| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SSLContext.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/ManagerFactoryParameters.html)   [**NEXT CLASS**](http://docs.google.com/javax/net/ssl/SSLContextSpi.html) | [**FRAMES**](http://docs.google.com/index.html?javax/net/ssl/SSLContext.html)    [**NO FRAMES**](http://docs.google.com/SSLContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.net.ssl**

Class SSLContext

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

public class **SSLContext**extends [Object](http://docs.google.com/java/lang/Object.html)

Instances of this class represent a secure socket protocol implementation which acts as a factory for secure socket factories or SSLEngines. This class is initialized with an optional set of key and trust managers and source of secure random bytes.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| protected | [**SSLContext**](http://docs.google.com/javax/net/ssl/SSLContext.html#SSLContext(javax.net.ssl.SSLContextSpi,%20java.security.Provider,%20java.lang.String))([SSLContextSpi](http://docs.google.com/javax/net/ssl/SSLContextSpi.html) contextSpi, [Provider](http://docs.google.com/java/security/Provider.html) provider, [String](http://docs.google.com/java/lang/String.html) protocol)            Creates an SSLContext object. |

| **Method Summary** | |
| --- | --- |
| [SSLEngine](http://docs.google.com/javax/net/ssl/SSLEngine.html) | [**createSSLEngine**](http://docs.google.com/javax/net/ssl/SSLContext.html#createSSLEngine())()            Creates a new SSLEngine using this context. |
| [SSLEngine](http://docs.google.com/javax/net/ssl/SSLEngine.html) | [**createSSLEngine**](http://docs.google.com/javax/net/ssl/SSLContext.html#createSSLEngine(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) peerHost, int peerPort)            Creates a new SSLEngine using this context using advisory peer information. |
| [SSLSessionContext](http://docs.google.com/javax/net/ssl/SSLSessionContext.html) | [**getClientSessionContext**](http://docs.google.com/javax/net/ssl/SSLContext.html#getClientSessionContext())()            Returns the client session context, which represents the set of SSL sessions available for use during the handshake phase of client-side SSL sockets. |
| static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) | [**getDefault**](http://docs.google.com/javax/net/ssl/SSLContext.html#getDefault())()            Returns the default SSL context. |
| [SSLParameters](http://docs.google.com/javax/net/ssl/SSLParameters.html) | [**getDefaultSSLParameters**](http://docs.google.com/javax/net/ssl/SSLContext.html#getDefaultSSLParameters())()            Returns a copy of the SSLParameters indicating the default settings for this SSL context. |
| static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) | [**getInstance**](http://docs.google.com/javax/net/ssl/SSLContext.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol)            Returns a SSLContext object that implements the specified secure socket protocol. |
| static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) | [**getInstance**](http://docs.google.com/javax/net/ssl/SSLContext.html#getInstance(java.lang.String,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) protocol, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns a SSLContext object that implements the specified secure socket protocol. |
| static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) | [**getInstance**](http://docs.google.com/javax/net/ssl/SSLContext.html#getInstance(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) provider)            Returns a SSLContext object that implements the specified secure socket protocol. |
| [String](http://docs.google.com/java/lang/String.html) | [**getProtocol**](http://docs.google.com/javax/net/ssl/SSLContext.html#getProtocol())()            Returns the protocol name of this SSLContext object. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/net/ssl/SSLContext.html#getProvider())()            Returns the provider of this SSLContext object. |
| [SSLSessionContext](http://docs.google.com/javax/net/ssl/SSLSessionContext.html) | [**getServerSessionContext**](http://docs.google.com/javax/net/ssl/SSLContext.html#getServerSessionContext())()            Returns the server session context, which represents the set of SSL sessions available for use during the handshake phase of server-side SSL sockets. |
| [SSLServerSocketFactory](http://docs.google.com/javax/net/ssl/SSLServerSocketFactory.html) | [**getServerSocketFactory**](http://docs.google.com/javax/net/ssl/SSLContext.html#getServerSocketFactory())()            Returns a ServerSocketFactory object for this context. |
| [SSLSocketFactory](http://docs.google.com/javax/net/ssl/SSLSocketFactory.html) | [**getSocketFactory**](http://docs.google.com/javax/net/ssl/SSLContext.html#getSocketFactory())()            Returns a SocketFactory object for this context. |
| [SSLParameters](http://docs.google.com/javax/net/ssl/SSLParameters.html) | [**getSupportedSSLParameters**](http://docs.google.com/javax/net/ssl/SSLContext.html#getSupportedSSLParameters())()            Returns a copy of the SSLParameters indicating the supported settings for this SSL context. |
| void | [**init**](http://docs.google.com/javax/net/ssl/SSLContext.html#init(javax.net.ssl.KeyManager%5B%5D,%20javax.net.ssl.TrustManager%5B%5D,%20java.security.SecureRandom))([KeyManager](http://docs.google.com/javax/net/ssl/KeyManager.html)[] km, [TrustManager](http://docs.google.com/javax/net/ssl/TrustManager.html)[] tm, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this context. |
| static void | [**setDefault**](http://docs.google.com/javax/net/ssl/SSLContext.html#setDefault(javax.net.ssl.SSLContext))([SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) context)            Sets the default SSL context. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### SSLContext

protected **SSLContext**([SSLContextSpi](http://docs.google.com/javax/net/ssl/SSLContextSpi.html) contextSpi,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider,  
 [String](http://docs.google.com/java/lang/String.html) protocol)

Creates an SSLContext object.

**Parameters:**contextSpi - the delegateprovider - the providerprotocol - the protocol

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

### getDefault

public static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) **getDefault**()  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns the default SSL context.

If a default context was set using the [SSLContext.setDefault()](http://docs.google.com/javax/net/ssl/SSLContext.html#setDefault(javax.net.ssl.SSLContext)) method, it is returned. Otherwise, the first call of this method triggers the call SSLContext.getInstance("Default"). If successful, that object is made the default SSL context and returned.

The default context is immediately usable and does not require [initialization](http://docs.google.com/javax/net/ssl/SSLContext.html#init(javax.net.ssl.KeyManager%5B%5D,%20javax.net.ssl.TrustManager%5B%5D,%20java.security.SecureRandom)).

**Returns:**the default SSL context **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if the [SSLContext.getInstance()](http://docs.google.com/javax/net/ssl/SSLContext.html#getInstance(java.lang.String)) call fails**Since:** 1.6

### setDefault

public static void **setDefault**([SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) context)

Sets the default SSL context. It will be returned by subsequent calls to [getDefault()](http://docs.google.com/javax/net/ssl/SSLContext.html#getDefault()). The default context must be immediately usable and not require [initialization](http://docs.google.com/javax/net/ssl/SSLContext.html#init(javax.net.ssl.KeyManager%5B%5D,%20javax.net.ssl.TrustManager%5B%5D,%20java.security.SecureRandom)).

**Parameters:**context - the SSLContext **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if context is null [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method does not allow SSLPermission("setDefaultSSLContext")**Since:** 1.6

### getInstance

public static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) protocol)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a SSLContext object that implements the specified secure socket protocol.

This method traverses the list of registered security Providers, starting with the most preferred Provider. A new SSLContext object encapsulating the SSLContextSpi implementation from the first Provider that supports the specified protocol is returned.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**protocol - the standard name of the requested protocol. See Appendix A in the  [Java Secure Socket Extension Reference Guide](http://docs.google.com/technotes/guides/security/jsse/JSSERefGuide.html#AppA)  for information about standard protocol names. **Returns:**the new SSLContext object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports a TrustManagerFactorySpi implementation for the specified protocol.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html)

Returns a SSLContext object that implements the specified secure socket protocol.

A new SSLContext object encapsulating the SSLContextSpi implementation from the specified provider is returned. The specified provider must be registered in the security provider list.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**protocol - the standard name of the requested protocol. See Appendix A in the  [Java Secure Socket Extension Reference Guide](http://docs.google.com/technotes/guides//security/jsse/JSSERefGuide.html#AppA)  for information about standard protocol names.provider - the name of the provider. **Returns:**the new SSLContext object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a SSLContextSpi implementation for the specified protocol is not available from the specified provider. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if the specified provider is not registered in the security provider list. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the provider name is null or empty.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [SSLContext](http://docs.google.com/javax/net/ssl/SSLContext.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a SSLContext object that implements the specified secure socket protocol.

A new SSLContext object encapsulating the SSLContextSpi implementation from the specified Provider object is returned. Note that the specified Provider object does not have to be registered in the provider list.

**Parameters:**protocol - the standard name of the requested protocol. See Appendix A in the  [Java Secure Socket Extension Reference Guide](http://docs.google.com/technotes/guides/security/jsse/JSSERefGuide.html#AppA)  for information about standard protocol names.provider - an instance of the provider. **Returns:**the new SSLContext object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a KeyManagerFactorySpi implementation for the specified protocol is not available from the specified Provider object. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the provider name is null.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getProtocol

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

Returns the protocol name of this SSLContext object.

This is the same name that was specified in one of the getInstance calls that created this SSLContext object.

**Returns:**the protocol name of this SSLContext object.

### getProvider

public final [Provider](http://docs.google.com/java/security/Provider.html) **getProvider**()

Returns the provider of this SSLContext object.

**Returns:**the provider of this SSLContext object

### init

public final void **init**([KeyManager](http://docs.google.com/javax/net/ssl/KeyManager.html)[] km,  
 [TrustManager](http://docs.google.com/javax/net/ssl/TrustManager.html)[] tm,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)  
 throws [KeyManagementException](http://docs.google.com/java/security/KeyManagementException.html)

Initializes this context. Either of the first two parameters may be null in which case the installed security providers will be searched for the highest priority implementation of the appropriate factory. Likewise, the secure random parameter may be null in which case the default implementation will be used.

Only the first instance of a particular key and/or trust manager implementation type in the array is used. (For example, only the first javax.net.ssl.X509KeyManager in the array will be used.)

**Parameters:**km - the sources of authentication keys or nulltm - the sources of peer authentication trust decisions or nullrandom - the source of randomness for this generator or null **Throws:** [KeyManagementException](http://docs.google.com/java/security/KeyManagementException.html) - if this operation fails

### getSocketFactory

public final [SSLSocketFactory](http://docs.google.com/javax/net/ssl/SSLSocketFactory.html) **getSocketFactory**()

Returns a SocketFactory object for this context.

**Returns:**the SocketFactory object **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the SSLContextImpl requires initialization and the init() has not been called

### getServerSocketFactory

public final [SSLServerSocketFactory](http://docs.google.com/javax/net/ssl/SSLServerSocketFactory.html) **getServerSocketFactory**()

Returns a ServerSocketFactory object for this context.

**Returns:**the ServerSocketFactory object **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the SSLContextImpl requires initialization and the init() has not been called

### createSSLEngine

public final [SSLEngine](http://docs.google.com/javax/net/ssl/SSLEngine.html) **createSSLEngine**()

Creates a new SSLEngine using this context.

Applications using this factory method are providing no hints for an internal session reuse strategy. If hints are desired, [createSSLEngine(String, int)](http://docs.google.com/javax/net/ssl/SSLContext.html#createSSLEngine(java.lang.String,%20int)) should be used instead.

Some cipher suites (such as Kerberos) require remote hostname information, in which case this factory method should not be used.

**Returns:**the SSLEngine object **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the underlying provider does not implement the operation. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the SSLContextImpl requires initialization and the init() has not been called**Since:** 1.5

### createSSLEngine

public final [SSLEngine](http://docs.google.com/javax/net/ssl/SSLEngine.html) **createSSLEngine**([String](http://docs.google.com/java/lang/String.html) peerHost,  
 int peerPort)

Creates a new SSLEngine using this context using advisory peer information.

Applications using this factory method are providing hints for an internal session reuse strategy.

Some cipher suites (such as Kerberos) require remote hostname information, in which case peerHost needs to be specified.

**Parameters:**peerHost - the non-authoritative name of the hostpeerPort - the non-authoritative port **Returns:**the new SSLEngine object **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the underlying provider does not implement the operation. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the SSLContextImpl requires initialization and the init() has not been called**Since:** 1.5

### getServerSessionContext

public final [SSLSessionContext](http://docs.google.com/javax/net/ssl/SSLSessionContext.html) **getServerSessionContext**()

Returns the server session context, which represents the set of SSL sessions available for use during the handshake phase of server-side SSL sockets.

This context may be unavailable in some environments, in which case this method returns null. For example, when the underlying SSL provider does not provide an implementation of SSLSessionContext interface, this method returns null. A non-null session context is returned otherwise.

**Returns:**server session context bound to this SSL context

### getClientSessionContext

public final [SSLSessionContext](http://docs.google.com/javax/net/ssl/SSLSessionContext.html) **getClientSessionContext**()

Returns the client session context, which represents the set of SSL sessions available for use during the handshake phase of client-side SSL sockets.

This context may be unavailable in some environments, in which case this method returns null. For example, when the underlying SSL provider does not provide an implementation of SSLSessionContext interface, this method returns null. A non-null session context is returned otherwise.

**Returns:**client session context bound to this SSL context

### getDefaultSSLParameters

public final [SSLParameters](http://docs.google.com/javax/net/ssl/SSLParameters.html) **getDefaultSSLParameters**()

Returns a copy of the SSLParameters indicating the default settings for this SSL context.

The parameters will always have the ciphersuites and protocols arrays set to non-null values.

**Returns:**a copy of the SSLParameters object with the default settings **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the default SSL parameters could not be obtained.**Since:** 1.6

### getSupportedSSLParameters

public final [SSLParameters](http://docs.google.com/javax/net/ssl/SSLParameters.html) **getSupportedSSLParameters**()

Returns a copy of the SSLParameters indicating the supported settings for this SSL context.

The parameters will always have the ciphersuites and protocols arrays set to non-null values.

**Returns:**a copy of the SSLParameters object with the supported settings **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the supported SSL parameters could not be obtained.**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SSLContext.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/ManagerFactoryParameters.html)   [**NEXT CLASS**](http://docs.google.com/javax/net/ssl/SSLContextSpi.html) | [**FRAMES**](http://docs.google.com/index.html?javax/net/ssl/SSLContext.html)    [**NO FRAMES**](http://docs.google.com/SSLContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).