| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Inet6Address.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/Inet4Address.html)   [**NEXT CLASS**](http://docs.google.com/java/net/InetAddress.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/Inet6Address.html)    [**NO FRAMES**](http://docs.google.com/Inet6Address.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3dy6vkm) | DETAIL: FIELD | CONSTR | [METHOD](#2s8eyo1) |

## **java.net**

Class Inet6Address

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.net.InetAddress](http://docs.google.com/java/net/InetAddress.html)  
 **java.net.Inet6Address**

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

public final class **Inet6Address**extends [InetAddress](http://docs.google.com/java/net/InetAddress.html)

This class represents an Internet Protocol version 6 (IPv6) address. Defined by  [*RFC 2373: IP Version 6 Addressing Architecture*](http://www.ietf.org/rfc/rfc2373.txt).

#### Textual representation of IP addresses

Textual representation of IPv6 address used as input to methods takes one of the following forms:

The preferred form is x:x:x:x:x:x:x:x, where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address. This is the full form. For example,

| 1080:0:0:0:8:800:200C:417A |  |
| --- | --- |

1. Note that it is not necessary to write the leading zeros in an individual field. However, there must be at least one numeral in every field, except as described below.

Due to some methods of allocating certain styles of IPv6 addresses, it will be common for addresses to contain long strings of zero bits. In order to make writing addresses containing zero bits easier, a special syntax is available to compress the zeros. The use of "::" indicates multiple groups of 16-bits of zeros. The "::" can only appear once in an address. The "::" can also be used to compress the leading and/or trailing zeros in an address. For example,

| 1080::8:800:200C:417A |  |
| --- | --- |

An alternative form that is sometimes more convenient when dealing with a mixed environment of IPv4 and IPv6 nodes is x:x:x:x:x:x:d.d.d.d, where the 'x's are the hexadecimal values of the six high-order 16-bit pieces of the address, and the 'd's are the decimal values of the four low-order 8-bit pieces of the standard IPv4 representation address, for example,

| ::FFFF:129.144.52.38 |  |
| --- | --- |
| ::129.144.52.38 |  |

where "::FFFF:d.d.d.d" and "::d.d.d.d" are, respectively, the general forms of an IPv4-mapped IPv6 address and an IPv4-compatible IPv6 address. Note that the IPv4 portion must be in the "d.d.d.d" form. The following forms are invalid:

| ::FFFF:d.d.d |  |
| --- | --- |
| ::FFFF:d.d |  |
| ::d.d.d |  |
| ::d.d |  |

The following form:

| ::FFFF:d |  |
| --- | --- |

is valid, however it is an unconventional representation of the IPv4-compatible IPv6 address,

| ::255.255.0.d |  |
| --- | --- |

1. while "::d" corresponds to the general IPv6 address "0:0:0:0:0:0:0:d".

For methods that return a textual representation as output value, the full form is used. Inet6Address will return the full form because it is unambiguous when used in combination with other textual data.

#### Special IPv6 address

| *IPv4-mapped address* | Of the form::ffff:w.x.y.z, this IPv6 address is used to represent an IPv4 address. It allows the native program to use the same address data structure and also the same socket when communicating with both IPv4 and IPv6 nodes.  In InetAddress and Inet6Address, it is used for internal representation; it has no functional role. Java will never return an IPv4-mapped address. These classes can take an IPv4-mapped address as input, both in byte array and text representation. However, it will be converted into an IPv4 address. |
| --- | --- |

#### Textual representation of IPv6 scoped addresses

The textual representation of IPv6 addresses as described above can be extended to specify IPv6 scoped addresses. This extension to the basic addressing architecture is described in [draft-ietf-ipngwg-scoping-arch-04.txt].

Because link-local and site-local addresses are non-global, it is possible that different hosts may have the same destination address and may be reachable through different interfaces on the same originating system. In this case, the originating system is said to be connected to multiple zones of the same scope. In order to disambiguate which is the intended destination zone, it is possible to append a zone identifier (or *scope\_id*) to an IPv6 address.

The general format for specifying the *scope\_id* is the following:

*IPv6-address*%*scope\_id*

The IPv6-address is a literal IPv6 address as described above. The *scope\_id* refers to an interface on the local system, and it can be specified in two ways.

1. *As a numeric identifier.* This must be a positive integer that identifies the particular interface and scope as understood by the system. Usually, the numeric values can be determined through administration tools on the system. Each interface may have multiple values, one for each scope. If the scope is unspecified, then the default value used is zero.
2. *As a string.* This must be the exact string that is returned by [NetworkInterface.getName()](http://docs.google.com/java/net/NetworkInterface.html#getName()) for the particular interface in question. When an Inet6Address is created in this way, the numeric scope-id is determined at the time the object is created by querying the relevant NetworkInterface.

Note also, that the numeric *scope\_id* can be retrieved from Inet6Address instances returned from the NetworkInterface class. This can be used to find out the current scope ids configured on the system.

**Since:** 1.4 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.net.Inet6Address)

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/net/Inet6Address.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this object against the specified object. |
| byte[] | [**getAddress**](http://docs.google.com/java/net/Inet6Address.html#getAddress())()            Returns the raw IP address of this InetAddress object. |
| static [Inet6Address](http://docs.google.com/java/net/Inet6Address.html) | [**getByAddress**](http://docs.google.com/java/net/Inet6Address.html#getByAddress(java.lang.String,%20byte%5B%5D,%20int))([String](http://docs.google.com/java/lang/String.html) host, byte[] addr, int scope\_id)            Create an Inet6Address in the exact manner of [InetAddress.getByAddress(String,byte[])](http://docs.google.com/java/net/InetAddress.html#getByAddress(java.lang.String,%20byte%5B%5D)) except that the IPv6 scope\_id is set to the given numeric value. |
| static [Inet6Address](http://docs.google.com/java/net/Inet6Address.html) | [**getByAddress**](http://docs.google.com/java/net/Inet6Address.html#getByAddress(java.lang.String,%20byte%5B%5D,%20java.net.NetworkInterface))([String](http://docs.google.com/java/lang/String.html) host, byte[] addr, [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) nif)            Create an Inet6Address in the exact manner of [InetAddress.getByAddress(String,byte[])](http://docs.google.com/java/net/InetAddress.html#getByAddress(java.lang.String,%20byte%5B%5D)) except that the IPv6 scope\_id is set to the value corresponding to the given interface for the address type specified in addr. |
| [String](http://docs.google.com/java/lang/String.html) | [**getHostAddress**](http://docs.google.com/java/net/Inet6Address.html#getHostAddress())()            Returns the IP address string in textual presentation. |
| [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) | [**getScopedInterface**](http://docs.google.com/java/net/Inet6Address.html#getScopedInterface())()            Returns the scoped interface, if this instance was created with with a scoped interface. |
| int | [**getScopeId**](http://docs.google.com/java/net/Inet6Address.html#getScopeId())()            Returns the numeric scopeId, if this instance is associated with an interface. |
| int | [**hashCode**](http://docs.google.com/java/net/Inet6Address.html#hashCode())()            Returns a hashcode for this IP address. |
| boolean | [**isAnyLocalAddress**](http://docs.google.com/java/net/Inet6Address.html#isAnyLocalAddress())()            Utility routine to check if the InetAddress in a wildcard address. |
| boolean | [**isIPv4CompatibleAddress**](http://docs.google.com/java/net/Inet6Address.html#isIPv4CompatibleAddress())()            Utility routine to check if the InetAddress is an IPv4 compatible IPv6 address. |
| boolean | [**isLinkLocalAddress**](http://docs.google.com/java/net/Inet6Address.html#isLinkLocalAddress())()            Utility routine to check if the InetAddress is an link local address. |
| boolean | [**isLoopbackAddress**](http://docs.google.com/java/net/Inet6Address.html#isLoopbackAddress())()            Utility routine to check if the InetAddress is a loopback address. |
| boolean | [**isMCGlobal**](http://docs.google.com/java/net/Inet6Address.html#isMCGlobal())()            Utility routine to check if the multicast address has global scope. |
| boolean | [**isMCLinkLocal**](http://docs.google.com/java/net/Inet6Address.html#isMCLinkLocal())()            Utility routine to check if the multicast address has link scope. |
| boolean | [**isMCNodeLocal**](http://docs.google.com/java/net/Inet6Address.html#isMCNodeLocal())()            Utility routine to check if the multicast address has node scope. |
| boolean | [**isMCOrgLocal**](http://docs.google.com/java/net/Inet6Address.html#isMCOrgLocal())()            Utility routine to check if the multicast address has organization scope. |
| boolean | [**isMCSiteLocal**](http://docs.google.com/java/net/Inet6Address.html#isMCSiteLocal())()            Utility routine to check if the multicast address has site scope. |
| boolean | [**isMulticastAddress**](http://docs.google.com/java/net/Inet6Address.html#isMulticastAddress())()            Utility routine to check if the InetAddress is an IP multicast address. |
| boolean | [**isSiteLocalAddress**](http://docs.google.com/java/net/Inet6Address.html#isSiteLocalAddress())()            Utility routine to check if the InetAddress is a site local address. |

| **Methods inherited from class java.net.**[**InetAddress**](http://docs.google.com/java/net/InetAddress.html) |
| --- |
| [getAllByName](http://docs.google.com/java/net/InetAddress.html#getAllByName(java.lang.String)), [getByAddress](http://docs.google.com/java/net/InetAddress.html#getByAddress(byte%5B%5D)), [getByAddress](http://docs.google.com/java/net/InetAddress.html#getByAddress(java.lang.String,%20byte%5B%5D)), [getByName](http://docs.google.com/java/net/InetAddress.html#getByName(java.lang.String)), [getCanonicalHostName](http://docs.google.com/java/net/InetAddress.html#getCanonicalHostName()), [getHostName](http://docs.google.com/java/net/InetAddress.html#getHostName()), [getLocalHost](http://docs.google.com/java/net/InetAddress.html#getLocalHost()), [isReachable](http://docs.google.com/java/net/InetAddress.html#isReachable(int)), [isReachable](http://docs.google.com/java/net/InetAddress.html#isReachable(java.net.NetworkInterface,%20int,%20int)), [toString](http://docs.google.com/java/net/InetAddress.html#toString()) |

| **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** |
| --- |

### getByAddress

public static [Inet6Address](http://docs.google.com/java/net/Inet6Address.html) **getByAddress**([String](http://docs.google.com/java/lang/String.html) host,  
 byte[] addr,  
 [NetworkInterface](http://docs.google.com/java/net/NetworkInterface.html) nif)  
 throws [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html)

Create an Inet6Address in the exact manner of [InetAddress.getByAddress(String,byte[])](http://docs.google.com/java/net/InetAddress.html#getByAddress(java.lang.String,%20byte%5B%5D)) except that the IPv6 scope\_id is set to the value corresponding to the given interface for the address type specified in addr. The call will fail with an UnknownHostException if the given interface does not have a numeric scope\_id assigned for the given address type (eg. link-local or site-local). See [here](http://docs.google.com/Inet6Address.html#scoped) for a description of IPv6 scoped addresses.

**Parameters:**host - the specified hostaddr - the raw IP address in network byte ordernif - an interface this address must be associated with. **Returns:**an Inet6Address object created from the raw IP address. **Throws:** [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html) - if IP address is of illegal length, or if the interface does not have a numeric scope\_id assigned for the given address type.**Since:** 1.5

### getByAddress

public static [Inet6Address](http://docs.google.com/java/net/Inet6Address.html) **getByAddress**([String](http://docs.google.com/java/lang/String.html) host,  
 byte[] addr,  
 int scope\_id)  
 throws [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html)

Create an Inet6Address in the exact manner of [InetAddress.getByAddress(String,byte[])](http://docs.google.com/java/net/InetAddress.html#getByAddress(java.lang.String,%20byte%5B%5D)) except that the IPv6 scope\_id is set to the given numeric value. The scope\_id is not checked to determine if it corresponds to any interface on the system. See [here](http://docs.google.com/Inet6Address.html#scoped) for a description of IPv6 scoped addresses.

**Parameters:**host - the specified hostaddr - the raw IP address in network byte orderscope\_id - the numeric scope\_id for the address. **Returns:**an Inet6Address object created from the raw IP address. **Throws:** [UnknownHostException](http://docs.google.com/java/net/UnknownHostException.html) - if IP address is of illegal length.**Since:** 1.5

### isMulticastAddress

public boolean **isMulticastAddress**()

Utility routine to check if the InetAddress is an IP multicast address. 11111111 at the start of the address identifies the address as being a multicast address.

**Overrides:**[isMulticastAddress](http://docs.google.com/java/net/InetAddress.html#isMulticastAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the InetAddress is an IP multicast address**Since:** JDK1.1

### isAnyLocalAddress

public boolean **isAnyLocalAddress**()

Utility routine to check if the InetAddress in a wildcard address.

**Overrides:**[isAnyLocalAddress](http://docs.google.com/java/net/InetAddress.html#isAnyLocalAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the Inetaddress is a wildcard address.**Since:** 1.4

### isLoopbackAddress

public boolean **isLoopbackAddress**()

Utility routine to check if the InetAddress is a loopback address.

**Overrides:**[isLoopbackAddress](http://docs.google.com/java/net/InetAddress.html#isLoopbackAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the InetAddress is a loopback address; or false otherwise.**Since:** 1.4

### isLinkLocalAddress

public boolean **isLinkLocalAddress**()

Utility routine to check if the InetAddress is an link local address.

**Overrides:**[isLinkLocalAddress](http://docs.google.com/java/net/InetAddress.html#isLinkLocalAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the InetAddress is a link local address; or false if address is not a link local unicast address.**Since:** 1.4

### isSiteLocalAddress

public boolean **isSiteLocalAddress**()

Utility routine to check if the InetAddress is a site local address.

**Overrides:**[isSiteLocalAddress](http://docs.google.com/java/net/InetAddress.html#isSiteLocalAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the InetAddress is a site local address; or false if address is not a site local unicast address.**Since:** 1.4

### isMCGlobal

public boolean **isMCGlobal**()

Utility routine to check if the multicast address has global scope.

**Overrides:**[isMCGlobal](http://docs.google.com/java/net/InetAddress.html#isMCGlobal()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the address has is a multicast address of global scope, false if it is not of global scope or it is not a multicast address**Since:** 1.4

### isMCNodeLocal

public boolean **isMCNodeLocal**()

Utility routine to check if the multicast address has node scope.

**Overrides:**[isMCNodeLocal](http://docs.google.com/java/net/InetAddress.html#isMCNodeLocal()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the address has is a multicast address of node-local scope, false if it is not of node-local scope or it is not a multicast address**Since:** 1.4

### isMCLinkLocal

public boolean **isMCLinkLocal**()

Utility routine to check if the multicast address has link scope.

**Overrides:**[isMCLinkLocal](http://docs.google.com/java/net/InetAddress.html#isMCLinkLocal()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the address has is a multicast address of link-local scope, false if it is not of link-local scope or it is not a multicast address**Since:** 1.4

### isMCSiteLocal

public boolean **isMCSiteLocal**()

Utility routine to check if the multicast address has site scope.

**Overrides:**[isMCSiteLocal](http://docs.google.com/java/net/InetAddress.html#isMCSiteLocal()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the address has is a multicast address of site-local scope, false if it is not of site-local scope or it is not a multicast address**Since:** 1.4

### isMCOrgLocal

public boolean **isMCOrgLocal**()

Utility routine to check if the multicast address has organization scope.

**Overrides:**[isMCOrgLocal](http://docs.google.com/java/net/InetAddress.html#isMCOrgLocal()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a boolean indicating if the address has is a multicast address of organization-local scope, false if it is not of organization-local scope or it is not a multicast address**Since:** 1.4

### getAddress

public byte[] **getAddress**()

Returns the raw IP address of this InetAddress object. The result is in network byte order: the highest order byte of the address is in getAddress()[0].

**Overrides:**[getAddress](http://docs.google.com/java/net/InetAddress.html#getAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**the raw IP address of this object.

### getScopeId

public int **getScopeId**()

Returns the numeric scopeId, if this instance is associated with an interface. If no scoped\_id is set, the returned value is zero.

**Returns:**the scopeId, or zero if not set.**Since:** 1.5

### getScopedInterface

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

Returns the scoped interface, if this instance was created with with a scoped interface.

**Returns:**the scoped interface, or null if not set.**Since:** 1.5

### getHostAddress

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

Returns the IP address string in textual presentation. If the instance was created specifying a scope identifier then the scope id is appended to the IP address preceded by a "%" (per-cent) character. This can be either a numeric value or a string, depending on which was used to createthe instance.

**Overrides:**[getHostAddress](http://docs.google.com/java/net/InetAddress.html#getHostAddress()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**the raw IP address in a string format.

### hashCode

public int **hashCode**()

Returns a hashcode for this IP address.

**Overrides:**[hashCode](http://docs.google.com/java/net/InetAddress.html#hashCode()) in class [InetAddress](http://docs.google.com/java/net/InetAddress.html) **Returns:**a hash code value for this IP address.**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)

### 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 IP address as this object.

Two instances of InetAddress represent the same IP address if the length of the byte arrays returned by getAddress is the same for both, and each of the array components is the same for the byte arrays.

**Overrides:**[equals](http://docs.google.com/java/net/InetAddress.html#equals(java.lang.Object)) in class [InetAddress](http://docs.google.com/java/net/InetAddress.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())

### isIPv4CompatibleAddress

public boolean **isIPv4CompatibleAddress**()

Utility routine to check if the InetAddress is an IPv4 compatible IPv6 address.

**Returns:**a boolean indicating if the InetAddress is an IPv4 compatible IPv6 address; or false if address is IPv4 address.**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Inet6Address.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/Inet4Address.html)   [**NEXT CLASS**](http://docs.google.com/java/net/InetAddress.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/Inet6Address.html)    [**NO FRAMES**](http://docs.google.com/Inet6Address.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3dy6vkm) | DETAIL: FIELD | CONSTR | [METHOD](#2s8eyo1) |

[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).