| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SSLServerSocket.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/net/ssl/SSLProtocolException.html)   [**NEXT CLASS**](http://docs.google.com/javax/net/ssl/SSLServerSocketFactory.html) | [**FRAMES**](http://docs.google.com/index.html?javax/net/ssl/SSLServerSocket.html)    [**NO FRAMES**](http://docs.google.com/SSLServerSocket.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#26in1rg) |

## **javax.net.ssl**

Class SSLServerSocket

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.net.ServerSocket](http://docs.google.com/java/net/ServerSocket.html)  
 **javax.net.ssl.SSLServerSocket**

public abstract class **SSLServerSocket**extends [ServerSocket](http://docs.google.com/java/net/ServerSocket.html)

This class extends ServerSockets and provides secure server sockets using protocols such as the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols.

Instances of this class are generally created using a SSLServerSocketFactory. The primary function of SSLServerSockets is to create SSLSockets by accepting connections.

SSLServerSockets contain several pieces of state data which are inherited by the SSLSocket at socket creation. These include the enabled cipher suites and protocols, whether client authentication is necessary, and whether created sockets should begin handshaking in client or server mode. The state inherited by the created SSLSocket can be overriden by calling the appropriate methods.

**Since:** 1.4 **See Also:**[ServerSocket](http://docs.google.com/java/net/ServerSocket.html), [SSLSocket](http://docs.google.com/javax/net/ssl/SSLSocket.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**SSLServerSocket**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#SSLServerSocket())()            Used only by subclasses. |
| protected | [**SSLServerSocket**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#SSLServerSocket(int))(int port)            Used only by subclasses. |
| protected | [**SSLServerSocket**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#SSLServerSocket(int,%20int))(int port, int backlog)            Used only by subclasses. |
| protected | [**SSLServerSocket**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#SSLServerSocket(int,%20int,%20java.net.InetAddress))(int port, int backlog, [InetAddress](http://docs.google.com/java/net/InetAddress.html) address)            Used only by subclasses. |

| **Method Summary** | |
| --- | --- |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getEnabledCipherSuites**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledCipherSuites())()            Returns the list of cipher suites which are currently enabled for use by newly accepted connections. |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getEnabledProtocols**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledProtocols())()            Returns the names of the protocols which are currently enabled for use by the newly accepted connections. |
| abstract  boolean | [**getEnableSessionCreation**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnableSessionCreation())()            Returns true if new SSL sessions may be established by the sockets which are created from this server socket. |
| abstract  boolean | [**getNeedClientAuth**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getNeedClientAuth())()            Returns true if client authentication will be *required* on newly accepted server-mode SSLSockets. |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getSupportedCipherSuites**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedCipherSuites())()            Returns the names of the cipher suites which could be enabled for use on an SSL connection. |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getSupportedProtocols**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedProtocols())()            Returns the names of the protocols which could be enabled for use. |
| abstract  boolean | [**getUseClientMode**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getUseClientMode())()            Returns true if accepted connections will be in SSL client mode. |
| abstract  boolean | [**getWantClientAuth**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getWantClientAuth())()            Returns true if client authentication will be *requested* on newly accepted server-mode connections. |
| abstract  void | [**setEnabledCipherSuites**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledCipherSuites(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] suites)            Sets the cipher suites enabled for use by accepted connections. |
| abstract  void | [**setEnabledProtocols**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledProtocols(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] protocols)            Controls which particular protocols are enabled for use by accepted connections. |
| abstract  void | [**setEnableSessionCreation**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnableSessionCreation(boolean))(boolean flag)            Controls whether new SSL sessions may be established by the sockets which are created from this server socket. |
| abstract  void | [**setNeedClientAuth**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean))(boolean need)            Controls whether accepted server-mode SSLSockets will be initially configured to *require* client authentication. |
| abstract  void | [**setUseClientMode**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))(boolean mode)            Controls whether accepted connections are in the (default) SSL server mode, or the SSL client mode. |
| abstract  void | [**setWantClientAuth**](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean))(boolean want)            Controls whether accepted server-mode SSLSockets will be initially configured to *request* client authentication. |

| **Methods inherited from class java.net.**[**ServerSocket**](http://docs.google.com/java/net/ServerSocket.html) |
| --- |
| [accept](http://docs.google.com/java/net/ServerSocket.html#accept()), [bind](http://docs.google.com/java/net/ServerSocket.html#bind(java.net.SocketAddress)), [bind](http://docs.google.com/java/net/ServerSocket.html#bind(java.net.SocketAddress,%20int)), [close](http://docs.google.com/java/net/ServerSocket.html#close()), [getChannel](http://docs.google.com/java/net/ServerSocket.html#getChannel()), [getInetAddress](http://docs.google.com/java/net/ServerSocket.html#getInetAddress()), [getLocalPort](http://docs.google.com/java/net/ServerSocket.html#getLocalPort()), [getLocalSocketAddress](http://docs.google.com/java/net/ServerSocket.html#getLocalSocketAddress()), [getReceiveBufferSize](http://docs.google.com/java/net/ServerSocket.html#getReceiveBufferSize()), [getReuseAddress](http://docs.google.com/java/net/ServerSocket.html#getReuseAddress()), [getSoTimeout](http://docs.google.com/java/net/ServerSocket.html#getSoTimeout()), [implAccept](http://docs.google.com/java/net/ServerSocket.html#implAccept(java.net.Socket)), [isBound](http://docs.google.com/java/net/ServerSocket.html#isBound()), [isClosed](http://docs.google.com/java/net/ServerSocket.html#isClosed()), [setPerformancePreferences](http://docs.google.com/java/net/ServerSocket.html#setPerformancePreferences(int,%20int,%20int)), [setReceiveBufferSize](http://docs.google.com/java/net/ServerSocket.html#setReceiveBufferSize(int)), [setReuseAddress](http://docs.google.com/java/net/ServerSocket.html#setReuseAddress(boolean)), [setSocketFactory](http://docs.google.com/java/net/ServerSocket.html#setSocketFactory(java.net.SocketImplFactory)), [setSoTimeout](http://docs.google.com/java/net/ServerSocket.html#setSoTimeout(int)), [toString](http://docs.google.com/java/net/ServerSocket.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### SSLServerSocket

protected **SSLServerSocket**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Used only by subclasses.

Create an unbound TCP server socket using the default authentication context.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket

### SSLServerSocket

protected **SSLServerSocket**(int port)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Used only by subclasses.

Create a TCP server socket on a port, using the default authentication context. The connection backlog defaults to fifty connections queued up before the system starts to reject new connection requests.

**Parameters:**port - the port on which to listen **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket

### SSLServerSocket

protected **SSLServerSocket**(int port,  
 int backlog)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Used only by subclasses.

Create a TCP server socket on a port, using the default authentication context and a specified backlog of connections.

**Parameters:**port - the port on which to listenbacklog - how many connections may be pending before the system should start rejecting new requests **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket

### SSLServerSocket

protected **SSLServerSocket**(int port,  
 int backlog,  
 [InetAddress](http://docs.google.com/java/net/InetAddress.html) address)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Used only by subclasses.

Create a TCP server socket on a port, using the default authentication context and a specified backlog of connections as well as a particular specified network interface. This constructor is used on multihomed hosts, such as those used for firewalls or as routers, to control through which interface a network service is provided.

**Parameters:**port - the port on which to listenbacklog - how many connections may be pending before the system should start rejecting new requestsaddress - the address of the network interface through which connections will be accepted **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket

| **Method Detail** |
| --- |

### getEnabledCipherSuites

public abstract [String](http://docs.google.com/java/lang/String.html)[] **getEnabledCipherSuites**()

Returns the list of cipher suites which are currently enabled for use by newly accepted connections.

If this list has not been explicitly modified, a system-provided default guarantees a minimum quality of service in all enabled cipher suites.

There are several reasons why an enabled cipher suite might not actually be used. For example: the server socket might not have appropriate private keys available to it or the cipher suite might be anonymous, precluding the use of client authentication, while the server socket has been told to require that sort of authentication.

**Returns:**an array of cipher suites enabled**See Also:**[getSupportedCipherSuites()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedCipherSuites()), [setEnabledCipherSuites(String [])](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledCipherSuites(java.lang.String%5B%5D))

### setEnabledCipherSuites

public abstract void **setEnabledCipherSuites**([String](http://docs.google.com/java/lang/String.html)[] suites)

Sets the cipher suites enabled for use by accepted connections.

The cipher suites must have been listed by getSupportedCipherSuites() as being supported. Following a successful call to this method, only suites listed in the suites parameter are enabled for use.

Suites that require authentication information which is not available in this ServerSocket's authentication context will not be used in any case, even if they are enabled.

SSLSockets returned from accept() inherit this setting.

**Parameters:**suites - Names of all the cipher suites to enable **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - when one or more of ciphers named by the parameter is not supported, or when the parameter is null.**See Also:**[getSupportedCipherSuites()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedCipherSuites()), [getEnabledCipherSuites()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledCipherSuites())

### getSupportedCipherSuites

public abstract [String](http://docs.google.com/java/lang/String.html)[] **getSupportedCipherSuites**()

Returns the names of the cipher suites which could be enabled for use on an SSL connection.

Normally, only a subset of these will actually be enabled by default, since this list may include cipher suites which do not meet quality of service requirements for those defaults. Such cipher suites are useful in specialized applications.

**Returns:**an array of cipher suite names**See Also:**[getEnabledCipherSuites()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledCipherSuites()), [setEnabledCipherSuites(String [])](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledCipherSuites(java.lang.String%5B%5D))

### getSupportedProtocols

public abstract [String](http://docs.google.com/java/lang/String.html)[] **getSupportedProtocols**()

Returns the names of the protocols which could be enabled for use.

**Returns:**an array of protocol names supported**See Also:**[getEnabledProtocols()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledProtocols()), [setEnabledProtocols(String [])](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledProtocols(java.lang.String%5B%5D))

### getEnabledProtocols

public abstract [String](http://docs.google.com/java/lang/String.html)[] **getEnabledProtocols**()

Returns the names of the protocols which are currently enabled for use by the newly accepted connections.

**Returns:**an array of protocol names**See Also:**[getSupportedProtocols()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedProtocols()), [setEnabledProtocols(String [])](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnabledProtocols(java.lang.String%5B%5D))

### setEnabledProtocols

public abstract void **setEnabledProtocols**([String](http://docs.google.com/java/lang/String.html)[] protocols)

Controls which particular protocols are enabled for use by accepted connections.

The protocols must have been listed by getSupportedProtocols() as being supported. Following a successful call to this method, only protocols listed in the protocols parameter are enabled for use.

SSLSockets returned from accept() inherit this setting.

**Parameters:**protocols - Names of all the protocols to enable. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - when one or more of the protocols named by the parameter is not supported or when the protocols parameter is null.**See Also:**[getEnabledProtocols()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnabledProtocols()), [getSupportedProtocols()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getSupportedProtocols())

### setNeedClientAuth

public abstract void **setNeedClientAuth**(boolean need)

Controls whether accepted server-mode SSLSockets will be initially configured to *require* client authentication.

A socket's client authentication setting is one of the following:

* client authentication required
* client authentication requested
* no client authentication desired

Unlike [setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean)), if the accepted socket's option is set and the client chooses not to provide authentication information about itself, *the negotiations will stop and the connection will be dropped*.

Calling this method overrides any previous setting made by this method or [setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean)).

The initial inherited setting may be overridden by calling [SSLSocket.setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setNeedClientAuth(boolean)) or [SSLSocket.setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setWantClientAuth(boolean)).

**Parameters:**need - set to true if client authentication is required, or false if no client authentication is desired.**See Also:**[getNeedClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getNeedClientAuth()), [setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean)), [getWantClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getWantClientAuth()), [setUseClientMode(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))

### getNeedClientAuth

public abstract boolean **getNeedClientAuth**()

Returns true if client authentication will be *required* on newly accepted server-mode SSLSockets.

The initial inherited setting may be overridden by calling [SSLSocket.setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setNeedClientAuth(boolean)) or [SSLSocket.setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setWantClientAuth(boolean)).

**Returns:**true if client authentication is required, or false if no client authentication is desired.**See Also:**[setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean)), [setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean)), [getWantClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getWantClientAuth()), [setUseClientMode(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))

### setWantClientAuth

public abstract void **setWantClientAuth**(boolean want)

Controls whether accepted server-mode SSLSockets will be initially configured to *request* client authentication.

A socket's client authentication setting is one of the following:

* client authentication required
* client authentication requested
* no client authentication desired

Unlike [setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean)), if the accepted socket's option is set and the client chooses not to provide authentication information about itself, *the negotiations will continue*.

Calling this method overrides any previous setting made by this method or [setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean)).

The initial inherited setting may be overridden by calling [SSLSocket.setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setNeedClientAuth(boolean)) or [SSLSocket.setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setWantClientAuth(boolean)).

**Parameters:**want - set to true if client authentication is requested, or false if no client authentication is desired.**See Also:**[getWantClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getWantClientAuth()), [setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean)), [getNeedClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getNeedClientAuth()), [setUseClientMode(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))

### getWantClientAuth

public abstract boolean **getWantClientAuth**()

Returns true if client authentication will be *requested* on newly accepted server-mode connections.

The initial inherited setting may be overridden by calling [SSLSocket.setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setNeedClientAuth(boolean)) or [SSLSocket.setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLSocket.html#setWantClientAuth(boolean)).

**Returns:**true if client authentication is requested, or false if no client authentication is desired.**See Also:**[setWantClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setWantClientAuth(boolean)), [setNeedClientAuth(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setNeedClientAuth(boolean)), [getNeedClientAuth()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getNeedClientAuth()), [setUseClientMode(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))

### setUseClientMode

public abstract void **setUseClientMode**(boolean mode)

Controls whether accepted connections are in the (default) SSL server mode, or the SSL client mode.

Servers normally authenticate themselves, and clients are not required to do so.

In rare cases, TCP servers need to act in the SSL client mode on newly accepted connections. For example, FTP clients acquire server sockets and listen there for reverse connections from the server. An FTP client would use an SSLServerSocket in "client" mode to accept the reverse connection while the FTP server uses an SSLSocket with "client" mode disabled to initiate the connection. During the resulting handshake, existing SSL sessions may be reused.

SSLSockets returned from accept() inherit this setting.

**Parameters:**mode - true if newly accepted connections should use SSL client mode.**See Also:**[getUseClientMode()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getUseClientMode())

### getUseClientMode

public abstract boolean **getUseClientMode**()

Returns true if accepted connections will be in SSL client mode.

**Returns:**true if the connection should use SSL client mode.**See Also:**[setUseClientMode(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setUseClientMode(boolean))

### setEnableSessionCreation

public abstract void **setEnableSessionCreation**(boolean flag)

Controls whether new SSL sessions may be established by the sockets which are created from this server socket.

SSLSockets returned from accept() inherit this setting.

**Parameters:**flag - true indicates that sessions may be created; this is the default. false indicates that an existing session must be resumed.**See Also:**[getEnableSessionCreation()](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#getEnableSessionCreation())

### getEnableSessionCreation

public abstract boolean **getEnableSessionCreation**()

Returns true if new SSL sessions may be established by the sockets which are created from this server socket.

**Returns:**true indicates that sessions may be created; this is the default. false indicates that an existing session must be resumed.**See Also:**[setEnableSessionCreation(boolean)](http://docs.google.com/javax/net/ssl/SSLServerSocket.html#setEnableSessionCreation(boolean))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SSLServerSocket.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/net/ssl/SSLProtocolException.html)   [**NEXT CLASS**](http://docs.google.com/javax/net/ssl/SSLServerSocketFactory.html) | [**FRAMES**](http://docs.google.com/index.html?javax/net/ssl/SSLServerSocket.html)    [**NO FRAMES**](http://docs.google.com/SSLServerSocket.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#26in1rg) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).