| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SocketFactory.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/ServerSocketFactory.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/net/SocketFactory.html)    [**NO FRAMES**](http://docs.google.com/SocketFactory.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**

Class SocketFactory

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

**Direct Known Subclasses:** [SSLSocketFactory](http://docs.google.com/javax/net/ssl/SSLSocketFactory.html)

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

This class creates sockets. It may be subclassed by other factories, which create particular subclasses of sockets and thus provide a general framework for the addition of public socket-level functionality.

Socket factories are a simple way to capture a variety of policies related to the sockets being constructed, producing such sockets in a way which does not require special configuration of the code which asks for the sockets:

* Due to polymorphism of both factories and sockets, different kinds of sockets can be used by the same application code just by passing it different kinds of factories.
* Factories can themselves be customized with parameters used in socket construction. So for example, factories could be customized to return sockets with different networking timeouts or security parameters already configured.
* The sockets returned to the application can be subclasses of java.net.Socket, so that they can directly expose new APIs for features such as compression, security, record marking, statistics collection, or firewall tunneling.

Factory classes are specified by environment-specific configuration mechanisms. For example, the *getDefault* method could return a factory that was appropriate for a particular user or applet, and a framework could use a factory customized to its own purposes.

**Since:** 1.4 **See Also:**[ServerSocketFactory](http://docs.google.com/javax/net/ServerSocketFactory.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**SocketFactory**](http://docs.google.com/javax/net/SocketFactory.html#SocketFactory())()            Creates a SocketFactory. |

| **Method Summary** | |
| --- | --- |
| [Socket](http://docs.google.com/java/net/Socket.html) | [**createSocket**](http://docs.google.com/javax/net/SocketFactory.html#createSocket())()            Creates an unconnected socket. |
| abstract  [Socket](http://docs.google.com/java/net/Socket.html) | [**createSocket**](http://docs.google.com/javax/net/SocketFactory.html#createSocket(java.net.InetAddress,%20int))([InetAddress](http://docs.google.com/java/net/InetAddress.html) host, int port)            Creates a socket and connects it to the specified port number at the specified address. |
| abstract  [Socket](http://docs.google.com/java/net/Socket.html) | [**createSocket**](http://docs.google.com/javax/net/SocketFactory.html#createSocket(java.net.InetAddress,%20int,%20java.net.InetAddress,%20int))([InetAddress](http://docs.google.com/java/net/InetAddress.html) address, int port, [InetAddress](http://docs.google.com/java/net/InetAddress.html) localAddress, int localPort)            Creates a socket and connect it to the specified remote address on the specified remote port. |
| abstract  [Socket](http://docs.google.com/java/net/Socket.html) | [**createSocket**](http://docs.google.com/javax/net/SocketFactory.html#createSocket(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) host, int port)            Creates a socket and connects it to the specified remote host at the specified remote port. |
| abstract  [Socket](http://docs.google.com/java/net/Socket.html) | [**createSocket**](http://docs.google.com/javax/net/SocketFactory.html#createSocket(java.lang.String,%20int,%20java.net.InetAddress,%20int))([String](http://docs.google.com/java/lang/String.html) host, int port, [InetAddress](http://docs.google.com/java/net/InetAddress.html) localHost, int localPort)            Creates a socket and connects it to the specified remote host on the specified remote port. |
| static [SocketFactory](http://docs.google.com/javax/net/SocketFactory.html) | [**getDefault**](http://docs.google.com/javax/net/SocketFactory.html#getDefault())()            Returns a copy of the environment's default socket factory. |

| **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** |
| --- |

### SocketFactory

protected **SocketFactory**()

Creates a SocketFactory.

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

### getDefault

public static [SocketFactory](http://docs.google.com/javax/net/SocketFactory.html) **getDefault**()

Returns a copy of the environment's default socket factory.

**Returns:**the default SocketFactory

### createSocket

public [Socket](http://docs.google.com/java/net/Socket.html) **createSocket**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates an unconnected socket.

**Returns:**the unconnected socket **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the socket cannot be created**See Also:**[Socket.connect(java.net.SocketAddress)](http://docs.google.com/java/net/Socket.html#connect(java.net.SocketAddress)), [Socket.connect(java.net.SocketAddress, int)](http://docs.google.com/java/net/Socket.html#connect(java.net.SocketAddress,%20int)), [Socket.Socket()](http://docs.google.com/java/net/Socket.html#Socket())

### createSocket

public abstract [Socket](http://docs.google.com/java/net/Socket.html) **createSocket**([String](http://docs.google.com/java/lang/String.html) host,  
 int port)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html)

Creates a socket and connects it to the specified remote host at the specified remote port. This socket is configured using the socket options established for this factory.

**Parameters:**host - the server hostport - the server port **Returns:**the Socket **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html) - if the host is not known**See Also:**[Socket.Socket(String, int)](http://docs.google.com/java/net/Socket.html#Socket(java.lang.String,%20int))

### createSocket

public abstract [Socket](http://docs.google.com/java/net/Socket.html) **createSocket**([String](http://docs.google.com/java/lang/String.html) host,  
 int port,  
 [InetAddress](http://docs.google.com/java/net/InetAddress.html) localHost,  
 int localPort)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html)

Creates a socket and connects it to the specified remote host on the specified remote port. The socket will also be bound to the local address and port supplied. This socket is configured using the socket options established for this factory.

**Parameters:**host - the server hostport - the server portlocalHost - the local address the socket is bound tolocalPort - the local port the socket is bound to **Returns:**the Socket **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html) - if the host is not known**See Also:**[Socket.Socket(String, int, java.net.InetAddress, int)](http://docs.google.com/java/net/Socket.html#Socket(java.lang.String,%20int,%20java.net.InetAddress,%20int))

### createSocket

public abstract [Socket](http://docs.google.com/java/net/Socket.html) **createSocket**([InetAddress](http://docs.google.com/java/net/InetAddress.html) host,  
 int port)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates a socket and connects it to the specified port number at the specified address. This socket is configured using the socket options established for this factory.

**Parameters:**host - the server hostport - the server port **Returns:**the Socket **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket**See Also:**[Socket.Socket(java.net.InetAddress, int)](http://docs.google.com/java/net/Socket.html#Socket(java.net.InetAddress,%20int))

### createSocket

public abstract [Socket](http://docs.google.com/java/net/Socket.html) **createSocket**([InetAddress](http://docs.google.com/java/net/InetAddress.html) address,  
 int port,  
 [InetAddress](http://docs.google.com/java/net/InetAddress.html) localAddress,  
 int localPort)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates a socket and connect it to the specified remote address on the specified remote port. The socket will also be bound to the local address and port suplied. The socket is configured using the socket options established for this factory.

**Parameters:**address - the server network addressport - the server portlocalAddress - the client network addresslocalPort - the client port **Returns:**the Socket **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs when creating the socket**See Also:**[Socket.Socket(java.net.InetAddress, int, java.net.InetAddress, int)](http://docs.google.com/java/net/Socket.html#Socket(java.net.InetAddress,%20int,%20java.net.InetAddress,%20int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SocketFactory.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/ServerSocketFactory.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/net/SocketFactory.html)    [**NO FRAMES**](http://docs.google.com/SocketFactory.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).