| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DatagramSocketImpl.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/java/net/DatagramSocket.html)   [**NEXT CLASS**](http://docs.google.com/java/net/DatagramSocketImplFactory.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/DatagramSocketImpl.html)    [**NO FRAMES**](http://docs.google.com/DatagramSocketImpl.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.net**

Class DatagramSocketImpl

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

**All Implemented Interfaces:** [SocketOptions](http://docs.google.com/java/net/SocketOptions.html)

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

Abstract datagram and multicast socket implementation base class.

**Since:** JDK1.1

| **Field Summary** | |
| --- | --- |
| protected  [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) | [**fd**](http://docs.google.com/java/net/DatagramSocketImpl.html#fd)            The file descriptor object. |
| protected  int | [**localPort**](http://docs.google.com/java/net/DatagramSocketImpl.html#localPort)            The local port number. |

| **Fields inherited from interface java.net.**[**SocketOptions**](http://docs.google.com/java/net/SocketOptions.html) |
| --- |
| [IP\_MULTICAST\_IF](http://docs.google.com/java/net/SocketOptions.html#IP_MULTICAST_IF), [IP\_MULTICAST\_IF2](http://docs.google.com/java/net/SocketOptions.html#IP_MULTICAST_IF2), [IP\_MULTICAST\_LOOP](http://docs.google.com/java/net/SocketOptions.html#IP_MULTICAST_LOOP), [IP\_TOS](http://docs.google.com/java/net/SocketOptions.html#IP_TOS), [SO\_BINDADDR](http://docs.google.com/java/net/SocketOptions.html#SO_BINDADDR), [SO\_BROADCAST](http://docs.google.com/java/net/SocketOptions.html#SO_BROADCAST), [SO\_KEEPALIVE](http://docs.google.com/java/net/SocketOptions.html#SO_KEEPALIVE), [SO\_LINGER](http://docs.google.com/java/net/SocketOptions.html#SO_LINGER), [SO\_OOBINLINE](http://docs.google.com/java/net/SocketOptions.html#SO_OOBINLINE), [SO\_RCVBUF](http://docs.google.com/java/net/SocketOptions.html#SO_RCVBUF), [SO\_REUSEADDR](http://docs.google.com/java/net/SocketOptions.html#SO_REUSEADDR), [SO\_SNDBUF](http://docs.google.com/java/net/SocketOptions.html#SO_SNDBUF), [SO\_TIMEOUT](http://docs.google.com/java/net/SocketOptions.html#SO_TIMEOUT), [TCP\_NODELAY](http://docs.google.com/java/net/SocketOptions.html#TCP_NODELAY) |

| **Constructor Summary** | |
| --- | --- |
| [**DatagramSocketImpl**](http://docs.google.com/java/net/DatagramSocketImpl.html#DatagramSocketImpl())() |

| **Method Summary** | |
| --- | --- |
| protected abstract  void | [**bind**](http://docs.google.com/java/net/DatagramSocketImpl.html#bind(int,%20java.net.InetAddress))(int lport, [InetAddress](http://docs.google.com/java/net/InetAddress.html) laddr)            Binds a datagram socket to a local port and address. |
| protected abstract  void | [**close**](http://docs.google.com/java/net/DatagramSocketImpl.html#close())()            Close the socket. |
| protected  void | [**connect**](http://docs.google.com/java/net/DatagramSocketImpl.html#connect(java.net.InetAddress,%20int))([InetAddress](http://docs.google.com/java/net/InetAddress.html) address, int port)            Connects a datagram socket to a remote destination. |
| protected abstract  void | [**create**](http://docs.google.com/java/net/DatagramSocketImpl.html#create())()            Creates a datagram socket. |
| protected  void | [**disconnect**](http://docs.google.com/java/net/DatagramSocketImpl.html#disconnect())()            Disconnects a datagram socket from its remote destination. |
| protected  [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) | [**getFileDescriptor**](http://docs.google.com/java/net/DatagramSocketImpl.html#getFileDescriptor())()            Gets the datagram socket file descriptor. |
| protected  int | [**getLocalPort**](http://docs.google.com/java/net/DatagramSocketImpl.html#getLocalPort())()            Gets the local port. |
| protected abstract  int | [**getTimeToLive**](http://docs.google.com/java/net/DatagramSocketImpl.html#getTimeToLive())()            Retrieve the TTL (time-to-live) option. |
| protected abstract  byte | [**getTTL**](http://docs.google.com/java/net/DatagramSocketImpl.html#getTTL())()  **Deprecated.** *use getTimeToLive instead.* |
| protected abstract  void | [**join**](http://docs.google.com/java/net/DatagramSocketImpl.html#join(java.net.InetAddress))([InetAddress](http://docs.google.com/java/net/InetAddress.html) inetaddr)            Join the multicast group. |
| protected abstract  void | [**joinGroup**](http://docs.google.com/java/net/DatagramSocketImpl.html#joinGroup(java.net.SocketAddress,%20java.net.NetworkInterface))([SocketAddress](http://docs.google.com/java/net/SocketAddress.html) mcastaddr, [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) netIf)            Join the multicast group. |
| protected abstract  void | [**leave**](http://docs.google.com/java/net/DatagramSocketImpl.html#leave(java.net.InetAddress))([InetAddress](http://docs.google.com/java/net/InetAddress.html) inetaddr)            Leave the multicast group. |
| protected abstract  void | [**leaveGroup**](http://docs.google.com/java/net/DatagramSocketImpl.html#leaveGroup(java.net.SocketAddress,%20java.net.NetworkInterface))([SocketAddress](http://docs.google.com/java/net/SocketAddress.html) mcastaddr, [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) netIf)            Leave the multicast group. |
| protected abstract  int | [**peek**](http://docs.google.com/java/net/DatagramSocketImpl.html#peek(java.net.InetAddress))([InetAddress](http://docs.google.com/java/net/InetAddress.html) i)            Peek at the packet to see who it is from. |
| protected abstract  int | [**peekData**](http://docs.google.com/java/net/DatagramSocketImpl.html#peekData(java.net.DatagramPacket))([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)            Peek at the packet to see who it is from. |
| protected abstract  void | [**receive**](http://docs.google.com/java/net/DatagramSocketImpl.html#receive(java.net.DatagramPacket))([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)            Receive the datagram packet. |
| protected abstract  void | [**send**](http://docs.google.com/java/net/DatagramSocketImpl.html#send(java.net.DatagramPacket))([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)            Sends a datagram packet. |
| protected abstract  void | [**setTimeToLive**](http://docs.google.com/java/net/DatagramSocketImpl.html#setTimeToLive(int))(int ttl)            Set the TTL (time-to-live) option. |
| protected abstract  void | [**setTTL**](http://docs.google.com/java/net/DatagramSocketImpl.html#setTTL(byte))(byte ttl)  **Deprecated.** *use setTimeToLive instead.* |

| **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)) |

| **Methods inherited from interface java.net.**[**SocketOptions**](http://docs.google.com/java/net/SocketOptions.html) |
| --- |
| [getOption](http://docs.google.com/java/net/SocketOptions.html#getOption(int)), [setOption](http://docs.google.com/java/net/SocketOptions.html#setOption(int,%20java.lang.Object)) |

| **Field Detail** |
| --- |

### localPort

protected int **localPort**

The local port number.

### fd

protected [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) **fd**

The file descriptor object.

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

### DatagramSocketImpl

public **DatagramSocketImpl**()

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

### create

protected abstract void **create**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Creates a datagram socket.

**Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if there is an error in the underlying protocol, such as a TCP error.

### bind

protected abstract void **bind**(int lport,  
 [InetAddress](http://docs.google.com/java/net/InetAddress.html) laddr)  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Binds a datagram socket to a local port and address.

**Parameters:**lport - the local portladdr - the local address **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if there is an error in the underlying protocol, such as a TCP error.

### send

protected abstract void **send**([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sends a datagram packet. The packet contains the data and the destination address to send the packet to.

**Parameters:**p - the packet to be sent. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while sending the datagram packet. [PortUnreachableException](http://docs.google.com/java/net/PortUnreachableException.html) - may be thrown if the socket is connected to a currently unreachable destination. Note, there is no guarantee that the exception will be thrown.

### connect

protected void **connect**([InetAddress](http://docs.google.com/java/net/InetAddress.html) address,  
 int port)  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Connects a datagram socket to a remote destination. This associates the remote address with the local socket so that datagrams may only be sent to this destination and received from this destination. This may be overridden to call a native system connect.

If the remote destination to which the socket is connected does not exist, or is otherwise unreachable, and if an ICMP destination unreachable packet has been received for that address, then a subsequent call to send or receive may throw a PortUnreachableException. Note, there is no guarantee that the exception will be thrown.

**Parameters:**address - the remote InetAddress to connect toport - the remote port number **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - may be thrown if the socket cannot be connected to the remote destination**Since:** 1.4

### disconnect

protected void **disconnect**()

Disconnects a datagram socket from its remote destination.

**Since:** 1.4

### peek

protected abstract int **peek**([InetAddress](http://docs.google.com/java/net/InetAddress.html) i)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Peek at the packet to see who it is from. Updates the specified InetAddress to the address which the packet came from.

**Parameters:**i - an InetAddress object **Returns:**the port number which the packet came from. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs [PortUnreachableException](http://docs.google.com/java/net/PortUnreachableException.html) - may be thrown if the socket is connected to a currently unreachable destination. Note, there is no guarantee that the exception will be thrown.

### peekData

protected abstract int **peekData**([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Peek at the packet to see who it is from. The data is copied into the specified DatagramPacket. The data is returned, but not consumed, so that a subsequent peekData/receive operation will see the same data.

**Parameters:**p - the Packet Received. **Returns:**the port number which the packet came from. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs [PortUnreachableException](http://docs.google.com/java/net/PortUnreachableException.html) - may be thrown if the socket is connected to a currently unreachable destination. Note, there is no guarantee that the exception will be thrown.**Since:** 1.4

### receive

protected abstract void **receive**([DatagramPacket](http://docs.google.com/java/net/DatagramPacket.html) p)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Receive the datagram packet.

**Parameters:**p - the Packet Received. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while receiving the datagram packet. [PortUnreachableException](http://docs.google.com/java/net/PortUnreachableException.html) - may be thrown if the socket is connected to a currently unreachable destination. Note, there is no guarantee that the exception will be thrown.

### setTTL

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected abstract void **setTTL**(byte ttl)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

**Deprecated.** *use setTimeToLive instead.*

Set the TTL (time-to-live) option.

**Parameters:**ttl - a byte specifying the TTL value **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while setting the time-to-live option.**See Also:**[getTTL()](http://docs.google.com/java/net/DatagramSocketImpl.html#getTTL())

### getTTL

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected abstract byte **getTTL**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

**Deprecated.** *use getTimeToLive instead.*

Retrieve the TTL (time-to-live) option.

**Returns:**a byte representing the TTL value **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while retrieving the time-to-live option**See Also:**[setTTL(byte)](http://docs.google.com/java/net/DatagramSocketImpl.html#setTTL(byte))

### setTimeToLive

protected abstract void **setTimeToLive**(int ttl)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Set the TTL (time-to-live) option.

**Parameters:**ttl - an int specifying the time-to-live value **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while setting the time-to-live option.**See Also:**[getTimeToLive()](http://docs.google.com/java/net/DatagramSocketImpl.html#getTimeToLive())

### getTimeToLive

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

Retrieve the TTL (time-to-live) option.

**Returns:**an int representing the time-to-live value **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while retrieving the time-to-live option**See Also:**[setTimeToLive(int)](http://docs.google.com/java/net/DatagramSocketImpl.html#setTimeToLive(int))

### join

protected abstract void **join**([InetAddress](http://docs.google.com/java/net/InetAddress.html) inetaddr)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Join the multicast group.

**Parameters:**inetaddr - multicast address to join. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while joining the multicast group.

### leave

protected abstract void **leave**([InetAddress](http://docs.google.com/java/net/InetAddress.html) inetaddr)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Leave the multicast group.

**Parameters:**inetaddr - multicast address to leave. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while leaving the multicast group.

### joinGroup

protected abstract void **joinGroup**([SocketAddress](http://docs.google.com/java/net/SocketAddress.html) mcastaddr,  
 [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) netIf)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Join the multicast group.

**Parameters:**mcastaddr - address to join.netIf - specifies the local interface to receive multicast datagram packets **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while joining the multicast group**Since:** 1.4

### leaveGroup

protected abstract void **leaveGroup**([SocketAddress](http://docs.google.com/java/net/SocketAddress.html) mcastaddr,  
 [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) netIf)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Leave the multicast group.

**Parameters:**mcastaddr - address to leave.netIf - specified the local interface to leave the group at **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs while leaving the multicast group**Since:** 1.4

### close

protected abstract void **close**()

Close the socket.

### getLocalPort

protected int **getLocalPort**()

Gets the local port.

**Returns:**an int representing the local port value

### getFileDescriptor

protected [FileDescriptor](http://docs.google.com/java/io/FileDescriptor.html) **getFileDescriptor**()

Gets the datagram socket file descriptor.

**Returns:**a FileDescriptor object representing the datagram socket file descriptor

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DatagramSocketImpl.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/java/net/DatagramSocket.html)   [**NEXT CLASS**](http://docs.google.com/java/net/DatagramSocketImplFactory.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/DatagramSocketImpl.html)    [**NO FRAMES**](http://docs.google.com/DatagramSocketImpl.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).