| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NetworkInterface.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/NetPermission.html)   [**NEXT CLASS**](http://docs.google.com/java/net/NoRouteToHostException.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/NetworkInterface.html)    [**NO FRAMES**](http://docs.google.com/NetworkInterface.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.net**

Class NetworkInterface

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

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

This class represents a Network Interface made up of a name, and a list of IP addresses assigned to this interface. It is used to identify the local interface on which a multicast group is joined. Interfaces are normally known by names such as "le0".

**Since:** 1.4

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/net/NetworkInterface.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this object against the specified object. |
| static [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) | [**getByInetAddress**](http://docs.google.com/java/net/NetworkInterface.html#getByInetAddress(java.net.InetAddress))([InetAddress](http://docs.google.com/java/net/InetAddress.html) addr)            Convenience method to search for a network interface that has the specified Internet Protocol (IP) address bound to it. |
| static [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) | [**getByName**](http://docs.google.com/java/net/NetworkInterface.html#getByName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Searches for the network interface with the specified name. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/net/NetworkInterface.html#getDisplayName())()            Get the display name of this network interface. |
| byte[] | [**getHardwareAddress**](http://docs.google.com/java/net/NetworkInterface.html#getHardwareAddress())()            Returns the hardware address (usually MAC) of the interface if it has one and if it can be accessed given the current privileges. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[InetAddress](http://docs.google.com/java/net/InetAddress.html)> | [**getInetAddresses**](http://docs.google.com/java/net/NetworkInterface.html#getInetAddresses())()            Convenience method to return an Enumeration with all or a subset of the InetAddresses bound to this network interface. |
| [List](http://docs.google.com/java/util/List.html)<[InterfaceAddress](http://docs.google.com/java/net/InterfaceAddress.html)> | [**getInterfaceAddresses**](http://docs.google.com/java/net/NetworkInterface.html#getInterfaceAddresses())()            Get a List of all or a subset of the InterfaceAddresses of this network interface. |
| int | [**getMTU**](http://docs.google.com/java/net/NetworkInterface.html#getMTU())()            Returns the Maximum Transmission Unit (MTU) of this interface. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/net/NetworkInterface.html#getName())()            Get the name of this network interface. |
| static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html)> | [**getNetworkInterfaces**](http://docs.google.com/java/net/NetworkInterface.html#getNetworkInterfaces())()            Returns all the interfaces on this machine. |
| [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) | [**getParent**](http://docs.google.com/java/net/NetworkInterface.html#getParent())()            Returns the parent NetworkInterface of this interface if this is a subinterface, or null if it is a physical (non virtual) interface or has no parent. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html)> | [**getSubInterfaces**](http://docs.google.com/java/net/NetworkInterface.html#getSubInterfaces())()            Get an Enumeration with all the subinterfaces (also known as virtual interfaces) attached to this network interface. |
| int | [**hashCode**](http://docs.google.com/java/net/NetworkInterface.html#hashCode())()            Returns a hash code value for the object. |
| boolean | [**isLoopback**](http://docs.google.com/java/net/NetworkInterface.html#isLoopback())()            Returns whether a network interface is a loopback interface. |
| boolean | [**isPointToPoint**](http://docs.google.com/java/net/NetworkInterface.html#isPointToPoint())()            Returns whether a network interface is a point to point interface. |
| boolean | [**isUp**](http://docs.google.com/java/net/NetworkInterface.html#isUp())()            Returns whether a network interface is up and running. |
| boolean | [**isVirtual**](http://docs.google.com/java/net/NetworkInterface.html#isVirtual())()            Returns whether this interface is a virtual interface (also called subinterface). |
| boolean | [**supportsMulticast**](http://docs.google.com/java/net/NetworkInterface.html#supportsMulticast())()            Returns whether a network interface supports multicasting or not. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/net/NetworkInterface.html#toString())()            Returns a string representation of the object. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [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)) |

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

### getName

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

Get the name of this network interface.

**Returns:**the name of this network interface

### getInetAddresses

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[InetAddress](http://docs.google.com/java/net/InetAddress.html)> **getInetAddresses**()

Convenience method to return an Enumeration with all or a subset of the InetAddresses bound to this network interface.

If there is a security manager, its checkConnect method is called for each InetAddress. Only InetAddresses where the checkConnect doesn't throw a SecurityException will be returned in the Enumeration.

**Returns:**an Enumeration object with all or a subset of the InetAddresses bound to this network interface

### getInterfaceAddresses

public [List](http://docs.google.com/java/util/List.html)<[InterfaceAddress](http://docs.google.com/java/net/InterfaceAddress.html)> **getInterfaceAddresses**()

Get a List of all or a subset of the InterfaceAddresses of this network interface.

If there is a security manager, its checkConnect method is called with the InetAddress for each InterfaceAddress. Only InterfaceAddresses where the checkConnect doesn't throw a SecurityException will be returned in the List.

**Returns:**a List object with all or a subset of the InterfaceAddresss of this network interface**Since:** 1.6

### getSubInterfaces

public [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html)> **getSubInterfaces**()

Get an Enumeration with all the subinterfaces (also known as virtual interfaces) attached to this network interface.

For instance eth0:1 will be a subinterface to eth0.

**Returns:**an Enumeration object with all of the subinterfaces of this network interface**Since:** 1.6

### getParent

public [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) **getParent**()

Returns the parent NetworkInterface of this interface if this is a subinterface, or null if it is a physical (non virtual) interface or has no parent.

**Returns:**The NetworkInterface this interface is attached to.**Since:** 1.6

### getDisplayName

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

Get the display name of this network interface. A display name is a human readable String describing the network device.

**Returns:**the display name of this network interface, or null if no display name is available.

### getByName

public static [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) **getByName**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Searches for the network interface with the specified name.

**Parameters:**name - The name of the network interface. **Returns:**A NetworkInterface with the specified name, or null if there is no network interface with the specified name. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - If an I/O error occurs. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the specified name is null.

### getByInetAddress

public static [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) **getByInetAddress**([InetAddress](http://docs.google.com/java/net/InetAddress.html) addr)  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Convenience method to search for a network interface that has the specified Internet Protocol (IP) address bound to it.

If the specified IP address is bound to multiple network interfaces it is not defined which network interface is returned.

**Parameters:**addr - The InetAddress to search with. **Returns:**A NetworkInterface or null if there is no network interface with the specified IP address. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - If an I/O error occurs. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the specified address is null.

### getNetworkInterfaces

public static [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html)> **getNetworkInterfaces**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns all the interfaces on this machine. Returns null if no network interfaces could be found on this machine. NOTE: can use getNetworkInterfaces()+getInetAddresses() to obtain all IP addresses for this node

**Returns:**an Enumeration of NetworkInterfaces found on this machine **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.

### isUp

public boolean **isUp**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns whether a network interface is up and running.

**Returns:**true if the interface is up and running. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### isLoopback

public boolean **isLoopback**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns whether a network interface is a loopback interface.

**Returns:**true if the interface is a loopback interface. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### isPointToPoint

public boolean **isPointToPoint**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns whether a network interface is a point to point interface. A typical point to point interface would be a PPP connection through a modem.

**Returns:**true if the interface is a point to point interface. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### supportsMulticast

public boolean **supportsMulticast**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns whether a network interface supports multicasting or not.

**Returns:**true if the interface supports Multicasting. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### getHardwareAddress

public byte[] **getHardwareAddress**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns the hardware address (usually MAC) of the interface if it has one and if it can be accessed given the current privileges.

**Returns:**a byte array containing the address or null if the address doesn't exist or is not accessible. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### getMTU

public int **getMTU**()  
 throws [SocketException](http://docs.google.com/java/net/SocketException.html)

Returns the Maximum Transmission Unit (MTU) of this interface.

**Returns:**the value of the MTU for that interface. **Throws:** [SocketException](http://docs.google.com/java/net/SocketException.html) - if an I/O error occurs.**Since:** 1.6

### isVirtual

public boolean **isVirtual**()

Returns whether this interface is a virtual interface (also called subinterface). Virtual interfaces are, on some systems, interfaces created as a child of a physical interface and given different settings (like address or MTU). Usually the name of the interface will the name of the parent followed by a colon (:) and a number identifying the child since there can be several virtual interfaces attached to a single physical interface.

**Returns:**true if this interface is a virtual interface.**Since:** 1.6

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Compares this object against the specified object. The result is true if and only if the argument is not null and it represents the same NetworkInterface as this object.

Two instances of NetworkInterface represent the same NetworkInterface if both name and addrs are the same for both.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object to compare against. **Returns:**true if the objects are the same; false otherwise.**See Also:**[InetAddress.getAddress()](http://docs.google.com/java/net/InetAddress.html#getAddress())

### hashCode

public int **hashCode**()

**Description copied from class:** [**Object**](http://docs.google.com/java/lang/Object.html#hashCode()) Returns a hash code value for the object. This method is supported for the benefit of hashtables such as those provided by java.util.Hashtable.

The general contract of hashCode is:

* Whenever it is invoked on the same object more than once during an execution of a Java application, the hashCode method must consistently return the same integer, provided no information used in equals comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application.
* If two objects are equal according to the equals(Object) method, then calling the hashCode method on each of the two objects must produce the same integer result.
* It is *not* required that if two objects are unequal according to the [Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) method, then calling the hashCode method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

As much as is reasonably practical, the hashCode method defined by class Object does return distinct integers for distinct objects. (This is typically implemented by converting the internal address of the object into an integer, but this implementation technique is not required by the JavaTM programming language.)

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

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

**Description copied from class:** [**Object**](http://docs.google.com/java/lang/Object.html#toString()) Returns a string representation of the object. In general, the toString method returns a string that "textually represents" this object. The result should be a concise but informative representation that is easy for a person to read. It is recommended that all subclasses override this method.

The toString method for class Object returns a string consisting of the name of the class of which the object is an instance, the at-sign character `@', and the unsigned hexadecimal representation of the hash code of the object. In other words, this method returns a string equal to the value of:

getClass().getName() + '@' + Integer.toHexString(hashCode())

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the object.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NetworkInterface.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/NetPermission.html)   [**NEXT CLASS**](http://docs.google.com/java/net/NoRouteToHostException.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/NetworkInterface.html)    [**NO FRAMES**](http://docs.google.com/NetworkInterface.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).