| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/HttpCookie.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/FileNameMap.html)   [**NEXT CLASS**](http://docs.google.com/java/net/HttpRetryException.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/HttpCookie.html)    [**NO FRAMES**](http://docs.google.com/HttpCookie.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.net**

Class HttpCookie

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

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

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

An HttpCookie object represents an http cookie, which carries state information between server and user agent. Cookie is widely adopted to create stateful sessions.

There are 3 http cookie specifications:

Netscape draft

RFC 2109 -  [*http://www.ietf.org/rfc/rfc2109.txt*](http://www.ietf.org/rfc/rfc2109.txt)

RFC 2965 -  [*http://www.ietf.org/rfc/rfc2965.txt*](http://www.ietf.org/rfc/rfc2965.txt)

HttpCookie class can accept all these 3 forms of syntax.

**Since:** 1.6

| **Constructor Summary** | |
| --- | --- |
| [**HttpCookie**](http://docs.google.com/java/net/HttpCookie.html#HttpCookie(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) value)            Constructs a cookie with a specified name and value. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/net/HttpCookie.html#clone())()            Create and return a copy of this object. |
| static boolean | [**domainMatches**](http://docs.google.com/java/net/HttpCookie.html#domainMatches(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) domain, [String](http://docs.google.com/java/lang/String.html) host)            The utility method to check whether a host name is in a domain or not. |
| boolean | [**equals**](http://docs.google.com/java/net/HttpCookie.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Test the equality of two http cookies. |
| [String](http://docs.google.com/java/lang/String.html) | [**getComment**](http://docs.google.com/java/net/HttpCookie.html#getComment())()            Returns the comment describing the purpose of this cookie, or null if the cookie has no comment. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCommentURL**](http://docs.google.com/java/net/HttpCookie.html#getCommentURL())()            Returns the comment url describing the purpose of this cookie, or null if the cookie has no comment url. |
| boolean | [**getDiscard**](http://docs.google.com/java/net/HttpCookie.html#getDiscard())()            Return the discard attribute of the cookie |
| [String](http://docs.google.com/java/lang/String.html) | [**getDomain**](http://docs.google.com/java/net/HttpCookie.html#getDomain())()            Returns the domain name set for this cookie. |
| long | [**getMaxAge**](http://docs.google.com/java/net/HttpCookie.html#getMaxAge())()            Returns the maximum age of the cookie, specified in seconds. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/net/HttpCookie.html#getName())()            Returns the name of the cookie. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPath**](http://docs.google.com/java/net/HttpCookie.html#getPath())()            Returns the path on the server to which the browser returns this cookie. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPortlist**](http://docs.google.com/java/net/HttpCookie.html#getPortlist())()            Return the port list attribute of the cookie |
| boolean | [**getSecure**](http://docs.google.com/java/net/HttpCookie.html#getSecure())()            Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol. |
| [String](http://docs.google.com/java/lang/String.html) | [**getValue**](http://docs.google.com/java/net/HttpCookie.html#getValue())()            Returns the value of the cookie. |
| int | [**getVersion**](http://docs.google.com/java/net/HttpCookie.html#getVersion())()            Returns the version of the protocol this cookie complies with. |
| boolean | [**hasExpired**](http://docs.google.com/java/net/HttpCookie.html#hasExpired())()            Reports whether this http cookie has expired or not. |
| int | [**hashCode**](http://docs.google.com/java/net/HttpCookie.html#hashCode())()            Return hash code of this http cookie. |
| static [List](http://docs.google.com/java/util/List.html)<[HttpCookie](http://docs.google.com/java/net/HttpCookie.html)> | [**parse**](http://docs.google.com/java/net/HttpCookie.html#parse(java.lang.String))([String](http://docs.google.com/java/lang/String.html) header)            Constructs cookies from set-cookie or set-cookie2 header string. |
| void | [**setComment**](http://docs.google.com/java/net/HttpCookie.html#setComment(java.lang.String))([String](http://docs.google.com/java/lang/String.html) purpose)            Specifies a comment that describes a cookie's purpose. |
| void | [**setCommentURL**](http://docs.google.com/java/net/HttpCookie.html#setCommentURL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) purpose)            Specifies a comment url that describes a cookie's purpose. |
| void | [**setDiscard**](http://docs.google.com/java/net/HttpCookie.html#setDiscard(boolean))(boolean discard)            Specify whether user agent should discard the cookie unconditionally. |
| void | [**setDomain**](http://docs.google.com/java/net/HttpCookie.html#setDomain(java.lang.String))([String](http://docs.google.com/java/lang/String.html) pattern)            Specifies the domain within which this cookie should be presented. |
| void | [**setMaxAge**](http://docs.google.com/java/net/HttpCookie.html#setMaxAge(long))(long expiry)            Sets the maximum age of the cookie in seconds. |
| void | [**setPath**](http://docs.google.com/java/net/HttpCookie.html#setPath(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Specifies a path for the cookie to which the client should return the cookie. |
| void | [**setPortlist**](http://docs.google.com/java/net/HttpCookie.html#setPortlist(java.lang.String))([String](http://docs.google.com/java/lang/String.html) ports)            Specify the portlist of the cookie, which restricts the port(s) to which a cookie may be sent back in a Cookie header. |
| void | [**setSecure**](http://docs.google.com/java/net/HttpCookie.html#setSecure(boolean))(boolean flag)            Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL. |
| void | [**setValue**](http://docs.google.com/java/net/HttpCookie.html#setValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) newValue)            Assigns a new value to a cookie after the cookie is created. |
| void | [**setVersion**](http://docs.google.com/java/net/HttpCookie.html#setVersion(int))(int v)            Sets the version of the cookie protocol this cookie complies with. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/net/HttpCookie.html#toString())()            Constructs a cookie header string representation of this cookie, which is in the format defined by corresponding cookie specification, but without the leading "Cookie:" token. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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** |
| --- |

### HttpCookie

public **HttpCookie**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) value)

Constructs a cookie with a specified name and value.

The name must conform to RFC 2965. That means it can contain only ASCII alphanumeric characters and cannot contain commas, semicolons, or white space or begin with a $ character. The cookie's name cannot be changed after creation.

The value can be anything the server chooses to send. Its value is probably of interest only to the server. The cookie's value can be changed after creation with the setValue method.

By default, cookies are created according to the RFC 2965 cookie specification. The version can be changed with the setVersion method.

**Parameters:**name - a String specifying the name of the cookievalue - a String specifying the value of the cookie **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the cookie name contains illegal characters or it is one of the tokens reserved for use by the cookie protocol [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if name is null**See Also:**[setValue(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setValue(java.lang.String)), [setVersion(int)](http://docs.google.com/java/net/HttpCookie.html#setVersion(int))

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

### parse

public static [List](http://docs.google.com/java/util/List.html)<[HttpCookie](http://docs.google.com/java/net/HttpCookie.html)> **parse**([String](http://docs.google.com/java/lang/String.html) header)

Constructs cookies from set-cookie or set-cookie2 header string. RFC 2965 section 3.2.2 set-cookie2 syntax indicates that one header line may contain more than one cookie definitions, so this is a static utility method instead of another constructor.

**Parameters:**header - a String specifying the set-cookie header. The header should start with "set-cookie", or "set-cookie2" token; or it should have no leading token at all. **Returns:**a List of cookie parsed from header line string **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if header string violates the cookie specification's syntax, or the cookie name contains llegal characters, or the cookie name is one of the tokens reserved for use by the cookie protocol [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the header string is null

### hasExpired

public boolean **hasExpired**()

Reports whether this http cookie has expired or not.

**Returns:**true to indicate this http cookie has expired; otherwise, false

### setComment

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

Specifies a comment that describes a cookie's purpose. The comment is useful if the browser presents the cookie to the user. Comments are not supported by Netscape Version 0 cookies.

**Parameters:**purpose - a String specifying the comment to display to the user**See Also:**[getComment()](http://docs.google.com/java/net/HttpCookie.html#getComment())

### getComment

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

Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.

**Returns:**a String containing the comment, or null if none**See Also:**[setComment(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setComment(java.lang.String))

### setCommentURL

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

Specifies a comment url that describes a cookie's purpose. The comment url is useful if the browser presents the cookie to the user. Comment url is RFC 2965 only.

**Parameters:**purpose - a String specifying the comment url to display to the user**See Also:**[getCommentURL()](http://docs.google.com/java/net/HttpCookie.html#getCommentURL())

### getCommentURL

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

Returns the comment url describing the purpose of this cookie, or null if the cookie has no comment url.

**Returns:**a String containing the comment url, or null if none**See Also:**[setCommentURL(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setCommentURL(java.lang.String))

### setDiscard

public void **setDiscard**(boolean discard)

Specify whether user agent should discard the cookie unconditionally. This is RFC 2965 only attribute.

**Parameters:**discard - true indicates to discard cookie unconditionally**See Also:**[getDiscard()](http://docs.google.com/java/net/HttpCookie.html#getDiscard())

### getDiscard

public boolean **getDiscard**()

Return the discard attribute of the cookie

**Returns:**a boolean to represent this cookie's discard attribute**See Also:**[setDiscard(boolean)](http://docs.google.com/java/net/HttpCookie.html#setDiscard(boolean))

### setPortlist

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

Specify the portlist of the cookie, which restricts the port(s) to which a cookie may be sent back in a Cookie header.

**Parameters:**ports - a String specify the port list, which is comma seperated series of digits**See Also:**[getPortlist()](http://docs.google.com/java/net/HttpCookie.html#getPortlist())

### getPortlist

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

Return the port list attribute of the cookie

**Returns:**a String contains the port list or null if none**See Also:**[setPortlist(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setPortlist(java.lang.String))

### setDomain

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

Specifies the domain within which this cookie should be presented.

The form of the domain name is specified by RFC 2965. A domain name begins with a dot (.foo.com) and means that the cookie is visible to servers in a specified Domain Name System (DNS) zone (for example, www.foo.com, but not a.b.foo.com). By default, cookies are only returned to the server that sent them.

**Parameters:**pattern - a String containing the domain name within which this cookie is visible; form is according to RFC 2965**See Also:**[getDomain()](http://docs.google.com/java/net/HttpCookie.html#getDomain())

### getDomain

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

Returns the domain name set for this cookie. The form of the domain name is set by RFC 2965.

**Returns:**a String containing the domain name**See Also:**[setDomain(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setDomain(java.lang.String))

### setMaxAge

public void **setMaxAge**(long expiry)

Sets the maximum age of the cookie in seconds.

A positive value indicates that the cookie will expire after that many seconds have passed. Note that the value is the *maximum* age when the cookie will expire, not the cookie's current age.

A negative value means that the cookie is not stored persistently and will be deleted when the Web browser exits. A zero value causes the cookie to be deleted.

**Parameters:**expiry - an integer specifying the maximum age of the cookie in seconds; if zero, the cookie should be discarded immediately; otherwise, the cookie's max age is unspecified.**See Also:**[getMaxAge()](http://docs.google.com/java/net/HttpCookie.html#getMaxAge())

### getMaxAge

public long **getMaxAge**()

Returns the maximum age of the cookie, specified in seconds. By default, -1 indicating the cookie will persist until browser shutdown.

**Returns:**an integer specifying the maximum age of the cookie in seconds**See Also:**[setMaxAge(long)](http://docs.google.com/java/net/HttpCookie.html#setMaxAge(long))

### setPath

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

Specifies a path for the cookie to which the client should return the cookie.

The cookie is visible to all the pages in the directory you specify, and all the pages in that directory's subdirectories. A cookie's path must include the servlet that set the cookie, for example, */catalog*, which makes the cookie visible to all directories on the server under */catalog*.

Consult RFC 2965 (available on the Internet) for more information on setting path names for cookies.

**Parameters:**uri - a String specifying a path**See Also:**[getPath()](http://docs.google.com/java/net/HttpCookie.html#getPath())

### getPath

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

Returns the path on the server to which the browser returns this cookie. The cookie is visible to all subpaths on the server.

**Returns:**a String specifying a path that contains a servlet name, for example, */catalog***See Also:**[setPath(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setPath(java.lang.String))

### setSecure

public void **setSecure**(boolean flag)

Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.

The default value is false.

**Parameters:**flag - if true, sends the cookie from the browser to the server using only when using a secure protocol; if false, sent on any protocol**See Also:**[getSecure()](http://docs.google.com/java/net/HttpCookie.html#getSecure())

### getSecure

public boolean **getSecure**()

Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol.

**Returns:**true if the browser can use any standard protocol; otherwise, false**See Also:**[setSecure(boolean)](http://docs.google.com/java/net/HttpCookie.html#setSecure(boolean))

### getName

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

Returns the name of the cookie. The name cannot be changed after creation.

**Returns:**a String specifying the cookie's name

### setValue

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

Assigns a new value to a cookie after the cookie is created. If you use a binary value, you may want to use BASE64 encoding.

With Version 0 cookies, values should not contain white space, brackets, parentheses, equals signs, commas, double quotes, slashes, question marks, at signs, colons, and semicolons. Empty values may not behave the same way on all browsers.

**Parameters:**newValue - a String specifying the new value**See Also:**[getValue()](http://docs.google.com/java/net/HttpCookie.html#getValue())

### getValue

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

Returns the value of the cookie.

**Returns:**a String containing the cookie's present value**See Also:**[setValue(java.lang.String)](http://docs.google.com/java/net/HttpCookie.html#setValue(java.lang.String))

### getVersion

public int **getVersion**()

Returns the version of the protocol this cookie complies with. Version 1 complies with RFC 2965/2109, and version 0 complies with the original cookie specification drafted by Netscape. Cookies provided by a browser use and identify the browser's cookie version.

**Returns:**0 if the cookie complies with the original Netscape specification; 1 if the cookie complies with RFC 2965/2109**See Also:**[setVersion(int)](http://docs.google.com/java/net/HttpCookie.html#setVersion(int))

### setVersion

public void **setVersion**(int v)

Sets the version of the cookie protocol this cookie complies with. Version 0 complies with the original Netscape cookie specification. Version 1 complies with RFC 2965/2109.

**Parameters:**v - 0 if the cookie should comply with the original Netscape specification; 1 if the cookie should comply with RFC 2965/2109 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if v is neither 0 nor 1**See Also:**[getVersion()](http://docs.google.com/java/net/HttpCookie.html#getVersion())

### domainMatches

public static boolean **domainMatches**([String](http://docs.google.com/java/lang/String.html) domain,  
 [String](http://docs.google.com/java/lang/String.html) host)

The utility method to check whether a host name is in a domain or not.

This concept is described in the cookie specification. To understand the concept, some terminologies need to be defined first:

effective host name = hostname if host name contains dot

                 or = hostname.local if not

Host A's name domain-matches host B's if:

* their host name strings string-compare equal; or
* A is a HDN string and has the form NB, where N is a non-empty name string, B has the form .B', and B' is a HDN string. (So, x.y.com domain-matches .Y.com but not Y.com.)

A host isn't in a domain (RFC 2965 sec. 3.3.2) if:

* The value for the Domain attribute contains no embedded dots, and the value is not .local.
* The effective host name that derives from the request-host does not domain-match the Domain attribute.
* The request-host is a HDN (not IP address) and has the form HD, where D is the value of the Domain attribute, and H is a string that contains one or more dots.

Examples:

* A Set-Cookie2 from request-host y.x.foo.com for Domain=.foo.com would be rejected, because H is y.x and contains a dot.
* A Set-Cookie2 from request-host x.foo.com for Domain=.foo.com would be accepted.
* A Set-Cookie2 with Domain=.com or Domain=.com., will always be rejected, because there is no embedded dot.
* A Set-Cookie2 with Domain=ajax.com will be accepted, and the value for Domain will be taken to be .ajax.com, because a dot gets prepended to the value.
* A Set-Cookie2 from request-host example for Domain=.local will be accepted, because the effective host name for the request- host is example.local, and example.local domain-matches .local.

**Parameters:**domain - the domain name to check host name withhost - the host name in question **Returns:**true if they domain-matches; false if not

### toString

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

Constructs a cookie header string representation of this cookie, which is in the format defined by corresponding cookie specification, but without the leading "Cookie:" token.

**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 form of the cookie. The string has the defined format

### equals

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

Test the equality of two http cookies.

The result is true only if two cookies come from same domain (case-insensitive), have same name (case-insensitive), and have same path (case-sensitive).

**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 reference object with which to compare. **Returns:**true if 2 http cookies equal to each other; otherwise, false**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Return hash code of this http cookie. The result is the sum of hash code value of three significant components of this cookie: name, domain, and path. That is, the hash code is the value of the expression:getName().toLowerCase().hashCode()

+ getDomain().toLowerCase().hashCode()

+ getPath().hashCode()

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**this http cookie's hash code**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)

### clone

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

Create and return a copy of this object.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this http cookie**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.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/HttpCookie.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/FileNameMap.html)   [**NEXT CLASS**](http://docs.google.com/java/net/HttpRetryException.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/HttpCookie.html)    [**NO FRAMES**](http://docs.google.com/HttpCookie.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).