Class java.net.MulticastSocket

```
java.lang.Object
  |
  +--java.net.DatagramSocket
        |
        +--java.net.MulticastSocket
```

public class **MulticastSocket**
extends DatagramSocket

The multicast datagram socket class is useful for sending and receiving IP multicast packets. A MulticastSocket is a (UDP) DatagramSocket, with additional capabilities for joining "groups" of other multicast hosts on the internet.

A multicast group is specified by a class D IP address, those in the range `224.0.0.1` to `239.255.255.255`, inclusive, and by a standard UDP port number. One would join a multicast group by first creating a MulticastSocket with the desired port, then invoking the `joinGroup(InetAddress groupAddr)` method:

```
// join a Multicast group and send the group salutations
...
byte[] msg = {'H', 'e', 'l', 'l', 'o'};
InetAddress group = InetAddress.getByName("228.5.6.7");
MulticastSocket s = new MulticastSocket(6789);
s.joinGroup(group);
DatagramPacket hi = new DatagramPacket(msg, msg.length,
                            group, 6789);
s.send(hi);
// get their responses!
byte[] buf = new byte[1000];
DatagramPacket recv = new DatagramPacket(buf, buf.length);
s.receive(recv);
...
// OK, I'm done talking - leave the group...
s.leaveGroup(group);
```

When one sends a message to a multicast group, **all** subscribing recipients to that host and port receive the message (within the time-to-live range of the packet, see below). The socket needn't be a member of the multicast group to send messages to it.

When a socket subscribes to a multicast group/port, it receives datagrams sent by other hosts to the group/port, as do all other members of the group and port. A socket relinquishes membership in a group by the leaveGroup(InetAddress addr) method. **Multiple MulticastSocket's** may subscribe to a multicast group and port concurrently, and they will all receive group datagrams.

Currently applets are not allowed to use multicast sockets.

Since:
JDK1.1

Constructor Summary

MulticastSocket() Create a multicast socket.
MulticastSocket(int port) Create a multicast socket and bind it to a specific port.

Method Summary

InetAddress	**getInterface**() Retrieve the address of the network interface used for multicast packets.
int	**getTimeToLive**() Get the default time-to-live for multicast packets sent out on the socket.
byte	**getTTL**() **Deprecated.** *use the getTimeToLive method instead, which allows you to get time-to-live values from 0 (excluded) to 255 (included).*
void	**joinGroup**(InetAddress mcastaddr) Joins a multicast group.
void	**leaveGroup**(InetAddress mcastaddr) Leave a multicast group.
void	**send**(DatagramPacket p, byte ttl) Sends a datagram packet to the destination, with a TTL (time- to-live) other than the default for the socket.
void	**setInterface**(InetAddress inf) Set the outgoing network interface for multicast packets on this socket, to other than the system default.
void	**setTimeToLive**(int ttl) Set the default time-to-live for multicast packets sent out on this socket.
void	**setTTL**(byte ttl) **Deprecated.** *use the setTimeToLive method instead, which allows you to set time-to-live values from 0 (excluded) to 255 (included).*

Methods inherited from class java.net.DatagramSocket
close , getLocalAddress , getLocalPort , getReceiveBufferSize , getSendBufferSize , getSoTimeout , receive , send , setReceiveBufferSize , setSendBufferSize , setSoTimeout

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

MulticastSocket

```
public MulticastSocket()
                throws IOException
```

Create a multicast socket.

MulticastSocket

```
public MulticastSocket(int port)
               throws IOException
```

Create a multicast socket and bind it to a specific port.

Parameters:
`local` - port to use

Method Detail

setTTL

```
public void setTTL(byte ttl)
            throws IOException
```

Deprecated. *use the setTimeToLive method instead, which allows you to set time-to-live values from 0 (excluded) to 255 (included).*
Set the default time-to-live for multicast packets sent out on this socket. The TTL sets the IP time-to-live for `DatagramPackets` sent to a MulticastGroup, which specifies how many "hops" that the packet will be forwarded on the network before it expires.

This method may only be used to set time-to-live value between 1 and 127. The behavior if the value is outside that range is undefined.

Parameters:
`ttl` - the time-to-live

---

setTimeToLive

```
public void setTimeToLive(int ttl)
                  throws IOException
```

Set the default time-to-live for multicast packets sent out on this socket. The TTL sets the IP time-to-live for `DatagramPackets` sent to a MulticastGroup, which specifies how many "hops" that the packet will be forwarded on the network before it expires.

The ttl is **must** be in the range `0 < ttl <= 255` or an IllegalArgumentException will be thrown.

Parameters:
`ttl` - the time-to-live

---

getTTL

```
public byte getTTL()
            throws IOException
```

Deprecated. *use the getTimeToLive method instead, which allows you to get time-to-live values from 0 (excluded) to 255 (included).*
Get the default time-to-live for multicast packets sent out on the socket. This method will truncate any time to live values greater than 127 to 127.

---

getTimeToLive

```
public int getTimeToLive()
                throws IOException
```

Get the default time-to-live for multicast packets sent out on the socket.

---

joinGroup

```
public void joinGroup(InetAddress mcastaddr)
               throws IOException
```

Joins a multicast group.

Parameters:
`mcastaddr` - is the multicast address to join
Throws:
IOException - is raised if there is an error joining or when address is not a multicast address.

---

leaveGroup

```
public void leaveGroup(InetAddress mcastaddr)
                throws IOException
```

Leave a multicast group.

Parameters:
`mcastaddr` - is the multicast address to leave
Throws:
IOException - is raised if there is an error leaving or when address is not a multicast address.

---

setInterface

```
public void setInterface(InetAddress inf)
                  throws SocketException
```

Set the outgoing network interface for multicast packets on this socket, to other than the system default. Useful for multihomed hosts.

---

getInterface

```
public InetAddress getInterface()
                        throws SocketException
```

Retrieve the address of the network interface used for multicast packets.

---

send

```
public void send(DatagramPacket p,
                 byte ttl)
          throws IOException
```

Sends a datagram packet to the destination, with a TTL (time- to-live) other than the default for the socket. This method need only be used in instances where a particular TTL is desired; otherwise it is preferable to set a TTL once on the socket, and use that default TTL for all packets. This method does **not** alter the default TTL for the socket.

Parameters:

`p` - is the packet to be sent. The packet should contain the destination multicast ip address and the data to be sent. One does not need to be the member of the group to send packets to a destination multicast address.

`ttl` - optional time to live for multicast packet. default ttl is 1.

Throws:

IOException - is raised if an error occurs i.e error while setting ttl.

See Also:

`DatagramSocket.send(java.net.DatagramPacket)`,
`DatagramSocket.receive(java.net.DatagramPacket)`
