o=myorg,c=US)
(uniquemember=cn=user1,ou=otherunit,o=myorg,c=US))"
```

- **Authorization based on LDAP attributes of user.** Use LDAP to check user passwords and verify that the user matches a set of attributes (the attribute that identifies the user as being able to access the resource is a part of the users own LDAP entry).

  Example:

  ```
  LDAPRequire filter "(&(jobtitle=accountant)(location=newyork))"
  ```

  To use this form of LDAPRequire, the IBM HTTP Server must use the ldap.user.cert.filter to translate the CN provided for the user into a DN. IBM HTTP Server must also search using the user DN as a base and use the search filter provided in the LDAPRequire directive. If any results are returned, authorization succeeds.

  Example:

  ```
  ldapsearch ... -b "cn=user1,ou=otherunit,o=myorg,c=US" "(&(jobtitle=accountant)
  (location=newyork))"
  ```

  Some attributes (sometimes called Dynamic Roles) in LDAP are calculated dynamically by the LDAP server and might have different semantics that are not valid in a search filter. Such values would fail if used in the preceding example and cannot be used for authorization on IBM HTTP Server.

- **Authentication only: Use LDAP to check user passwords only.**

  You can use the Require directive to limit to specific users or maintain a flat group file using AuthGroupFile.

5. Edit the `ldap.prop` configuration file. If you do not have one yet, you can use the `ldap.prop.sample` file that ships with IBM HTTP Server. If you have questions about the correct values, check with your LDAP server administrator. Update the following directives with values that are correct for your environment:

   a. Enter the Web server connection information.

b. When using SSL, or LDAPS, or LDAP over SSL:
   - Change ldap.transport to an SSL value
   - Change ldap.URL to include the LDAPS protocol and the proper port value, for example, 636.
   - Configure the SSL key database to be used, for example:

     ldap.key.fileName=/*path_to*/*key.kdb*

     ldap.key.file.password.stashFile=/*path_to*/stashfile

     Where *stashfile* is created by the bin/ldapstash command.

     ldap.key.label=*label*

     Where *label* is the value appearing in IKEYMAN for the referenced *key.kdb*.

Results

Searches that use the mod_ibm_ldap directives maintain a pool of server connections that authenticate as the ldap.application.dn user. The first connection is created when the first LDAP-protected request is received. Connections will be held open a specified number of seconds (ldap.idleConnection.timeout) for subsequent searches on that connection or connections for other requests.

If you are reading logs or looking at an IP trace, the following sequence of events should occur:

- IBM HTTP Server starts.
- If LDAP_TRACE_FILE is set, it will have a few entries for LDAP_obtain_config
- The first request for LDAP-protected resource is received.
- IBM HTTP Server binds to LDAP using the ldap.application.dn username and the password stashed in ldap.application.password.stashFile (Application Connection)
- IBM HTTP Server performs a search over this connection to translate the username typed in by the user, or the contents of their client certificate, into a Distinguished Name (DN) using the *user.*.filter* settings.
- IBM HTTP Server binds to the LDAP server as username/password provided by the client to check authentication (This is a ″user connection″ to the LDAP server)
- If any LDAPRequire directives are in effect for this request, IBM HTTP Server processes them in the manner described in the preceeding procedure.
- IBM HTTP Server unbinds the user connection
- The application connection is maintained for the next request

Lightweight Directory Access Protocol

This section addresses questions about what Lightweight Directory Access Protocol (LDAP) is and how it works, and provides high level overviews of X.500 and LDAP.

LDAP is a standard protocol that provides a means of storing and retrieving information about people, groups, or objects on a centralized X.500 or LDAP directory server. X.500 enables that information to be organized and queried, using LDAP, from multiple web servers using a variety of attributes. LDAP queries can be as simple or complex as is required to identify a desired individual entity or group of entities. LDAP reduces required system resources by including only a functional subset of the original X.500 Directory Access Protocol (DAP).

The IBM HTTP Server LDAP module enables the use of an X.500 directory server for authentication and authorization purposes. IBM HTTP Server can use this capability to limit access of a resource to a controlled set of users. This capability reduces the administrative overhead usually required to maintain user and group information for each individual Web server.

You can configure the IBM HTTP Server LDAP module to use TCP/IP or Secure Sockets Layer (SSL) connections to the X.500 directory server. The LDAP module can be configured to reference a single LDAP server or multiple servers.

X.500 overview. X.500 provides a directory service with components that are capable of more efficient retrieval. LDAP uses two of these components: The information model, which determines the form and character, and the namespace, which enables information indexing and referencing.

The X.500 directory structure differs from other directories in information storage and retrieval. This directory service associates information with attributes. A query based on attributes generates and passes to the LDAP server, and the server returns the respective values. LDAP uses a simple, string-based approach for representing directory entries.

An X.500 directory consists of typed entries that are based on the ObjectClass attribute. Each entry consists of attributes. The ObjectClass attribute identifies the type of entry, for example, a person or organization, that determines the required and optional attributes.

You can divide entries, arranged in a tree structure, among servers in geographical and organizational distribution. The directory service names entries, according to their position within the distribution hierarchy, by a distinguished name (DN).

Lightweight Directory Access Protocol overview. Accessing an X.500 directory requires the Directory Access Protocol (DAP). However, DAP requires large amounts of system resources and support mechanisms to handle the complexity of the protocol. To enable desktop workstations to access the X.500 directory service, LDAP was introduced.

LDAP, a client and server-based protocol can handle some of the heavy resources required by DAP clients. An LDAP server can only return results or errors to the client, requiring little from the client. If unable to answer a client request, an LDAP Server must chain the request to another X.500 server. The server must complete the request, or return an error to the LDAP server, which in turn passes the information to the client.

IBM HTTP Server supports the following LDAP servers:

- iPlanet/Netscape Directory Server
- IBM SecureWay® Directory Server
- Microsoft Active Directory

Querying the Lightweight Directory Access Protocol server

The Lightweight Directory Access Protocol (LDAP) accesses the X.500 directory using text strings called filters. When these query strings pass to the LDAP server, the server returns the requested portions of the specified entity.

About this task

LDAP filters use attributes to simplify queries to the LDAP server. For example, you can use a filter such as ″objectclass=person″ to limit your query to entities that represent people as opposed to groups or equipment.

- To authorize a user as a member of a group, add the following directive to the configuration file:

  ```
  LDAPRequire group "group_name"
  ```

  For example:

  ```
  LDAPRequire group "Administrative Users"
  ```

- To authorize a user by filter, add the following directive to the configuration file:

```
LDAPRequire filter "ldap_search_filter"
```

For example, to enable access to the resource by a programmer in your department:

```
LDAPRequire filter"(&(objectclass=person)(cn=*)(ou=programmer)(o=department))"
```

Or, to enable access for John Doe only:

```
LDAPRequire filter "(&(objectclass=person)(cn=John Doe))"
```

Secure Sockets Layer and the Lightweight Directory Access Protocol module

IBM HTTP Server provides the ability to use a secure connection between the LDAP module running in the Web server and the LDAP directory server. If this feature is enabled, any communication between the Web server and the directory server is encrypted.

To enable this feature, edit the `ldap.prop` LDAP configuration file and change the value of `ldap.transport` to `SSL`. Create or obtain a certificate database file (`X.kdb`) and a password stash file (`Y.sth`). You can use IKEYMAN to obtain a key database file. You must use the **ldapstash** program to create the stash file. You will also need to change the values for `ldap.URL` and `ldap.group.URL` to use port 636 instead of port 389.

The key database file contains the certificates which establish identity. The LDAP server can require that the Web server provide a certificate before allowing queries. When using a certificate with an SSL connection between the LDAP module and the LDAP server, the user ID that IBM HTTP Server is configured to use must have write permission to the key database file containing the certificate.

Certificates establish identity to prevent other users from stealing or overwriting your certificates (and therefore your identity). If someone has read permission to the key database file, they can retrieve the user's certificates and masquerade as that user. Grant read or write permission only to the owner of the key database file.

SSL certificate revocation list

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.

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.
- 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] <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.

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

LDAP directives

These configuration parameters control the Lightweight Directory Access Protocol (LDAP) feature in IBM HTTP Server.

- “LdapCodepageDir directive” on page 137
- “LdapConfigfile directive” on page 137
- “LDAPRequire directive” on page 137
- “Ldap.application.authType directive” on page 138
- “Ldap.application.DN directive” on page 138
- “Ldap.application.password.stashFile directive” on page 138
- “Ldap.cache.timeout directive” on page 138
- “Ldap.group.attribute directive” on page 139
- “Ldap.group.dnattribute directive” on page 139
- “Ldap.group.memberattribute directive” on page 139
- “Ldap.group.memberAttributes directive” on page 139
- “Ldap.group.name.filter directive” on page 140
- “Ldap.group.search.depth directive” on page 140
- “Ldap.group.URL directive” on page 140
- “Ldap.idleConnection.timeout directive” on page 141
- “Ldap.key.file.password.stashfile directive” on page 141
- “Ldap.key.fileName directive” on page 141
- “Ldap.key.label directive” on page 142
- “Ldap.LdapReferralhoplimit directive” on page 142
- “Ldap.LdapReferrals directive” on page 142
- “Ldap.realm directive” on page 142
- “Ldap.search.timeout directive” on page 143
- “Ldap.transport directive” on page 143

- “Ldap.url directive” on page 144
- “Ldap.user.authType directive” on page 144
- “Ldap.user.cert.filter directive” on page 144
- “Ldap.user.name.fieldSep directive” on page 145
- “Ldap.user.name.filter directive” on page 145
- “Ldap.version directive” on page 146
- “Ldap.waitToRetryConnection.interval directive” on page 146

Note: 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.

LdapCodepageDir directive

Codepages are now automatically installed in the IHS installation directory and are referenced relative to the IHS installation directory, as opposed to the configured server root directory as in previous versions.

LdapConfigfile directive

The LdapConfigFile directive indicates the name of the LDAP properties file associated with a group of LDAP parameters.

Syntax	`LdapConfigFile <Fully qualified path to configuration file>`
Scope	Single instance per directory stanza
Default	`c:\program files\ibm http server\conf\ldap.prop.sample`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Fully qualified path to a single configuration file. Use this directive in the `httpd.conf` file.

LDAPRequire directive

The uire directive is used to restrict access to a resource that is controlled by LDAP authentication to a specified collection of users. It can either use groups that are defined in LDAP by using the group type, or it can use an LDAP filter type to designate a collection of users with a similar set of attribute values.

Syntax	`uire filter <filter name>` or `uire group <group1 [group2.group3....]>`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	`uire filter "(&(objectclass=person)(cn=*)(ou=IHS)(o=IBM))"`, or `uire group "sample group"`. Use this directive in the `httpd.conf` file.

If the group type is used, and multiple group values are specified, the group validation is a logical AND of the groups. A user must be a member of *sample Group1* and *sample Group2* if a logical OR of groups is required. For example, if a user is a member of *sample Group1* or *sample Group2*, then a new LDAP group, *our department group*, should be created on the LDAP server that has *sample Group1* and *sample*

Group2 as its members. You would then use the directive: uire group *our Department Group* .

Ldap.application.authType directive

The Ldap.application.authType directive specifies the method for authenticating the Web server to the LDAP server.

Syntax	`ldap.application.authType=None`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	• None: If the LDAP server does not require the Web server to authenticate. • Basic: Uses the distinguished name (DN) of the Web server as the user ID, and the password stored in the stash file, as the password.

Ldap.application.DN directive

The Ldap.application.DN directive indicates the distinguished name (DN) of the Web server. Use this name as the user name when accessing an LDAP server using basic authentication. Use the entry specified in the LDAP server to access the directory server.

Syntax	`ldap.application.DN=cn=ldapadm,ou=ihs test,o=IBM,c=US`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Distinguished name

Ldap.application.password.stashFile directive

The Ldap.application.password.stashFile directive indicates the name of the stash file containing the encrypted password for the application to authenticate to the LDAP server when Server Authentication type is Basic.

Syntax	`ldap.application.password.stashFile=c:\IHS\ldap.sth`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Fully qualified path to the stash file. You can create this stash file with the **ldapstash** command.

Ldap.cache.timeout directive

The ldap.cache.timeout directive caches responses from the LDAP server. If you configure the Web server to run as multiple processes, each process manages its own copy of the cache.

Syntax	`ldap.cache.timeout= <secs>`
Scope	Single instance per directory stanza
Default	600

Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	The maximum length of time, in seconds, a response returned from the LDAP server remains valid.

Ldap.group.attribute directive

The ldap.group.attributes directive indicates the filter used to determine if a distinguished name (DN) is an actual group through an LDAP search.

Syntax	`ldap.group.memberattribute = <attribute>`
Scope	Single instance per directory stanza
Default	`uniquegroup`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	An ldap attribute - See the `ldap.prop.sample` directive for more information on the use of this directive.

Ldap.group.dnattribute directive

The ldap.group.dnattributes specifies the filter used to determine, through an LDAP search, if a distinguished name (DN) is an actual group.

Syntax	`ldap.group.memberattribute = <ldap filter>`
Scope	Single instance per directory stanza
Default	`groupofnames groupofuniquenames`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	An ldap filter - See the `ldap.prop.sample` directive for more information on the use of this directive.

Ldap.group.memberattribute directive

The ldap.group.memberattribute directive specifies the attribute to retrieve unique groups from an existing group.

Syntax	`ldap.group.memberattribute = <ldap filter>`
Scope	Single instance per directory stanza
Default	`groupofnames groupofuniquenames`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	An ldap filter - See the `ldap.prop.sample` directive for more information on the use of this directive.

Ldap.group.memberAttributes directive

The ldap.group.memberAttributes directive serves as a means to extract group members, once the function finds a group entry in an LDAP directory.

Syntax	`ldap.group.memberAttributes= attribute [attribute2....]`
Scope	Single instance per directory stanza
Default	`member` and `uniquemember`

Module mod_ibm_ldap
Multiple instances in the configuration file yes
Values Must equal the distinguished names of the group members. You can use more than one attribute to contain member information.

Ldap.group.name.filter directive

The ldap.group.name.filter directive indicates the filter LDAP uses to search for group names.

Syntax ldap.group.name.filter = *<group name filter>*
Scope Single instance per directory stanza
Default (&(cn=%v1) (|(objectclass=groupOfNames) (objectclass=groupOfUniqueNames))
Module mod_ibm_ldap
Multiple instances in the configuration file yes
Values An LDAP filter. See Querying the LDAP server using LDAP search filters.

Ldap.group.search.depth directive

The ldap.group.search.depth directive searches subgroups when specifying the uire group <group> directives. Groups can contain both individual members and other groups.

Syntax ldap.group.search.depth = *<integer depth>*
Scope Single instance per directory stanza
Default 1
Module mod_ibm_ldap
Multiple instances in the configuration file yes
Values An integer. When doing a search for a group, if a member in the process of authentication is not a member of the required group, any subgroups of the required group are also searched. For example:

```
group1 >group2 (group2 is a member of group1)
group2 >group3 (group3 is a member of group2)
group3 >jane   (jane is a member of group3)
```

If you search for jane and require her as a member of group1, the search fails with the default ldap.search.depth value of 1. If you specify ldap.group.search.depth>2, the search succeeds.

Use ldap.group.search.depth=<depth to search --number> to limit the depth of subgroup searches. This type of search can become very intensive on an LDAP server. Where group1 has group2 as a member, and group2 has group1 as a member, this directive limits the depth of the search. In the previous example, group1 has a depth of 1, group2 has a depth of 2 and group3 has a depth of 3.

Ldap.group.URL directive

The ldap.group.URL directive specifies a different location for a group on the same LDAP server. You cannot use this directive to specify a different LDAP server from that specified in the ldap.URL directive.

Syntax ldap.group.URL = ldap://*<hostname:port>*/*<BaseDN>*

Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	• host name: Host name of the LDAP server. • port number: Optional port number on which the LDAP server listens. The default for TCP connections is 389. If you use SSL, you must specify the port number. • BaseDN: Provides the root of the LDAP tree in which to perform the search for groups.

Note:
This property becomes required if the LDAP URL for groups differs from the URL specified by the ldap.URL property.

Ldap.idleConnection.timeout directive

The ldap.idleConnection.timeout directive caches connections to the LDAP server for performance.

Syntax	`ldap.idleConection.timeout = <secs>`
Scope	Single instance per directory stanza
Default	600
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Length of time, in seconds, before an idle LDAP server connection closes due to inactivity.

Ldap.key.file.password.stashfile directive

The ldap.key.file.password.stashfile directive indicates the stash file containing the encrypted keyfile password; use the ldapstash command to create this stash file.

Syntax	`ldap.key.file.password.stashfile =d:\ <Key password file name>`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Fully qualified path to the stash file.

Ldap.key.fileName directive

The ldap.key.fileName directive indicates the file name of the key file database. This option becomes required when you use Secure Sockets Layer (SSL).

Syntax	`ldap.key.fileName=d:\<Key file name>`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Fully qualified path to the key file.

Ldap.key.label directive

The ldap.key.file.password.stashfile directive indicates the certificate label name the Web server uses to authenticate to the LDAP server.

Syntax	`My Server Certificate`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	A valid label used in the key database file. This label becomes required only when using Secure Sockets Layer (SSL) and the LDAP server requests client authentication from the Web server.

Ldap.LdapReferralhoplimit directive

The ldap.LdapReferralHopLimit directive indicates the maximum number of referrals to follow. LDAP authentication will fail if the specified limit is exceeded.

Syntax	`ldap.LdapReferralHopLimit = <number_of_hops>`
Scope	Single instance per directory stanza
Default	10
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	0 to 10

Set the LdapReferrals directive `on` to use the LdapReferralhoplimit directive.

Note: An LdapReferralhoplimit value of 0 will cause authentication to fail if any referrals are encountered.

The LdapReferralhoplimit directive is not meaningful when the LdapReferrals directive is `off` (default).

Ldap.LdapReferrals directive

The ldap.LdapReferrals directive indicates whether referrals (which redirect a client request to another LDAP server) will be chased for searches while performing LDAP queries.

Syntax	`ldap.LdapReferrals = off | on`
Scope	Single instance per directory stanza
Default	off
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	On or off

Ldap.realm directive

he ldap.key.realm directive indicates the name of the protected area, as seen by the requesting client.

Syntax	`ldap.realm==<Protection Realm>`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes

Values	A description describing the protected page.

uire directive

The uire directive is used to restrict access to a resource that is controlled by LDAP authentication to a specified collection of users. It can either use groups that are defined in LDAP by using the group type, or it can use an LDAP filter type to designate a collection of users with a similar set of attribute values.

Syntax	`uire filter <filter name>` or `uire group <group1 [group2.group3....]>`
Scope	Single instance per directory stanza
Default	None
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	`uire filter "(&(objectclass=person)(cn=*)(ou=IHS)(o=IBM))"`, or `uire group "sample group"`. Use this directive in the `httpd.conf` file.

If the group type is used, and multiple group values are specified, the group validation is a logical AND of the groups. A user must be a member of *sample Group1* and *sample Group2* if a logical OR of groups is required. For example, if a user is a member of *sample Group1* or *sample Group2*, then a new LDAP group, *our department group*, should be created on the LDAP server that has *sample Group1* and *sample Group2* as its members. You would then use the directive: uire group *our Department Group* .

Ldap.search.timeout directive

The ldap.search.timeout directive indicates the maximum time, in seconds, to wait for an LDAP server to complete a search operation.

Syntax	`ldap.search.timeout = <secs>`
Scope	Single instance per directory stanza
Default	10
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Length of time, in seconds.

Ldap.transport directive

The ldap.transport directive indicates the transport method used to communicate with the LDAP server.

Syntax	`ldap.transport = TCP`
Scope	Single instance per directory stanza
Default	TCP
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	TCP or SSL

Ldap.url directive

The ldap.url directive indicates the URL of the LDAP server to authenticate against.

Syntax

`ldap.url = ldap://<hostname:port>/<BaseDN>`

where:

- hostname: Represents the host name of the LDAP server.
- port: Represents the optional port number on which the LDAP server listens. The default for TCP connections is 389. You must specify the port number if you use SSL.
- BaseDN: Provides the root of the LDAP tree in which to perform the search for users.

  For example: `ldap.URL=ldap://<ldap.ibm.com:489/o=Ace Industry, c=US>`

Scope Single instance per directory stanza
Default None
Module `mod_ibm_ldap`
Multiple instances in the configuration file yes

Ldap.user.authType directive

The ldap.usr.authType directive indicates the method for authenticating the user requesting a Web server. Use this name as the user name when accessing an LDAP server.

Syntax `ldap.user.authType = BasicIfNoCert`
Scope Single instance per directory stanza
Default Basic
Module `mod_ibm_ldap`
Multiple instances in the configuration file yes
Values Basic, Cert, BasicIfNoCert

Ldap.user.cert.filter directive

The ldap.usr.cert.filter directive indicates the filter used to convert the information in the client certificate passed over Secure Sockets Layer (SSL) to a search filter for and LDAP entry.

Syntax `ldap.user.cert.filter=(&(objectclass=person)(cn=%v1))`
Scope Single instance per directory stanza
Default "(&(objectclass=person) (cn=%v1, ou=%v2, o=%v3,c=%v4))"
Module `mod_ibm_ldap`
Multiple instances in the configuration file yes
Values An LDAP filter. See Querying the LDAP server using LDAP search filters.

Secure Socket Layer (SSL) certificates include the following fields, all of which you can convert to a search filter:

Certificate field	Variable
common name	%v1
organizational unit	%v2

organization	%v3
country	%v4
locality	%v5
state or country	%v6
serial number	%v7

When you generate the search filter, you can find the field values in the matching variable fields (%v1, %v2). The following table shows the conversion:

User certificate	**Filter conversion**
Certificate	cn=Road Runner, o=Acme Inc, c=US
Filter	(cn=%v1, o=%v3, c=%v4)
Resulting query	(cn=RoadRunner, o=Acme, Inc, c=US)

Ldap.user.name.fieldSep directive

The ldap.usr.name.fieldSep directive indicates characters as valid field separator characters when parsing the user name into fields.

Syntax `ldap.user.name.fieldSep=/`
Scope Single instance per directory stanza
Default The space, comma, and the tab (/t) character.
Module `mod_ibm_ldap`
Multiple instances in the configuration file yes
Values Characters. If '/' represents the only field separator character and the user enters `"Joe Smith/Acme"`, then '`%v2`' equals `"Acme"`.

Ldap.user.name.filter directive

The ldap.usr.name.filter directive indicates the filter used to convert the user name entered in a search filter for an LDAP entry.

Syntax `ldap.user.name.filter=<`*`user name filter`*`>`
Scope Single instance per directory stanza
Default `"((objectclass=person) (cn=%v1 %v2))"`, where *%v1* and *%v2* represent characters entered by the user.

For example, if the user enters `"Paul Kelsey"`, the resulting search filter becomes `"((objectclass=person)(cn=Paul Kelsey))"`. You can find search filter syntax described in Querying the LDAP server using LDAP search filters.

However, because the Web server cannot differentiate between multiple returned entries, authentication fails when the LDAP server returns more than one entry. For example, if the user makes the ldap.user.name.filter= `"((objectclass=person)(cn=%v1* %v2*))"` and enters **Pa Kel**, the resulting search filter becomes `"(cn=Pa* Kel*)"`. The filter finds multiple entries such as (cn=Paul Kelsey) and (cn=Paula Kelly) and authentication fails. You must modify your search filter.

Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	An LDAP filter. See Querying the LDAP server using LDAP search filters.

Ldap.version directive

The ldap.version directive indicates the version of the LDAP protocol used to connect to the LDAP server. the protocol version used by the LDAP server determines the LDAP version.

Note: This directive is optional.

Syntax	`ldap.version=3`
Scope	Single instance per directory stanza
Default	`ldap.version=3`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	2 or 3

Ldap.waitToRetryConnection.interval directive

The ldap.waitToRetryConnection.interval directive indicates the time the Web server waits between failed attempts to connect.

If an LDAP server goes down, the Web server continues to try to connect.

Syntax	`ldap.waitToRetryConnection.interval=<`*secs*`>`
Scope	Single instance per directory stanza
Default	`300`
Module	`mod_ibm_ldap`
Multiple instances in the configuration file	yes
Values	Time (in seconds)

Related reference

Distributed platforms z/OS “mod_ibm_ldap directives migration” on page 149
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.

Converting your directives from mod_ibm_ldap to mod_ldap

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.

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 2. 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
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.

Table 2. LDAP configuration directives conversion (continued)

mod_ibm_ldap	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.

   `<ihsinst>bin/apachectl -t`

   Note: 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.

   Note: 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.

```
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

 LDAPTrustedMode SSL

 <Directory /path/to/htdocs>
  AuthzLDAPAuthoritative on
  AuthBasicProvider ldap
  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)
 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
 LDAPTrustedClientCert CMS_LABEL my_cert3
 AuthLDAPURL ldaps://groups_ldap.server.org:636/o=OurOrg,c=US?cn?sub?
  (|(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

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.

Note: 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, 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`.

Note: 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 top 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
```

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
```

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
```

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=OurOrg,c=US
ldap.group.URL=ldaps://groups_ldap.server.org:636/o=OurOrg,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:

```
LDAPTrustedGlobalCert CMS_KEYFILE /path/to/keyfile.kdb myKDBpassword
```

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
```

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
```

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
```

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)
```

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

Chapter 18, “Authenticating with LDAP on IBM HTTP Server using **mod_ibm_ldap** (Distributed systems),” on page 131
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 31
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.

Chapter 19. 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

Note: 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/httpserv/manual70/mod/mod_ldap.html
- http://publib.boulder.ibm.com/httpserv/manual70/mod/mod_authnz_ldap.html

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.

   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.

   Distributed platforms

   ```
   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
   ```

Note: 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 Chapter 3, “Performing required z/OS system configurations,” on page 21 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`

   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/httpserv/manual70/mod/mod_authnz_ldap.html

Chapter 20. Authenticating with SAF on IBM HTTP Server (z/OS systems)

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
LoadModule authz_default_module modules/mod_authz_default.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.

1. Determine the directory location you want to limit access to. For example: <Location ″/admin-bin″>.
2. 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
   ```

3. Optional: Specify Require saf-user or Require saf-group to restrict access to a specific SAF user or group.

SAF directives

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 160
- “AuthSAFExpiration directive” on page 160

- “AuthSAFReEnter directive” on page 161
- “SAFRunAs directive” on page 161

AuthSAFAuthoritative directive

The AuthSAFAuthoritative directive sets whether authorization is passed to lower level modules.

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.

Syntax	`AuthSAFExpiration` *`short_phrase`*
Default	`off`
Context	directory, .htaccess
Module	`mod_authnz_saf`
Values	`off` or *`short_phrase`*

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 "zwasa051_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 `zwasa051_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 `zwasa051_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 `zwasa051_SAF Re-enter new password`.

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.

Syntax	`AuthSAFReEnter` *`short_phrase`*
Default	`Re-enter new password`
Context	directory, .htaccess
Module	`mod_authnz_saf`
Values	`off` or *`short_phrase`*

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 AuthSAFExpiration 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
AuthSAFExpiration "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.

Syntax	`SAFRunAs` *`value`*
Default	`off`
Context	directory, .htaccess

Module
mod_authnz_saf

Values
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.

<*surrogate ID*>: The server will run the request under the ID associated with the SAF surrogate ID specified.

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 ADMINS
AuthzGroupFileAuthoritative Off
AuthSAFAuthoritative Off
</Location>
```

If you want to require a request to run using the SAF privileges associated wit 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 46
These configuration parameters control the FastCGI feature in IBM HTTP Server.

“SSL directives” on page 79
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 21. Troubleshooting IBM HTTP Server

This section describes how to start troubleshooting IBM HTTP Server.

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

This topic contains troubleshooting information on 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 - reference Apache FAQ.
- 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 WASDuplicateSocket, 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

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 V1R6.0 MVS

Extended Addressability Guide for more information. For recommended MEMLIMIT values, see Chapter 3, “Performing required z/OS system configurations,” on page 21.

If you do not set the MEMLIMIT parameter, all parts of the Web server will disable, 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

This topic contains troubleshooting information for known problems with the cryptographic hardware.

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 it created. To determine what shared memory was created, typeipcs -a and for a size of 270760. This was the memory created by **pkcsslotd**.
3. Export EXPSHM=ON.
4. Start the pkcs11 process: /etc/rc.pkcsw11
5. Restart IBM HTTP Server: ./apachectl start

Symptoms of poor server response time

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 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 top of the listen queue. When this condition occurs, the following error 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

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 platforms **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

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

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:

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 right 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.

Chapter 22. Viewing error messages for a target server that fails to start

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

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.

Value	Reason
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

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

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: **SSL0200E: Handshake Failed, <*code*>.**
  - Reason: The handshake failed when the SSL library returned an unknown error.
  - Solution: Report this problem to Service.
- Message: **SSL0201E: Handshake Failed, Internal error - Bad handle.**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.

- 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 Service.
- 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 Service.
- 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 Service 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 Service.
- 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.**
  - Report problem to Service. Reason: An unknown error has occurred in the SSL library.
  - Solution: Report the problem to Service.
- 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 Service.
- 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 Service.
- 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 Chapter 12, “Securing with SSL communications,” on page 67.
- 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 Service.
- Message: **SSL0239E: Handshake Failed, Internal error - write failed.**
  - Reason: The write failed.
  - Solution: None. Report this error to Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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 Service.
- 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.

SSL initialization messages

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 platforms
  - 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 Chapter 3, “Performing required z/OS system configurations,” on page 21
- 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: Iinitialization 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

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 occured**
  - 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.

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.

Notices

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.

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

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. For a current list of IBM trademarks, visit the IBM Copyright and trademark information Web site (www.ibm.com/legal/copytrade.shtml).

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.