



## Site Map

Welcome!

[Where to find out more](#)

[Overview](#)

[Documentation](#)

[New functions in this release](#)

[Other sources of information](#)

[Licensing](#)

[Acknowledgments](#)

[Installation Instructions by operating system:](#)

[on AIX](#)

[on HP](#)

[on Linux](#)

[Linux for S/390: SuSE distribution Init Run command entries](#)

[on Solaris](#)

[on Windows NT and Windows 2000](#)

[From Web download site](#)

[Additional Step for Installing on Windows NT Backup Domain Controller \(BDC\)](#)

[National Language Support](#)

[Getting Started with IBM HTTP Server](#)

[Getting started with LDAP \(Not valid on HP or Linux\)](#)

[Protecting files or directories](#)

[Using key ring files](#)

[SSL and LDAP module](#)

[Creating an LDAP connection](#)

[Supported LDAP servers on IBM HTTP Server](#)

[Getting started quickly with secure connections](#)

[Obtaining certificates](#)

[Buying certificates from an external CA provider](#)

[Creating a self-signed certificate](#)

[Get started quickly without secure connections by operating system:](#)

[on AIX](#)

[on HP](#)

[on Linux](#)

[on Solaris](#)

[on Windows NT and Windows 2000](#)

[Building DSOs or DLLs with IBM HTTP Server](#)

[Compilers](#)

[Components for building on UNIX platform](#)

[Components for building on Windows platform](#)

[Build methods](#)

[Things to consider](#)

[Writing and building dynamic modules on Windows NT and Windows 2000 platforms](#)

[Source files included in the installation](#)

[Building a module](#)

[Configure the server](#)

[Enable client authentication](#)

[Set and view cipher specs](#)

[Setting up Fast Response Cache Accelerator](#)

[Customizing cache management with the Cache Accelerator](#)

[Enabling and configuring the Cache Accelerator](#)

[Customizing logging](#)

[Log Fast Response Cache Accelerator Requests \(valid on AIX, Windows NT and Windows 2000 platforms\)](#)

[Overview](#)

[Specifying the log path and name](#)

[Troubleshooting](#)

[What to do first](#)

[Viewing Logs](#)

[Error messages](#)

[Viewing error messages](#)

[Identifying GSKit certificate support limitations](#)

[Known problems with hardware crypto support](#)

[Known problems on HP platform](#)

[Known problems on Solaris platform](#)

[Known problems on Windows operating systems](#)

[Security on IE V5.01x](#)

[Contacting Customer Service](#)

[Using the Key Management Utility \(IKEYMAN\)](#)

[Before you begin](#)

[Reviewing security configuration example](#)

[Setting up your system environment](#)

[Using the IKEYMAN graphical user interface](#)

## Starting IKEYMAN

Using IKEYMAN or the IKEYCMD line interface

User interface task reference

Using IKEYMAN to store keys on your PKCS11 device

On Linux for S/390: Using the IKEYCMD command line interface

IKEYCMD command line syntax

IKEYCMD command line parameter overview

IKEYCMD command line options overview

Command line invocation

User property file

Apache 2.0 process model overview

Symptoms of an inadequate number of processes for server load

Controlling the number of Apache processes

KeepAlive affects availability of Apache processes to handle requests

Setting expiration dates on static content

apachectl utility (Not valid on Windows NT or Windows 2000 operating systems)

Description

Location

Options

Usage

Default

Notes

Fast Response Cache Accelerator restrictions

Caching

Operational

Supported CA software

Certificate authorities

Using the key management utility (IKEYMAN)

Before you begin

Reviewing security configuration example

Setting up your system environment

Using the IKEYMAN graphical user interface

Starting IKEYMAN

Using IKEYMAN or the IKEYCMD line interface

User interface task reference

Using IKEYMAN to store keys on your PKCS11 device

On Linux for S/390: Using the IKEYCMD command line interface

IKEYCMD command line syntax

[IKEYCMD command line parameter overview](#)

[IKEYCMD command line options overview](#)

[Command line invocation](#)

[User property file](#)

[CipherSpecs](#)

[Browser configuration](#)

[Specifications and key sizes](#)

[Valid cipher specifications](#)

[Client authentication](#)

[Levels of client authentication](#)

[Types of access control](#)

[Configuration files](#)

[Configuration file character support](#)

[AfpA Directives](#)

[AfpAAdvancedTuning](#)

[AfpACache](#)

[AfpAEnable](#)

[AfpALogFile](#)

[AfpAPort](#)

[Apache directives supported by the IBM HTTP Server](#)

[FastCGI directives](#)

[FastCgiAccessChecker](#)

[FastCgiAccessCheckerAuthoritative](#)

[FastCgiAuthenticator](#)

[FastCgiAuthenticatorAuthoritative](#)

[FastCgiAuthorizer](#)

[FastCgiAuthorizerAuthoritative](#)

[FastCgiConfig](#)

[FastCgiExternalServer](#)

[FastCgilpcDir](#)

[FastCgiServer](#)

[FastCgiSuexec](#)

[LDAP directives](#)

[LdapConfigFile](#)

[LDAPRequire](#)

[ldap.application.authType](#)

[ldap.application.DN](#)

[ldap.application.password.stashFile](#)

- ldap.cache.timeout
- ldap.group.memberAttributes
- ldap.group.name.filter
- ldap.group.URL
- ldap.idleConnection.Timeout
- ldap.key.file.password.stashfile
- ldap.key.fileName
- ldap.key.label
- ldap.realm
- ldap.search.timeout
- ldap.transport
- ldap.url
- ldap.user.authType
- ldap.user.cert.filter
- ldap.user.name.fieldSep
- ldap.user.name.filter
- ldap.waitForRetryConnection.interval
- ldap.version

## Windows NT Performance Monitor

## SSL Directives

- Keyfile
- LogLevel
- SSLAcceleratorDisable
- SSLCacheDisable
- SSLCacheEnable
- SSLCacheErrorLog
- SSLCachePath
- SSLCachePortFilename
- SSLCacheTraceLog
- SSLCipherBan
- SSLCipherRequire
- SSLCipherSpec
- SSLClientAuth
- SSLClientAuthGroup
- SSLClientAuthRequire
- SSLCRLHostname
- SSLCRLPort
- SSLCRLUserID

- [SSLDisable](#)
- [SSLEnable](#)
- [SSLFakeBasicAuth](#)
- [SSLPKCSDriver](#)
- [SSLServerCert](#)
- [SSLStashfile](#)
- [SSLV2Timeout](#)
- [SSLV3Timeout](#)
- [SSLVersion](#)

## Environment variables

- [SSL handshake environment variables](#)
  - [Server certificate environment variables](#)
  - [Client certificate environment variables](#)

## IKEYMAN

### Before you begin

- [Reviewing security configuration example](#)
- [Setting up your system environment](#)

### Using the IKEYMAN graphical user interface

#### Starting IKEYMAN

#### Using IKEYMAN or the IKEYCMD line interface

##### User interface task reference

##### Using IKEYMAN to store keys on your PKCS11 device

##### On Linux for S/390: Using the IKEYCMD command line interface

##### IKEYCMD command line syntax

##### IKEYCMD command line parameter overview

##### IKEYCMD command line options overview

##### Command line invocation

##### User property file

## Key sizes

## LDAP

### Introduction and concepts

#### X.500 Overview

#### LDAP Overview

### Querying the LDAP server

#### LDAP search filters

#### Examples of LDAP search filters

#### Installing the LDAP Client

### Configuring LDAP on the IBM HTTP Server

[Using LDAP to protect files](#)

[Protection options](#)

[Without protection](#)

[With password protection](#)

[With secure SSL connections](#)

[Cache session IDs \(UNIX only\)](#)

[SSL](#)

[Security concepts](#)

[What is a secure communication?](#)

[What is encryption?](#)

[What is authentication?](#)

[What is a public key infrastructure?](#)

[What is the secure sockets layer protocol?](#)

[Defining SSL for multiple-IP virtual hosts](#)

[Enabling certificate revocation list on SSL](#)

[Directives needed to set up certificate revocation list](#)

[Directives supported on global server and virtual host](#)

[Enabling cryptographic devices for SSL](#)

[Using SSL password prompting](#)

[Getting started](#)

[Initializing the IBM 4758 device on AIX platform](#)

[Initializing the IBM 4758 device on Windows NT and Windows 2000 platforms](#)

[Initializing PKCS11 token on Windows platforms](#)

[Using IKEYMAN to store keys on a PKCS11 device](#)

[Configuring the IBM HTTP Server to use accelerator devices](#)

[Configuring the IBM HTTP Server to use accelerator and key storage devices](#)

[Configuring the IBM HTTP Server to use key storage devices](#)

[Starting and stopping the server on UNIX with apachectl utility](#)

[Starting and stopping the server on Windows NT and Windows 2000 operating systems](#)

[Reference Links: Acronym list](#)

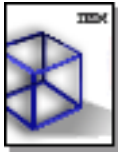
[Reference Links: Apache 2.0 documentation](#)

[Reference Links: Glossary](#)

[IBM Notices](#)

[Trademarks](#)

[Notices](#)



## **Welcome to the IBM HTTP Server**

Welcome to the IBM HTTP Server help. This help describes concepts and tasks related to using the IBM HTTP Server.

Click any topic in the navigation area to the left.

If this information does not display correctly, ensure you have a properly configured browser, supporting HTML V4.51 and Java applets. If you use Netscape Navigator, ensure that you have Version 4.07 or later. If you use Internet Explorer, ensure you have Version 5.0 or later.

## **Where to find out more**

For information about the Apache Server, refer to the [Apache User's Guide](#) and the [IBM HTTP Server Web site](#).

(C) Copyright International Business Machines Corporation 2002. All rights reserved.





## Looking at the product

This section takes a high level look at the IBM HTTP Server, including new functions for this release, locating the IBM HTTP Server documentation and other valuable sources of information. Licensing and acknowledgments are also noted here. Links to related topics appear at the end of this section.

- [Identifying IBM HTTP Server offerings](#)
- [Looking at changes to Apache 2.0 and the Apache Portable Runtime](#)
- [Identifying new functions in this release](#)
- [Upgrading from Apache 1.3 to Apache 2.0](#)
- [Finding IBM HTTP Server documentation](#)
- [Locating other sources of information](#)
- [Licensing](#)
- [Acknowledging copyrighted work](#)
- [Finding related information](#)

## Identifying IBM HTTP Server offerings

The IBM HTTP Server, powered by Apache, is a Web server based on the Apache Web server developed by the Apache Software Foundation (ASF) (<http://httpd.apache.org>). The IBM HTTP Server includes several functions not available in the Apache Web server:

- InstallShield for multiple platforms, enabling consistent installation of the IBM HTTP Server to alternate directories.
- Secure Sockets Layer (SSL) secure connections support.
- Fast Response Cache Accelerator (FRCA) support (Windows and AIX 5.x operating systems). IBM has extended Apache to work with the IBM Fast Response Cache Accelerator, or *Cache Accelerator*. The Cache Accelerator acts as an in-kernel mini HTTP GET engine and HTTP content cache. The FRCA serves static Web pages significantly faster than the Apache Web server, without the Cache Accelerator, enabled by default. Disable the Cache Accelerator completely through configuration file directives.
- Dynamic content generation with FastCGI.
- Simultaneous installation of the IBM HTTP Server in multiple languages on all platforms.

- Web server LDAP authentication protection through an LDAP module.

## Looking at changes to Apache 2.0 and the Apache Portable Runtime

IBM HTTP Server V2.0 is based on a recent General Availability (GA) release of Apache 2.0 and the Apache Portable Runtime (APR), distributed by the Apache Software Foundation. IBM makes some limited changes to Apache 2.0 and APR when creating IBM HTTP Server V2.0. These changes can be broadly classified into two categories:

1. Bug fixes:

IBM HTTP Server is based on GA releases of Apache 2.0 and the APR. The IBM development team selectively backports fixes from the current Apache 2.0 and APR development source code repositories (hosted at <http://cvs.apache.org/>) into IBM HTTP Server. Security fixes receive the highest priority.

2. Changes to support IBM modules:

IBM maintains some changes to support specific IBM requirements.

IBM HTTP Server V2.0 source code is 100% compatible with the release of Apache 2.0 and the APR upon which it is based. By maintaining the proper compiler settings, modules compiled to be dynamically loaded into Apache 2.0 will work with IBM HTTP Server V2.0, subject to the normal restrictions the Apache Module Magic Number imposes.

### Changes to Apache 2.0:

IBM does not make any significant changes to the Apache 2.0 code.


### Changes to Apache Portable Runtime:

IBM has added a socket IO redirection layer application programming interface (API) to the APR. This new API does not change the existing API of APR, as distributed by the Apache Software Foundation. This change enables IBM to introduce some serviceability enhancements and solve some problems with cleanly integrating SSL and the FRCA kernel cache accelerator, without requiring additional changes to the Apache 2.0 code.

The `ihs_patch` file, which installs in the `readme` directory, contains all source code changes made to Apache 2.0 and APR. The `CHANGES_HTTPD`, `CHANGES_APR`, and `CHANGES_APRUTIL` files, which install in the `readme` directory, contain a text description of the changes.

Source code for IBM-specific modules is not provided.

**WINDOWS**

 The IBM HTTP Server, powered by Apache for the Windows operating system, does not run on Windows 95 or Windows 98 operating systems. If you need an Apache Web server for either the Windows 95 or Windows 98 operating system, you can get the source and binary installation images, from the Apache Web site: <http://httpd.apache.org/dist/>.

## Identifying new functions in this release

New functions for this release include:

- Fast Response Cache Accelerator on AIX 5.x
- FastCGI

FastCGI, a language independent, scalable, open extension to Common Gateway Interface (CGI), provides high performance and persistence without the limitations of server-specific APIs.

## Upgrading from Apache 1.3 to Apache 2.0

For a comprehensive list of tips for upgrading Apache 1.3 to Apache 2.0, visit <http://httpd.apache.org/docs-2.0/upgrading.html>.

## Finding IBM HTTP Server documentation

The documentation available for this release, in HTML format resides in the `<IHS install root>/manual/ibm/<Lang>` directory. For the most recent documentation, visit the [IBM HTTP Server Web site](#).

## Locating other sources of information

Several good Internet news groups cover HTTP servers in general. These groups have information about all the popular Web servers and could help you find answers to your questions, so check the news group archives first:

- USENET `news.software.ibm.com`, `ibm.websphere.http-servers`
- <http://www.dejanews.com/>
- `comp.infosystems.www.servers`
- USENET newsgroup `comp.infosystems.www.server.ms-windows`
- USENET newsgroup `comp.infosystems.www.servers.unix`

## Licensing

See the file called `LICENSE. IBM`

## [Acknowledging copyrighted work](#)

We wish to acknowledge the following copyrighted works that make up portions of the IBM HTTP Server, based on Apache software:

Portions of this software were developed at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc. (Bellcore).

This package contains software written and copyrighted by Henry Spencer. See the file called `src/regex/COPYRIGHT`.

The NT port was started with code provided to the IBM HTTP Server based on Apache Group by Ambarish Malpani, of ValiCert, Inc. ([www.valicert.com](http://www.valicert.com)).

### Finding related information

- [Getting started with the IBM HTTP Server](#)
- [Installing and uninstalling the IBM HTTP Server](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



## Getting started with the IBM HTTP Server

This section provides information on getting started with the IBM HTTP Server. Links to related topics appear at the end of this section.

- [Starting the IBM HTTP Server](#)
- [Finding related information](#)

### [Starting the IBM HTTP Server](#)

Issue the following commands from the default directories, or take the appropriate steps to start the IBM HTTP Server, based on the operating system. For example:



```
/usr/IBMIHS/bin/apachectl  
start
```

---



```
/opt/IBMIHS/bin/apachectl  
start
```

---




```
/opt/IBMIHS/bin/apachectl  
start
```

---




```
/opt/IBMIHS/bin/apachectl  
start
```

---

 You can now install the server in any directory of your choosing. Issue the following command from that directory: *directory of choice/bin/apachectl start*


1. Click **Start > Programs > IBM HTTP Server > Start Server**. A message box indicating the server has started briefly appears.
2. Open a browser and type in your server name in the URL box.

WINDOWS

 If you use the developer installation option, then the IBM HTTP Server does not install as a service. You have to run the `apache.exe` file from a command line.

*If the IBM HTTP Server does not start on the Windows operating system:*

WINDOWS

1. Go to **Services** in the Control Panel.
2. Double-click **IBM HTTP Server**.
3. Click .
4. Highlight your user ID from the list, and click **Add**.
5. Click **Start**.

*If the IBM HTTP Server does not start on the Windows operating system:*

WINDOWS

1. Go to **Services** in the Control Panel.
2. Double-click **IBM HTTP Server**. The IBM HTTP Server Properties window appears.
3. Click **Log On** tab.
4. Click **Browse**. The Select User window appears.
5. Highlight your user ID from the list, and click **OK**. The IBM HTTP Server Properties window appears.
6. Click **OK**.
7. Click **Start**.

### Finding related information

- [Getting started quickly with secure connections](#)
- [Locating glossary terms](#)
- [Using the Secure Sockets Layer protocol for secure communications](#)

---

[\(Back to the top\)](#)



## **Getting started quickly with secure connections**

This section provides information to help you get started with secure connections. This information includes how to obtain certificates, create self-signed certificates and set up the Secure Sockets Layer (SSL). Links to related topics appear at the end of this section.

- [Obtaining certificates](#)
  - [Buying a certificate from an external certificate authority provider](#)
  - [Creating a self-signed certificate](#)
- [Setting up Secure Sockets Layer using the configuration file](#)
- [Starting a secure virtual host](#)
- [Finding related information](#)

### **Obtaining certificates**

When you set up secure connections, associate your public key with a digitally signed certificate from a certificate authority (CA), designated as a trusted CA on your server.

You can obtain a certificate two ways:

- [Buy a certificate from an external CA provider](#)
- [Create a self-signed certificate](#)

### ***Buying 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 [Listing trusted CAs on the IBM HTTP Server](#) for a list.

Use [IKEYMAN](#) to create a new key pair and certificate request to send to an external CA. Then define SSL settings in the Security folder in the Administration Server.

### **Creating a self-signed certificate**

To create a self-signed certificate, you can use your [key management utility](#)



(IKEYMAN), or you can purchase [certificate authority software](#) from a CA provider.

## [Setting up Secure Sockets Layer using the default configuration file](#)

To set up Secure Sockets Layer (SSL) using the default configuration file (`<install_root>/conf/httpd.conf`):



1. Specify the `SSLEnable` directive in the configuration file, to enable SSL.
2. Specify a `Keyfile` directive and any SSL directives you want to enable.
3. If you run the IBM HTTP Server on the Linux for PowerPC (PPC) operating system, you need to add the **Listen 0.0.0.0:443** directive to the configuration file to enable SSL. If you do not specify this directive, you will receive a `PEER_ID_NOT_SET` error in the error log when you try to connect to the server.
4. Restart the server.

## [Starting a secure virtual host](#)

To start a secure virtual host:

1. Specify the `SSLEnable` directive in 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 to enable for that particular virtual host. You can specify any directive, with the exception of the cache directives, inside a virtual host.
3. Restart the server.

### **Finding related information**

- [Associating your public key with certificate authorities](#)
- [Defining Secure Sockets Layer for multiple virtual hosts](#)
- [Locating glossary terms](#)
- [Understanding Secure Sockets Layer environment variables](#)
- [Using Secure Sockets Layer directives](#)
- [Using the Secure Sockets Layer protocol for secure communications](#)
- [Using SSL password prompting](#)

[\(Back to the top\)](#)



## Associating your public key with certificate authorities

This section contains information regarding trusted certificate authorities (CAs) on the IBM HTTP Server, along with supported CA software. Links to related information appear at the end of this section.

- [Listing trusted certificate authorities on the IBM HTTP Server](#)
- [Identifying supported certificate authority software](#)
- [Finding related information](#)

Associate your public key with a digitally signed certificate from a certificate authority (CA), 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. The IBM HTTP Server supports the following external CAs:

- [Thawte](#)
- [VeriSign](#)

## [Listing trusted certificate authorities on the IBM HTTP Server](#)

By default, the following list contains designated trusted CAs on the IBM HTTP Server:

- RSA Secure Server Certification Authority
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium 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 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Test CA Root Certificate

## Identifying supported certificate authority software

The IBM HTTP Server supports the following certificate authority (CA) software:

- Any X.509-compliant certificate authority
- [Entrust](#)
- [Netscape Certificate Server](#)
- [Tivoli PKI](#)
- [XCert](#)

### **Finding related information**


- [Authenticating clients](#)
- [Creating a self-signed certificate](#)
- [Enabling a certificate revocation list in Secure Sockets Layer](#)
- [Enabling client authentication](#)
- [Getting started quickly with secure connections](#)
- [Locating glossary terms](#)
- [Using the Secure Sockets Layer protocol for secure communications](#)

---

[\(Back to the top\)](#)



## Authenticating clients

The IBM HTTP Server supports three levels of client authentication and two types of access control, based on client certificate information. This section discusses setting the desired client authentication level and the access control type, along with associated notes . Links to related information appear at the end of this section.

- [Setting client authentication levels](#)
- [Setting access control types](#)
- [Finding related information](#)

### Setting client authentication levels

Set the level of client authentication with the `SSLClientAuth` directive:

- `Required, (2)`
- `Optional, (1)`
- `None, (0)`

You can add a second argument, `crl`, to use a certificate revocation list. For example, `SSLClientAuth 1 crl`

### Choosing the Required level

If you choose the Required level of client authentication, the secure server requests a certificate, from all clients making an HTTPS request. The server validates clients by checking for a *trusted Certificate Authority root certificate* in the local key database. A trusted Certificate Authority (CA) root certificate is a certificate signed by a certificate authority, designated as a trusted CA on your server.

The server establishes a secure connection if the client has a valid certificate. The server denies the request if the client has an expired certificate, or if a certificate authority (CA) that is not designated as a trusted CA on the server, signs the certificate.



SSL client authentication increases network traffic.

### Choosing the Optional level

If you choose the Optional level of client authentication, the server requests a client certificate. If the client does not provide a certificate, the server still establishes a secure connection. The server denies the request if the client has provided an expired certificate, or if a certificate authority (CA) that is not designated as a trusted CA on the server, signs the certificate.



SSL client authentication increases network traffic.

### Choosing None

If you choose None, the secure server does not request certificates from clients.

### Specifying certificate revocation list as Second Argument

Specifying certificate revocation list (CRL) as a second argument to the `SSLClientAuth` directive enables CRL checking. You cannot enable CRL if "None" has been selected as the first argument.

### Setting access control types

Set the access control type with the `SSLFakeBasicAuth`, or `SSLClientAuthRequire` directives.



The `SSLClientAuthRequire` directive is the preferred type of client authentication.

## SSLFakeBasicAuth directive

Using the SSLFakeBasicAuth directive is not recommended. Password files generated for use with Apache SSL code, or mod\_ssl and Apache, do not work with the IBM HTTP Server because of differing distinguished name formats.

The SSLFakeBasicAuth type provides a simple method for client authentication. If you specify SSLFakeBasicAuth, the client certificate distinguished name and the password ("password") are Base64-encoded and placed in the authorization header. Put the mod\_ibm\_ssl module first in the module list, so that subsequent authentication modules have the fake basic authentication user ID and password available. Basic authentication support within a specified virtual host does not work because the client distinguished name and the password, `password`, overwrite the user ID and password supplied by a user.

To display the distinguished name from a client certificate, create a CGI program to print out the SSL\_CLIENT\_DN environment variable.

## SSLClientAuthGroup directive

This directive enables you to specify a logic string of specific client certificate attributes and group them together as a single unit. This ability enables a certain set of client certificate attributes access to multiple objects on that server.

The syntax is **SSLClientAuthGroup (name) (expression)**

Use parentheses to group comparisons. If the value of the attribute contains a non-alphanumeric character, delimit the value with quotes.

Valid attributes follow:

```
CommonName
Country
Email
Group
IssuerCommonName
IssuerCountry
IssuerEmail
IssuerLocality
IssuerOrg
IssuerOrgUnit
IssuerStateOrProvince
Locality
Org
OrgUnit
StateOrProvince
```

The following short names are also valid:

```
CN, C, E, G, ICN, IC, IE, IL, IO, IOU, IST, L, O, OU, ST
```

## SSLClientAuthRequire directive

The more extensive SSLClientAuthRequire support enables the webmaster to define logical expressions containing the x509 attributes. Comparisons between these logical expressions and the client certificate information determine object access. Before processing occurs, GSKit validates the client certificate to ensure that a trusted certificate authority signed the certificate.

The SSLClientAuthRequire directive enables a webmaster to build a logical expression consisting of attribute checks linked with AND, OR, and NOTs. You can also use parentheses. For example:

```
SSLClientAuthRequire (CommonName = "Fred Smith" OR CommonName = "John Deere") AND Org = IBM
```

means that only client certificates containing a common name of either Fred Smith, or John Deere, and an

organization of IBM can have object access.

For the attribute checks, the only valid comparisons are equal and not equal (= and !=). You can link each attribute check with AND, OR, or NOT (also &&, ||, and !). When you specify multiple SSLClientAuthRequire directives for one resource, the resource acts as if Boolean AND operators join the values.

Use parentheses to group comparisons. If the value of the attribute contains a nonalphanumeric character, delimit the value with quotes.

Valid attributes follow:

```
CommonName
Country
Email
IssuerCommonName
IssuerCountry
IssuerEmail
IssuerLocality
IssuerOrg
IssuerOrgUnit
IssuerStateOrProvince
Locality
Org
OrgUnit
StateOrProvince
```

The following short names are also valid:

```
CN, C, E, ICN, IC, IE, IL, IO, IOU, IST, L, O, OU, ST
```

### Finding related information


- [Associating your public key with certificate authorities](#)
- [Authenticating clients](#)
- [Enabling a certificate revocation list in Secure Sockets Layer](#)
- [Enabling client authentication](#)
- [Locating glossary terms](#)
- [Understanding Secure Sockets Layer environment variables](#)
- [Using Secure Sockets Layer directives](#)
- [Using the SSLFakeBasicAuth directive](#)
- [Using the SSLClientAuthRequire directive](#)
- [Using the SSLClientAuthGroup directive](#)

---

[\(Back to the top\)](#)



## Using Secure Sockets Layer directives

This section provides information on using SSL directives. This information includes specific syntax, descriptions, scopes and associated notes . Links to related topics appear at the end of this section.

- [Keyfile](#)
- [LogLevel](#)
- [SSLAcceleratorDisable](#)
- [SSLCacheDisable](#)
- [SSLCacheEnable](#)
- [SSLCacheErrorLog](#)
- [SSLCachePath](#)
- [SSLCachePortFilename](#)
- [SSLCacheTraceLog](#)
- [SSLCipherBan](#)
- [SSLCipherRequire](#)
- [SSLCipherSpec](#)
- [SSLClientAuth](#)
- [SSLClientAuthGroup](#)
- [SSLClientAuthRequire](#)
- [SSLCRLHostname](#)
- [SSLCRLPort](#)
- [SSLCRLUserID](#)
- [SSLDisable](#)
- [SSLEnable](#)
- [SSLFakeBasicAuth](#)
- [SSLFIPSDisable](#)
- [SSLFIPSEnable](#)
- [SSLPKCSDriver](#)
- [SSLServerCert](#)
- [SSLStashfile](#)
- [SSLV2Timeout](#)
- [SSLV3Timeout](#)
- [SSLVersion](#)
- [Finding related information](#)

### Keyfile

- Description: Sets the key file to use.
- Default: No default
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global base and virtual host
- Syntax: `Keyfile [prompt]/fully qualified path to key file/keyfile.kdb`
- Values: File name of the key file. Use the `prompt` option to enable the HTTP server to prompt you for the Key file password during start up. See [Using SSL Password Prompting](#).

### LogLevel

- Description: Adjusts the verbosity of the messages recorded in the error logs. When you specify a particular level, the server reports messages from all other levels of higher significance. For example, when you specify `LogLevel info`, the server reports messages with log levels of `notice` and `warn`. Specifying at




least `level crit` is recommended.

- Default: `LogLevel error`
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Allowed. Order of preference is top to bottom, first to last. If the client does not support cipher specifications, the connection closes.
- Scope: Server configuration, virtual host
- Syntax: `LogLevel level`
- Values: The following available levels appear in order of decreasing significance:

<u>Level</u>	<u>Description</u>	<u>Example</u>
<code>emerg</code>	Emergencies: system rendered unusable.	"Child cannot open lock file. Exiting"
<code>alert</code>	Take immediate action.	"getpwuid: could not determine user name from uid"
<code>crit</code>	Critical conditions.	"socket: Failed to get a socket, exiting child"
<code>error</code>	Error conditions.	"Premature end of script headers"
<code>warn</code>	Warning conditions.	"child process 1234 did not exit, sending another SIGHUP"
<code>notice</code>	Normal, but significant condition.	"httpd: caught SIGBUS, attempting to dump core in ..."
<code>info</code>	Informational.	"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."
<code>debug</code>	Debug-level messages.	"Opening configuration file ..."

## SSLAcceleratorDisable

- Description: Disables the accelerator device.
- Default: Accelerator device is enabled
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Not allowed
- Scope: Virtual and global
- Syntax: `SSLAcceleratorDisable`
- 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.

## SSLCacheDisable

- Description: Disables the external SSL session ID cache
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical Apache server instance, allowed only outside of virtual host stanzas
- Syntax: SSLCacheDisable
- Values: None

 Valid only in UNIX environments.

UNIX

## SSLCacheEnable

- Description: Enables the external SSL session ID cache
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical Apache server instance, allowed only outside of virtual host stanzas
- Syntax: SSLCacheEnable
- Values: None

 Valid only in UNIX environments.

UNIX

## SSLCacheErrorLog

- Description: Sets the file name for session ID cache error logging
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical server instance, allowed only outside of virtual host stanzas.
- Syntax: SSLCacheErrorLog /usr/HTTPServer/log/sidd\_log
- Values: Valid file name

 Not valid on Windows operating system.

UNIX

## SSLCachePath

- Description: Specifies the path to the session ID caching daemon executable.
- Default: <server-root>/bin/sidd
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical IBM HTTP Server
- Syntax: SSLCachePath /usr/HTTPServer/bin/sidd
- Values: Valid path name.

 Not valid on Windows operating system.

UNIX

## SSLCachePortFilename

- Description: Sets the file name for the UNIX domain socket used for communication between the server instances and the session ID cache daemon.
- Default: If this directive is not specified and the cache is enabled, then the server attempts to use the file: `<server-root>/logs/siddport`
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical Apache server instance, allowed only outside of virtual host stanzas.
- Syntax: `SSLCachePortFilename /usr/HTTPServer/logs/siddport`
- Values: Valid file name. ⚠ The Web server deletes this file during startup; do not use an existing file name.  
⚠ Valid only on UNIX platform.

UNIX

## SSLCacheTraceLog

- Description: Specifies the trace log to which session ID trace messages log.
- Default: None
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Not allowed
- Scope: One per physical IBM HTTP Server
- Syntax: `SSLCacheTraceLog /usr/IBMIHS/log/sidd-trace.log`
- Values: Valid path name.  
⚠ Not valid on Windows operating systems.

## SSLCipherBan

- Description: Denies access to an object, if the client attempts the specified cipher.
- Default: None
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Allowed per directory stanza. Order of preference is top to bottom.
- Scope: Multiple instances per directory stanza
- Syntax: `SSLCipherBan <cipher specification>`
- Values: See [SSL Version 2 Cipher Specifications](#), [SSL Version 3 and TLS Version 1 Cipher Specifications](#)

## SSLCipherRequire

- Description: Allows limited access to objects according to specified ciphers.
- Default: None
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Allowed per directory stanza. Order of preference is top to bottom.
- Scope: Multiple instances per directory stanza
- Syntax: `SSLCipherRequire <cipher specification>`
- Values: See [SSL Version 2 Cipher Specifications](#), [SSL Version 3 and TLS Version 1 Cipher Specifications](#)

## SSLCipherSpec


- Description: Specifies a cipher specification that you can use in a secure transaction.
- Default: If nothing is specified, the server uses all cipher specifications available from the installed GSK library.
- Module: `mod_ibm_ssl`

- Multiple instances in the configuration file: Allowed. Order of preference is top to bottom, first to last. If the client does not support the cipher specifications, the connection closes.
- Scope: Virtual host
- Syntax: `SSLCipherSpec shortname` or `SSLCipherSpec longname`
- Values: See [SSL Version 2 Cipher Specifications](#), [SSL Version 3 and TLS Version 1 Cipher Specifications](#)


### **Version 2 Cipher Specifications**

<b><u>Short name</u></b>	<b><u>Long name</u></b>	<b><u>Description</u></b>
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**


<b><u>Short name</u></b>	<b><u>Long name</u></b>	<b><u>Description</u></b>
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)
36 (See  )	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RC2 MD5 (40-bit)
32	SSL_RSA_WITH_NULL_SHA	
31	SSL_RSA_WITH_NULL_MD5	
30	SSL_NULL_WITH_NULL_NULL	

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

 Cipher specification 36 requires Netscape Navigator V4.07; it does not work on earlier versions of Netscape browsers.

## SSLClientAuth

- Description: Sets the mode of client authentication to use (none (0), optional (1), or required (2)).
- Default: `SSLClientAuth none`
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: One instance per virtual host
- Scope: Virtual host
- Syntax: `SSLClientAuth <level required> [crl]`
- 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

- Description: Enables you to group client certificate attributes together for use in the `SSLClientAuthRequire` directive.
- Default: None
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: Allowed. The function joins these directives by "AND".
- Scope: Multiple instances per directory stanza
- Syntax: `<SSLClientAuthGroup group name> <logic string>`
- Values: Logical expression consisting of attribute checks linked with AND, OR, NOT, and parentheses.

### **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.

A listing of valid attributes follows:

- CommonName
- Country
- Email
- Group
- IssuerCommonName
- IssuerCountry
- IssuerEmail
- IssuerLocality
- IssuerOrg
- IssuerOrgUnit
- IssuerStateOrProvince
- Locality
- Org
- OrgUnit
- StateOrProvince

A listing of valid short names follows:

CN, C, E, G, ICN, IC, IE, IL, IO, IOU, IST, L, O, OU, ST

## SSLClientAuthRequire

- Description: Enables extensive validation of client certificate information before serving an object
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in a configuration file: Allowed. The function joins these directives by "AND".
- Scope: Directory
- Syntax: `SSLClientAuthRequire CommonName = Richard`
- Values: [Logical expression](#) consisting of attribute checks linked with AND, OR, NOT, and parentheses.

## ***Description of valid logical expressions***

For example:

```
SSLClientAuthRequire (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.

A listing of valid attributes follow:

- CommonName
- Country
- Email
- IssuerCommonName
- IssuerCountry
- IssuerEmail
- IssuerLocality
- IssuerOrg
- IssuerOrgUnit
- IssuerStateOrProvince
- Locality

- Org
- OrgUnit
- StateOrProvince

A listing of valid short names follows:

CN, C, E, ICN, IC, IE, IL, IO, IOU, IST, L, O, OU, ST

## SSLCRLHostname

- Description: TCP/IP name, or address of LDAP server, where CRL database resides.
- Default: SSLCRLHostname is disabled by default.
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global server or virtual host
- Syntax: `SSLCRLHostname <TCP/IP name or address>`
- Values: TCP/IP name or address of LDAP server

## SSLCRLPort

- Description: Port of LDAP server, where CRL database resides.
- Default: SSLCRLPort is disabled by default.
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global server or virtual host
- Syntax: `SSLCRLPort <port number>`
- Values: Port of LDAP server; default=389

## SSLCRLUserID

- Description: User ID to send to the LDAP server, where CRL database resides.
- 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
- Scope: Global server or virtual host
- Syntax: `SSLCRLUserID <[prompt]userid>`
- 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. See [Using SSL Password Prompting](#).

## SSLDisable

- Description: Disables SSL for this virtual host.
- Default: SSL is disabled by default.
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global server or virtual host
- Syntax: `SSLDisable`
- Values: None

## SSLEnable

- Description: Enables SSL for this virtual host.
- Default: SSL is disabled by default.

- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server>
- Scope: Global server or virtual host
- Syntax: SSLEnable
- Values: None

## SSLFakeBasicAuth

- Description: 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 the password `password`.
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Within a directory stanza, used along with AuthName, AuthType, and require directives.
- Syntax: SSLFakeBasicAuth
- Values: None

## SSLFIPSDisable

- Description: Disables Federal Information Processing Standards (FIPS).
- Default: FIPS is disabled by default.
- Scope: Virtual and global.
- Syntax: SSLFIPSDisable

## SSLFIPSEnable

- Description: Enables Federal Information Processing Standards (FIPS).
- Default: FIPS is disabled by default.
- Scope: Virtual and global.
- Syntax: SSLFIPSEnable

## SSLPKCSDriver

- Description: Identifies the fully qualified name to the module, or driver used to access the PKCS11 device
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global server, or virtual host
- 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.
- Values: Path and name of PKCS11 module, or driver.

The default locations of the modules for each PKCS11 device follow, by 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



## SSLServerCert

- Description: Sets the server certificate to use for this virtual host
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host
- Scope: IP-based virtual hosts
- Syntax: `SSLServerCert [prompt]my_certificate_label; on PKCS11 device - SSLServerCert mytokenlabel:mykeylabel`
- Values: Certificate label. Use the /prompt option to enable the HTTP server to prompt you for the Crypto token password during start up. See [Using SSL Password Prompting](#).

Use no delimiters around the certificate label. Ensure that the label is contained on one line; leading and trailing white space is ignored.

## SSLStashfile

- Description: Indicates path to file with file name, containing the encrypted password for opening the PKCS11 device.
- Default: None
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Virtual host and global server
- Syntax: `sslstash [-c] <file> <function> <password>`, where:
  - **-c** = Create a new stash file. If not specified, the server updates an existing stash file
  - **File** = Fully qualified name of the file to create or update
  - **Function** = Function with which to use the password Valid values include *crl* or *crypto*
  - **Password** = The password to stash
  - **Usage** - `sslstash -c conf\pkcs11.passwd crypto pkcs11`
- Values: Path with file name

Locate an **sslstash** command in the `bin` directory of the IBM HTTP Server, for UNIX, and the server installation root for the Windows platform. Use this command to store the password for the PKCS11 device. The stash file created after using the **sslstash** command can hold two different passwords for two different functions: *crl* and *cryptology*.

## SSLV2Timeout

- Description: Sets the timeout for SSL Version 2 session IDs
- Default: 40
- Module: mod\_ibm\_ssl
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global base and virtual host

- Syntax: `SSLV2Timeout 60`
- Values: 0 to 100 seconds

## SSLV3Timeout

- Description: Sets the timeout for SSL Version 3 session IDs
- Default: 120
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: Global base and virtual host
- Syntax: `SSLV3Timeout 1000`
- Values: 0 to 86400 seconds

## SSLVersion

- Description: Enables object access rejection, if the client attempts to connect with an SSL protocol version other than the one specified.
- Default: None
- Module: `mod_ibm_ssl`
- Multiple instances in the configuration file: One instance per virtual host and global server
- Scope: One per directory stanza
- Syntax: `SSLVersion ALL`
- Values: `SSLV2|SSLV3|TLSV1|ALL`

### **Finding related information**

- [Authenticating clients](#)
- [Caching session IDs](#)
- [Enabling client authentication](#)
- [Enabling session ID caching](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



## Using SSL Password Prompting

For users who are concerned with storing passwords in stash files, the IBM HTTP Server provides the ability to enter the passwords required for the IBM SSL module during server initialization. Entering the `/prompt` option on the appropriate directive will direct the HTTP Server to prompt the user to enter the appropriate passwords during start-up. Users are prompted for passwords only during full start-up. This option does not work during restart.

Prompting can be enabled for the following functions:

- The Key Database file password
- The password associated with the user ID used for communication with an LDAP server for Certificate Revocation List (CRL) processing
- The password associated with the Crypto Card token

## Caching session IDs

To offset the expense of SSL handshakes between a client and server, cache session IDs. Cached session IDs enable a client and server to communicate with a shortened handshake.

You can set timeout values that indicate how long entries last in the session ID cache.

### Finding related information

- [Enabling session ID caching](#)
- [Locating glossary terms](#)
- [Using the SSLV2Timeout directive](#)
- [Using the SSLV3Timeout directive](#)

## Enabling session ID caching

**Cached session IDs** enable a client and server to communicate with a shortened handshake.



WINDOWS

To enable session ID caching on Windows platforms:

1. Set the timeout value that applies to the session ID cache to a value greater than 0. Specify the **SSLV2Timeout** directive with valid values between 0 and 100, and the **SSLV3Timeout** directive with valid values between 0 and 86400. These values appear in seconds.
2. Save the configuration file and restart the server.

To enable session ID caching on UNIX platforms:



UNIX

1. Accept the default, or specify the **SSLCacheEnable** directive in the configuration file outside of a virtual host stanza.
2. Assign a name to the port for the session ID cache, by specifying the **SSLCachePortFilename**, if the default name in the `<server-root>/logs` directory appears unacceptable.
3. Set the timeout value that applies to the session ID cache. Specify the **SSLV2Timeout** directive with valid values between 0 and 100, and the **SSLV3Timeout** directive with valid values between 0 and 86400. These values appear in seconds.
4. Decide whether to log caching errors. To enable logging of errors that can occur during session ID caching, or retrieval from the cache, specify the **SSLCacheErrorLog** directive in the configuration file outside of a virtual host stanza.
5. Save the configuration file and restart the server.

### Finding related information

- [Caching session IDs](#)
- [Finding the default and sample configuration file](#)
- [Locating glossary terms](#)
- [Using Secure Sockets Layer directives](#)



## Finding the default and sample configuration files

- [Supporting configuration file characters](#)
- [Finding related information](#)

Locate the `httpd.conf` configuration file in the `conf` directory of your server installation. This file also installs as a default configuration file, in case you need to use another copy of the original file.

The product provides a sample configuration file called `httpd.conf.sample`, illustrating basic IBM module directives and advanced security options.

### [Supporting configuration file characters](#)

The IBM HTTP Server configuration file, `httpd.conf`, supports only single-byte characters (SBCS). This restriction applies to all operating system platforms.

### **Finding related information**


- [Getting started quickly with secure connections](#)
  - [Getting started quickly without secure connections](#)
  - [Getting started with the IBM HTTP Server](#)
  - [Locating glossary terms](#)
  - [Setting up advanced security options](#)
-



## Getting started quickly without secure connections

Starting the server without secure connections does not require any changes to the configuration files.

- [Getting started on the AIX operating system](#)
- [Getting started on the HP operating system](#)
- [Getting started on the Linux operating system](#)
- [Getting started on the Solaris operating system](#)
- [Getting started on the Windows operating systems](#)
- [Finding related information](#)

 The following directories listed under each operating system are the defaults. You can now install the IBM HTTP Server in any directory of your choosing.

### Getting started on the AIX operating system

Go to the command prompt in the `usr/IBMIHS/bin` directory and type `./apachectl start`. The `apachectl` utility supports start and other options.

### Getting started on the HP operating system

Go to the command prompt in the `/opt/IBMIHS/bin` directory. Type `./apachectl start`. The `apachectl` utility supports start and other options.

### Getting started on the Linux operating system

Go to the command prompt in the `/opt/IBMIHS/bin` directory. Type `./apachectl start`. The `apachectl` utility supports start and other options.


### Getting started on the Solaris operating system

Go to the command prompt in the `/opt/IBMIHS/bin` directory. Type `./apachectl start`. The `apachectl` utility supports start and other options.

### Getting started on the Windows operating systems

The server installs as a service, and runs automatically. When you reboot, the server starts. You can control the server by either:

- Using the standard Windows Service Control panels.
- Using the **net start** and **net stop** commands. For more information about these commands, see the Windows help file. Access these commands from the Start menu, clicking **Start > Programs > IBM HTTP Server**.

 You have the option to run the server from the command line. You do not need to run the server as a service. See [Starting and stopping on Windows operating systems](#) for details.

### Finding related information

- [Getting started with the IBM HTTP Server](#)
- [Locating glossary terms](#)
- [Starting and stopping the IBM HTTP Server on UNIX systems](#)

---

[\(Back to the top\)](#)





## Starting and stopping the IBM HTTP Server on UNIX systems

This section provides information on using the Apachectl utility. Links to related information appear at the end of this section.

- [Using the Apachectl utility](#)
- [Finding related information](#)

UNIX

### Using the Apachectl utility

The Apachectl utility starts and stops the IBM HTTP Server on UNIX systems.

**Location:** You can find the Apachectl utility located in the `base_directory/bin` directory on the following operating systems:

- AIX: Locate the base directory at `/usr/IBMIHS`.
- HP: Locate the base directory at `/opt/IBMIHS`.
- Linux: Locate the base directory at `/opt/IBMIHS`.
- Solaris: Locate the base directory at `/opt/IBMIHS`.

**Options:** The Apachectl utility contains the following options:

- **Start:** Starts server.
- **Stop:** Stops server.
- **Restart:** Restarts server by sending SIGHUP, or starts the server if it is not running.
- **Fullstatus:** Dumps a full status screen; requires the Lynx text browser and the `mod_status` module.
- **Status:** Dumps a short status screen; requires the Lynx text browser and the `mod_status` module.
- **Graceful:** Restarts server by sending SIGUSR1, or starts the server if it is not running, without shutting down completely.
- **Configtest:** Performs a configuration syntax test. This option does not check semantics; see error log when starting server.
- **Help:** Provides help on the Apachectl utility.

**Syntax:** `./apachectl graceful`

**Default:** None

## Finding related information

- [Getting started quickly with secure connections](#)
- [Getting started quickly without secure connections](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



## Glossary

▶ [Link to Acronym List](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

### A

**absolute path** The full path of an object that begins with the root directory.

**abstract syntax notation one (ASN.1)** In the Distributed Computing Environment (DCE), a data representation scheme that enables complicated types to be defined and enables values of these types to be specified.

**access control lists (ACL)** A file attribute that contains the basic and extended permissions that control access to the file.

**adapter** A part that electrically or physically connects a device to a computer or to another device.

**agent** A function that represents a requester to a server. An agent can be present in both a source and a target system.

**algorithm** A finite set of well-defined rules for the solution of a problem in a finite number of steps.

**alias** An alternative name that can be used instead of the primary name.

**APache eXtenSion (APXS)** A support program that simplifies the creation of dynamic shared object (DSO) files for Apache modules (especially for third-party modules). It can be used to build DSO-based modules outside of the Apache source tree.

**argument** An expression that is passed to a function or subroutine for evaluation.

**asymmetric keys** In computer security, the two keys in a key pair. The keys are called asymmetric because one key holds more of the encryption pattern than the other does.

**attribute** (1) A characteristic that identifies and describes a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object. (2) Information within a managed object that is visible at the object boundary. An attribute has a type, which indicates the range of information given by the attribute, and a value, which is within that range.

**authentication** In computer security, verification of the identity of a user or the user's eligibility to access an object.

[\(Back to the top\)](#)

## B

**base64** Base64 is a command line utility which encodes and decodes files in this format. It can be used within a pipeline as an encoding or decoding filter, and is most commonly used in this manner as part of an automated mail processing system.

**basic encoding rules (BER)** A set of rules used to encode abstract syntax notation one (ASN.1) values as strings of octets.

**Boolean** A value of 0 or 1 represented internally in binary notation. Any operation in which each of the operands and the result take one of two values.

**browser** A client program that initiates requests to a Web server and displays the information that the server returns.

**buffer** A routine or an area of storage that compensates for the different speeds of data flow or timings of events, when transferring data from one device to another.

[\(Back to the top\)](#)

## C

**cache** To place, hide, or store frequently used information locally for quick retrieval.

**cache accelerator** Provides support for caching on multiple Web servers and on servers with multiple IP addresses.

**certificate authority (CA)** In computer security, an organization that issues certificates. The certificate authority authenticates the certificate owner's identity and the services that the owner is authorized to use. It also manages the issuance of new certificates and revokes certificates from unauthorized users who are no longer authorized to use them. A certificate authority is considered to be trusted when a user accepts any certificate issued by that certificate authority as proof of the certificate owner's identity.

**certificate revocation list (CRL)** A list of certificates that need to be revoked before their expiration date.

**cipher** In Cryptographic Support, data that is unintelligible to all except those who have the key to decode it to plaintext.

**ciphertext** In Cryptographic Support, data that is unintelligible to all except those who have the key to decode it to plaintext.

The output of an encryption function. Encryption transforms plaintext into ciphertext.

**class** (1) In object-oriented design or programming, a model or template that can be instantiated to create objects with a common definition and therefore, common properties, operations, and behavior. An object is an instance of a class. (2) In the AIX operating system, pertaining to the I/O characteristics of a device. System devices are classified as block or character devices

**Common Gateway Interface (CGI)** A standard for the exchange of information between a Web server and computer programs that are external to it. The external programs can be written in any programming language that is supported by the operating system on which the Web server is running.

**component** A reusable object or program that performs a specific function and is designed to work with other components and applications.

**connector** In a query management command, the TO word in the EXPORT command, the FROM word in the IMPORT command, or the AS word in the SAVE DATA command.

**container** A Java run-time environment for enterprise beans. A container, which runs on an Enterprise JavaBeans server, manages the life cycles of enterprise bean objects, coordinates distributed transactions, and implements object security.

**conversational monitor system** An operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under the control of the VM control program.

**cryptographic support** The IBM licensed program that provides support for the encryption and decryption of data, according to the Data Encryption Algorithm, and for the management of cryptographic keys and personal identification numbers (PINs).

[\(Back to the top\)](#)

## D

**daemon** A program that runs unattended to perform continuous or periodic systemwide functions, such as network control.

**Data Encryption Standard (DES)** In computer security, the National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm.

**data link control (DLC)** The protocol layer used by nodes on a data link to accomplish an orderly exchange of information.

**decrypt** In cryptographic support, to convert ciphertext into plaintext.

**default** A value, attribute, or option that is automatically supplied or assumed by the

system or program when no value is specified by the user.

**delimited identifier** A sequence of characters enclosed within double quotation marks (").

**digest** Data that has been organized into a format that provides for quick access to each piece of data.

**digital certificate** A form of personal identification that can be verified electronically. Only the certificate owner who holds the corresponding private key can present a certificate for authentication through a Web browser session. Anyone can verify that the certificate is valid by using a readily available public key.

**digital signature** Information that is encrypted with an entity private key and is appended to a message to assure the recipient of the authenticity and integrity of the message. The digital signature proves that the message was signed by the entity that owns, or has access to, the private key or shared secret symmetric key.

**directive** A statement that is used in the configuration file for a Web server to define a particular setting for the server.

**Directory Access Protocol (DAP)** The X.500 protocol that a directory user agent uses to obtain directory information from a remote directory system agent.

**distinguished name (DN)** In computer security, information that uniquely identifies the owner of a certificate.

**domain** An object, icon, or container that contains other objects representing the resources of a domain. You can use the domain object to manage those resources.

**dynamic link library (DLL)** A file containing executable code and data bound to a program at load time or run time, rather than during linking. Several applications can share the code and data in a dynamic link library simultaneously.

**dynamic shared object (DSO)** A mechanism which provides a way to build a piece of program code in a special format for loading at run time into the address space of an executable program. The DSO gets knowledge of the executable program symbol set as if it had been statically linked with it in the first place

**dump** (1) To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging. (2) To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer. (3) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

[\(Back to the top\)](#)

## E

**emulation** The imitation of one computing system by another system through the use of software and hardware that allow the latter to run programs written for the former.

**encoding** The underlying part of a code page that defines: a) the coding space (the number and allowable value of code points in a code page); b) the rules for sharing the coding space between control and graphic characters; and c) the rules related to the specific options permitted in that scheme.

**encrypt** In Cryptographic Support, to systematically scramble information so that it cannot be read without knowing the coding key.

**enterprise bean** A nonvisual software component that conforms to the Sun Microsystems, Inc. Enterprise JavaBeans architecture. An enterprise bean is designed to be installed on a server and accessed remotely from a client. It realizes the standard component architecture for building distributed object-oriented business applications in the Java programming language.

**entity** Any concrete or abstract thing of interest, including associations among things; for example, a person, object, event, or process that is of interest in the context under consideration, and about which data may be stored in a database.

**environment variable** A variable that specifies how an operating system or another program runs, or the devices that the operating system recognizes.

**error** A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

**error log** (1) A data set or file in a product or system where error information is stored for later access. (2) A form in a maintenance library that is used to record error information about a product or system. (3) A record of machine checks, device errors, and volume statistical data.

**extension** A class of objects designated by a specific term or concept; denotation.

[\(Back to the top\)](#)

## F

**Fast Common Gateway Interface Protocol (FastCGI)** The Fast Common Gateway Interface (FastCGI) is an enhancement to the existing Common Gateway Interface (CGI), which is a standard for interfacing external applications with Web servers.

**filter** (1) A device or program that separates data, signals, or material in accordance with specified criteria. (2) On the AIX operating system, a command that reads standard input data, modifies the data, and sends it to the display screen.

**firmware** Proprietary code that is usually delivered as microcode as part of an

operating system. Firmware is more efficient than software loaded from an alterable medium and more adaptable to change than pure hardware circuitry.

**flag** 1) To mark an information item for selection for further processing. (2) A character that signals the occurrence of some condition, such as the end of a word.

**folder** A container used to organize objects.

**fully qualified domain name (FQDN)** In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is mycomputer.city.company.com.

[\(Back to the top\)](#)

## G

**group** A collection of users who can share access authorities for protected resources.

[\(Back to the top\)](#)

## H

**handler** A function that is registered by the application programmer that the system or the application calls when certain events occur in the system or application.

**handshake** A Secure Sockets Layer (SSL) session always begins with an exchange of messages called the SSL handshake. The handshake allows the server to authenticate itself to the client by using public key techniques, and then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

**hashing** Link sequences are of length  $\log n$ . Hashing is a method which overcomes the  $\log n$  barrier. The idea is that the position of a key within the data structure is computed directly from the value of the key.

**header** System-defined control information that precedes user data.

**hierarchical** A way to organize data on computer systems using a hierarchy of containers, often called folders (directories) and files. In this scheme, folders may contain other folders and files. The successive containment of folders within folders creates the levels of organization, which is the hierarchy.

**host name** In the Internet suite of protocols, the name that is given to a machine. Sometimes, host name is used to mean fully qualified domain name (FQDN). Other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if rchland.vnet.ibm.com is the fully qualified domain name, either of the



following can be considered the host name: (a) rchland.vnet.ibm.com, or (b) rchland.

**Hypertext Transfer Protocol (HTTP)** In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

**Hypertext Transport Protocol Secure (HTTPS)** A TCP/IP protocol that is used by World Wide Web servers and Web browsers to transfer and display hypermedia documents securely across the Internet.

[\(Back to the top\)](#)

## I

**instance** In object-oriented programming, an object created by instantiating a class.

**invocation** The activation of a program or procedure.

[\(Back to the top\)](#)

## J

**Java** An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

**Java Development Kit (JDK)** A software package that can be used to write, compile, debug, and run Java applets and applications.

**Java Runtime Environment (JRE)** A subset of the Java Development Kit (JDK) that contains the core executables and files that constitute the standard Java platform. The JRE includes the Java Virtual Machine (JVM), core classes, and supporting files.

**Java Virtual Machine (JVM)** A software implementation of a central processing unit (CPU) that runs compiled Java code (applets and applications).

[\(Back to the top\)](#)

## K

**kernel** The part of an operating system that performs basic functions such as allocating hardware resources.

**key** In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data.

**key database** Exists as a file that the server uses to store one or more key pairs and certificates. You can use one key database for all your key pairs and certificates, or create multiple databases.

**key file** In the Distributed Computing Environment (DCE), a file that contains encryption keys for noninteractive principals.

**key pair** In computer security, a public key and a private key. When the key pair is used for encryption, the sender uses the public key to encrypt the message, and the recipient uses the private key to decrypt the message. When the key pair is used for signing, the signer uses the private key to encrypt a representation of the message, and the recipient uses the public key to decrypt the representation of the message for signature verification.

**key ring** In computer security, a file that contains public keys, private keys, trusted roots, and certificates.

[\(Back to the top\)](#)

## L

**Layered Service Provider (LSP)** A *service provider* is an installed protocol stack, not to be confused with a service, which is a server application. A *base protocol* is a protocol (such as TCP) capable of performing data communications with a remote endpoint. A *layered protocol* is a protocol that cannot stand alone; it relies on a base protocol for services. SSL is an example of a layered protocol. Layered protocols are only used through an interface for service providers. These are *layered service providers*.

**Lightweight Directory Access Protocol (LDAP)** In TCP/IP, a protocol that enables users to locate people, organizations, and other resources in an Internet directory or intranet directory.

**local area network (LAN)** (1) A computer network located on a user's premises within a communication across the LAN boundary may be subject to some form of regulation. (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

**long name** The expanded name of the presentation space or emulation session.

[\(Back to the top\)](#)

## M

**machine translation** A translation productivity tool that works by breaking down sentences or other text segments, analyzing them in context and then recreating their meaning in the target language. Machine translation works best on large volumes of well written texts from narrow subject areas.

**memory load control** A facility, added in AIX Version 3.2, that detects memory over-commitment and temporarily reduces the number of running processes, thus

avoiding thrashing.

**method** In object-oriented design or programming, the software that implements the behavior specified by an operation.

**microcode** A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable.

**mode** A method of operation in which the actions that are available to a user are determined by the state of the system.

**module** A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

[\(Back to the top\)](#)

## N

**name space** The scope within which a name provides the intended identification. Here is an analogy: given names are intended to uniquely identify members of a family; in this case, the name space is the family.

**native** Pertaining to the relationship between a transport user and a transport provider that are both based on the same transport protocol.

**net mask** A 32-bit mask used to identify the most local portion of a local area network (LAN)

**node** The smallest unit of valid, complete structure in an XML document. The nodes that include a tag set, along with any required attributes, attribute values, and content, constitute an element.

[\(Back to the top\)](#)

## O

**object** (1) In object-oriented design or programming, a concrete realization of a class that consists of data and the operations associated with that data.

(2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both.

**object class** A categorization or grouping of objects that share similar behaviors and circumstances.

**object identifier (OID)** An administratively assigned data value of the type defined in abstract syntax notation 1 (ASN.1).

**octet** A byte composed of eight binary elements.

**Open Systems Interconnection (OSI)** The interconnection of open systems in

accordance with standards of the International Organization for Standardization (ISO) for the exchange of information.

**operand** An entity on which an operation is performed.

[\(Back to the top\)](#)

## P

**parse** To break down a string of information such as a command or file into its constituent parts.

**path** A list of one or more directory names and an object name (such as the name of a file) that are separated by an operating system-specific character, such as the slash (/) in UNIX operating systems, the backslash (\) in Windows operating systems, and the semicolon (;) in OS/2 operating systems.

**Peripheral Component Interconnect (PCI)** A computer bussing architecture that defines electrical and physical standards for electronic interconnection.

**permissions** In the Distributed Computing Environment (DCE), the modes of access to a protected object. The number and meaning of permissions with respect to an object are defined by the access control list (ACL) manager of the object.

**pipeline** A serial arrangement of processors or a serial arrangement of registers within a processor. Each processor or register performs part of a task and passes results to the next processor; several parts of different tasks can be performed at the same time.

**PKCS12** Sometimes referred to as PFX files; PKCS#12 files are used by several programs including Netscape, MSIE and MS Outlook.

**plug-in** A self-contained software component that modifies (adds or changes) function in a particular software system. When a user adds a plug-in to a software system, the foundation of the original software system remains intact. The development of plug-ins requires well defined application programming interfaces (APIs).

**port** (1) A system or network access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. One or more ports are controlled by a single data link control (DLC) process. (4) In the Internet suite of protocols, a specific logical connector between the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) and a higher level protocol or application. (5) To modify a computer program to enable it to run on a different platform.

**port number** In the Internet suite of protocols, the identifier for a logical connector between an application entity and the transport service.

**presentation space** A conceptual two-dimensional surface in storage on which data for a portion of the display surface is represented.

**principal** In DCE Security, an entity that can communicate securely with another entity. In the Distributed computing Environment (DCE), principals are represented as entries in the Registry database and include users, servers, computers, and authentication surrogates.

**private key** In secure communication, an algorithmic pattern used to encrypt messages that only the corresponding public key can decrypt. The private key is also used to decrypt messages that were encrypted by the corresponding public key. The private key is kept on the user's system and is protected by a password.

**process ID** A unique number assigned to a process by the operating system. The number is used internally by processes to communicate.

**property** A characteristic or attribute that describes a unit of information.

**proxy server** A server that receives requests intended for another server and that acts on the behalf of the client behalf (as the client proxy) to obtain the requested service. A proxy server is often used when the client and the server are incompatible for direct connection. For example, the client is unable to meet the security authentication requirements of the server but should be permitted some services.

**public key** In secure communication, an algorithmic pattern used to decrypt messages that were encrypted by the corresponding private key. A public key is also used to encrypt messages that only the corresponding private key can decrypt. Users broadcast their public keys to everyone with whom they must exchange encrypted messages.

**public key infrastructure (PKI)** An infrastructure that supports digital signatures and other public key-enabled security services.

[\(Back to the top\)](#)

## R

**redirect** To divert data from a process to a file or device to which it would not normally go.

**relative path** A path that begins with the working directory.

**root certificate** In SET programs, the certificate at the top of the certificate chain hierarchy.

**root node** In a graphical representation of data as a tree, a node that has no

parents but typically has children.

[\(Back to the top\)](#)

## S

**scope** Specification of the boundary within which system resources can be used.

**Secure Sockets Layer (SSL)** A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corporation and RSA Data Security, Inc.

**Secure Hash Algorithm (SHA)** The current approved hash algorithm produces a message digest of 160 bits.

**server-side includes** A facility for including dynamic information in documents sent to clients, such as current date, the last modification date of the file, and the size or last modification of other files.

**shim** A thin, often tapered, piece of material, such as metal, used to fill in space between things for support, adjustment, or leveling.

**short name** In Personal Communications, the one-letter name (A through Z) of the presentation space or emulation session.

**Simple Mail Transfer Protocol (SMTP)** In the Internet suite of protocols, an application protocol for transferring mail among users of the Internet.

**small computer system interface (SCSI)** A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

**socket** A method of communication between two processes. A socket is an identifier that the application uses to uniquely identify an end point of communications. The user associates a protocol address with the socket by associating a socket address with the socket.

**stanza** A group of lines in a file that together have a common function or define a part of the system. Stanzas are usually separated by blank lines or colons, and each stanza has a name.

**stash file** A file that hides other data files within.

**string** A sequence of elements of the same nature, such as characters considered as a whole. For example, character string, binary string, and hexadecimal string.

**subdirective** Similar to directives, except that they do not have their own class. The directive is responsible for fetching and processing the subdirective arguments.

**subgroup** A subset of a group.

**subnet** An interconnected, but independent segment of a network that is identified by its Internet Protocol (IP) address.

**subtree** A section of a directory hierarchy, which is also called a directory tree. The subtree typically starts at a particular directory and includes all subdirectories and objects below that directory in the directory hierarchy; that is, any subdirectories or objects connected to the directory, or to any lower level of its subdirectories.

**symmetric keys** In computer security, the two keys in a key pair. The keys are called symmetric because each key holds as much of the encryption pattern as the other does.

**syntax** The rules for the construction of a command or statement.

[\(Back to the top\)](#)

## T

**target** In advanced program-to-program communications, the program or system to which a request for processing is sent.

**thrashing** A condition, caused by a high level of memory over-commitment, in which the system is spending all of its time writing out virtual memory pages and reading them back in. The application programs make no progress because their pages do not stay in memory long enough to be used. Memory load control is intended to avoid or stop thrashing.

**thread** A stream of computer instructions that is in control of a process. A multithread process begins with one stream of instructions (one thread) and can later create other instruction streams to perform tasks.

**timeout** (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

**token** A particular message or bit pattern that signifies permission to transmit.

**tree structure** A data structure that represents entities in nodes, with at most one parent node for each node, and with only one root node.

**trusted root** A certificate signed by a certificate authority (CA), designated as a trusted CA on your server.

[\(Back to the top\)](#)

## U

**utility** In programming, a program that performs a common service function.

[\(Back to the top\)](#)

## V

**variable** A name used to represent data that can be changed while the program or procedure is running.

**virtual host** Refers to the practice of maintaining more than one server on one machine, differentiated by their apparent host name.

[\(Back to the top\)](#)

## W

**wildcard** A special character such as an asterisk (\*) or a question mark (?) that you can use to represent one or more characters. Any character or set of characters can replace a pattern matching character.

[\(Back to the top\)](#)

## X

**X.500** The directory services standard of International Telecommunication Union (ITU), International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC).

**x509** The x509 command is a multipurpose certificate utility. It can be used to display certificate information, convert certificates to various forms, sign certificate requests, or edit certificate trust settings.

[\(Back to the top\)](#)





## Acronym List

ACL: access control list  
API: application programming interface  
APXS: APache eXtenSion  
ASCII: American National Standard Code for Information Interchange  
ASN.1: abstract syntax notation one  
BER: Basic encoding rules  
BHAPI: BSAFE Hardware API  
CA: certificate authority  
CGI: Common Gateway Interface  
CLF: Common Log Format  
CMS: conversational monitor system  
CRL: certificate revocation list  
DAP: Directory Access Protocol  
DCE: Distributed Computing Environment  
DES: Data Encryption Standard  
DFS: distributed File System  
DLC: data link control  
DLL: dynamic link library  
DN: distinguished name  
DNS: Domain Name System  
DSO: dynamic shared object  
ECLF: Extended Common Log Format  
FastCGI: Fast Common Gateway Interface Protocol  
FIPS: Federal Information Processing Standard  
FRCA: Fast Response Cache Accelerator  
HTTP: Hypertext Transfer Protocol  
HTTPS: Hypertext Transport Protocol Secure  
IEC: International Electrotechnical Commission  
IETF: Internet Engineering Task Force  
IP: Internet Protocol  
ISO: International Organization for Standardization  
ITU: International Telecommunication Union  
JDK: Java Development Kit  
JRE: Java Runtime Environment  
JVM: Java Virtual Machine  
LAN: local area network  
LDAP: Lightweight Directory Access Protocol


LSP: Layered Service Provider  
MAC: message authentication code  
MIME: Multipurpose Internet Mail Extensions  
NIST: National Institute of Standards and Technology  
NFS: Network File System  
NLS: National Language Support  
OID: Object ID  
OSI: Open Systems Interconnection  
PCI: Peripheral Component Interconnect  
PIN: personal identification number  
PKI: public key infrastructure  
PPP: Point-to-Point Protocol  
SCSI: Small Computer System Interface  
SHA: Secure Hash Algorithm  
SIDD: session ID cache daemon  
SMI: Structure of Management Information  
SMTP: Simple Mail Transfer Protocol  
SSI: server-side includes  
SSL: Secure Sockets Layer  
TCL: transmission control layer  
TCP: Transmission Control Protocol  
TCP/IP: Transmission Control Protocol/Internet Protocol  
UDP: User Datagram Protocol  
V-CLF: Common Log Format with virtual host information  
V-ECLF: Extended Common Log Format with virtual host information

[\(Back to the top\)](#)



## Starting and stopping on Windows operating systems

The server installs as a service, and runs automatically. When you reboot, the server starts itself. You can control the server using the standard Windows Service Control panels.

 If you use the developer installation option, then the IBM HTTP Server does not install as a service. You have to run the `apache.exe` file from a command line.

You can also use the **net start IBM HTTP Server 2.x** and **net stop IBM HTTP Server 2.x** commands, where x is the appropriate version of the IBM HTTP Server, such as 2.0.46. For more information about these commands, see the Windows help files. You can also access these commands from the Start menu. Find the default location by clicking **Start > Programs > IBM HTTP Server**.

### Finding related information

- [Running the Windows server from the command line](#)
- [Getting started with the IBM HTTP Server](#)
- [Locating glossary terms](#)



## Setting advanced security options

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](#).
3. [Define Secure Sockets Layer \(SSL\) for multiple-IP virtual hosts](#).
4. [Enable session ID caching](#).

### Finding related information

- [Associating your public key with certificate authorities](#)
- [Enabling certificate revocation list in Secure Sockets Layer](#)
- [Getting started quickly with secure connections](#)
- [Locating glossary terms](#)
- [Using cipher specifications](#)



## Enabling 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.

For each virtual host, choose the level and the type of client authentication.


- [Choosing the level of client authentication](#)
- [Choosing the type of client authentication protection](#)
- [Finding related information](#)

## Choosing 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.

<i>None</i>	The server requests no client certificate from the client.
<i>Optional</i>	The server requests, but does not require, a client certificate. If presented, the client certificate must prove valid.
<i>Required</i>	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

1. Specify one of the following directives in the configuration file, for each virtual host stanza:
  - **SSLClientAuthRequire:**  
Recommended. Refer to the description of [SSLClientAuthRequire](#).  
For example, `SSLClientAuthRequire CommonName=Richard`
  - **SSLFakeBasicAuth:**  
Not recommended. Refer to the description of [SSLFakeBasicAuth](#). If you specify `SSLFakeBasicAuth`, ensure that the `mod_ibm_ssl` module appears last in the module list.
2. Save the configuration file and restart the server.

### Finding related information

- [Authenticating clients](#)
- [Defining Secure Sockets Layer for multiple-IP virtual hosts](#)
- [Enabling a certificate revocation list in Secure Sockets Layer](#)
- [Enabling session ID caching](#)
- [Locating glossary terms](#)
- [Setting and viewing cipher specifications](#)

---

[\(Back to the top\)](#)



## Defining Secure Sockets Layer 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.

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.

### Finding related information

- [Enabling client authentication](#)
- [Enabling session ID caching](#)
- [Locating glossary terms](#)
- [Setting and viewing cipher specifications](#)



## Setting and viewing cipher specifications

This section describes setting and viewing cipher specifications for secure transactions. Links to related topics appear at the end of this section.

- [Specifying cipher specifications](#)
- [Viewing configured cipher specifications](#)
- [Finding related information](#)

For each virtual host, set the cipher specification to use during secure transactions. The specified cipher specifications validate against the level of the GSK toolkit 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.

### [Specifying cipher specifications](#)

1. Specify a value for each virtual host stanza in the configuration file, 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 an [SSL Version 2](#), or [SSL Version 3 cipher specification](#).

2. Save the configuration file and restart the server.

### [Viewing configured cipher specification](#)

To see which cipher specifications the server uses for secure transactions, look at the informational messages in the error log.

1. Specify to include informational messages in the error log by using the [LogLevel](#) directive in the configuration file:

```
LogLevel info
```

2. Look in the error log for messages in this format:

```
TimeStamp info_message mod_ibm_ssl: Using Version 2/3 Cipher: long name/short name.
```

The order that the cipher specifications appear in the error log from top to bottom represents the attempted order of the cipher specifications.

### **Finding related information**

- [Authenticating clients](#)
- [Defining SSL for multiple-IP virtual hosts](#)
- [Enabling client authentication](#)
- [Enabling session ID caching](#)
- [Locating glossary terms](#)
- [Using cipher specifications](#)


---

[\(Back to the top\)](#)





## Using cipher specifications

This section contains information regarding cipher specifications, including browser configuration, key sizes, valid cipher specifications and associated notes . Links to related information appear at the end of this section.

- [Configuring the browser](#)
- [Identifying cipher specifications and key sizes](#)
- [Listing valid cipher specifications](#)
  - [Secure Sockets Layer Version 2 Cipher Specifications](#)
    - [North American Edition](#)
  - [Secure Sockets Layer Version 3 and Transport Layer Security Version 1 Cipher Specifications](#)
    - [North American Edition](#)
    - [International Export Edition](#)
  - [FIPS Approved NIST SSLV3 and TLSV1 \(only available with SSLFIPSEnable\)](#)
- [Finding related information](#)

## Configuring the browser

The following directives require browser configuration:

- [SSLCipherBan](#)
- [SSLCipherRequire](#)
- [SSLCipherSpec](#)
- [SSLVersion](#)

## Identifying cipher specifications and key sizes

The Secure Sockets Layer (SSL) cipher specification indicates the data encryption algorithm and key size usage. SSL V3 includes the hashing algorithm. For example, cipher specification DES SHA (56 bit) uses the DES encryption algorithm, a 56-bit key size and the SHA hashing algorithm. For more detailed information on cipher specifications, go to [SSL V3.0 Specifications](#).

## Listing valid cipher specifications

The following section provides a listing of currently valid cipher specifications.

### ***SSL Version 2 cipher specifications***

- SSL\_DES\_192\_EDE3\_CBC\_WITH\_MD5
- SSL\_RC4\_128\_WITH\_MD5
- SSL\_RC2\_CBC\_128\_CBC\_WITH\_MD5
- SSL\_DES\_64\_CBC\_WITH\_MD5
- SSL\_RC4\_128\_EXPORT40\_WITH\_MD5
- SSL\_RC2\_CBC\_128\_CBC\_EXPORT40\_WITH\_MD5

---

### North American Edition (U.S. and Canada)

<u>Short name</u>	<u>Long name</u>	<u>Description</u>
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)


---

### ***Secure Sockets Layer Version 3 and Transport Layer Security Version 1 Cipher Specifications***

- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD4
- SSL\_RSA\_WITH\_RC4\_128\_MD5
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC2\_CBC\_40\_MD5
- SSL\_RSA\_WITH\_NULL\_SHA
- SSL\_RSA\_WITH\_NULL\_MD5
- SSL\_NULL\_WITH\_NULL\_NULL
- TLS\_RSA\_EXPORT1024\_WITH\_RC4\_56\_SHA
- TLS\_RSA\_EXPORT1024\_WITH\_DES\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA

---

### North American Edition (US and Canada)


<u>Short name</u>	<u>Long name</u>	<u>Description</u>
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)
 36	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RC2 MD5 (40 bit)

32	SSL_RSA_WITH_NULL_SHA	
31	SSL_RSA_WITH_NULL_MD5	
30	SSL_NULL_WITH_NULL_NULL	
62	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	
64	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	
2F	TLS_RSA_WITH_AES_128_CBC_SHA	AES SHA (128 bit)
35b	TLS_RSA_WITH_AES_256_CBC_SHA	AES SHA (128 bit)



Cipher specification 36 requires Netscape Navigator V4.07 or later.

### International Export Edition

<u>Short name</u>	<u>Long name</u>	<u>Description</u>
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)
 36	SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RC2 MD5 (40 bit)

32	SSL_RSA_WITH_NULL_SHA	
31	SSL_RSA_WITH_NULL_MD5	
30	SSL_NULL_WITH_NULL_NULL	
62	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	RC4 SHA Export1024 (56 bit)
64	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	DES SHA Export1024 (56 bit)



Cipher specification 36 requires Netscape Navigator V4.07 or later.


### ***FIPS Approved NIST SSLV3 and TLSV1 (only available with SSLFIPSEnable)***

- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_FIPS\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_FIPS\_WITH\_DES\_CBC\_SHA

### **North American Edition (US and Canada)**

<b><u>Short name</u></b>	<b><u>Long name</u></b>	<b><u>Description</u></b>
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 (128 bit)

---

 2F	TLS_RSA_WITH_AES_128_CBC_SHA	AES SHA (128 bit)
39	SSL_RSA_WITH_DES_CBC_SHA	DES SHA (56 bit)
FE	SSL_RSA_FIPS_WITH_DES_CBC_SHA	DES SHA (56 bit)

---



Not supported in versions available outside of North America.

### Finding related information

- [Locating glossary terms](#)
- [Setting advanced security options](#)
- [Setting and viewing cipher specifications](#)
- [Understanding Secure Sockets Layer environment variables](#)
- [Using Secure Sockets Layer directives](#)
- [Using the SSLCipherRequire directive](#)
- [Using the SSLCipherSpec directive](#)
- [Using the SSLVersion directive](#)

---

[\(Back to the top\)](#)



## Enabling a certificate revocation list in Secure Sockets Layer

This section provides information on identifying directives for certificate revocation list (CRL) and those supported in global servers and virtual hosts. Links to related topics appear at the end of this section.

- [Identifying directives needed to set up certificate revocation list](#)
- [Identifying directives supported in global server and virtual host](#)
- [Finding related information](#)

Certificate revocation provides the ability to revoke a client certificate given to the IHS 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 the 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 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 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.

## Finding related information

- [Associating your public key with certificate authorities](#)
- [Authenticating clients](#)
- [Enabling client authentication](#)
- [Locating glossary terms](#)
- [Understanding Secure Sockets Layer environment variables](#)
- [Using Secure Sockets Layer directives](#)
- [Using SSL Password Prompting](#)

---

[\(Back to the top\)](#)





## Understanding Secure Sockets Layer environment variables

This section provides information about the Secure Sockets Layer (SSL) environment variables. Links to related topics appear at the end of this section.

- [Looking at SSL handshake environment variables](#)
- [Looking at server certificate environment variables](#)
- [Looking at client certificate environment variables](#)
- [Finding related information](#)

SSL-specific environment variables get exposed to common gateway interface (CGI) applications and server-side includes (SSI) processed pages. You can categorize these variables into three types:

- Variables for information regarding the SSL handshake
- Variables for exposing the server certificate information
- Variables for exposing client certificate information, if you enable client authentication.

When making a valid SSL request, the SSL handshake environment variables and the server certificate environment variables are set. Setting client authentication to either *optional* or *require*, results in the client certificate environment variables setting.

### [Looking at SSL handshake environment variables](#)

A list of SSL handshake environment variables, with their descriptions and values follows:

- HTTPS
  - Description: Indicates an SSL connection.
  - Values: String contains either `ON`, for an SSL connection, or `OFF`, if not.
- HTTPS\_CIPHER
  - Description: Contains the cipher used in the SSL handshake.
  - Values: See the [list of valid cipher specifications](#).
- HTTPS\_KEYSIZE
  - Description: Indicates the size of the key.
  - Values: See the table below.
- HTTPS\_SECRETKEYSIZE
  - Description: Indicates the actual strength of the key.

- Values: See the table below.
- **SSL\_CIPHER**
  - Description: Acts as a duplicate of HTTPS\_CIPHER.
  - Values: See the [list of valid cipher specifications](#).
- **SSL\_PROTOCOL\_VERSION**
  - Description: Contains the protocol version.
  - Values: String contains either SSLV2, SSLV3, or TLSV1.

## Values for HTTPS\_KEYSIZE and HTTPS\_SECRETKEYSIZE

### For 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

### For 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

## [Looking at server certificate environment variables](#)

A list of server certificate environment variables with their associated descriptions and values follows:

- **SSL\_SERVER\_C**
  - Description: Contains the country attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_CN**
  - Description: Contains the common name attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_DN**
  - Description: Contains the distinguished name of the server certificate used in the IP-based virtual host which received the request
  - Values: String or empty
- **SSL\_SERVER\_EMAIL**
  - Description: Contains the e-mail attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_L**
  - Description: Contains the locality attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_O**
  - Description: Contains the organization attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_OU**
  - Description: Contains the organizational unit attribute of the server certificate
  - Values: String or empty
- **SSL\_SERVER\_ST**
  - Description: Contains the state or province attribute of the server certificate
  - Values: String or empty

## [Looking at client certificate environment variables](#)

A list of the client certificate environment variables, with their associated descriptions and values follows:

- **SSL\_CLIENT\_C**
  - Description: Contains the client certificate country
  - Values: String or empty
- **SSL\_CLIENT\_CERTBODY**

- Description: Contains the client certificate
  - Values: String containing the complete client certificate as a string
- **SSL\_CLIENT\_CERTBODYLEN**
  - Description: Contains the length of the client certificate
  - Values: Integer
- **SSL\_CLIENT\_CN**
  - Description: Contains the client certificate common name
  - Values: String or empty
- **SSL\_CLIENT\_DN**
  - Description: Contains the distinguished name from the client certificate
  - Values: String or empty
- **SSL\_CLIENT\_EMAIL**
  - Description: Contains the client certificate e-mail
  - Values: String or empty
- **SSL\_CLIENT\_IC**
  - Description: Contains the country name of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_ICN**
  - Description: Contains the common name of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_IDN**
  - Description: Contains the distinguished name of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_IEMAIL**
  - Description: Contains the e-mail address of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_IL**
  - Description: Contains the locality of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_IO**
  - Description: Contains the organization name of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_IOU**
  - Description: Contains the organizational unit name of the client certificate issuer

- Values: String or empty
- **SSL\_CLIENT\_IPC**
  - Description: Contains the postal code of the client certificate issuer
  - Values: String and empty
- **SSL\_CLIENT\_IST**
  - Description: Contains the state or province of the client certificate issuer
  - Values: String or empty
- **SSL\_CLIENT\_L**
  - Description: Contains the client certificate locality
  - Values: String or empty
- **SSL\_CLIENT\_NEWSESSIONID**
  - Description: Indicates whether this session ID is new
  - Values: String containing "TRUE" or "FALSE"
- **SSL\_CLIENT\_O**
  - Description: Contains the client certificate organization
  - Values: String or empty
- **SSL\_CLIENT\_OU**
  - Description: Contains the client certificate organizational unit
  - Values: String or empty
- **SSL\_CLIENT\_PC**
  - Description: Contains the client certificate postal code
  - Values: String and empty
- **SSL\_CLIENT\_SERIALNUM**
  - Description: Contains the client certificate serial number
  - Values: String or empty
- **SSL\_CLIENT\_SESSIONID**
  - Description: Contains the session ID
  - Values: String or empty
- **SSL\_CLIENT\_ST**
  - Description: Contains the client certificate state or province
  - Values: String or empty

### Finding related information

- [Listing valid cipher specifications](#)
- [Locating glossary terms](#)
- [Setting advanced security options](#)

[\(Back to the top\)](#)



## **Using the Key Management Utility**

This section provides information on planning, preparation and use of the Key Management Utility (IKEYMAN) utility. Links to related topics appear at the end of this section.

- [Planning to use the Key Management Utility](#)
- [Reviewing security configuration examples](#)
- [Setting your system environment](#)
- [Using the Key Management Utility graphical user interface](#)
- [Starting the Key Management Utility](#)
- [Using the Key Management Utility line interface](#)
- [Referencing user interface tasks](#)
- [Using the Key Management Utility to store keys on your PKCS11 device](#)
- [Using the Key Management Utility Command Line Interface on Linux for S/390 operating system](#)
- [Looking at the Key Management Utility command line syntax](#)
- [Reviewing Key Management Utility command line parameters](#)
- [Reviewing Key Management Utility command line options](#)
- [Using command line invocation](#)
- [Working with user properties file](#)
- [Finding related information](#)

### **Planning to use the Key Management Utility**

To have a secure network connection, create a key for secure network communications and receive a certificate from a certificate authority (CA), designated as a trusted CA on your server. 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.

Use IKEYMAN for configuration tasks related to public and private key creation and management. You cannot use IKEYMAN for configuration options that update the server configuration file, `httpd.conf`. For options that update the server configuration file, use the IBM Administration Server.



*Linux for S/390 users:* Use the IKEYCMD Command Line Interface to perform similar functions to IKEYMAN. See [Using the IKEYCMD Command Line Interface](#) for more detailed information regarding IKEYCMD.

### **Reviewing security configuration examples**

This section provides detailed information on tasks 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.

### **Setting your system environment**

The IKEYMAN GUI is Java-based and needs an IBM Developer Kit for the Java platform, or Java Runtime Environment (JRE) to run. Ensure you have Developer Kit level V1.4 or later for IKEYMAN support. Set your system environment using the following guidelines:

- Set the JAVA\_HOME variable to the location of the Java Developer Kit on the machine. If you are using WebSphere, set JAVA\_HOME to the to the JRE shipped with WebSphere:

```
EXPORT JAVA_HOME=the IBM Developer Kit for the Java platform home directory
full path name
```

For example, on Linux:

```
export JAVA_HOME=/opt/WebSphere/AppServer/java
```

- On Windows platforms: To bring up IKEYMAN, click Start > Programs > IBM HTTP Server 1.3.28 > Start Key Management Utility
- On UNIX platforms: Run `<IHS install root>/bin/ikeyman` to bring up the IKEYMAN GUI. You cannot invoke IKEYMAN from any directory because the IKEYMAN script that brings up the IKEYMAN GUI is no longer in the `/usr/bin` directory.

**Note:** If you are unable to bring up IKEYMAN, do the following:

1. If you are not using the JRE that comes with WebSphere, and the `gskikm.jar` file exists in your JAVA\_HOME directory, rename and move the `$JAVA_HOME/jre/lib/ext/gskikm.jar` file to a directory that is not visible to the JDK classpath, external directory, and bootclasspath. For example, on Linux: `mv $JAVA_HOME/jre/lib/ext/gskikm.jar to /gskfiles/gskikm.jar.org`.
2. Set JAVA\_HOME to the home of the Developer Kit for the Java platform located on your machine. If you are using WebSphere, JAVA\_HOME should be set to the JRE that is shipped with WebSphere. For example, on Linux: `export JAVA_HOME=/opt/WebSphere/AppServer/java`
3. Ensure that the `C:\Program Files\IBM\Java141\jre\lib\security\java.security` file has the following providers for GSKit:
 

```
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.spi.IBMCMSProvider
```

 If you plan to use cryptographic hardware for GSKit, add the following providers in this order:
 

```
security.provider.1=com.ibm.spi.IBMCMSProvider
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.crypto.pkcs11.provider.IBMPKCS11
```
4. If you are not using an IBM JDK or if the IBM JDK files are older than the GSKit files, copy all the jar files from `ibm\gsk7\classes\jre\lib\ext` to `Java141\jre\lib\ext`.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information regarding IKEYCMD.

To run IKEYMAN on the Linux for S/390 operating system, set up environment variables to use the IKEYCMD command line interface as follows:

1. Set your PATH to where your Java or JRE executable resides:

```
EXPORT PATH=/opt/IBMJava/bin:$PATH
```

2. Set the following CLASSPATH environment variable:

```
EXPORT CLASSPATH=/usr/local/ibm/gsk/classes/cfwk.zip:/usr/local/IBM/
gsk/classes/gsk7cls.jar:$CLASSPATH
```

Once completed, IKEYCMD should run from any directory. To run an IKEYCMD command, use the following syntax:

```
java com.ibm.gsk.ikeyman.ikeycmd <command>
```

You can substitute `jre` for `java`, depending on whether you use a JRE executable, or the IBM Developer Kit for the Java platform. **Example:**

```
jre com.ibm.gsk.ikeyman.ikeycmd <command>
```

Each IKEYCMD, except `create database`, requires that you specify the key database and password for the key database because the database opens with each command. See [Using the IKEYCMD command line interface](#), for more detailed information on IKEYCMD.

**Note:** If you are unable to open IKEYMAN, do the following:

1. Rename and move the `$JAVA_HOME/jre/lib/ext/gskikm.jar` to a directory that is not visible to the JDK classpath, `extdirs`, `bootclasspath`, e.g. for Linux: `mv $JAVA_HOME/jre/lib/ext/gskikm.jar to /gskfiles/gskikm.jar.org`
2. Set the JAVA\_HOME to the location of the Java Developer Kit on the machine. If you are using WebSphere, set JAVA\_HOME to the to the JRE shipped with WebSphere:

```
EXPORT JAVA_HOME=the IBM Developer Kit for the Java platform home directory
full path name
```

For example, for Linux:



```
export JAVA_HOME=/opt/WebSphere/AppServer/java
```

Ensure that the C:\Program Files\IBM\Java141\jre\lib\security\java.security file has the following providers for GSKit:

```
security.provider.2=com.ibm.crypto.provider.IBMJCE
```

```
security.provider.3=com.ibm.spi.IBMCMSProvider
```

If you plan to use cryptographic hardware for GSKit, add the following providers in this order:

```
security.provider.1=com.ibm.spi.IBMCMSProvider
```

```
security.provider.2=com.ibm.crypto.provider.IBMJCE
```

```
security.provider.3=com.ibm.jsse.IBMJSSEProvider
```

```
security.provider.4=com.ibm.crypto.pkcs11.provider.IBMPKCS11
```

3. If you are not using an IBM JDK or if the IBM JDK files are older than the GSKit files, copy all the jar files from `ibm\gsk7\classes\jre\lib\ext` to `Java141\jre\lib\ext`.

## [Using the Key Management Utility graphical user interface](#)

The following section describes how to get started and use IKEYMAN or the IKEYCMD command line interface.


### [Starting the Key Management Utility](#)

To start the IKEYMAN graphical user interface:

Type `<IHS root>/bin/ikeyman` on the command line, or change to the `<IHS root>/bin` directory and type `ikeyman` on the command line.

#### WINDOWS

Go to the start user interface and click **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.

### [Using the Key Management Utility command line interface](#)

To have a secure network connection, create a key for secure network communications and receive a certificate from a certificate authority (CA), designated as a trusted CA on your server. Use IKEYMAN, or IKEYCMD on Linux for S/390 operating systems, to create the key database file, public and private key pair and certificate request. After you receive the CA-signed certificate, use IKEYMAN, or IKEYCMD on Linux for S/390 operating systems, to receive the certificate into the key database where you created the original certificate request.

This section provides a quick reference of IKEYMAN and IKEYCMD tasks and common task descriptions.

### [Referencing user interface tasks](#)

A summarization of the IKEYMAN user interface and IKEYCMD command line interface tasks follow:

<b><u>IKEYMAN and IKEYCMD task</u></b>	<b><u>For instructions, go to:</u></b>
Create a new key database and specify the database password	<a href="#">"Creating a new key database"</a>
Create a new key pair and certificate request	<a href="#">"Creating a new key pair and certificate request"</a>
Create a self-signed certificate	<a href="#">"Creating a self-signed certificate"</a>
Export a key to another database or PKCS12 file	<a href="#">"Exporting keys"</a>

Import a key from another database or PKCS12 file	"Importing keys"
List certificate authorities (CAs) and certificate requests	"Listing CAs"
Open a key database	"Opening a key database"
Receive a CA-signed certificate into a key database	"Receiving a CA-signed certificate"
Show the default key in a key database	"Showing the default key in a key database"
Store the root certificate of a CA	"Storing a CA certificate"
Store the encrypted database password in a stash file	"Storing the encrypted database password in a stash file"

## Creating a new key database

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.

To create a new key database:

1. Start the IKEYMAN GUI. Refer to [Starting the Key Management Utility](#) for platform-specific instructions.
2. Click **key database file** from the main UI, then click **New**.
3. Enter your key database name in the New dialog box, or click **key.kdb** if you use the default. Click **OK**.
4. Enter your correct password in the Password Prompt dialog box, then enter to confirm the password. Click **OK**.

UNIX  
WINDOWS



*Linux for S/390 users:*

See [Using the IKEYCMD Command Line Interface](#) for more detailed information regarding IKEYCMD. Each key database operation requires a password. Even though a database of the type `sslight` requires a specified password, you can use a NULL string password, specified as `""`. To create a new key database using the IKEYCMD command line interface, enter the following command:

```
Java com.ibm.gsk.ikeyman.ikeycmd -keydb -create -db <file name>.kdb -pw <password>
-type cms -expire <days> -stash
```

where:

-type: Represents the type of key database. The IBM HTTP Server only handles a CMS key database.

-expire: Represents days before the password expires.

-stash: Indicates password stashing for the key database. Stashing the password is required for the IBM HTTP Server.

When you specify the `-stash` option during the key database creation, the password stashes in a file with a file name built as follows:

```
<file name of key database>.sth
```


For example, if the database created is named `keydb.kdb`, the stash file name is `keydb.sth`.

## Setting the database password

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 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 should contain at least six characters and contain at least two nonconsecutive numbers. Make sure 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 SSL password prompting. See [Using 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.

## [Changing the database password](#)

To change the database password:

1. Start the IKEYMAN GUI. 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 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**.



*Linux for S/390 users:* See [Using the IKEYCMD Command Line Interface](#), for more detailed information on IKEYCMD.

To change the database password, type:

```
Java com.ibm.gsk.ikeyman.ikeycmd -keydb -changePW dB <file name> .kdb -pw <password> -
new_pw
<new_password> -expire <days> -stash
```

where:

-new\_pw: Represents the new key database password. This password must differ from the old password.

-expire: Represents the number of days before the password expires.

-stash: Indicates password stashing for the key database. Stashing the password is required for the IBM HTTP Server.

## [Registering a key database with the server](#)

The initial configuration setting for the default key database name is `key.kdb`. If you use `key.kdb` as your default key database name, you do not need to register the database with the server. The server uses the initial setting on the KeyFile directive in the configuration file. If you do not use `key.kdb` as your default key database name, or, if you create additional key databases, you must register those databases.

UNIX

WINDOWS

## [Creating a new key pair and certificate request](#)

You find key pairs and certificate requests stored in a key database. To create a public and private key pair and certificate request:

1. If you have not created the key database, see [Creating a new key database](#) for instructions.
2. Start the IKEYMAN GUI. Refer to [Starting the Key Management Utility](#) for platform-specific instructions.
3. Click **Key Database File**, from the main UI, 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 UI, then click **New Certificate Request**.
7. In the New Key and Certificate Request dialog box, enter the following:
  - Key Label: Enter a descriptive comment to identify the key and certificate in the database.
  - Key size
  - Organization Name
  - Organization Unit (optional)
  - Locality (optional)
  - State/Province (optional)
  - Zip code (optional)
  - Country: Enter a country code. Specify at least two characters. **Example:** US

- o 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 appears.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information about IKEYCMD.

To create a public and private key pair and certificate request:

1. Enter the following command:

```
Java com.ibm.gsk.ikeyman.ikeycmd -certreq -create dB <dB_name>.kdb -pw
<password> -size <1024 | 512> -dn<distinguished_name>
-file <file name> -label <label>
```

where:

-size: Represents a key size of 512, or 1024.

-label: Represents a label attached to a certificate or a certificate request.

-dn: Represents 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.

**Example:**

```
"CN=weblinux.raleigh.ibm.com,O=IBM,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
```

where:

-file: Represents the name of the file where the certificate request stores.

2. Verify that the certificate successfully created.
  1. View the contents of the certificate request file you created.
  2. Make sure the key database recorded the certificate request:

```
Java com.ibm.gsk.ikeyman.ikeycmd -certreq -list dB <file name>
-pw <password>
```

You should see the label listed that you just created.

3. Send the newly created file to a certificate authority.

## Creating a self-signed certificate

It usually takes two to three weeks to get a certificate from a well known CA. While waiting for an issued certificate, 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.

UNIX

**WINDOWS** To create a self-signed certificate:

1. See [Creating a new key database](#) for instructions, if you have not created the key database.
2. Start the IKEYMAN GUI. Refer to [Starting the Key Management Utility](#) for platform-specific instructions.
3. Click **Key Database File** from the main UI, 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. Enter your password in the Password Prompt dialog box, 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:
  - o Key Label: Enter a descriptive comment used to identify the key and certificate in the database.
  - o Key Size
  - o Common Name: Enter the fully qualified host name of the Web server as the common name. **Example:** www.myserver.com.
  - o Organization Name
  - o Organization Unit (Optional)
  - o Locality (Optional)
  - o State/Province (Optional)
  - o Zip code (Optional)
  - o Country: Enter a country code. Specify at least two characters. **Example:**US
  - o Validity Period

8. Click **OK**.

*Linux for S/390 users:* See [Using the Key Management Utility command line interface](#) for more detailed information about IKEYCMD.

To create a self-signed certificate:

Enter the following command:

```
Java com.ibm.gsk.ikeyman.ikeycmd -cert -create dB <dB_name>.kdb -pw <password>
-size <1024 | 512> -dn<distinguished name> -label <label> -default_cert
<yes or no>
```

where:

-size: Indicates a key size of 512, or 1024

-label: Indicates a descriptive comment used to identify the key and certificate in the database.

-dn: 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

**Example:**

```
"CN=weblinux.raleigh.ibm.com,O=IBM,OU=IBM HTTP Server,L=RTP,ST=NC,C=US"
```

-default\_cert: Enter *yes*, if you want this certificate as the default certificate in the key database. Enter *no*, if not.

UNIX

WINDOWS

## Exporting keys

To export keys to another key database:

1. Start the IKEYMAN GUI. 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 using the default. Click **OK**.
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:
  - o Click **Export Key**.
  - o Click the target database type.
  - o Enter the file name, or use the Browse option.
  - o Enter the correct location.
7. Click **OK**.
8. Click **OK** in the Password Prompt dialog box, to export the selected key to another key database.

To export keys to a PKCS12 file:

1. Enter *ikeyman* on a command line on the UNIX platform, 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:
  - o Click **Export KeyM**.
  - o Click the PKCS12 database file type.
  - o Enter the file name, or use the Browse option.
  - o 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.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information about IKEYCMD.

UNIX

WINDOWS

## Importing keys

To import keys from another key database:

1. Start the IKEYMAN GUI. 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 the **key.kdb** file, if using 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:
  - o Click the **Import Key**.
  - o Click the key database file type.
  - o Enter the file name, or use the Browse option.
  - o Select the correct location.
7. Click **OK**.
8. Enter the correct password in the Password Prompt dialog box, and click **OK**.
9. Click the correct label name in the Select from Key Label list, and click **OK**.

UNIX

**WINDOWS** To import keys from a PKCS12 file:

1. Start the IKEYMAN GUI. 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 the **key.kdb** file, if using the default. Click **OK**.
4. Enter your password and click **OK**.
5. Click **Personal Certificates** in the Key Database content frame, then click the **Export/Import** button on the label.
6. In the Export/Import Key window:
  - o Click **Import Key**.
  - o Click **PKCS12**.
  - o Enter the file name, or use the Browse option.
  - o Select the correct location.
7. Click **OK**.
8. In the Password Prompt dialog box, enter the correct password, then click **OK**.



*Linux for S/390 users:* See [Using the IKEYCMD Command Line Interface](#) for more detailed information about IKEYCMD.

UNIX

## **WINDOWS** [Listing certificate authorities](#)

To display a list of trusted certificate authorities (CAs) in a key database:

1. Start the IKEYMAN GUI. 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 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 **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.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information about IKEYCMD.

To display a list of trusted CAs in a key database:

```
Java com.ibm.gsk.ikeyman.ikeycmd -cert -list CA dB <dbname>.kdb -pw <password>
-type CMS
```

UNIX

**WINDOWS**

## [Opening a key database](#)

To open an existing key database:

1. Start the IKEYMAN GUI. Refer to [Starting the Key Management Utility](#) for platform-specific instructions.
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 appears in the File Name text box.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information about IKEYCMD.


No explicit opening of a key database occurs. For each command, specify database and password options. These specifications provide the information needed to operate in a key database.

## [Receiving a signed certificate from a certificate authority](#)

Use this procedure to receive an electronically mailed certificate from a certificate authority (CA), designated as a trusted CA on your server. 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 Premium 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 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 3 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 [Storing a CA certificate](#) to receive intermediate CA certificates.

 If the CA 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 who is not a trusted CA. For instructions, see [Storing a CA certificate](#).

UNIX

**WINDOWS** To receive the CA-signed certificate into a key database:

1. Start the IKEYMAN GUI. 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 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**.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information about IKEYCMD.

To receive the CA-signed certificate into a key database, enter the following command:

```
Java com.ibm.gsk.ikeyman.ikeycmd -cert -receive -file <file name> dB <dB_name>
.kdb -pw <password> -format <ascii | binary> -default_cert <yes | no>
```

where:

- format: Represents where a certificate authority can provide a CA certificate, in either ASCII or binary format
- label: Represents the label attached to the CA certificate.
- trust: Indicates whether you can trust this CA. Use enable options when receiving a CA certificate.
- file: Indicates file containing the CA certificate.

UNIX  
WINDOWS

## Showing the default key in a key database

To display the default key entry:

1. Start the IKEYMAN GUI. 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 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.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information on IKEYCMD.

To display the default key entry:

```
Java com.ibm.gsk.ikeyman.ikeycmd -cert -getdefault dB <dbname>.kdb -pw <password>
```

UNIX  
WINDOWS

## Storing a certificate authority certificate

To store a certificate from a CA who is not a trusted CA:

1. Start the IKEYMAN GUI. 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 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 **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**.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information on IKEYCMD.

To store a certificate from a CA who is not a trusted CA:

```
Java com.ibm.gsk.ikeyman.ikeycmd -cert -add dB <file name>.kdb -pw <password>
-label <label> -format <ASCII | binary> -trust <enable | disable> -file
<file>
```

where:

- label: Represents the label attached to the certificate or the certificate request
- format: Indicates that the certificate authorities can supply a binary ASCII file
- trust: Indicates whether you can trust this CA. This value should be Yes.

UNIX  
WINDOWS

## Storing the encrypted database password in a stash file



For a secure network connection, store the encrypted database password in a stash file.

To store the password while a database creates:

1. Start the IKEYMAN GUI. 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 New dialog box, or click the **key.kdb** file, if using the default. Click **OK**.
4. Enter your correct password in the Password Prompt dialog box, then enter 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.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information on IKEYCMD.

To store the password while creating a database:

```
Java com.ibm.gsk.ikeyman.ikeycmd -keydb -create dB <path_to_dB>/<dB_name>.kdb
-pw <password> -type CMS -expire <days> -stash
```

UNIX

**WINDOWS** To store the password after creating a database:

1. Start the IKEYMAN GUI. 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 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.



*Linux for S/390 users:* See [Using the IKEYCMD command line interface](#) for more detailed information on IKEYCMD.

To store the password after creating a database:

```
Java com.ibm.gsk.ikeyman.ikeycmd -keydb -stashpw dB <dB_name>.kdb -pw <password>
```



## [On Linux for S/390: Using the Key Management Utility command line interface](#)

On the Linux for S/390 operating system, **IKEYCMD**, the Java command line interface to IKEYMAN, provides the necessary options to create and manage keys, certificates and certificate requests. If you act as your own CA, you can use IKEYCMD 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.

Use IKEYCMD for configuration tasks related to public and private key creation and management. You cannot use IKEYCMD for configuration options that update the server configuration file, `httpd.conf`. For options that update the server configuration file, use the IBM Administration Server.

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

## [Looking at the Key Management Utility command line syntax](#)

The syntax of the Java command line interface follows:

```
Java [-Dikeycmd.properties=<properties_file>] com.ibm.gsk.ikeyman.ikeycmd
<object> <action> [options]
```

where:

<b>-Dikeycmd.properties</b>	Specifies the name of an optional properties file to use for this Java invocation. A default properties file, <code>ikeycmd.properties</code> , 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 IKEYCMD invocations
-version	Displays version information for IKEYCMD

*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.

### [Reviewing Key Management Utility command line parameters](#)

The following table describes each *action* possible on a specified *object*.

<u>Object</u>	<u>Actions</u>	<u>Description</u>
-keydb	-changepw	Change the password for a key database
	-convert	Convert the 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
	details	List the detailed information for a specific 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 that 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
-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 IKEYCMD command
-version	Display IKEYCMD version information

### **Reviewing Key Management Utility command line options**


The following table shows each option that can exist on the command line. The options are listed as a complete group. However, their use depends on the *object* and *action* specified on the command line.

<b><u>Option</u></b>	<b><u>Description</u></b>
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): <pre>"CN=Jane Doe,O=IBM,OU=Java Development,L=Endicott,ST=NY,ZIP=13760,C=country"</pre>
-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 <i>object</i> ).
-format	Format of a certificate (either <i>ASCII</i> for Base64_encoded ASCII or <i>binary</i> 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 <a href="#">Creating a new key database</a> .
-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 <i>-target</i> specifies a key database. See <a href="#">Creating a new key database</a> .
-target_type	Type of database specified by <i>-target</i> operand (see <i>-type</i> ).
-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.

## Using command line invocation

A list of each command line invocation, with the optional parameters specified in *italics* follows.

 For simplicity, the actual Java invocation, `Java com.ibm.gsk.ikeyman.iKeycmd`, is omitted from each of the command invocations.

```
-keydb -changepw dB <file name> -pw <password> -new_pw <new_password> -stash
      -expire <days>
-keydb -convert dB <file name> -pw <password> -old_format <CMS | webdb>
      -new_format <CMS>
-keydb -create dB <file name> -pw <password> -type <CMS | sslight> -expire
      <days> -stash
-keydb -delete dB <file name> -pw <password>
-keydb -stashpw dB <file name> -pw <password>

-cert -add dB <file name> -pw <password> -label <label> -file <file name> -format
      <ASCII | binary> -trust <enable | disable>
-cert -create dB <file name> -pw <password> -label <label> -dn <distinguished_name>
      -size <1024 | 512> -x509version <3 | 1 | 2> -default_cert <no | yes>
-cert -delete dB <file name> -pw <password> -label <label>
-cert -details dB <file name> -pw <password> -label <label>
-cert -export dB <file name> -pw <password> -label <label> -type <CMS | sslight>
      -target <file name> -target_pw <password> -target_type <CMS | sslight |
pkcs12>
      -encryption <strong | weak>
-cert -extract dB <file name> -pw <password> -label <label> -target <file name>
      -format <ASCII | binary>
-cert -getdefault dB <file name> -pw <password>
-cert -import dB <file name> -pw <password> -label <label> -type <CMS | sslight>
      -target <file name> -target_pw <password> -target_type <CMS | sslight>
-cert -import -file <file name> -type <pkcs12> -target <file name> -target_pw
<password>
      -target_type <CMS | sslight>
-cert -list <all | personal | CA | site> dB <file name> -pw <password> -type
      <CMS | sslight>
-cert -modify dB <file name> -pw <password> -label <label> -trust <enable | disable>
-cert -receive -file <file name> dB <file name> -pw <password> -format <ASCII |
binary>
      -default_cert <no | yes>
-cert -setdefault dB <file name> -pw <password> -label <label>
-cert -sign -file <file name> dB <file name> -pw <password> -label <label> -target
<file name>
      -format <ASCII | binary> -expire <days>

-certreq -create dB <file name> -pw <password> -label <label> -dn
<distinguished_name>
      -size <1024 | 512> -file <file name>
-certreq -delete dB <file name> -pw <password> -label <label>
-certreq -details dB <file name> -pw <password> -label <label>
-certreq -extract dB <file name> -pw <password> -label <label> -target <file name>
-certreq -list dB <file name> -pw <password>
-certreq -recreate dB <file name> -pw <password> -label <label> -target <file name>

-help
-version
```

## [Working with user properties file](#)

To eliminate some of the typing on the Java command line interface invocations, specify user properties in a properties file. Specify the properties file on the Java command line invocation through the `-Dikeycmd.properties` Java option. A sample properties file, `ikeycmd.properties`, is supplied as a sample to enable Java applications to modify default settings for their application.

### Finding related information

- [Enabling cryptographic hardware for the Secure Sockets Layer](#)
- [Locating glossary terms](#)
- [Using the Secure Sockets Layer protocol for secure communications](#)

---

[\(Back to the top\)](#)




## Enabling cryptographic hardware for the Secure Sockets Layer




This section provides information on enabling cryptographic hardware for the Secure Sockets Layer (SSL). Links to related topics appear at the end of this section.

- [Getting started](#)
- [Initializing IBM cryptographic hardware \(IBM 4758 and IBM e-business Cryptographic Accelerator\) on the AIX operating system](#)
- [Initializing IBM tokens on Windows operating systems](#)
- [Using IKEYMAN to create keys for a PKCS11 device](#)
- [Configuring the IBM HTTP Server to use nCipher and Rainbow accelerator devices](#)
- [Configuring the IBM HTTP Server to use PKCS11 devices](#)
- [Finding related information](#)

Managing cryptographic keys and storing them on cryptographic hardware provides a highly secure architecture for secure online transactions. This capability greatly increases performance and security in a Web server using SSL.

 For this example, *key storage devices* are defined as those devices where the key is only visible on the hardware device.


The following cryptographic devices 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, please check with the hardware vendors for specific applications of these devices.

Device	Key Storage?	Acceleration Support?	Notes 
Rainbow Cryptoswift PCI with BSAFE Interface Model	No	Yes	Use with SSLAcceleratorDisable directive only. Supported on HP, Solaris, and the Windows operating systems.

nCipher nFast Accelerator with BHAPI plug-in under BSAFE 4.0	No	Pure accelerator	Requires either a SCSI or PCI-based nForce unit; use with SSLAcceleratorDisable directive only. Supported on Solaris and Windows operating systems.
nCipher nForce Accelerator, <i>accelerator mode</i>	No	Yes	Uses the BHAPI and BSAFE interface. Supported on Solaris and Windows operating systems.
nCipher nForce Accelerator, <i>Key stored accelerator mode</i>	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 with PKCS#11 Interface	Yes	No	Uses the PKCS11 interface. Supported on AIX and Windows operating systems.



Support for the following adapters has been tested with WebSphere Application Server Version 4.0.2 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.

## Getting started

The IBM 4758 requires the PKCS11 support software for the host machine and internal firmware. You also need the manual which explains software installation and card coprocessor microcode loading. The support software and manual do not come with the IBM 4758 card, but you can download them from the following Web site: <http://www-3.ibm.com/security/cryptocards/index.shtml>. From the download site, obtain the PKCS#11 Model 002/023 software and the PKCS#11 Installation manual.

After installing the support software on your machine and loading the microcode on the IBM 4758, initialize the card.

Configure the 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. For the IBM hardware cryptographic devices - IBM 4758 card, available on AIX and Windows operating systems, and IBM e-business Cryptographic Accelerator, the PKCS11 module ships with the `bos.pkcs11` package on AIX.

Install the `devices.pci.14109f00` device for the IBM 4758 and the `devices.pci.1410e601` device for the IBM e-business Cryptographic Accelerator. AIX V4.3.3 maintenance level09 is recommended when using 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/html/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 - `/opt/nfast/toolkits/pkcs11/libcknfast.so`
- HP-UX - `/opt/nfast/toolkits/pkcs11/libcknfast.sl`
- Linux - `/opt/nfast/toolkits/pkcs11/libcknfast.so`
- SUN - `/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`

## [Initializing IBM cryptographic hardware \(IBM 4758 and IBM e-business Cryptographic Accelerator\) on the AIX operating system](#)

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: [Download AIX fixes](#). For Version 4.3, click **AIX 4.3 OS, Java, compilers > Download selective fixes**. For Version 5.1, click **AIX 5.1 OS, Java, compilers > Download selective fixes**. This 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.

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. To initialize your token:

1. Select **Initialize your token**
2. Set a security officer and User PIN, if not already set
3. Initialize your user PIN. See [Chapter 5: Token Initialization from the PKCS11 manual](#) for more detailed information.

## WINDOWS [Initializing IBM tokens on Windows operating systems](#)

To initialize the IBM 4758 card on Windows operating systems, obtain the PKCS11 software for these operating systems from <http://www-3.ibm.com/security/cryptocards/html/ordersoftware.shtml>.

You can use the `TOKUTIL.EXE` utility that installs with the PKCS11 software to initialize your card on Windows operating systems.

Refer to [Chapter 5: Token Initialization from the PKCS11](#) for more details.


⚠ Make sure you have the `cryptoki.dll` module in your path.

## [Using IKEYMAN to store keys on a PKCS11 device](#)

To create keys for your PKCS11 device, provide an `ikmuser.properties` file for IKEYMAN. To provide this file:



1. Copy the `ikmuser.sample` file that ships with the IBM HTTP Server and GSKit, typically installed in the following directories:
  - AIX = `/usr/opt/ibm/gskta/classes`
  - HP = `/opt/ibm/gsk7/classes`
  - Linux = `/usr/local/ibm/gsk7/classes`
  - Solaris = `/opt/ibm/gsk7/classes`
  - Windows = `C:\Program Files\ibm\gsk7\classes`

to a file called `ikmuser.properties` in the `classes` directory. 

Cryptographic token may not work if the `ikmuser.properties` file does not reside in the `classes` directory.

2. Edit the `ikmuser.properties` file to set the `DEFAULT_CRYPTOGRAPHIC_MODULE` property to the name of the module managing your PKCS11 device. For example:

```
DEFAULT_CRYPTOGRAPHIC_MODULE=C:\pkcs11\bin\NT\cryptoki.dll
```

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

The module is normally installed on your system when you install the software for your PKCS11 device.

3. Save the `ikmuser.properties` file

As long as you have the `ikmuser.properties` file located in the `classes` directory, the device reads the `ikmuser.properties` file contents, whenever you bring up IKEYMAN.


**Note:** The cryptographic token is no longer be a separate item on the IKEYMAN GUI menu. It is treated as one of the keystore type. You can specify the PKCS11 module name by specifying the property of `DEFAULT_CRYPTOGRAPHIC_MODULE` in the `ikmuser.properties` file as before. However, IKEYMAN will no longer try to load the DLL/LIB at startup time to decide whether to support the cryptographic token. The value of `DEFAULT_CRYPTOGRAPHIC_MODULE` will be used only for the default value shown on the GUI.

When you open the Cryptographic Token, IKEYMANi will first retrieve the value of `DEFAULT_CRYPTOGRAPHIC_MODULE` in the `ikmuser.properties` file and pre-fill the value in the "File Name" and "Location" fields in the **Key Database File -> Open** dialog box of Ikeyman GUI. You can modify the value in the **File Name** and **Location** fields or press the **Browse** button. If the specified DLL/LIB cannot be loaded, an error message will be displayed as follows:

Once the specified DLL/LIB is loaded successfully, you will be able to use IKEYMAN. After opening a cryptographic token successfully, IKEYMAN will display the certificates stored in the cryptographic token.

When IKEYMAN comes up, the IBM Key Management window has an additional menu item called *cryptographic token*.


1. Click **Cryptographic Token** from your IBM Key Management window.
2. Click **Open**.
3. The Open Cryptographic Token window displays. Select your cryptographic token label and enter the user pin and password you specified when initializing your token with the configuration utility.
4. If you want to open an existing secondary key database, select **Open an existing secondary key database file**, specify a file name, and location. If not, disable this function by removing the check mark. If you want to create a new secondary key database file, select **Create new secondary key database file**, specify a conversational monitor system (CMS) key database file, the file name, and location. If not, make sure to clear the check mark by this option. Click **OK**.
5. Proceed with the steps as if you had opened a key database. You can continue with the same steps to create a self-signed certificate, or add a new digital-signed certificate. Instead of using **Key Database > Open**, use **Cryptographic Token > Open**.

 With the IBM HTTP Server, you must specify a key file to perform encryption. If you use PKCS11 devices, this key file should hold your signer certificates for your personal certificate, created using PKCS11 device.

## [Configuring the IBM HTTP Server to use nCipher and Rainbow accelerator devices](#)

The IBM HTTP Server enables nCipher and Rainbow accelerator devices by default. To disable your accelerator device, add the following directive to your configuration file:  
SSLAcceleratorDisable

## [Configuring the IBM HTTP Server to use PKCS11 devices](#)

 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`.

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 updates
  - `file`: Represents a fully qualified name of the file to create or update
  - `function`: Represents 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>`

## Finding related information

- [Choosing protection options](#)
- [Enabling a certificate revocation list in the Secure Sockets Layer](#)
- [Locating glossary terms](#)
- [Troubleshooting](#)
- [Using Secure Sockets Layer directives](#)
- [Using SSL Password Prompting](#)

---

[\(Back to the top\)](#)



## Choosing protection options

Choose from the following options to set up protection for server resources:

- [Without protection](#)
- [With password protection](#)
- [With secure SSL connections](#)
- [Finding related information](#)

UNIX

### Without protection

Use the default if you do not want protection. Without protection, any client can access any file on your server machine.

If you want to limit access to certain files, consider using password protection.

### With password protection

If you want to require a valid password before displaying a particular file, or directory of files, set up password protection. To set password protection see the documentation in the Apache User's Guide for the [mod\\_auth module](#).

For example, if your server serves stock quotes, you want to verify that the client requesting the quote has paid for the quotation service. Password protection limits access to certain files.

### With secure SSL connections

Implementing SSL requires additional steps. If you want data to travel over the Internet in encrypted form, [set up secure connections](#).

For example, your server provides a catalog for a mail order business and buyers submit credit card numbers over the Internet. You want to set up a secure connection for the credit card information.

### **Finding related information**

- [Getting started quickly with secure connections](#)
- [Locating glossary terms](#)
- [Starting and stopping the IBM HTTP Server on UNIX systems](#)

[\(Back to the top\)](#)



## Troubleshooting

This section provides information to help you identify problems, gather or look at information regarding these problems, and get additional help if you need it. Links to related topics appear at the end of this section.

- [Knowing what to do first](#)
- [Experiencing an IBM HTTP Server Service logon failure on Windows operating systems](#)
- [Identifying symptoms of poor server response time](#)
- [Identifying error messages](#)
  - [Initialization messages](#)
  - [Handshake messages](#)
  - [Configuration messages](#)
  - [I/O messages](#)
  - [Cache messages](#)
  - [Secure Sockets Layer stash utility errors](#)
- [Viewing Error Messages from a Target Server Start](#)
- [Identifying GSKit certificate support limitations](#)
- [Looking at known problems with hardware cryptographic support](#)
- [Looking at known problems on the HP platform](#)
- [Looking at known problems on the Solaris operating system](#)
- [Looking at known problems on the Linux for PowerPC operating system](#)
- [Looking at known problems on the UNIX platform](#)
- [Looking at known problems on Windows operating systems](#)
- [Configuring security on Internet Explorer V5.01x](#)
- [Contacting Customer Service and Support](#)

### Knowing what to do first

Perform the following steps:



- Ensure you run the IBM Developer Kit, Java Edition V1.4, or the Java Runtime Environment (JRE) V1.4 on AIX, Linux, HP, Solaris and Windows operating systems. You must install and run the IBM Developer Kit, Java edition, or the JRE yourself.
- 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:

- On AIX: /usr/IBMIHS/logs/error\_log
- On HP: /opt/IBMIHS/logs/error\_log
- On Linux: /opt/IBMIHS/logs/error\_log
- On Solaris: /opt/IBMIHS/logs/error\_log
- On Windows: <server\_root>\logs\error.log

## 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. 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.

1. Click **Start > Settings > Control Panel**.
2. Open **Administrative Tools**.
3. Open **Services**. The local user you select is created in Local Users and Groups, under Computer Management.
4. Click **Service > Actions > Properties**.
5. Choose the **Log on** tab.
6. Select this account option and click **Browse**, to select the user to associate with the service.

WINDOWS

WINDOWS

## Identifying 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. This condition will be logged in a single error message in the error log:

- The message on AIX, Linux, Solaris, or HP-UX is: "Server reached MaxClients setting, consider raising the MaxClients setting"
- The message on Windows platforms is: "Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting"

If you configured Apache to listen on multiple ports, you can find responses slow on one port (port 80, for example) but adequate on another port (port 443, for example). The disparity in response time results from each port having its own listen queue. You may have a deep port 80 queue and a shallow port 443 queue. Apache does not attempt to balance the number of connections received from each queue. When a connection becomes available, Apache and the operating system consume from either queue at random.

To address this scenario, add more Apache threads to handle inbound connections. This is accomplished by increasing the number of processes or increasing the number of threads per process.

## Identifying error messages

SSL error messages are in the format: SSLnnnnX, where:

- nnnn: Equals a four digit error number
  - 0100-0199: Indicates an initialization message
  - 0200-0299: Indicates a handshake message
  - 0300-0399: Indicates configuration messages
  - 0400-0499: Indicates I/O messages
  - 0600-0699: Indicates caching messages
  - 0700-0799: Indicates SSL Stash utility messages
- 
- X: Message level
  - I: Informational
  - E: Error

- W: Warning
- S: Severe

## Initialization messages

The following messages appear due to initialization problems:

- Message: **SSL0100S: GSK could not initialize, <errorCode>**.
  - Reason: Initialization failed when the SSL library returned an unknown error.
  - Solution: None. Report this problem to Service.
- Message: **SSL0101S: 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: An unknown error has occurred in the SSL library.
  - Solution: Report the problem to Service.
- **Message:SSL0118E: Initialization error, Open failed due to cipher error.**
  - Reason: An unknown error has occurred in the SSL library.
  - Solution: Report the problem to Service.
- **Message:SSL0119E: Initialization error, 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:SSL0120E: Initialization error, Key file has an invalid internal format.**
  - Reason: Key file has an invalid format.
  - Solution: Recreate key file.
- **Message:SSL0121E: Initialization error, 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:SSL0122E: Initialization error, 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:SSL0123E: Initialization error, 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:SSL0124E: Initialization error, The default key in the key file has an expired certificate. Use IKEYMAN to remove certificates that are expired.**
  - Reason: The certificate has expired.
  - Solution: Use IKEYMAN to select another certificate as the default.
- **Message:SSL0125E: Initialization error, There was an error loading one of the GSKdynamic link libraries.**
  - Reason: An open of the SSL environment resulted in an error because one of the GSKdynamic link libraries could not be loaded.
  - Solution: Contact support to make sure the GSKit is installed correctly.
- **Message: SSL0126E: Initialization error, Invalid date.**
  - Reason: The system date was set to an invalid date.
  - Solution: Change the system date to a valid date.
- **Message:SSL0127E: Initialization error, No ciphers specified.**
  - Reason: SSLV2 and SSLV3 are disabled.
  - Solution: None. Report this problem to Service.
- **Message:SSL0128E: Initialization error, No certificate.**
  - 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 a certificate.
- **Message:SSL0129E: Initialization error, The received certificate was formatted incorrectly.**
  - Reason: The client did not specify a valid certificate.
  - Solution: Client problem.
- **Message:SSL0130E: Initialization error, 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:SSL0131I: Initialization error, 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:SSL0132E: Initialization error, 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:SSL0133W: Initialization error, An incorrectly formatted SSL message was received.**
  - Reason: The SSL message is incorrectly formatted.
  - Solution: Check and correct the SSL format.
- **Message:SSL0134W: Initialization error, Could not verify MAC.**
  - Reason: The communication between the client and the server failed.
  - Solution: Retry the connection from the client.
- **Message:SSL0135W: Initialization error, 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 IBM HTTP Server does not support.
  - Solution: Retry the connection from the client using a SSL Version 2 or 3, or TLS 1 protocol. Try another certificate.
- **Message:SSL0136W: Initialization error, Invalid certificate signature.**
- **Message:SSL0137W: Initialization error, Invalid certificate sent by client.**
  - Reason: The client did not specify a valid certificate.
  - Solution: Client problem.
- **Message:SSL0138W: Initialization error, Invalid peer.**
- **Message:SSL0139W: Initialization error, Permission denied.**
- **Message:SSL0140W: Initialization error, The self-signed certificate is not valid.**
- **Message:SSL0141E: Initialization error, Internal error - read failed.**
  - Reason: The read failed.
  - Solution: None. Report this error to Service.
- **Message:SSL0142E: Initialization error, Internal error - write failed.**
  - Reason: The write failed.
  - Solution: None. Report this error to Service.
- **Message:SSL0143I: Initialization error, Socket has been closed.**
  - Reason: The client closed the socket before the protocol completed.
  - Solution: Retry connection between client and server.
- **Message:SSL0144E: Initialization error, 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:SSL0145E: Initialization error, 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:SSL0146E: Initialization error, 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:SSL0147E: Initialization error, Invalid security type combination.**
  - 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:SSL0148E: Initialization error, Internal error - SSL Handle creation failure.**
  - Reason: There was an internal error in the security libraries.
  - Solution: None. Report this problem to Service.
- **Message:SSL0149E: Initialization error, 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:SSL0150E: Initialization error, 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 LDAP server is available.
- **Message:SSL0151E: Initialization error, 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:SSL0152E: Initialization error, 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:SSL0153E: Initialization error, The PKCS#11 driver failed to find the token 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:SSL0154E: Initialization error, 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 PKCS#11 token or initialize the device.
- **Message:SSL0155E: Initialization error, The password/pin to access the PKCS#11 token is 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:SSL0156E: Initialization error, 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:SSL0157E: Initialization error, 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:SSL0158E: Initialization error, Internal error - The attribute has a negative length: <function>.**
  - Reason: The length value passed to the function is negative, which is invalid.
  - Solution: None. Report this problem to Service.
- **Message:SSL0159E: Initialization error, The enumeration value is invalid for the specified enumeration type: <function>.**
  - Reason: The function call contains an invalid function ID.
  - Solution: None. Report this problem to Service.
- **Message:SSL0160E: Initialization error, 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:SSL0161E: Initialization error, 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:SSL0162W: Setting the LD\_LIBRARY\_PATH for GSK failed. (SOLARIS2)**
- **Message:SSL0162W: Setting the LD\_LIBRARY for GSK failed. (LINUX)**
- **Message:SSL0162W: Setting the LIBPATH for GSK failed. (AIX)**
- **Message:SSL0162W: Setting the SHLIB\_PATH for GSK failed. (HPUX11)**
  - Reason: Memory allocation failure.
  - Solution: The process is low on memory and should be restarted.
- **Message:SSL0163W: Setting the LD\_LIBRARY\_PATH for GSK failed, could not append /usr/lib.(SOLARIS2)**
- **Message:SSL0163W: Setting the LD\_LIBRARY for GSK failed, could not append /usr/lib.(LINUX)**
- **Message:SSL0163W: Setting the LIBPATH for GSK failed, could not append /usr/opt/ibm/gskkm/lib.(AIX)**
- **Message:SSL0163W: Setting the SHLIB\_PATH for GSK failed, could not append /usr/lib.(HPUX11)**
  - Reason: Memory allocation failure.
  - Solution: The process is low on memory and should be restarted.
- **Message:SSL0164W: Error accessing Registry, <function> returned <code>.**
  - Reason: Memory allocation failure.
  - Solution: The process is low on memory and should be restarted.
- **Message:SSL0165W: Storage allocation failed.**
  - Reason: Memory allocation failure.
  - Solution: The process is low on memory and should be restarted.
- **Message:SSL0166E: Failure attempting to load GSK library.**
  - Reason: Either the GSK toolkit is not installed, a permissions problem exists, or the file does not exist.
  - Solution: Install the GSK toolkit, and check permissions on the library.
- **Message:SSL0167E: GSK function address undefined.**
  - Reason: Incorrect version of the GSK installed.
  - Solution: Install the correct version of the GSK.

- **Message:SSL0168E: SSL initialization for server: <server>, port: <port> failed due to a configuration error**
  - Reason: There was an error parsing the SSLClientAuthRequire directive.
  - Solution: Check the syntax of this directive.
- **Message:SSL0169E: Key file does not exist: <key file>.**
  - Reason: The file name specified for key file directive does not exist.
  - Solution: Check the key file directive. Use a fully qualified path and file name. If there are blanks in the path or file name, the directive should be enclosed in quotes.
- **Message:SSL0170E: GSK could not initialize, no key file specified.**
  - Reason: There is no key database file listed for this Virtual host.
  - Solution: Use the Keyfile directive to configure the key database file to use for SSL.
- **Message:SSL0171E: CRL cannot be specified as an option for the SSLClientAuth directive on HP-UX because the IBM HTTP Server does not support CRL on HP-UX**
  - Reason: Client certificate revocation checking is not supported on HP.
  - Solution: Remove the CRL option from the SSLClientAuth directive.
- **Message:SSL0172E: If CRL is turned on, you must specify an LDAP host name for the SSLCRLHostname directive"**
  - Reason: Certificate Revocation List (CRL) checking was enabled by the CRL option on the SSLClientAuth directive but the LDAP server containing the CRL was not specified.
  - Solution: Specify the LDAP server address using the SSLCRLHostname directive.
- **Message:SSL0173E: Failure obtaining supported cipher specs from the GSK library.**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.
- **Message:SSL0174I: No CRL password found in the stash file: <file name>.**
  - Reason: Certificate revocation list checking has been enabled which requires accessing an LDAP server but there is no password in the SSL Stash file to use to authorize the Web server to the LDAP server.
  - Solution: If accessing the LDAP server using an anonymous bind this message can be ignored. For authorized access, a password must be stashed in a file using the SSLStash utility.
- **Message:SSL0174I: No CRYPTO password found in the stash file: <file**

**name>.**

- Reason: SSL has been configured to use a PKCS 11 type cryptographic card but there is no password in the SSL stash file to use to access the cryptographic card token.
- Solution: Stash the password a file using the SSLStash utility.
- **Message:SSL0175E: fopen failed for stash file: <file name>**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.
- **Message:SSL0176E: fread failed for the stash file: <file name>**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.
- **Message:SSL0177E: stash\_recover <file>,\<function>\, pw\_buf, NULL > failed, invalid version <version>.**
  - Reason: The SSL stash file was created with an incompatible level of the SSLStash utility.
  - Solution: Create a new stash file using the SSLStash utility included with this version of the IBM HTTP Server.
- **Message:SSL0178E: stash\_recover <file>,\<function>\", pw\_buf, NULL > failed with invalid function.**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.
- **Message:SSL0179E: Unknown return code from stash\_recover(), <code>**
  - Reason: An internal error has occurred.
  - Solution: Report this problem to Service.
- **Message:SSL0180S: Unable to start session ID cache: %s\n**
- **Message:SSL0181S: Unable to fork for startup of session ID cache\n**

## **Handshake messages**

The following messages appear due to handshake failures:

- **Message:SSL0200E: Handshake Failed, <code>.**
  - Reason: The handshake failed when the SSL library returned an unknown error.
  - Solution: None. 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: 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:**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: Handshake Failed, The default key in the key file has an expired certificate. Use IKEYMAN to remove certificates that are expired**
  - Reason: An expired certificate exists in the key file and is the default.
  - Solution: Use IKEYMAN to select another certificate as the default.
- 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, Invalid date.**
  - Reason: The system date was set to an invalid date.
  - Solution: 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.
  - 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, Invalid certificate sent by client.**
  - Reason: The client did not specify a valid certificate.
  - Solution: Client problem.
- 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:**SSL0244E: Handshake Failed, Invalid security type combination.**
  - 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.

## **Configuration messages**

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>**

## I/O messages

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.

## Cache messages

The following message appears due to caching problems:

- Message:**SSL0600S: Unable to connect to session ID cache**
  - Reason: The server was not able to connect to the Session ID caching daemon.
  - Solution: Verify that the daemon was successfully started.

## Secure Sockets Layer Stash utility errors

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.

## Viewing error messages from a target server start

If you encounter an error starting a target server, the error message, line number in the configuration file and the actual line text that caused the error display. To view the line text error in context:

1. Click **View Configuration > Edit Configuration**.
2. Select the text.
3. Copy the text.
4. Go to **View Configuration > Edit Configuration** and press **Ctrl + F** for Find.
5. Paste the text.
6. Click **OK**.

## Identifying GSKit certificate support limitations

Ikeyman cannot be used to create certificates with key sizes larger than 1024 bits. However, certificates with key sizes up to 4096 can be imported into the key database.

## Looking at known problems with hardware cryptographic support

You must have the `bos.pkcs11` package installed on the AIX platform, 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.

The `ikmuser.sample` file shipped with the GSKit Toolkit, typically installs in the following directories, depending on the platform:

- AIX: `/usr/opt/ibm/gskta/classes`
- HP: `/opt/ibm/gsk7/classes`
- Linux: `/usr/local/ibm/gsk7/classes`
- Solaris: `/opt/ibm/gsk7/classes`



- Windows: C:\Program Files\ibm\gsk7\classes

Renaming this file to `ikmuser.properties` in the `classes` directory, enables IKEYMAN to use it for a cryptographic token.

If you are having problems using the IBM eBusiness Cryptographic Accelerator Device with IBM HTTP Server 2.0, do the following:

1. Reboot the machine.
2. Kill `pkcsslotd` and the shared memory it created. To determine what shared memory was created, type `ipcs -a` and for a size of 270760. This was the memory created by `pkcsslotd`.
3. Do an export `EXPSHM=ON`.
4. Start the `pkcs11` process: `/etc/rc.pkcs11`
5. Restart the IBM HTTP Server: `./apachectl start`

## Looking at known problems on the HP platform



You cannot install one version of GSKit onto another. Delete the current GSKit files from your system before installing a new GSKit version.

### **Identifying LDAP Secure Sockets Layer limitation with Netscape LDAP server**

The LDAP client has a limitation when using Secure Sockets Layer (SSL) to communicate to a Netscape directory server. If the Netscape directory server has client authentication enabled, the connection fails. If the IBM HTTP Server uses SSL with LDAP, to check authentication information on a Netscape Directory Server, ensure that client authentication is not enabled on the directory server.



## Looking at known problems on the Solaris platform

A known problem on the Solaris operating system includes specifying a valid `ServerName` directive.

On some Solaris machines (level unknown), an error is received at IBM HTTP Server startup (`apachectl`). The error indicates that the `ServerName` directive is not set in the IBM HTTP Server configuration file, `httpd.conf`. To resolve this problem, supply a valid `ServerName` directive.



## Looking at known problems on the Linux PowerPC platform



## Looking at known problems on the UNIX platform

## Getting the suexec module to work

The suexec module does not work unless IBM HTTP Server V2.0 is installed to the default location.

### Running the `<ihs install root>/bin/httpd` command

Source the `<ihs install root>/bin/envvars` file first to ensure you can run the `<ihs install root>/bin/httpd` command to start the IBM HTTP Server. To source the envvars file, enter `. <ihs install root>/bin/envvars` at the command line. The envvars file contains the path to the libraries needed to run the `<ihs install root>/bin/httpd` command.

## WINDOWS

## [Looking at known problems on Windows operating systems](#)

### 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 WSADuplicateSocket, the cause of the failure. The shim is not unloaded when Aventail Connect is shut down.

Fix the problem by doing one of the following:

- Explicitly unloading the shim
- Rebooting the machine
- Temporarily removing the Aventail Connect V3.x shim

## [Configuring security on Internet Explorer V5.01x](#)

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

## [Contacting Customer Service and Support](#)

For help, see the [WebSphere Application Server support page](#).

You can also contact the IBM Software Support Center (1-800-IBM-SERV in the US and Canada). For more information on software support services and contact numbers in other countries, refer to the [Software Support Handbook](#).

### **Finding related information**


- [Enabling cryptographic devices for Secure Sockets Layer](#)
- [Getting started](#)
- [Locating glossary terms](#)
- [Setting and viewing cipher specifications](#)
- [Using cipher specifications](#)
- [Using Lightweight Directory Access Protocol directives](#)
- [Using Secure Sockets Layer directives](#)

---

[\(Back to the top\)](#)



## Using Lightweight Directory Access Protocol directives


This section provides information on the Lightweight Directory Access Protocol (LDAP) directives. These directives work on all supported platforms. The information includes specific directive descriptions, values, defaults, and special notes . Links to related topics appear at the end of this section.

- Working with LDAP directives:
  - [LdapConfigFile](#)
  - [LDAPRequire](#)
  - [ldap.application.authType](#)
  - [ldap.application.DN](#)
  - [ldap.application.password.stashFile](#)
  - [ldap.cache.timeout](#)
  - [ldap.group.attributes](#)
  - [ldap.group.dnattributes](#)
  - [ldap.group.memberattribute](#)
  - [ldap.group.memberAttributes](#)
  - [ldap.group.name.filter](#)
  - [ldap.group.URL](#)
  - [ldap.idleConnection.timeout](#)
  - [ldap.key.file.password.stashfile](#)
  - [ldap.key.fileName](#)
  - [ldap.key.label](#)
  - [ldap.realm](#)
  - [ldap.search.depth](#)
  - [ldap.search.timeout](#)
  - [ldap.transport](#)
  - [ldap.url](#)
  - [Ldap.user.authType](#)
  - [ldap.user.cert.filter](#)
  - [ldap.user.name.fieldSep](#)
  - [ldap.user.name.filter](#)
  - [ldap.version](#)
  - [ldap.waitToRetryConnection.interval](#)
- [Finding related information](#)

### LdapConfigFile


- Description: Indicates the name of the LDAP properties file associated with a group of LDAP parameters.

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

 Use this directive in the `httpd.conf` file.

## LDAPRequire

- Description: Indicates the group when using LDAP authentication.
- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `LDAPRequire filter <filter name> or LDAPRequire group <group1 [group2.group3....]>`
- Values: `LDAPRequire filter "&(objectclass=person)(cn=*)(ou=IHS)(o=IBM)"`, or `LDAPRequire group "sample group"`.

 Use this directive in the `httpd.conf` file.

## ldap.application.authType


- Description: Specifies the method for authenticating the Web server to the LDAP server.
- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.application.authType=None`
- 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

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

## ldap.application.password.stashFile

- Description: 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.
- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.application.password.stashFile=c:\IHS\ldap.sth`
- Values: Fully qualified path to the stash file.

 You can create this stash file with the **ldapstash** command.

### [ldap.cache.timeout](#)

- Description: 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.
- Default: 600
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.cache.timeout= <secs>`
- Values: The maximum length of time, in seconds, a response returned from the LDAP server remains valid.

### [ldap.group.attributes](#)

- Description: Indicates the filter used to determine if a distinguished name (DN) is an actual group through an LDAP search.
- Default: `groupofnames groupofuniquenames`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.group.attribute= attribute1 [attribute2...]`
- Values: Filter name

### [ldap.group.dnattributes](#)

- Description: Filter used to determine, via an LDAP search, if a DN is an actual group
- Default: `groupofnames groupofuniquenames`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.group.memberattribute= <ldap filter>`
- Values: An ldap filter. - See sample `ldap.prop.sample` for more information on the use of this directive.

### [ldap.group.memberattribute](#)

- Description: Attribute specified to retrieve unique groups from an existing group
- Default: `uniquegroup`

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

## [ldap.group.memberAttributes](#)

- Description: Serves as a means to extract group members, once the function finds a group entry in an LDAP directory.
- Default: `member` and `uniqueMember`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.group.memberAttributes= attribute [attribute2....]`
- 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](#)

- Description: Indicates the filter LDAP uses to search for group names.
- Default:  
`(&(cn=%v1)(|(objectclass=groupOfNames)(objectclass=groupOfUniqueNames))`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.group.name.filter= <group name filter>`
- Values: An LDAP filter. See [Querying the LDAP server using LDAP search filters](#).

## [ldap.group.URL](#)

- Description: 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.
- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.group.URL=ldap://<hostName:Port>/<BaseDN>`
- Values:
  - `hostName`: 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.

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



## ldap.idleConnection.timeout

- Description: Caches connections to the LDAP server for performance.
- Default: 600
- Module: mod\_ibm\_ldap
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: ldap.idleConnection.timeout= <secs>
- Values: Length of time, in seconds, before an idle LDAP server connection closes because of inactivity.

## ldap.key.file.password.stashfile


- Description: Indicates the stash file containing the encrypted keyfile password; use the **ldapstash** command to create this stash file.
- Default: None
- Module: mod\_ibm\_ldap
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: ldap.key.file.password.stashfile =d:\<Key password file name>
- Values: Fully qualified path to the stash file.

## ldap.key.fileName

- Description: Indicates the file name of the key file database. This option becomes required when you use SSL.
- Default: None
- Module: mod\_ibm\_ldap
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: ldap.key.fileName=d:\<Key file name>
- Values: Fully qualified path to the key file.

## ldap.key.label

- Description: Indicates the certificate label name the Web server uses to authenticate to the LDAP server.
- Default: None
- Module: mod\_ibm\_ldap
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: My Server Certificate
- 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.realm

- Description: Indicates the name of the protected area, as seen by the requesting client.

- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.realm==<Protection Realm>`
- Values: A description describing the protected page.

## [ldap.search.depth](#)

- Description: Searches subgroups when specifying `LdapRequire group <group>` directives. Groups can contain both individual members and other groups.
- Default: 1
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.search. = <secs>`
- 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.search.timeout](#)

- Description: Indicates the maximum time, in seconds, to wait for an LDAP server to complete a search operation.
- Default: 10
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.search.timeout = <secs>`
- Values: Length of time, in seconds.

## [ldap.transport](#)

- Description: Indicates the transport method used to communicate with the LDAP server.
- Default: TCP
- Module: `mod_ibm_ldap`

- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.transport=TCP`
- Values: TCP or SSL

## ldap.url


- Description: Indicates the URL of LDAP server to authenticate against.
- Default: None
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- 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`

## Ldap.user.authType

- Description: Indicates the method for authenticating the user requesting a Web server. Use this name as the user name when accessing an LDAP server.
- Default: `Basic`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `Ldap.user.authType=BasicIfNoCert`
- Values: `Basic`, `Cert`, `BasicIfNoCert`

## ldap.user.cert.filter

- Description: Indicates the filter used to convert the information in the client certificate passed over SSL to a search filter for an LDAP entry.
- Default: `"(&(objectclass=person) (cn=%v1, ou=%v2, o=%v3, c=%v4))"`.
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.user.cert.filter=(&(objectclass=person) (cn=%v1))`
- Values: An LDAP filter. See [Querying the LDAP server using LDAP search filters](#).

 SSL certificates include the following fields, all of which you can convert to a search filter:

<u>Certificate field</u>	<u>Variable</u>
common name	<code>%v1</code>
organizational unit	<code>%v2</code>


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:

<u>User</u>	<u>Filter Conversion</u>
<u>Certificate</u> 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

- Description: Indicates characters as valid field separator characters, when parsing the user name into fields.
- Default: The space, comma, and the tab (/t) character.
- Module: mod\_ibm\_ldap
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: ldap.user.name.fieldSep=/  
/
- Values: Characters

 If '/' represents the only field separator character and the user inputs "Joe Smith/Acme," then '%v1' equals "Joe Smith" and '%v2' equals "Acme."

### ldap.user.name.filter

- Description: Filter used to convert the user name as input by the user to a search filter for an LDAP entry.
- Default: "((objectclass=person) (cn=%v1 %v2))"  
where: %v1 and %v2 represent the words typed by the user.

For example, if the user types "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 types in **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
- Scope: Single instance per directory stanza.
- Syntax: `ldap.user.name.filter=<User Name Filter>`
- Values: An LDAP filter. See [Querying the LDAP server using LDAP search filters](#).

## Ldap.version

- Description: 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. This directive is optional.
- Default: `ldap.version=3`
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.version=3`
- Values: 2 or 3

## Ldap.waitToRetryConnection.interval

- Description: Indicates the time the Web server waits between failed attempts to connect. If an LDAP server goes down, the Web server continually thrashes, trying to connect.
- Default: 300
- Module: `mod_ibm_ldap`
- Multiple instances in the configuration file: Yes
- Scope: Single instance per directory stanza.
- Syntax: `ldap.waitToRetryConnection.interval=<secs>`
- Values: Time (in seconds)

## **Finding related information**


- [Getting started with the Lightweight Directory Access Protocol](#)
- [Locating glossary terms](#)
- [Using the Lightweight Directory Access Protocol](#)


---

[\(Back to the top\)](#)



## Using the Lightweight Directory Access Protocol

This section provides information on the Lightweight Directory Access Protocol (LDAP). This information includes basic concepts, overview subjects and associated notes , along with querying, installing and configuring procedures. Links to related information appear at the end of this section.

 This section is applicable to all supported operating systems.

- [Introducing the Lightweight Directory Access Protocol](#)
  - [X.500 overview](#)
  - [Lightweight Directory Access Protocol overview](#)
- [Querying the Lightweight Directory Access Protocol server using Lightweight Directory Access Protocol search filters](#)
  - [Looking at examples of Lightweight Directory Access Protocol search filters](#)
- [Installing the Lightweight Directory Access Protocol client](#)
- [Configuring Lightweight Directory Access Protocol on the IBM HTTP Server](#)
- [Finding related information](#)

## Introducing the Lightweight Directory Access Protocol

The following section addresses questions about what LDAP is and how it works, and provides high level overviews of X.500 and LDAP.

### **What is LDAP?**

The Lightweight Directory Access Protocol (LDAP) exists as an information directory where you define users and groups once and share them across multiple machines and multiple applications.

### **How does LDAP work?**

The IBM HTTP Server LDAP plug-in enables the directory to perform authentication and authorization required when accessing a protected resource. This capability greatly decreases the administrative overhead for maintaining user and group information locally for each Web server.

The IBM HTTP Server supports LDAP, a protocol that provides access to the X.500 directory, over a TCP or SSL connection. LDAP lets you store information in a directory service and perform queries in a database. When you use X.500

directories and LDAP, any LDAP-enabled application can store information, such as user authentication information, once other applications using the LDAP server can recognize this data.

LDAP reduces required system resources, by including only a functional subset of the original X.500 Directory Access Protocol (DAP).

IBM HTTP Server LDAP support offers a choice of LDAP configurations including:

- A single LDAP server
- Different LDAP servers accessed for different requests. For example, requests can come in from two different IP addresses, and the Web server can contact a different LDAP server for each request.

This information assumes you have an existing X.500 directory service available, for example, the IBM SecureWay X.500 directory.

## [X.500 overview](#)

X.500 provides a directory service with components 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 others 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*, based on the ObjectClass attribute. Each entry consists of attributes. The ObjectClass attribute identifies the type of entry, for example, person or organization, which 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 a certain protocol, for example 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 X.500 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.

## [Querying the Lightweight Directory Access Protocol server using Lightweight Directory Access Protocol search filters](#)

LDAP accesses the X.500 directory through human readable strings. When these query strings pass to the LDAP server, the server returns the distinguished name of the entry.

You find LDAP entries typed, or classified, by an ObjectClass attribute to simplify searches. For example, you can search an LDAP directory with `objectclass=acl` to locate all entries using *access control lists*.

A search filter for an LDAP entry has the following structure:

- Filters must begin and end with parentheses. See the following examples which show the placement of parentheses in complex queries.
- Filters can contain the following Boolean comparisons:
  - & - Boolean AND
  - | - Boolean OR
  - ! - Boolean NOT
- A particular object class, can require an attribute name.
- Filters can contain the following equality expressions
  - = - equal to
  - ~= - approximately equal to
  - >= - greater
  - <= - less
- Filters must contain the value of the attribute on which to search. This value can contain wildcards.

For more information on LDAP search filters, see RFC 1960.

## [Looking at examples of Lightweight Directory Access Protocol search filters](#)

The following LDAP search filter: `(cn=Joe Smith)` searches the directory service for the common name of Joe Smith. Possible matches include:

Joe Smith

The following search filter: `(!(cn=Jane Doe))` queries the directory service for



entries whose common name does not equal Jane Doe. Possible matches include:

Joe Schmoe

Adam Fosset

any name other than Jane Doe

The following search filter: `(&objectClass=acl)((sn=Johnson)` queries all access control list (ACL) entries matching a surname of Johnson. Possible matches include:


Peter Johnson

Davey Johnson

The following search filter: `(o=univ*of*carolin*)` queries the organization attribute. Possible matches include:

University of North Carolina Chapel Hill


University of South Carolina

 LDAP can return more than one entry. However, the Web server does not authenticate when multiple entries return. If the directory queried by this example contained both the University of North Carolina and the University of South Carolina, the server returns both entries, and authentication fails. You must alter the search filter.

## [Installing the Lightweight Directory Access Protocol client](#)

Running the IBM HTTP Server LDAP module requires a separate download of the LDAP client. You can obtain the LDAP client by downloading the client from a Web site and installing the client outside of the IBM HTTP Server installation.

WINDOWS

 If you download the client from a Web site, you can see references to the SecureWay directory. The SecureWay directory contains the LDAP client, which provides access to LDAP-capable servers. These instructions pertain to the Windows operating system only.

WINDOWS

To download the LDAP client from a Web site:

1. Open a browser and go to: [LDAP Download](#)
2. Click the applicable platform version. The SecureWay Directory Evaluation Code window appears.
3. Click **Download**.
4. Click the applicable V4.1 platform version of the listed directory server software and Client Software Developer Kits.
5. Click **Download**.
6. Select the appropriate language for the platform choice at the SecureWay directory and Client SDK Version 4.1 window, by clicking the **down arrow key** and clicking the language.
7. Click **Continue**.
8. Register for the download and provide a user ID and password, if asked at the next window.
9. Unzip the file.
10. Install the client, following the instructions in the Installation and Configuration Guide.



You do not need to download the LDAP Client on the AIX, HP, Linux and Solaris operating systems.

## [Configuring Lightweight Directory Access Protocol on the IBM HTTP Server](#)

To configure LDAP on the IBM HTTP Server to protect files, follow these steps:

1. To define by user:
  1. Launch the IBM Administration Server.
  2. Go to **Access Permissions > General Access** and insert the file, LdapConfigFile (C:/Program Files/IBM HTTP Server/conf/ldap.prop) in the LDAP: Configuration File field. This file is required.
  3. Enter the authentication realm name for the directory in the Authentication Realm Name field.

2. To define by group:

```
LDAPRequire group "group_name"
```


```
Example: LDAPRequire group "Administrative Users"
```

3. To define by filter:

```
LDAPRequire filter "ldap_search_filter"
```

```
For example: LDAPRequire filter
```

```
" (&(objectclass=person)(cn=*)(ou=IHS)(o=IBM)) "
```

 The LDAPRequire directive only works if manually inserted into the `httpd.conf` file.

4. To create an LDAP connection, provide information about the LDAP server used. Edit your LDAP properties file, `sample ldap.prop`, found in the IBM HTTP Server `conf` directory and insert the applicable directives:
  - Enter the Web server connection information
  - Enter client connection information
  - Enter timeout settings
5. Click **Submit** to continue, or **Reset** to clear the form.

### Finding related information

- [Getting started with the Lightweight Directory Access Protocol](#)
- [Locating glossary terms](#)
- [Using Lightweight Directory Access Protocol directives](#)

---

[\(Back to the top\)](#)



---

## Getting started with the Lightweight Directory Access Protocol

This section discusses the functions involved in getting started with Lightweight Directory Access Protocol (LDAP). Links to related topics appear at the end of this section.

- [Protecting files or directories with user or group information on a Lightweight Directory Access Protocol server](#)
- [Using key ring files](#)
- [Using Secure Sockets Layer and the Lightweight Directory Access Protocol module](#)
- [Creating a Lightweight Directory Access Protocol connection](#)
- [Identifying supported Lightweight Directory Access Protocol servers on the IBM HTTP Server](#)
- [Finding related information](#)

### [Protecting files or directories with user or group information on a Lightweight Directory Access Protocol server](#)

You can protect files and directories with user or group information by defining through a user, group, or filter:

#### ***To define by user:***

Manually insert the following directives into your configuration file, under a directory or Location stanza:

- `LdapConfigFile | path to ldap configuration file |`
- `AuthName | name |`
- `AuthType: basic`
- `Require valid -user`

#### ***To define by group:***

```
LDAPRequire group "group_name"
```


```
For example: LDAPRequire group "Administrative Users"
```

## To define by filter:

```
LDAPRequire filter "ldap_search_filter"
```

For example: LDAPRequire

```
filter "( &(objectclass=person)(cn=*)(ou=IHS)(o=IBM) ) "
```

 LDAPRequire only works if manually inserted into the `httpd.conf` file.

## Using key ring files

To use the `mod_ibm_ssl` and `mod_ibm_ldap` files when configuring LDAP to use SSL for communicating with the LDAP server, both the `mod_ibm_ssl` and `mod_ibm_ldap` files must use the same key ring file. If you enable SSL connections to the Web server and also use SSL as the transport between the Web server and the LDAP server, the key ring files used for both modules can merge into one key ring file. The configuration of each module can specify a different default certificate.

## Using Secure Sockets Layer and the Lightweight Directory Access Protocol module

When using Secure Sockets Layer (SSL) between the Lightweight Directory Access Protocol (LDAP) module and the LDAP directory server, the key database file must have write permission. The key database file contains the certificates which establish identity, and in a secure environment, the LDAP server can require the Web server to provide a certificate for querying the LDAP server for authentication information. The key database file must have write permission by the UNIX user ID on which the Web server runs.

Certificates establish identity, to prevent other certificates from stealing or overwriting your certificates. 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.

The LDAP module requires the password to the user's key database, even if a stash file exists. Use the **ldapstash** command to create an LDAP stash file, containing the key database file password.

## Creating a Lightweight Directory Access Protocol connection

To create an LDAP connection, provide information about the LDAP server.

1. Edit your sample LDAP properties file, `ldap.prop`, located in the IBM HTTP Server `conf` directory. Insert the applicable directives.
2. Enter the Web server connection information.
3. Enter client connection information.

4. Enter timeout settings.

## [Identifying supported Lightweight Directory Access Protocol servers on the IBM HTTP Server](#)

The IBM HTTP Server supports the following LDAP servers:

- [iPlanet/Netscape Directory Server](#)
- [IBM SecureWay Directory Server](#)

### **Finding related information**

- [Locating glossary terms](#)
- [Using the Lightweight Directory Access Protocol](#)
- [Using Lightweight Directory Access Protocol directives](#)

---

[\(Back to the top\)](#)



## Using the Secure Sockets Layer protocol for secure communications

This section provides foundation information for the Secure Sockets Layer (SSL) protocol, including a general overview of the protocol, and explanations of security concepts. Links to related topics appear at the end of this section.

- [Using SSL for secure communications](#)
- [Reviewing security concepts](#)
  - [Communicating securely](#)
  - [Understanding encryption](#)
  - [Understanding authentication](#)
  - [Identifying a Public Key Infrastructure](#)
  - [Looking at the Secure Sockets Layer protocol](#)
- [Finding related information](#)

SSL represents an encryption system used on servers to ensure privacy when transmitting information across the World Wide Web. SSL-enabled servers encrypt sensitive data into ciphertext before sending it to clients, preventing third parties from reading the data, even if they intercept this data en route. Clients receiving data from the server then decrypt the ciphertext to read the data. Using SSL on a Web server helps ensure that information transmitted between a client, such as a Web browser and a server, such as a Web server, remains private, and enables the clients to authenticate the identity of the server.

### [Using SSL for secure communications](#)

For a server and client to use SSL for secure communications, the server must have a public and private key pair and a certificate. The server uses its private key to sign messages to clients. The server sends its public key to clients so these clients can verify that the signed messages come from the server and so they can encrypt messages to the server. The server then decrypts these messages with its private key.

To send its public key to clients, the server needs a certificate issued by a *certificate authority (CA)*. The certificate contains a server public key, the *distinguished name* associated with the server certificate, the serial number or issue date of the certificate, and the expiration date of the certificate.

A certificate authority (CA) is a trusted third party (or a designated internal authority) that issues certificates. The CA verifies the identity of the server and digitally signs the certificate with its private key and uses its public key to ensure that the certificate is valid. A signed certificate binds server identity to a pair of electronic keys, used to encrypt and sign digital information. The certificate authority private key signs the certificate to verify server identity.

To operate a Web server in secure SSL mode, you must first obtain a signed certificate for your system, from a certificate authority. VeriSign, Inc. represents one of a number of companies that acts as a certificate authority. However, you can use a signed certificate of the appropriate format from any certification authority.

When you set up secure connections, associate your public key with a digitally signed certificate from a certificate authority (CA), designated as a trusted CA on your server.

## [Reviewing security concepts](#)

This section provides an overview of security concepts.

## [Communicating securely](#)

The rapid growth of electronic commerce over the Internet has led to an increasing demand for secure network communications. In addition, intra-company communications over private networks often contain confidential information that needs protection.

A secure network communication has the following characteristics:

### **Access control**

Only authorized parties protect and access resources. Restricting access on the basis of passwords, IP address, host names, or SSL client authentication ensures access control.

### **Authenticity**

You know who you are talking to and that you can trust that person. Authentication, using digital signature and digital certificates, ensures authenticity.

### **Information integrity**

Messages do not get altered during transmission. Without information integrity, you have no guarantee that the message you sent matches the message received. Digital signature ensures integrity.

### **Privacy and confidentiality**



Information conveyed from party to party during a transaction remains private and confidential, even if it gets into the wrong hands. Encryption ensures privacy and confidentiality.

## [Understanding encryption](#)

Encryption in its simplest form involves scrambling a message so that no one can read the message until unscrambled by the receiver. The sender uses an algorithmic pattern, or 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 encrypted with your public key because only you have the private key. Key pairs are stored in a key database 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 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 to send its public key to clients. A certificate authority (CA) issues this certificate and verifies the identity of the server.

## Understanding 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 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
- The expiration date

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. You can add, or remove trusted CAs, using the IBM Key Management Utility 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 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.

## **Identifying a Public Key Infrastructure**

A *Public Key Infrastructure (PKI)* represents a system of digital certificates, certificate authorities, registration authorities, certificate management service, and X.500 directories. PKI verifies the identity and authority of each party 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.

## **Looking at the Secure Sockets Layer protocol**

The Secure Sockets Layer (SSL) protocol was developed by Netscape Communications Corporation. SSL ensures that data 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.

Once 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. This functionality enables a retail company on the Internet to allow users to look through their merchandise without security, but then fill out order forms and send their credit card numbers using security.

### Finding related information

- [Associating your public key with certificate authorities](#)
- [Authenticating Secure Sockets Layer clients and Secure Sockets Layer Version 3](#)
- [Getting started quickly with secure connections](#)
- [Linking to Security Web site](#)
- [Locating glossary terms](#)
- [Retrieving Lightweight Directory Access Protocol information](#)
- [Using certificates](#)
- [Using cipher specifications and key sizes](#)


---

[\(Back to the top\)](#)



IBM  
HTTP Server

## Installing and uninstalling the IBM HTTP Server


This file contains procedures for installing and uninstalling the IBM HTTP Server, along with associated notes .

- [Installing the IBM HTTP Server](#)
- [Uninstalling the IBM HTTP Server](#)
- [Finding related information](#)

### [Installing the IBM HTTP Server](#)

To install the IBM HTTP Server:

1. Ensure you have the IBM Developer Kit, Java Technology Edition Version 1.4, installed on your machine.

 This Developer Kit ships with the WebSphere Application Server on the CD and is available from the following Web site:


<http://www.ibm.com/java/jdk>.

2. For AIX, ensure that you have the xIC.rte 6.0 runtime. This is a prerequisite of GSKit7. You can download this runtime from <https://techsupport.services.ibm.com/server/aix.fdc>.
3. Create a new directory to uncompress the IBM HTTP Server install image.
4. Download the IBM HTTP Server install image.
5. Uncompress the install image in this new directory. For example, if the compressed file name is `IHS.tar` or `IHS.zip`, type `tar -xf IHS.tar` or use a zip utility.

A listing of the following files appears, based on your operating system:

- `gskit.sh`
- `setup.jar`
- A GSKit run-time executable:
  - AIX: `gskta.rte`
  - HP-UX: `gsk7bas` directory
  - Linux for Intel: `gsk7bas_295-7.0-1.10.i386.rpm`
  - Linux for S/390: `gsk7bas-7.0-1.10.s390.rpm`
  - Linux for PowerPC: `gsk7bas-7.0.-1.10.ppc32.rpm`
  - Solaris: `gsk7bas` directory
  - Windows: `skit` directory

6. Go to the directory where you uncompressed the install image and type `java -jar setup.jar`.

 If the system cannot find the `java.exe` file, set your path to point to the Java product installed on your machine. For example: Set your path to: `export PATH=$PATH:/usr/java14/java/bin` before running: `java -jar setup.jar`.

You can also choose to do a silent installation. To do a silent installation, type: `java -jar setup.jar -silent -options silent.res`. To customize the install options, edit the `silent.res` text file. All options are set to **true** by default. To disable an option, set its value to **false**.

7. Choose the language in which to run the installation.

The Welcome to the InstallShield Wizard for the IBM HTTP Server appears.

8. Click **Next**. The license agreement appears.
9. Click **I accept the terms of the license agreement**, or **I do not accept the terms of the license agreement**.
10. Click **Next**.
11. Specify the directory name. The default directories follow:
  - o AIX: `/usr/IBMIHS/`
  - o HP\_UX: `/opt/IBMIHS/`
  - o Linux: `/opt/IBMHIHS/`
  - o Linux for S/390: `/opt/IBMIHS/`
  - o Linux for PowerPC: `/opt/IBMIHS/`
  - o Solaris: `/opt/IBMIHS/`
  - o Windows: `skit directory 2.0\`

12. Click **Next**.

The option appears for you to perform a typical, custom, or developer installation.

13. Select the type of installation you would like to perform:
  - o Typical
  - o Custom: Enables you to select multiple language installations simultaneously.
  - o Developer: Non-administrator

14. Click **Next**.

If you selected the typical installation, a list appears with everything included in the installation, along with the size of the image.

If you selected the custom installation, a list of components appears and you can clear the box next to the components you do not want to install.

15. Click **Next**. The following message appears: Installing IBM HTTP

Server. Please wait. You can click **Cancel** to stop the installation.

The message Updating the inventory appears. You can click **Cancel** to stop the inventory update.

16. Click **Finish**.

 You can now install in any path, on UNIX platforms, and can have more than one installation for the same machine.

## [Uninstalling the IBM HTTP Server](#)

To uninstall the IBM HTTP Server:

1. Go to the directory where you installed the IBM HTTP Server. Change to the `_uninst` directory, located in the root directory.
2. Type `java -jar uninstall.jar` on all platforms.

You can also choose to do a silent uninstall. To do a silent uninstall, type `java -jar uninstall.jar -silent`

The uninstall process on UNIX systems does not automatically uninstall the GSKit. You have to uninstall the GSKit manually by using the native uninstall method.

### Finding related information

- [Troubleshooting](#)
- [Understanding National Language Support](#)

---

[\(Back to the top\)](#)



## Understanding National Language Support

This section provides important information regarding National Language Support (NLS). This information includes documentation locations and configuration options per platform. Links to related topics appear at the end of this section.

- [Locating NLS documentation](#)
- [Understanding options for configuration of languages for the IBM HTTP Server](#)
  - [Using the installed script](#)
  - Manually editing the configuration files and the IKEYMAN start script:
    - [On the AIX operating system](#)
    - [On the HP operating system](#)
    - [On the Linux operating system](#)
    - [On the Solaris operating system](#)
  - [Manually editing IBM HTTP Server configuration files on Windows operating systems](#)
  - [Finding related information](#)

UNIX

The IBM HTTP Server supports nine languages, in addition to US English:

- Brazilian Portuguese
- Chinese Simplified
- Chinese Traditional
- French
- German
- Italian
- Japanese
- Korean
- Spanish



## Locating NLS documentation

You can find documentation supporting all nine languages located in language-specific subdirectories. For the IBM HTTP Server, these directories follow:

- AIX operating system: `/usr/IBMIHS/htdocs/<Lang>`
- HP operating system: `/opt/IBMIHS/htdocs/<Lang>`
- Linux operating system: `/opt/IBMIHS/htdocs/<Lang>`
- Solaris operating system: `/opt/IBMIHS/htdocs/<Lang>`
- Windows operating systems: `c:\Program Files\IBM HTTP Server 2.0\htdocs\<Lang>`

## Understanding options for configuration of languages for the IBM HTTP Server

You can choose from two options to configure languages for the IBM HTTP Server on UNIX platforms: Using a script installed with the IBM HTTP Server product, or manually editing the configuration files and the IKEYMAN start script.

### Using the installed script

A script named `<setuplang>`, installs with the IBM HTTP Server on UNIX platforms. The SETUPLANG script handles changes for the IBM HTTP Server.

This script makes the necessary changes to the configuration files for the IBM HTTP Server. These necessary changes help to recognize the language directories. Execute the following command for each operating system:

- AIX: `/usr/IBMIHS/bin/setuplang`
- HP: `/opt/IBMIHS/bin/setuplang`
- Linux: `/opt/IBMIHS/bin/setuplang`
- Solaris: `/opt/IBMIHS/bin/setuplang`

## Manually editing IBM HTTP Server configuration files and the IKEYMAN start script on the AIX operating system

To manually edit the IBM HTTP Server configuration files and the IKEYMAN start script, on the AIX operating system:

1. Edit the IBM HTTP Server configuration file, `HTTPD.CONF`, and change the directive, `DocumentRoot` from: `DocumentRoot /usr/IBMIHS/htdocs/en_US` to: `DocumentRoot /usr/IBMIHS/htdocs/<Lang>`  
where: `<Lang>` represents the language-specific directory. For example, the representation of the French directory follows:  
`DocumentRoot /usr/IBMIHS/htdocs/Fr_FR.`
2. Edit the IBM HTTP Server IKEYMAN start script, `/usr/bin/ikeyman`.  
Change the first entry of `export IKMLANG=` from: `export IKMLANG=en_US` to: `export IKMLANG=<Lang>`  
where: `<Lang>` represents the language-specific directory. For example, to use the French directory, specify:  
`export IKMLANG=Fr_FR`

## Option 2: Manually editing IBM HTTP Server configuration files and the IKEYMAN start script on the HP operating system

To manually edit the IBM HTTP Server configuration files and the IKEYMAN start script on the HP-UX operating system:

1. Edit the IBM HTTP Server configuration file, `HTTPD.CONF`, and change the directive, `DocumentRoot` from: `DocumentRoot /opt/IBMIHS/htdocs/en_US` to: `DocumentRoot /opt/IBMIHS/htdocs/<Lang>`  
where: `<Lang>` represents the language-specific directory. For example, the representation of the French directory follows:  
`DocumentRoot /opt/IBMIHS/htdocs/Fr_FR`
2. Edit the IBM HTTP Server IKEYMAN start script, `/usr/bin/ikeyman`, and change the first entry of `export IKMLANG=` from: `export IKMLANG=en_US` to: `export IKMLANG=<Lang>`  
where: `<Lang>` represents the language-specific directory. For example, to use the French directory, specify:  
`export IKMLANG=Fr_FR`

## Option 2: Manually editing the IBM HTTP Server configuration files and the IKEYMAN start script on the Linux operating system

To manually edit the IBM HTTP Server configuration files and the IKEYMAN start script on the Linux operating system:

1. Edit the IBM HTTP Server configuration file, `HTTPD.CONF`, and change the directive, `DocumentRoot`, from: `DocumentRoot /opt/IBMIHS/htdocs/en_US` to: `DocumentRoot /opt/IBMIHS/htdocs/<Lang>` where: `<Lang>` represents the language-specific directory. For example, the representation of the French directory follows: `DocumentRoot /opt/IBMIHS/htdocs/Fr_FR`.
2. Edit the IBM HTTP Server IKEYMAN start script, `/usr/bin/ikeyman`, and change the first entry of `export IKMLANG=` from: `export IKMLANG=en_US` to: `export IKMLANG=<Lang>` where: `<Lang>` represents the language-specific directory. For example, to use the French directory, specify:  
`export IKMLANG=Fr_FR`

## Option 2: Manually editing the IBM HTTP Server configuration files and the IKEYMAN start script on the Solaris operating system

To manually edit the IBM HTTP Server configuration files and the IKEYMAN start script on the Solaris operating system:

1. Edit the IBM HTTP Server configuration file, `HTTPD.CONF`, and change the directive, `DocumentRoot`, from: `DocumentRoot /opt/IBMIHS/htdocs/en_US` to: `DocumentRoot /opt/IBMIHS/htdocs/<Lang>` where: `<Lang>` represents the language-specific directory. For example, the representation of the French directory follows:  
`DocumentRoot /opt/IBMIHS/htdocs/Fr_FR`
2. Edit the IBM HTTP Server IKEYMAN start script, `/usr/bin/ikeyman`, and change the first entry of `export IKMLANG=`, from: `export IKMLANG=en_US` to: `export IKMLANG=<Lang>` where: `<Lang>` represents the language-specific directory. For example, to use the French directory, specify:  
`export IKMLANG=Fr_FR`

## Manually editing IBM HTTP Server configuration files on Windows operating systems

WINDOWS

To manually edit the IBM HTTP Server configuration files on Windows operating systems:

1. Edit the IBM HTTP Server configuration file, `HTTPD.CONF`, and change the `DocumentRoot` directive from:

```
DocumentRoot "C:\Program Files\IBM HTTP Server\htdocs\en_US"
```

to:

```
DocumentRoot "C:\Program Files\IBM HTTP Server\<Lang>"
```

where: *<Lang>* represents the language-specific directory. For example, the representation of the French directory follows:

```
DocumentRoot "C:\Program Files\IBM HTTP Server\htdocs\Fr_FR"
```

2. Change the `Include` directive from:

```
Include conf/admin.msg.en_US
```

to:

```
Include "conf/admin.msg.<Lang>"
```

where: *<Lang>* represents the language-specific directory. For example, to use the French directory, specify:

```
Include "conf/admin.msg.fr_FR"
```

### Finding related information

- [Installing and uninstalling the IBM HTTP Server](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



## **Building dynamic shared object modules or dynamic link libraries with the IBM HTTP Server**

This section provides information on building dynamic shared object (DSO) modules and dynamic link libraries (DLL). This information includes identifying compilers, locating build components and understanding build options. Links to related information appear at the end of this section.

UNIX

WINDOWS

- [Identifying viable compilers](#)
- [Locating build components for the UNIX platform](#)
- [Locating build components for the Windows operating systems](#)
- [Understanding build method options](#)
- [Building third-party modules to run as dynamic shared object modules or dynamic link libraries](#)
- [Finding related information](#)

Build Apache modules and third-party modules as dynamic shared object modules (DSOs), or dynamic link libraries (DLLs), for execution with the IBM HTTP Server. Apache modules that statically link with Apache during Apache build and installation, cannot statically link with the IBM HTTP Server. The IBM HTTP Server ships as an installation image with executables that you cannot rebuild, since the source does not ship with the installation image. The IBM HTTP Server does ship the header files necessary to compile and build a DSO, or DLL that executes as an IBM HTTP Server module.

The product executable, when compared to Apache, contains source code changes. The majority of these changes exist as hooks to accommodate the Fast Response Cache Accelerator (FRCA) and Secure Sockets Layer (SSL).

To assist in building DSOs and DLLs, review the general information provided below:

## Identifying viable compilers

Apache modules and third-party module testing incorporated the compilers and compiler levels listed below. Other compilers can work, but testing was limited to these environments:

- AIX - C for AIX V5.0.2.3 or VisualAge C++ Professional V5.0.2.3
- HP - HP\_UX aC++ Compiler (A.03.xx)
- Linux for Intel - gcc-2.95.x
- Linux for PowerPC - gcc 3.2.2
- Linux for S/390 - gcc 3.2.2
- Solaris - SunWorkShop V5.0
- Windows - Microsoft Visual C++ 6.0



UNIX

## Locating build components for the UNIX platform

Locations of the key components for building DSOs and DLLs on the UNIX platform follow:

- Locate the header files in the `include` directory.
- Locate the APXS script in the `bin` directory.

## Locating build components for the Windows operating systems

WINDOWS

Use the library files and the header files with the IBM HTTP Server on Windows operating systems.

- Locate the library files in the `src/include` directory.
- Locate the header files in the `src/os/win32` directory.

UNIX

WINDOWS

## Understanding build method options

You have several options available to you for building dynamic modules:

- **Configuration Scripts:** Some Apache modules include configuration scripts with the module source. These configuration scripts make compiling and installing modules easy and the module writer usually supports them.

Sometimes on the UNIX platform, these configuration scripts have a `WITH_APXS` option. If you use this option, ensure you point to the APXS script installed with the IBM HTTP Server, in the `bin` directory.

Check the configuration script parameters if you experience problems building or running your DSO-built module.

- The `example_module` directory:
  - UNIX platform: The IBM HTTP Server ships a sample module (`mod_example.c`) in the `example_module` directory. See [Apache APXS](#) for more information.
  - Windows operating systems - The IBM HTTP Server ships a sample module and project file for building a DLL on Windows operating systems. The configured project file finds header files in the `src\include` directory and resolves references to the `src\Corer\ApacheCore.lib` file. See [Building dynamic modules on Windows operating systems](#) for more information.
- APXS Script (UNIX platform only):

The installation process places APXS, a Perl script, in the IBM HTTP Server `bin` directory. This script builds dynamically shared objects on AIX, HP, Linux, and Solaris operating systems.

To use the APXS script, ensure that you have Perl script V5.003 or later, installed. Ensure the path to the Perl executable on the first line of the APXS script is correct, for example, `/usr/bin/local/perl`. See [Apache APXS](#) for information on using the APXS script on specific platforms. You may have to change this line to accommodate your Perl installation directory.

## [Building third-party modules to run as dynamic shared object modules or dynamic link libraries](#)

If you already have an existing Apache or IBM HTTP Server version installed in the default directory, `/usr/lib/apache` or `/usr/IBMIHS`, third-party module configuration scripts can look for and use parameters based on this existing Apache or IBM HTTP Server directory and these executables. To help avoid this problem, do one of the following:

- Delete your Apache directory.
- Ensure you use third-party configuration script options, `prefix=<installation root>`

where: `prefix` represents the installation root for the IBM HTTP Server.

### Finding related information

- [Using the APXS Script](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)





## Building dynamic modules on Windows operating systems

This section provides information on building dynamic modules on Windows operating systems. This information includes identifying source files, building a module and identifying restrictions. Links to related topics appear at the end of this section.

- [Identifying source files included in the installation](#)
- [Building a module](#)
- [Identifying restrictions](#)
- [Finding related information](#)

To write a module that works with the server for Windows operating systems, use the header files included when you install the IBM HTTP Server. See [Building dynamic shared object modules or dynamic link libraries with the IBM HTTP Server](#) for more detailed information.

### Identifying source files included in the installation

Installing the source code also creates an `src` directory in the server root directory. This directory contains the following directories:

- `include`: Contains the majority of header files used to create the IBM binary distribution of the Web server.
- `os\win32`: Contains platform-specific includes for the Windows platform distribution of the Web server.
- `src\lib`: Contains the libraries. Linking to these libraries enables your module to run with the binary distribution of the IBM HTTP Server.

### Building a module

To build a module:

1. Add the `src\include` directory to your list of include paths.
2. Link with the libraries, found in the `src\lib` directory.

### Identifying restrictions

The following restrictions apply, when building a module to run with the IBM HTTP Server:

1. You must link against the binary libraries installed by the server installation program in the `src\lib` directory.
2. You must use the header files supplied by the server installation program in the `src\include` directory.
3. You cannot modify or add fields to the header files in the `src\include` directory.

## Finding related information

- [Building dynamic shared object modules or dynamic link libraries with the IBM HTTP Server](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



## Configuring the server

You can [get started quickly](#) without making any configuration changes. If you need to encrypt the data being exchanged between the clients and the server, use the Secure Sockets Layer (SSL). You can [get started with secure connections](#) by making only a few configuration changes.

If you run the IBM HTTP Server on the Linux for PowerPC (PPC) operating system, you need to add the **Listen 0.0.0.0:443** directive to the configuration file to enable SSL. If you do not specify this directive, you will receive a PEER\_ID\_NOT\_SET error in the error log when you try to connect to the server.

If you run the IBM HTTP Server on Windows operating systems, you can easily [configure the Fast Response Cache Accelerator](#) to boost performance.

You can also make many other configuration changes with [Apache directives](#).

### Finding related information

- [Getting started quickly with secure connections](#)
  - [Identifying Apache directives supported by the IBM HTTP Server](#)
  - [Identifying Fast Response Cache Accelerator restrictions](#)
  - [Locating glossary terms](#)
  - [Using the Secure Sockets Layer protocol for secure communications](#)
-

IBM  
HTTP Server

WINDOWS

## Setting up the Fast Response Cache Accelerator

This section describes tuning tips to help improve the performance of your server using the Fast Response Cache Accelerator (FRCA), referred to as the *Cache Accelerator*. The Cache Accelerator is available only on Windows operating systems. Not all tips have been tested, nor will all tips have the same impact in all environments. Use your judgment when deciding which tips to implement on your server. Make a backup copy of your server configuration file, before making any modifications. Links to related topics appear at the end of this section.

- [Customizing cache management with the Cache Accelerator](#)
- [Enabling and configuring the Cache Accelerator](#)
- [Customizing logging for the Cache Accelerator](#)
- [Enabling high speed caching of servlets and JavaServer Pages files](#)
- [Finding related information](#)

### Customizing cache management with the Cache Accelerator

The Cache Accelerator can improve the performance of the IBM HTTP Server when serving static content, such as text and image files.

Because the Cache Accelerator cache automatically loads during server operation, you do not have to list the files to cache in your server configuration file. The server automatically caches changed pages again and removes outdated pages from the cache.

The Cache Accelerator supports servers with multiple IP addresses. Currently, support is not available for running the Cache Accelerator on a proxy server.

### Enabling and configuring the Cache Accelerator

By default, the IBM HTTP Server enables the Cache Accelerator. To disable the Cache Accelerator, remove or comment out the [Afpable](#) and [Afpacache](#) directives using the Administration Server.

### Customizing logging for the Cache Accelerator

By default, the Cache Accelerator records that a Web browser has accessed a cached file. If you do not need this access logging, turn the logging off for better

server performance. To set Cache Accelerator logging off, edit the `httpd.conf` configuration file and insert a comment character (`#`) at the beginning of the `AfpaLogFile` line.

## [Enabling high speed caching of servlets and JavaServer Pages files](#)

You can use the IBM HTTP Server Cache Accelerator in conjunction with WebSphere Application Server to cache certain dynamically generated servlet and JavaServer Pages (JSP) files. For details on how to enable this capability, see the External Caching description in the WebSphere Application Server InfoCenter documentation. Follow the instructions for adding the Afpa adapter bean and for configuring cacheable servlets and JSPs.

### Finding related information

- [Identifying Fast Response Cache Accelerator restrictions](#)
- [Locating glossary terms](#)
- [Logging Fast Response Cache Accelerator requests](#)
- [Using AFPA directives](#)

---

[\(Back to the top\)](#)


[\(Back to the top\)](#)



IBM  
HTTP Server



## Using AFPA directives

This section provides information on AFPA directives. These AFPA directives control the Fast Response Cache Accelerator, also referred to as the *Cache Accelerator*. The information includes specific directive syntax, scopes, defaults and associated notes . Links to related topics appear at the end of this section.

- Working with:
  - [AfpaAdvancedTuning \(Windows only\)](#)
  - [AfpaBindLogger \(AIX only\)](#)
  - [AfpaCache \(AIX and Windows only\)](#)
  - [AfpaEnable \(AIX and Windows only\)](#)
  - [AfpaLogFile \(AIX and Windows only\)](#)
  - [AfpaLogging \(AIX only\)](#)
  - [AfpaMaxCache \(AIX only\)](#)
  - [AfpaMinCache \(AIX only\)](#)
  - [AfpaRevalidationTimeout \(AIX only\)](#)
  - [AfpaSendServerHeader \(AIX only\)](#)
    - [Listing of switches and defaults](#)
    - [Detailing switch descriptions](#)
    - [Using optimal settings for typical 1-, 2-, and 4-way machines](#)
- [Finding related information](#)

WINDOWS

## [AfpaAdvancedTuning](#)

- **Syntax** - AfpAdvancedTuning *tuning\_string*
- **Scope** - One per physical Apache server
- **Default** - None
- **Notes** -
  - Valid on Windows only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpAdvancedTuning directive has advanced tuning parameters that require an extensive understanding of Web server performance issues. Using these switches incorrectly could lead to system instability and poor performance. The default settings are considered the optimal settings for the most demanding scenarios. The default settings are aggressive.

### ***Brief description of switches and their defaults:***

- **/bufs** - Number of logging buffers (each log buf is 65536; all log buffers are written every 5 seconds), default = 285
- **/size** - Maximum cached file size, default = 92160
- **/conns** - Number of connection endpoints, default = 6500
- **/ttl** - Time in seconds items are kept hot, default = 180
- **/threads** - Number of worker threads per CPU, default = 3
- **/active** - Maximum worker threads active per CPU, default = 10

### ***Long description of switches:***

- The **"/bufs"** switch specifies the number of logging buffers used by AFPA, for access logging. The buffers are necessary to store access logs until they are written to disk. AFPA writes all log buffers to disk every 5 seconds. The higher the throughput, the more memory required to store log entries. The number of buffers required is roughly equal to  $(\text{tps} * \text{Ls} * 5) / 65536$ , where "tps" is the expected transactions per second and "Ls" is the typical size data necessary to log the transaction. Assuming log entries are no more than 256 bytes and the number of logging buffers is 285 (the default), AFPA could process 14592 transactions per second, without requiring more log buffers.
- The **"/size"** switch specifies the maximum size of files AFPA keeps in its primary cache. Files larger than this size are cached in the AFPA secondary cache. The primary cache is backed by pinned memory.
- The **"/conns"** switch specifies the number of preallocated sockets used by AFPA. AFPA pre-allocates sockets for performance reasons. Choose the number of pre-allocated sockets to approximate the expected transactions per

second.

- The **"/ttl"** switch specifies in seconds the maximum lifetime (time to live) of a file in the AFPA primary cache. When this time elapses, the file is removed from the AFPA primary cache. Setting this value to zero prevents AFPA from aging files from the primary cache.
- The **"/threads"** switch specifies the number of worker threads used by AFPA to process requests.
- The **"/active"** switch specifies the number of worker threads concurrently active and not blocked on I/O.

## Example of Optimized Settings for typical 1, 2, and 4-way machines

- 1-way: AfpAdvanced Tuning **"/bufs 68 /size 75000 /conn 8000"**
- 2-way: AfpAdvancedTuning **"/bufs 93 /size 75000 /conn 10000"**
- 4-way: AfpAdvancedTuning **"/bufs 123 /size 65000 /conn 12000"**



## AfpaBindLogger

- **Description** - AfpaBindLogger [-1,0,1,...,n]
- **Scope** - One per physical Apache server
- **Default** - (-1)
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.
  - This command only binds the kernel logging thread to a processor.

AfpaBindLogger allows you to bind the Fast Response Cache 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.



WINDOWS

## AfpaCache



- **Description** - Turns Fast Response Cache Accelerator on or off
- **Scope** - Server configuration, virtual host, directory, per-directory configuration file
- **Syntax** - On or off
- **Usage** - AfpaCache on
- **Default** - None
- **Override** - Options
- **Multiple instances in the configuration file** - Allowed
- **Notes** -
  - Valid on AIX and Windows.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpaCache directive turns the Fast Response Cache Accelerator on or off for a particular scope (such as a directory). This directive applies to all descendants in a scope, unless otherwise modified by another directive.



WINDOWS

## AfpaEnable

- **Syntax** - AfpaEnable
- **Description** - Enables Fast Response Cache Accelerator
- **Scope** - One per physical Apache server
- **Default** - Fast Response Cache Accelerator disabled
- **Notes** -
  - Valid on AIX and Windows.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpaEnable directive enables the Fast Response Cache Accelerator (Cache Accelerator). If the directive is present, the Cache Accelerator listens on either the TCP port specified by the Port directive, or the default port (80). The Cache Accelerator listens on the ports of all active TCP/IP adapters on the server. If the port is bound to a particular TCP/IP adapter, the Cache Accelerator is disabled.



WINDOWS

## AfpaLogFile

- **Description** - Defines the Cache Accelerator log file name, location, and logging format
- **Scope** - One entry per physical Apache server
- **Values** - *file\_path\_and\_name log\_format*
- **Default** - `/tmp/afpa.logb`
- **Log Formats** -
  - 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
  - BINARY = Binary log with virtual host information (AIX only)
- **Multiple instances in the config file** - Not allowed
- **Notes** -
  - Valid on AIX and Windows.
  - The AFPA directives control the Fast Response Cache Accelerator function.

Defaults to Cache Accelerator logging disabled, if `AfpaLogFile` directive is not present.

The current date is used as the filetype for the log file. The log file is automatically rolled over at midnight each day.



## AfpaLogging

- **Description** - `AfpaLogging` on or off
- **Scope** - One per physical Apache server
- **Values** - On or off
- **Default** - (-1)
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The `AfpaLogging` directive turns the Fast Response Cache Accelerator logging on or off.



## AfpaMaxCache

- **Syntax** - AfpaMaxCache [size]
- **Scope** - One per physical Apache Server
- **Default** - None
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpaMaxCache directive specifies the maximum file size inserted into the Fast Response Cache Accelerator cache.



## AfpaMinCache

- **Syntax** - AfpaMinCache [size]
- **Scope** - One per physical Apache server
- **Default** - None
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpaMinCache directive specifies the minimum file size inserted into the Fast Response Cache Accelerator cache.



## AfpaRevalidationTimeout

- **Syntax** - AfpRevalidationTimeout <seconds>
- **Scope** - Global
- **Default** - 60
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

AfpRevalidationTimeout sets the time interval for revalidation of a cached object. Once an object is cached in the kernel after the time interval has expired, the kernel forces the next request of the object to set up the Apache Server for revalidation. The time interval is expressed in seconds.



## AfpSendServerHeader

- **Syntax** - AfpSendServerHeader true or false
- **Scope** - One per physical Apache Server
- **Default** - True
- **Notes** -
  - Valid on AIX only.
  - The AFPA directives control the Fast Response Cache Accelerator function.

The AfpSendServerHeader directive specifies whether or not the Fast Response Cache Accelerator sends the HTTP Server header in the response.

### Related information...

- [Enable session ID caching](#)
- [Fast Response Cache Accelerator](#)
- [Set up the Cache Accelerator access log](#)
- [Tune and manage your server](#)



## Identifying Fast Response Cache Accelerator restrictions

This section discusses the caching and operational restrictions for the Fast Response Cache Accelerator, sometimes referred to as the *Cache Accelerator*. Links to related topics appear at the end of this section.

- [Understanding caching restrictions](#)
- [Identifying operational restrictions](#)
- [Finding related information](#)

### Understanding caching restrictions

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:

- The Cache Accelerator supports only limited multilanguage content negotiation. Caching occurs for only a single language version, where a given URL maps to multiple translated versions.
- The Cache Accelerator listens on all the IP adapters of a server, on the port specified by the [AfpaPort](#) directive. You cannot configure the Cache Accelerator to listen on some IP adapters, but not others.
- The Cache Accelerator does not cache files on, or log to locally mounted remote file systems, like the Network File System (NFS).
- The Cache Accelerator does not support IP-based virtual host caching. Disable the Cache Accelerator for these directories. Use the [AfpaCache off](#) directive in any IP-based virtual host definitions.

### Identifying operational restrictions

The following operational restrictions apply:

- The IBM HTTP Server Cache Accelerator does not run as a proxy server.
- The IBM HTTP Server access log does not integrate the Cache Accelerator access log with the Apache access log.

- When you enable the Cache Accelerator, set the [MaxRequestsPerChild](#) directive to 0, to work around a problem that occurs when the child process stops and restarts on a loaded Web server.
- On a given machine, only one active instance of the server can have the Cache Accelerator enabled.
- Do not install the IBM HTTP Server on a machine running the IBM Netfinity Web Server Accelerator.
- Only access logging facilities exist for monitoring the Cache Accelerator.

 Support exists for name-based virtual hosts.

### Finding related information

- [Locating glossary terms](#)
- [Logging Fast Response Cache Accelerator requests](#)
- [Setting up the Fast Response Cache Accelerator](#)
- [Using the AfpCache directive](#)
- [Using the AfpEnable directive](#)
- [Using the AfpLogFile directive](#)

---

[\(Back to the top\)](#)



## Logging Fast Response Cache Accelerator requests

 The Fast Response Cache Accelerator (FRCA), also known as the *Cache Accelerator*, is supported on AIX and Windows operating systems.

This section provides information on logging FRCA requests. A general overview discusses the Cache Accelerator access log path and name. Links to related topics appear at the end of this section.

- [Understanding the Cache Accelerator access log](#)
- [Finding related information](#)

### Understanding the Cache Accelerator access log

The IBM HTTP Server can optionally create a log file that records requests served by the Cache Accelerator.

Enable the Cache Accelerator access log if you want to maintain a record of requests served by the Cache Accelerator. Use the normal Apache logging directives to log requests that are not served by the Cache Accelerator to separate log files.

This log provides a useful way to verify that caching is enabled and to identify cached files.

Even though you can cache a particular file, it might not always be served from the cache.

The log file has a date stamp automatically appended to its name. Every day 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.

For each request served out of the Cache Accelerator, a log entry in the access log shows the:

- Source host address
- Date and time of the request
- HTTP method of the request and what was requested
- HTTP return code, which indicates whether the request was honored

- Size of the returned data

A log entry can also optionally show the:

- Target virtual host (use formatting option V-CLF or V-ECLF)
- Referrer (use formatting option ECLF or V-ECLF)
- User agent (use formatting option ECLF or V-ECLF)

## Finding related information

- [Locating glossary terms](#)
- [Setting up the Fast Response Cache Accelerator](#)
- [Using the AfpalLogFile directive](#)

---

[\(Back to the top\)](#)





## Identifying Apache directives supported by the IBM HTTP Server

A list of the Apache directives supported by the IBM HTTP Server follows. A link to [related information](#) regarding Apache directives appears at the bottom of this file.

- [AcceptMutex](#)
- [AcceptPathInfo](#)
- [AccessFileName](#)
- [Action](#)
- [AddAlt](#)
- [AddAltByEncoding](#)
- [AddAltByType](#)
- [AddCharset](#)
- [AddDefaultCharset](#)
- [AddDescription](#)
- [AddEncoding](#)
- [AddHandler](#)
- [AddIcon](#)
- [AddIconByEncoding](#)
- [AddIconByType](#)
- [AddInputFilter](#)
- [AddLanguage](#)
- [AddModuleInfo](#)
- [AddOutputFilter](#)
- [AddType](#)
- [Alias](#)
- [AliasMatch](#)
- [allow](#)
- [AllowCONNECT](#)
- [AllowOverride](#)
- [Anonymous](#)

- [Anonymous\\_Authoritative](#)
- [Anonymous\\_LogEmail](#)
- [Anonymous\\_MustGiveEmail](#)
- [Anonymous\\_NoUserID](#)
- [Anonymous\\_VerifyEmail](#)
- [AuthAuthoritative](#)
- [AuthDBMAuthoritative](#)
- [AuthDBMGroupFile](#)
- [AuthDBMType](#)
- [AuthDBMUserFile](#)
- [AuthGroupFile](#)
- [AuthName](#)
- [AuthType](#)
- [AuthUserFile](#)
- [BrowserMatch](#)
- [BrowserMatchNoCase](#)
- [CacheNegotiatedDocs](#)
- [CheckSpelling](#)
- [ContentDigest](#)
- [CookieDomain](#)
- [CookieExpires](#)
- [CookieLog](#)
- [CookieName](#)
- [CookieStyle](#)
- [CookieTracking](#)
- [CoreDumpDirectory](#)
- [CustomLog](#)
- [Dav](#)
- [DavDepthInfinity](#)
- [DavLockDB](#)
- [DavMinTimeout](#)
- [DefaultIcon](#)
- [DefaultLanguage](#)

- [DefaultType](#)
- [DeflateBufferSize](#)
- [DeflateFilterNote](#)
- [DeflateMemLevel](#)
- [DeflateWindowSize](#)
- [deny](#)
- [<Directory>](#)
- [<DirectoryMatch>](#)
- [DirectoryIndex](#)
- [DocumentRoot](#)
- [EnableMMAP](#)
- [ErrorDocument](#)
- [ErrorLog](#)
- [ExpiresActive](#)
- [ExpiresByType](#)
- [ExpiresDefault](#)
- [ExtendedStatus](#)
- [FileETag](#)
- [<Files>](#)
- [<FilesMatch>](#)
- [ForceLanguagePriority](#)
- [ForceType](#)
- [Group](#)
- [Header](#)
- [HeaderName](#)
- [HostnameLookups](#)
- [IdentityCheck](#)
- [<IfDefine>](#)
- [<IfModule>](#)
- [ImapBase](#)
- [ImapDefault](#)
- [ImapMenu](#)
- [Include](#)

- [IndexIgnore](#)
- [IndexOptions](#)
- [IndexOrderDefault](#)
- [KeepAlive](#)
- [KeepAliveTimeout](#)
- [LanguagePriority](#)
- [<Limit>](#)
- [<LimitExcept>](#)
- [LimitRequestBody](#)
- [LimitRequestFields](#)
- [LimitRequestFieldsize](#)
- [LimitRequestLine](#)
- [LimitXMLRequestBody](#)
- [Listen](#)
- [ListenBacklog](#)
- [LoadFile](#)
- [LoadModule](#)
- [<Location>](#)
- [<LocationMatch>](#)
- [LockFile](#)
- [LogFormat](#)
- [LogLevel](#)
- [MaxClients](#)
- [MaxKeepAliveRequests](#)
- [MaxRequestsPerChild](#)
- [MaxSpareServers](#)
- [MaxSpareThreads](#)
- [MetaDir](#)
- [MetaFiles](#)
- [MetaSuffix](#)
- [MimeMagicFile](#)
- [MinSpareServers](#)
- [MinSpareThreads](#)

- [MultiViewsMatch](#)
- [NameVirtualHost](#)
- [NoProxy](#)
- [Options](#)
- [order](#)
- [PassEnv](#)
- [PidFile](#)
- [Proxy](#)
- [ProxyBlock](#)
- [ProxyDomain](#)
- [ProxyErrorOverride](#)
- [ProxyIOBufferSize](#)
- [ProxyMatch](#)
- [ProxyMatch](#)
- [ProxyMaxForwards](#)
- [ProxyPass](#)
- [ProxyPassReverse](#)
- [ProxyPreserveHost](#)
- [ProxyReceiveBufferSize](#)
- [ProxyRemote](#)
- [ProxyRemoteMatch](#)
- [ProxyRequests](#)
- [ProxyTimeout](#)
- [ProxyVia](#)
- [ReadmeName](#)
- [Redirect](#)
- [RedirectMatch](#)
- [RedirectPermanent](#)
- [RedirectTemp](#)
- [RemoveCharset](#)
- [RemoveEncoding](#)
- [RemoveHandler](#)
- [RemoveInputFilter](#)

- [RemoveLanguage](#)
- [RemoveOutputFilter](#)
- [RemoveType](#)
- [RequestHeader](#)
- [require](#)
- [RewriteBase](#)
- [RewriteCond](#)
- [RewriteEngine](#)
- [RewriteLock](#)
- [RewriteLog](#)
- [RewriteLogLevel](#)
- [RewriteMap](#)
- [RewriteOptions](#)
- [RewriteRule](#)
- [RLimitCPU](#)
- [RLimitMEM](#)
- [RLimitNPROC](#)
- [Satisfy](#)
- [ScoreBoardFile](#)
- [Script](#)
- [ScriptAlias](#)
- [ScriptAliasMatch](#)
- [ScriptInterpreterSource](#)
- [ScriptLog](#)
- [ScriptLogBuffer](#)
- [ScriptLogLength](#)
- [ScriptSock](#)
- [SendBufferSize](#)
- [ServerAdmin](#)
- [ServerAlias](#)
- [ServerLimit](#)
- [ServerName](#)
- [ServerPath](#)

- [ServerRoot](#)
- [ServerSignature](#)
- [ServerTokens](#)
- [SetEnv](#)
- [SetEnvIf](#)
- [SetEnvIfNoCase](#)
- [SetHandler](#)
- [SetInputFilter](#)
- [SetOutputFilter](#)
- [SSIEndTag](#)
- [SSIErrorMsg](#)
- [SSIStartTag](#)
- [SSITimeFormat](#)
- [SSIUndefinedEcho](#)
- [StartServers](#)
- [SuexecUserGroup](#)
- [ThreadLimit](#)
- [ThreadsPerChild](#)
- [TimeOut](#)
- [TransferLog](#)
- [TypesConfig](#)
- [UnsetEnv](#)
- [UseCanonicalName](#)
- [User](#)
- [UserDir](#)
- [VirtualDocumentRoot](#)
- [VirtualDocumentRootIP](#)
- [<VirtualHost>](#)
- [VirtualScriptAlias](#)
- [VirtualScriptAliasIP](#)
- [XBitHack](#)

## Finding related information

- [Locating glossary terms](#)
- [Using Apache directives](#)

---

[\(Back to the top\)](#)





## Looking at the Apache V2.0 process model in the IBM HTTP Server

- [Controlling the number of IBM HTTP Server processes](#)
- [Affecting availability of IBM HTTP Server processes to handle requests using the KeepAlive feature](#)
- [Setting expiration dates on static content](#)
- [Finding related information](#)

When the Apache *parent* process starts, it forks a number of child processes. Each of these child processes creates a number of threads which are responsible for accepting connections off of the listening sockets. When the system receives a connection, the system wakes up one of the threads to handle the connection. Each Apache thread can handle one connection.

For example, Apache needs 1000 threads to handle 1000 concurrently connected clients, or connections.

### [Controlling the number of IBM HTTP Server threads](#)

You can use several configuration directives to control the number of IBM HTTP Server threads, or the number of concurrently supported clients. The most important directives follow:

- [StartServers](#)  
Specifies the initial number of child processes to start when the server starts. Apache automatically increases the number of child processes as server load increases. The maximum number of child processes is the setting for MaxClients, divided by the setting for ThreadsPerChild.
- [MaxClients](#)  
Specifies the *maximum* number of Apache processes that can run at once. Limiting the number of Apache processes can prevent overrunning your hardware capabilities. Some installations will need to use the ServerLimit directive to allow a large setting for MaxClients.
- [ThreadsPerChild](#)  
Specifies the number of threads that will be created in each child process. Some installations will need to use the ThreadLimit directive to allow a large setting for ThreadsPerChild.

## Affecting availability of IBM HTTP Server processes to handle requests using the KeepAlive feature

HTTP V1.1 has a feature known as *Connection KeepAlive*. In a non-KeepAlive connection, the browser starts a TCP connection and sends a single request to the server. The server responds and then takes down the connection. Each request includes the bring-up and take down of a TCP connection. The KeepAlive feature maintains connections for a number of requests, controlled by configuration directives. This control reduces network overhead related to startup and tear down of TCP connections. The server assumes that for a KeepAlive connection, the browser sends another request. The server attempts to read the next request off the network. When the next request becomes unavailable, the process blocks on the read, waiting for the next request.

Consider a typical scenario where a user goes to a Web site resulting in the fetching and rendering of an information page on the browser. The user lingers while reading the page. The user does not actively request more pages, but the server blocks on a network read waiting for the next request. This blocked process becomes unavailable to handle requests from other clients. The user can follow a link off the Web site, get a cup of coffee and never send another request. You can configure the IBM HTTP Server to wait a specific time for the next request using:

- [KeepAliveTimeout](#)  
The default equals 15 seconds. This setting means the server stops waiting for the next request, shuts down the connection to the inactive client and makes the process available to handle other requests.

Some heavily loaded sites disable the [KeepAlive](#) feature entirely.

Assess the needs of your site and set `KeepAliveTimeout` appropriately. Give the browser enough time to request all the elements of a page over the KeepAlive connection, but not wait too long for the user to initiate the next page request. Some recommend setting `KeepAliveTimeout` to 5 seconds as a good compromise between balancing available processes with minimizing network I/O.

## Setting expiration dates on static content

You can reduce connection requests to your site by setting document expiration dates with `mod_expires`. If the browser has cached a previous visit, this action can save another connection request to your site.


### Finding related information

- [Locating glossary terms](#)
- [Identifying Apache directives supported by the IBM HTTP Server](#)

[\(Back to the top\)](#)



## Using Fast Common Gateway Interface directives

This section provides information on Fast Common Gateway Interface (FastCGI) directives and examples for setting up FastCGI. The information includes specific directive syntax, scopes, defaults and associated notes . Links to related topics appear at the end of this section.

- [Samples for Setting Up FastCGI](#)
- Working with the FastCGI directives:
  - [FastCgiAccessChecker](#)
  - [FastCgiAccessCheckerAuthoritative](#)
  - [FastCgiAuthenticator](#)
  - [FastCgiAuthenticatorAuthoritative](#)
  - [FastCgiAuthorizer](#)
  - [FastCgiAuthorizerAuthoritative](#)
  - [FastCgiConfig](#)
  - [FastCgiExternalServer](#)
  - [FastCgilpcDir](#)
  - [FastCgiServer](#)
  - [FastCgiSuexec](#)
- [Finding related information](#)

## [Samples for Setting Up FastCGI](#)

### **For Windows:**

```
LoadModule fastcgi_module modules/mod_fastcgi.dll
```

```
ScriptAlias /fcgi-bin/ "c:/Program Files/IBM HTTP Server2.0/fcgi-bin/"
FastCGIConfig -autoUpdate
```

```
<Directory "c:/Program Files/IBM HTTP Server2.0/fcgi-bin">
    AllowOverride None
    Options +ExecCGI
    SetHandler fastcgi-script
</Directory>
```

```
FastCGIServer "c:/Program Files/IBM HTTP Server2.0/fcgi-bin/echo" -processes 1
```

In order to refresh a running FastCGI application, use the Task Manager to end the process.

### **For AIX, Solaris, Linux, Linux/intel, Linux/390, Linux/PPC, and HP:**

```
<IfModule mod_fastcgi.c>
```

```
LoadModule fastcgi_module libexec/mod_fastcgi.so
AddModule mod_fastcgi.c
```

```
ScriptAlias /fcgi-bin/ /usr/HTTPServer/fcgi-bin/
```

```
FastCGIConfig -autoUpdate
```

Add the following lines to the config file (note that you may have to create the path):

```
<Directory /usr/HTTPServer/fcgi-bin>
AllowOverride None
Options None
SetHandler fastcgi-script
</Directory>
```

```
FastCgiServer /usr/HTTPServer/fcgi-bin/hello -processes 1
```

```
</IfModule>
```

In order to refresh a running FastCGI application, use the **-f** option on the **cp** command. For example:

```
cp -f echo.new echo.old
```

## FastCgiAccessChecker

- Description: Defines a FastCGI application as a per-directory access validator.
- Default: Directory
- Module: mod\_fastcgi
- Multiple instances in the configuration file: Yes
- Scope: Directory, Location
- Syntax: `FastCgiAccessChecker file name [-compat]`
- Values: File name

The Apache Access phase precedes user authentication and the HTTP headers submitted with the request determine the decision to enable access to the requested resource. Use FastCGI-based authorizers when a dynamic component exists as part of the access validation decision, like the time, or the status of a domain account.

If the FastCGI application file name does not have a corresponding static or external server definition, the application starts as a dynamic FastCGI application. If the file name does not begin with a slash (/), then the application assumes that the file name is relative to the [ServerRoot](#).

Use the `FastCgiAccessChecker` directive within [Directory](#) or [Location](#) containers. For example:

```
<Directory htdocs/protected>
FastCgiAccessChecker fcgi-bin/access-checker
</Directory>
```

`Mod_fastcgi` sends nearly all of the standard environment variables typically available to CGI and FastCGI request handlers. All headers returned by a FastCGI access-checker application in a successful response (Status: 200), pass to subprocesses, or CGI and FastCGI invocations, as environment variables. All headers returned in an unsuccessful response pass to the client. Obtain FastCGI specification compliant behavior by using the `-compat` option.

`Mod_fastcgi` sets the environment variable `FCGI_APACHE_ROLE` to `ACCESS_CHECKER`, to indicate the Apache-specific authorizer phase performed.

The HTTP Server does not support custom failure responses from FastCGI authorizer applications. See the [ErrorDocument](#) directive for a workaround. A FastCGI application can serve the document.

## FastCgiAccessCheckerAuthoritative

- Description: Enables access checking passing to lower level modules.

- **Default:** `FastCgiAccessCheckerAuthoritative On`
- **Module:** `mod_fastcgi`
- **Multiple instances in the configuration file:** Yes
- **Scope:** Directory
- **Syntax:** `FastCgiAccessCheckerAuthoritative On | Off`
- **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

- **Description:** Defines a FastCGI application as a per-directory authenticator.
- **Default:** None
- **Module:** `mod_fastcgi`
- **Multiple instances in the configuration file:** Yes
- **Scope:** Directory
- **Syntax:** `FastCgiAuthenticator file name [-compat]`
- **Values:** File name

Authenticators verify the requester, by matching the provided user name and password against a list, or database of known users and passwords. Use FastCGI based authenticators when the user database gets 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 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 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 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 `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

- Description: 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.
- Default: `FastCgiAuthenticatorAuthoritative On`
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Directory
- Syntax: `FastCgiAuthenticatorAuthoritative On | Off`
- 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

- Description: Defines a FastCGI application as a per-directory authorizer
- Default: None
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Directory
- Syntax: `FastCgiAuthorizer file name [-compat]`
- 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](#)

- Description: 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.
- Default: `FastCgiAuthorizerAuthoritative On`
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Directory
- Syntax: `FastCgiAuthorizerAuthoritative On | Off`
- 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](#)

- Description: Defines the default parameters for all dynamic FastCGI applications.
- Default: None
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Server configuration
- Syntax: `FastCgiConfig option option ...`

The `FastCgiConfig` directive does not affect static or external applications.

- 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  $(1 - \text{gainValue})$ , so making values smaller, weights them more heavily compared to the current value, which is scaled by `gainValue`.

**initial-env *name[=value]* none**

A name-value pair passed in the initial environment when instances of the application spawn. To pass a variable from the Apache environment, do not provide the "=" (if the variable is not actually in the environment, it is defined without a value). To define a variable without a value, provide the "=" without any value. This option is repeatable.

**init-start-delay *n* (1 second)**

The minimum number of seconds between the spawning of instances of this application. This delay decreases the demand placed on the system at server initialization.

**killInterval *n* (300 seconds)**

The `killInterval` determines how often the dynamic application instance killing policy is implemented within the process manager. Lower numbers result in a more aggressive policy, while higher numbers result in a less aggressive policy.

**listen-queue-depth *n* (100)**

The depth of the `listen()` queue, also known as the *backlog*, shared by all instances of this application. A deeper listen queue allows the server to cope with transient load fluctuations without rejecting requests; it does not increase throughput. Adding additional application instances can increase throughput and performance, depending upon the application and the host.

**maxClassProcesses *n* (10)**

The maximum number of dynamic FastCGI application instances allowed to run for any one FastCGI application.

**maxProcesses *n* (50)**

The maximum number of dynamic FastCGI application instances allowed to run at any time.

**minProcesses *n* (5)**

The minimum number of dynamic FastCGI application instances the process manager allows to run at any time, without killing them due to lack of demand.

**multiThreshold *n* (50)**

An integer between 0 and 100 used to determine whether to terminate any instance of a FastCGI application. If the application has more than one instance currently running, this attribute helps to decide whether to terminate one of them. If only one instance remains, `singleThreshold` is used instead.

**pass-header *header* none**

The name of an HTTP Request Header passed in the `request` environment. This option makes the contents of headers available to a CGI environment.

**priority *n* (0)**

The process priority assigned to the application instances using `setpriority()`.

**processSlack *n* (5 seconds)**

If the sum of all currently running dynamic FastCGI applications exceeds `maxProcesses - processSlack`, the process manager invokes the killing policy. This action improves performance at higher loads, by killing some of the most inactive application instances before reaching the `maxProcesses` value.

**restart *none***

This option causes the process manager to restart dynamic applications upon failure, similar to static applications.

**Restart-delay *n* (5 seconds)**

The minimum number of seconds between the respawning of failed instances of this application. This delay prevents a broken application from soaking up too much of the system.

**singleThreshold *n* (0)**

An integer between 0 and 100, used to determine whether the last instance of a FastCGI application can terminate. If the process manager computed load factor for the application is lower than the specified threshold, the last instance is terminated. Specify a value closer to 1, to make your executables run in the idle mode for a long time. If memory or CPU time is a concern, a value closer to 100 is more applicable. A value of 0, prevents the last instance of an application from terminating; this value is the default. Changing this default is not recommended, especially if you set the `-appConnTimeout` option.

**startDelay *n* (3 seconds)**

The number of seconds the Web server waits while trying to connect to a dynamic FastCGI application. If the interval expires, the process manager is notified with hope that another instance of the application starts. Set the `startDelay` value smaller than the `appConnTimeout` value, to be effective.

**updateInterval *n* (300 seconds)**

The `updateInterval` decides how often statistical analysis is performed to determine the fate of dynamic FastCGI applications.

## FastCgiExternalServer

- Description: Defines file name as an external FastCGI application. Operates the same as the [Fastcgiserver](#) directive, except that the CGI application is running in another process, outside the Web server.
- Default: None
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Server configuration
- Syntax: `FastCgiExternalServer file name -host hostnameport [-appConnTimeout n]`  
`FastCgiExternalServer file name -socket file name [-appConnTimeout n]`
- 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()`. 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 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***

UNIX

**UNIX platform:** 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.

WINDOWS

**Windows operating systems:** The name of the pipe the application uses for communicating with the Web server. The name is relative to the `FastCgilpcDir` option. The `-socket` and `-port` options are mutually exclusive.

## FastCgilpcDir

UNIX

WINDOWS

- Description: Specifies directory as the place to store the UNIX socket files used for communication between the applications and the Web server.
- Default:None
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Server configuration
- Syntax:
  - UNIX platform: `FastCgiIpcDir` *directory*
  - Windows operating systems: `FastCgilpcDir` *name*
- Values: *directory* or *name*

UNIX

**UNIX platform:** The `FastCgiIpcDir` directive specifies *directory* as the place to store and find, in the case of external FastCGI applications, the UNIX socket files used for communication between the applications and the Web server. If the directory does not begin with a slash (/) then it is assumed 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

**Windows operating systems:** The `FastCgiIpcDir` directive specifies *name* as the root for the named pipes used for communication between the application and the Web server. Put the *name* in the form `>\.\pipe\pipename`. The *pipename* part 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

- Description: 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`.
- Default:None
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Server configuration
- Syntax:`FastCgiServer file name [options]`
- Values:

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 (gen'd)***

**UNIX platform:** 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.

**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 `FastCgilpcDir` 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](#).

## FastCgiSuexec

- Description: Supports the suexec-wrapper
- Default: `FastCgiSuexec Off`
- Module: `mod_fastcgi`
- Multiple instances in the configuration file: Yes
- Scope: Server configuration
- Syntax: `FastCgiSuexec On | Off | file name`
- 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.

### Finding related information

- [Working with the Fast Common Gateway Interface protocol](#)
- [Locating glossary terms](#)

---

[\(Back to the top\)](#)



---

## Working with the Fast Common Gateway Interface protocol

This section provides an information overview of the Fast Common Gateway Interface (FastCGI) protocol, FastCGI applications and the FastCGI Web site. Links to related topics appear at the end of this section.

- [Learning about the Fast Common Gateway Interface protocol](#)
- [Using Fast Common Gateway Interface applications](#)
- [Sending mail to the Fast Common Gateway Interface Web site](#)
- [Finding related information](#)

### [Learning about the Fast Common Gateway Interface protocol](#)

This third-party module provides support for the FastCGI protocol. FastCGI, a language independent, scalable, open extension to Common Gateway Interface (CGI), provides high performance and persistence without the limitations of server-specific APIs.

This open protocol does not limit FastCGI applications to a particular development language. FastCGI application libraries currently exist for Perl, C/C++, Java, Python and the transmission control layer (TCL).

### [Using Fast Common Gateway Interface 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.

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.

### [Sending mail to the Fast Common Gateway Interface Web site](#)

For more information on FastCGI, go to the [FastCGI Web site](#). To receive FastCGI related announcements and notifications of module updates, send mail to [fastcgi-announce-request@idle.com](mailto: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](mailto: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.

### Finding related information

- [Locating glossary terms](#)
- [Using Fast Common Gateway Interface directives](#)
- [Visiting the Fast Common Gateway Interface Web site.](#)

---

[\(Back to the top\)](#)



IBM  
HTTP Server

WINDOWS

## Working with the Windows performance monitor

The IBM HTTP Server includes hooks for the Windows performance monitor. These hooks enable the Windows performance monitor to observe the current state of an active IBM HTTP Server.

### Viewing Server Statistics

To view server statistics through the Windows performance monitor, include the `LoadModule` directive corresponding to the module, `status_module`, in the `httpd.conf` file. To view statistics, load this module.

To view all possible statistics, include the `ExtendedStatus` directive, set to **On**. Without the `ExtendedStatus` directive, you can only see a few statistics.

If you want to view statistics for the Administration Server, include the `LoadModule status_module` and the `ExtendedStatus` directive, set to **On**, in the `admin.conf` file.

### Finding related information

- [Locating glossary terms](#)
- [Finding the default and sample configuration files](#)



## **IBM Notices**

- [Trademarks](#)
- [Notices](#)
- [Legal Web page](#)

### **Trademarks**

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX

IBM

VisualAge

WebSphere

Domino Go Webserver is a trademark of Lotus Development Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both.

Microsoft, Windows, and the Windows logo 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.

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

### **Notices**

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area.

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. Furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country, or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

---

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

---

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

---

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been

estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

---

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, International Programming License Agreement, or any equivalent agreement between us.

---

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

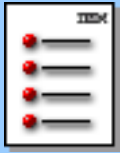
---

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

---

#### COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.



## **IBM HTTP Server**

### **Overview**

### **How to**

[Install](#)

[National Language Support](#)

**Get started**

[With IBM HTTP Server](#)

[With LDAP](#)

[With secure connections](#)

[Setting up advanced security](#)

[Without secure connections](#)

[Build DSO modules or DLLs](#)

**Build dynamic modules**

[Windows operating systems](#)

[Configure the server](#)

[Enable client authentication](#)

[Set and view cipher specs](#)

[Set up password protection](#)

[Set up Cache Accelerator \(AFPA\)](#)

[Log Cache Accelerator requests](#)

[Troubleshoot](#)

[Use IKEYMAN](#)

### **Tell me about**

[Apache Performance Tuning for UNIX](#)

[apachectl utility](#)

[Cache Accelerator](#)

[Cache Accelerator restrictions](#)

[CA software](#)

[Certificate authorities](#)

[Certificates, self-signed](#)

[Cipher specs](#)

[Client authentication](#)

[Configuration files](#)

**Directives**

[AFPA](#)

[Apache directives](#)

[FastCGI](#)

[LDAP](#)

[SSL](#)

**IKEYMAN**

[Key sizes](#)

[LDAP](#)

[Protection options](#)

[Session ID caching \(UNIX only\)](#)

**SSL**

[Defining for multiple-IP virtual hosts](#)

[Enabling CRL in SSL](#)

[Enabling crypto devices for SSL](#)

[Using SSL password prompting](#)

[SSL environment variables](#)

**Starting and stopping the server**

[on UNIX, with apachectl](#)

[Windows](#)

**Reference links**

[Acronym list](#)

[Apache Documentation](#)

[Glossary](#)

**Notices and trademarks**

## IBM HTTP Server™ Version 2.0.47



### The IBM HTTP Server

Become acquainted with the product features.



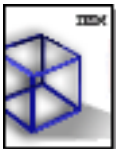
### Installation

Follow the steps for installation on your platform. Learn the basics for National Language Support enablement.



### System administration

Learn how to configure the product, and other components in your environment that interact with the product.



### IBM HTTP Server concepts

Introduce yourself to the IBM HTTP Server. Learn the basics for getting started quickly.



### Programming

Understand the requirements for building dynamic shared object modules (DSOs) or dynamic link libraries (DLLs) with the IBM HTTP Server.



### Troubleshooting

Diagnose and fix problems quickly.



## **Setting protection for server resources**

To set password protection for server resources:

1. Define password files.
2. Set up authentication rules.

### ***Defining password files***

A password file contains user names and passwords.

1. From the Administration Server, click **Authentication Files > Individuals**.
2. Define users.
3. Click **Submit**.
4. Restart the server.

### ***Setting up authentication rules***

See the documentation in the Apache User's Guide for the [mod\\_auth module](#).

#### **Finding related information**

- [Choosing protection options](#)
- [Getting started quickly with secure connections](#)
- [Locating glossary terms](#)