java.nio.channels

**Interface MulticastChannel**

* All Superinterfaces:

[AutoCloseable](http://download.oracle.com/javase/7/docs/api/java/lang/AutoCloseable.html), [Channel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html), [Closeable](http://download.oracle.com/javase/7/docs/api/java/io/Closeable.html), [NetworkChannel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html)

All Known Implementing Classes:

[DatagramChannel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/DatagramChannel.html)

public interface MulticastChannel

extends [NetworkChannel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html)

A network channel that supports Internet Protocol (IP) multicasting.

IP multicasting is the transmission of IP datagrams to members of a *group* that is zero or more hosts identified by a single destination address.

In the case of a channel to an [IPv4](http://download.oracle.com/javase/7/docs/api/java/net/StandardProtocolFamily.html#INET) socket, the underlying operating system supports [*RFC 2236: Internet Group Management Protocol, Version 2 (IGMPv2)*](http://www.ietf.org/rfc/rfc2236.txt). It may optionally support source filtering as specified by [*RFC 3376: Internet Group Management Protocol, Version 3 (IGMPv3)*](http://www.ietf.org/rfc/rfc3376.txt). For channels to an [IPv6](http://download.oracle.com/javase/7/docs/api/java/net/StandardProtocolFamily.html#INET6) socket, the equivalent standards are [*RFC 2710: Multicast Listener Discovery (MLD) for IPv6*](http://www.ietf.org/rfc/rfc2710.txt) and [*RFC 3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6*](http://www.ietf.org/rfc/rfc3810.txt).

The [join(InetAddress,NetworkInterface)](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MulticastChannel.html#join%28java.net.InetAddress,%20java.net.NetworkInterface%29) method is used to join a group and receive all multicast datagrams sent to the group. A channel may join several multicast groups and may join the same group on several [interfaces](http://download.oracle.com/javase/7/docs/api/java/net/NetworkInterface.html). Membership is dropped by invoking the [drop](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#drop%28%29) method on the returned [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html). If the underlying platform supports source filtering then the [block](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#block%28java.net.InetAddress%29) and [unblock](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#unblock%28java.net.InetAddress%29) methods can be used to block or unblock multicast datagrams from particular source addresses.

The [join(InetAddress,NetworkInterface,InetAddress)](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MulticastChannel.html#join%28java.net.InetAddress,%20java.net.NetworkInterface,%20java.net.InetAddress%29) method is used to begin receiving datagrams sent to a group whose source address matches a given source address. This method throws [UnsupportedOperationException](http://download.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html) if the underlying platform does not support source filtering. Membership is *cumulative* and this method may be invoked again with the same group and interface to allow receiving datagrams from other source addresses. The method returns a [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) that represents membership to receive datagrams from the given source address. Invoking the key's [drop](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#drop%28%29) method drops membership so that datagrams from the source address can no longer be received.

**Platform dependencies**

The multicast implementation is intended to map directly to the native multicasting facility. Consequently, the following items should be considered when developing an application that receives IP multicast datagrams:

* 1. The creation of the channel should specify the [ProtocolFamily](http://download.oracle.com/javase/7/docs/api/java/net/ProtocolFamily.html) that corresponds to the address type of the multicast groups that the channel will join. There is no guarantee that a channel to a socket in one protocol family can join and receive multicast datagrams when the address of the multicast group corresponds to another protocol family. For example, it is implementation specific if a channel to an [IPv6](http://download.oracle.com/javase/7/docs/api/java/net/StandardProtocolFamily.html#INET6) socket can join an [IPv4](http://download.oracle.com/javase/7/docs/api/java/net/StandardProtocolFamily.html#INET) multicast group and receive multicast datagrams sent to the group.
  2. The channel's socket should be bound to the [wildcard](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html#isAnyLocalAddress%28%29) address. If the socket is bound to a specific address, rather than the wildcard address then it is implementation specific if multicast datagrams are received by the socket.
  3. The [SO\_REUSEADDR](http://download.oracle.com/javase/7/docs/api/java/net/StandardSocketOption.html#SO_REUSEADDR) option should be enabled prior to [binding](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#bind%28java.net.SocketAddress%29) the socket. This is required to allow multiple members of the group to bind to the same address.

**Usage Example:**

// join multicast group on this interface, and also use this

// interface for outgoing multicast datagrams

NetworkInterface ni = NetworkInterface.getByName("hme0");

DatagramChannel dc = DatagramChannel.open(StandardProtocolFamily.INET)

.setOption(StandardSocketOption.SO\_REUSEADDR, true)

.bind(new InetSocketAddress(5000))

.setOption(StandardSocketOption.IP\_MULTICAST\_IF, ni);

InetAddress group = InetAddress.getByName("225.4.5.6");

MembershipKey key = dc.join(group, ni);

Since:

1.7

* + **Method Summary**

|  |  |
| --- | --- |
| Methods | |
| **Modifier and Type** | **Method and Description** |
| void | [**close**](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MulticastChannel.html#close%28%29)()  Closes this channel. |
| [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) | [**join**](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MulticastChannel.html#join%28java.net.InetAddress,%20java.net.NetworkInterface%29)([InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html" \o "class in java.net) group, [NetworkInterface](http://download.oracle.com/javase/7/docs/api/java/net/NetworkInterface.html) interf)  Joins a multicast group to begin receiving all datagrams sent to the group, returning a membership key. |
| [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) | [**join**](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MulticastChannel.html#join%28java.net.InetAddress,%20java.net.NetworkInterface,%20java.net.InetAddress%29)([InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html" \o "class in java.net) group, [NetworkInterface](http://download.oracle.com/javase/7/docs/api/java/net/NetworkInterface.html) interf, [InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html) source)  Joins a multicast group to begin receiving datagrams sent to the group from a given source address. |

* + - **Methods inherited from interface java.nio.channels.**[**NetworkChannel**](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html)

[bind](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#bind%28java.net.SocketAddress%29), [getLocalAddress](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#getLocalAddress%28%29), [getOption](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#getOption%28java.net.SocketOption%29), [setOption](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#setOption%28java.net.SocketOption,%20T%29), [supportedOptions](http://download.oracle.com/javase/7/docs/api/java/nio/channels/NetworkChannel.html#supportedOptions%28%29)

* + - **Methods inherited from interface java.nio.channels.**[**Channel**](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html)

[isOpen](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html#isOpen%28%29)

* + **Method Detail**
    - **close**
    - void close()

throws [IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html)

Closes this channel.

If the channel is a member of a multicast group then the membership is [dropped](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#drop%28%29). Upon return, the [membership-key](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) will be [invalid](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#isValid%28%29).

This method otherwise behaves exactly as specified by the [Channel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html) interface.

**Specified by:**

[close](http://download.oracle.com/javase/7/docs/api/java/lang/AutoCloseable.html#close%28%29) in interface [AutoCloseable](http://download.oracle.com/javase/7/docs/api/java/lang/AutoCloseable.html" \o "interface in java.lang)

**Specified by:**

[close](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html#close%28%29) in interface [Channel](http://download.oracle.com/javase/7/docs/api/java/nio/channels/Channel.html)

**Specified by:**

[close](http://download.oracle.com/javase/7/docs/api/java/io/Closeable.html#close%28%29) in interface [Closeable](http://download.oracle.com/javase/7/docs/api/java/io/Closeable.html)

Throws:

[IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html) - If an I/O error occurs

* + - **join**
    - [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) join([InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html" \o "class in java.net) group,
    - [NetworkInterface](http://download.oracle.com/javase/7/docs/api/java/net/NetworkInterface.html) interf)

throws [IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html)

Joins a multicast group to begin receiving all datagrams sent to the group, returning a membership key.

If this channel is currently a member of the group on the given interface to receive all datagrams then the membership key, representing that membership, is returned. Otherwise this channel joins the group and the resulting new membership key is returned. The resulting membership key is not [source-specific](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#sourceAddress%28%29).

A multicast channel may join several multicast groups, including the same group on more than one interface. An implementation may impose a limit on the number of groups that may be joined at the same time.

Parameters:

group - The multicast address to join

interf - The network interface on which to join the group

Returns:

The membership key

Throws:

[IllegalArgumentException](http://download.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - If the group parameter is not a [multicast](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html#isMulticastAddress%28%29) address, or the group parameter is an address type that is not supported by this channel

[IllegalStateException](http://download.oracle.com/javase/7/docs/api/java/lang/IllegalStateException.html) - If the channel already has source-specific membership of the group on the interface

[UnsupportedOperationException](http://download.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html) - If the channel's socket is not an Internet Protocol socket

[ClosedChannelException](http://download.oracle.com/javase/7/docs/api/java/nio/channels/ClosedChannelException.html) - If this channel is closed

[IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html) - If an I/O error occurs

[SecurityException](http://download.oracle.com/javase/7/docs/api/java/lang/SecurityException.html) - If a security manager is set, and its [checkMulticast](http://download.oracle.com/javase/7/docs/api/java/lang/SecurityManager.html#checkMulticast%28java.net.InetAddress%29) method denies access to the multiast group

* + - **join**
    - [MembershipKey](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html) join([InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html" \o "class in java.net) group,
    - [NetworkInterface](http://download.oracle.com/javase/7/docs/api/java/net/NetworkInterface.html) interf,
    - [InetAddress](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html) source)

throws [IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html)

Joins a multicast group to begin receiving datagrams sent to the group from a given source address.

If this channel is currently a member of the group on the given interface to receive datagrams from the given source address then the membership key, representing that membership, is returned. Otherwise this channel joins the group and the resulting new membership key is returned. The resulting membership key is [source-specific](http://download.oracle.com/javase/7/docs/api/java/nio/channels/MembershipKey.html#sourceAddress%28%29).

Membership is *cumulative* and this method may be invoked again with the same group and interface to allow receiving datagrams sent by other source addresses to the group.

Parameters:

group - The multicast address to join

interf - The network interface on which to join the group

source - The source address

Returns:

The membership key

Throws:

[IllegalArgumentException](http://download.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - If the group parameter is not a [multicast](http://download.oracle.com/javase/7/docs/api/java/net/InetAddress.html#isMulticastAddress%28%29) address, the source parameter is not a unicast address, the group parameter is an address type that is not supported by this channel, or the source parameter is not the same address type as the group

[IllegalStateException](http://download.oracle.com/javase/7/docs/api/java/lang/IllegalStateException.html) - If the channel is currently a member of the group on the given interface to receive all datagrams

[UnsupportedOperationException](http://download.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html) - If the channel's socket is not an Internet Protocol socket or source filtering is not supported

[ClosedChannelException](http://download.oracle.com/javase/7/docs/api/java/nio/channels/ClosedChannelException.html) - If this channel is closed

[IOException](http://download.oracle.com/javase/7/docs/api/java/io/IOException.html) - If an I/O error occurs

[SecurityException](http://download.oracle.com/javase/7/docs/api/java/lang/SecurityException.html) - If a security manager is set, and its [checkMulticast](http://download.oracle.com/javase/7/docs/api/java/lang/SecurityManager.html#checkMulticast%28java.net.InetAddress%29) method denies access to the multiast group