| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InetSocketAddress.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/InetAddress.html)   [**NEXT CLASS**](http://docs.google.com/java/net/InterfaceAddress.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/InetSocketAddress.html)    [**NO FRAMES**](http://docs.google.com/InetSocketAddress.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

## **java.net**

Class InetSocketAddress

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

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

public class **InetSocketAddress**extends [SocketAddress](http://docs.google.com/java/net/SocketAddress.html)

This class implements an IP Socket Address (IP address + port number) It can also be a pair (hostname + port number), in which case an attempt will be made to resolve the hostname. If resolution fails then the address is said to be *unresolved* but can still be used on some circumstances like connecting through a proxy.

It provides an immutable object used by sockets for binding, connecting, or as returned values.

The *wildcard* is a special local IP address. It usually means "any" and can only be used for bind operations.

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

| **Constructor Summary** | |
| --- | --- |
| [**InetSocketAddress**](http://docs.google.com/java/net/InetSocketAddress.html#InetSocketAddress(java.net.InetAddress,%20int))([InetAddress](http://docs.google.com/java/net/InetAddress.html) addr, int port)            Creates a socket address from an IP address and a port number. |
| [**InetSocketAddress**](http://docs.google.com/java/net/InetSocketAddress.html#InetSocketAddress(int))(int port)            Creates a socket address where the IP address is the wildcard address and the port number a specified value. |
| [**InetSocketAddress**](http://docs.google.com/java/net/InetSocketAddress.html#InetSocketAddress(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) hostname, int port)            Creates a socket address from a hostname and a port number. |

| **Method Summary** | |
| --- | --- |
| static [InetSocketAddress](http://docs.google.com/java/net/InetSocketAddress.html) | [**createUnresolved**](http://docs.google.com/java/net/InetSocketAddress.html#createUnresolved(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) host, int port)            Creates an unresolved socket address from a hostname and a port number. |
| boolean | [**equals**](http://docs.google.com/java/net/InetSocketAddress.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this object against the specified object. |
| [InetAddress](http://docs.google.com/java/net/InetAddress.html) | [**getAddress**](http://docs.google.com/java/net/InetSocketAddress.html#getAddress())()            Gets the InetAddress. |
| [String](http://docs.google.com/java/lang/String.html) | [**getHostName**](http://docs.google.com/java/net/InetSocketAddress.html#getHostName())()            Gets the hostname. |
| int | [**getPort**](http://docs.google.com/java/net/InetSocketAddress.html#getPort())()            Gets the port number. |
| int | [**hashCode**](http://docs.google.com/java/net/InetSocketAddress.html#hashCode())()            Returns a hashcode for this socket address. |
| boolean | [**isUnresolved**](http://docs.google.com/java/net/InetSocketAddress.html#isUnresolved())()            Checks whether the address has been resolved or not. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/net/InetSocketAddress.html#toString())()            Constructs a string representation of this InetSocketAddress. |

| **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)) |

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

### InetSocketAddress

public **InetSocketAddress**(int port)

Creates a socket address where the IP address is the wildcard address and the port number a specified value.

A valid port value is between 0 and 65535. A port number of zero will let the system pick up an ephemeral port in a bind operation.

**Parameters:**port - The port number **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the port parameter is outside the specified range of valid port values.

### InetSocketAddress

public **InetSocketAddress**([InetAddress](http://docs.google.com/java/net/InetAddress.html) addr,  
 int port)

Creates a socket address from an IP address and a port number.

A valid port value is between 0 and 65535. A port number of zero will let the system pick up an ephemeral port in a bind operation.

A null address will assign the *wildcard* address.

**Parameters:**addr - The IP addressport - The port number **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the port parameter is outside the specified range of valid port values.

### InetSocketAddress

public **InetSocketAddress**([String](http://docs.google.com/java/lang/String.html) hostname,  
 int port)

Creates a socket address from a hostname and a port number.

An attempt will be made to resolve the hostname into an InetAddress. If that attempt fails, the address will be flagged as *unresolved*.

If there is a security manager, its checkConnect method is called with the host name as its argument to check the permissiom to resolve it. This could result in a SecurityException.

A valid port value is between 0 and 65535. A port number of zero will let the system pick up an ephemeral port in a bind operation.

**Parameters:**hostname - the Host nameport - The port number **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the port parameter is outside the range of valid port values, or if the hostname parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager is present and permission to resolve the host name is denied.**See Also:**[isUnresolved()](http://docs.google.com/java/net/InetSocketAddress.html#isUnresolved())

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

### createUnresolved

public static [InetSocketAddress](http://docs.google.com/java/net/InetSocketAddress.html) **createUnresolved**([String](http://docs.google.com/java/lang/String.html) host,  
 int port)

Creates an unresolved socket address from a hostname and a port number.

No attempt will be made to resolve the hostname into an InetAddress. The address will be flagged as *unresolved*.

A valid port value is between 0 and 65535. A port number of zero will let the system pick up an ephemeral port in a bind operation.

**Parameters:**host - the Host nameport - The port number **Returns:**a InetSocketAddress representing the unresolved socket address **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the port parameter is outside the range of valid port values, or if the hostname parameter is null.**Since:** 1.5 **See Also:**[isUnresolved()](http://docs.google.com/java/net/InetSocketAddress.html#isUnresolved())

### getPort

public final int **getPort**()

Gets the port number.

**Returns:**the port number.

### getAddress

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

Gets the InetAddress.

**Returns:**the InetAdress or null if it is unresolved.

### getHostName

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

Gets the hostname.

**Returns:**the hostname part of the address.

### isUnresolved

public final boolean **isUnresolved**()

Checks whether the address has been resolved or not.

**Returns:**true if the hostname couldn't be resolved into an InetAddress.

### toString

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

Constructs a string representation of this InetSocketAddress. This String is constructed by calling toString() on the InetAddress and concatenating the port number (with a colon). If the address is unresolved then the part before the colon will only contain the hostname.

**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 this object.

### equals

public final 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 address as this object.

Two instances of InetSocketAddress represent the same address if both the InetAddresses (or hostnames if it is unresolved) and port numbers are equal. If both addresses are unresolved, then the hostname & the port number are compared.

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

### hashCode

public final int **hashCode**()

Returns a hashcode for this socket address.

**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 socket 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)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InetSocketAddress.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/InetAddress.html)   [**NEXT CLASS**](http://docs.google.com/java/net/InterfaceAddress.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/InetSocketAddress.html)    [**NO FRAMES**](http://docs.google.com/InetSocketAddress.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

[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).