| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/URL.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/URISyntaxException.html)   [**NEXT CLASS**](http://docs.google.com/java/net/URLClassLoader.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/URL.html)    [**NO FRAMES**](http://docs.google.com/URL.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#lnxbz9) |

## **java.net**

Class URL

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

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

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

Class URL represents a Uniform Resource Locator, a pointer to a "resource" on the World Wide Web. A resource can be something as simple as a file or a directory, or it can be a reference to a more complicated object, such as a query to a database or to a search engine. More information on the types of URLs and their formats can be found at:

[*http://www.socs.uts.edu.au/MosaicDocs-old/url-primer.html*](http://www.socs.uts.edu.au/MosaicDocs-old/url-primer.html)

In general, a URL can be broken into several parts. The previous example of a URL indicates that the protocol to use is http (HyperText Transfer Protocol) and that the information resides on a host machine named www.socs.uts.edu.au. The information on that host machine is named /MosaicDocs-old/url-primer.html. The exact meaning of this name on the host machine is both protocol dependent and host dependent. The information normally resides in a file, but it could be generated on the fly. This component of the URL is called the *path* component.

A URL can optionally specify a "port", which is the port number to which the TCP connection is made on the remote host machine. If the port is not specified, the default port for the protocol is used instead. For example, the default port for http is 80. An alternative port could be specified as:

http://www.socs.uts.edu.au:80/MosaicDocs-old/url-primer.html

The syntax of URL is defined by [*RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax*](http://www.ietf.org/rfc/rfc2396.txt), amended by [*RFC 2732: Format for Literal IPv6 Addresses in URLs*](http://www.ietf.org/rfc/rfc2732.txt). The Literal IPv6 address format also supports scope\_ids. The syntax and usage of scope\_ids is described [here](http://docs.google.com/Inet6Address.html#scoped).

A URL may have appended to it a "fragment", also known as a "ref" or a "reference". The fragment is indicated by the sharp sign character "#" followed by more characters. For example,

http://java.sun.com/index.html#chapter1

This fragment is not technically part of the URL. Rather, it indicates that after the specified resource is retrieved, the application is specifically interested in that part of the document that has the tag chapter1 attached to it. The meaning of a tag is resource specific.

An application can also specify a "relative URL", which contains only enough information to reach the resource relative to another URL. Relative URLs are frequently used within HTML pages. For example, if the contents of the URL:

http://java.sun.com/index.html

contained within it the relative URL:

FAQ.html

it would be a shorthand for:

http://java.sun.com/FAQ.html

The relative URL need not specify all the components of a URL. If the protocol, host name, or port number is missing, the value is inherited from the fully specified URL. The file component must be specified. The optional fragment is not inherited.

The URL class does not itself encode or decode any URL components according to the escaping mechanism defined in RFC2396. It is the responsibility of the caller to encode any fields, which need to be escaped prior to calling URL, and also to decode any escaped fields, that are returned from URL. Furthermore, because URL has no knowledge of URL escaping, it does not recognise equivalence between the encoded or decoded form of the same URL. For example, the two URLs:

http://foo.com/hello world/ and http://foo.com/hello%20world

would be considered not equal to each other.

Note, the [URI](http://docs.google.com/java/net/URI.html) class does perform escaping of its component fields in certain circumstances. The recommended way to manage the encoding and decoding of URLs is to use [URI](http://docs.google.com/java/net/URI.html), and to convert between these two classes using [toURI()](http://docs.google.com/java/net/URL.html#toURI()) and [URI.toURL()](http://docs.google.com/java/net/URI.html#toURL()).

The [URLEncoder](http://docs.google.com/java/net/URLEncoder.html) and [URLDecoder](http://docs.google.com/java/net/URLDecoder.html) classes can also be used, but only for HTML form encoding, which is not the same as the encoding scheme defined in RFC2396.

**Since:** JDK1.0 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.net.URL)

| **Constructor Summary** | |
| --- | --- |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) spec)            Creates a URL object from the String representation. |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) host, int port, [String](http://docs.google.com/java/lang/String.html) file)            Creates a URL object from the specified protocol, host, port number, and file. |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String,%20java.net.URLStreamHandler))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) host, int port, [String](http://docs.google.com/java/lang/String.html) file, [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html) handler)            Creates a URL object from the specified protocol, host, port number, file, and handler. |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) host, [String](http://docs.google.com/java/lang/String.html) file)            Creates a URL from the specified protocol name, host name, and file name. |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.net.URL,%20java.lang.String))([URL](http://docs.google.com/java/net/URL.html) context, [String](http://docs.google.com/java/lang/String.html) spec)            Creates a URL by parsing the given spec within a specified context. |
| [**URL**](http://docs.google.com/java/net/URL.html#URL(java.net.URL,%20java.lang.String,%20java.net.URLStreamHandler))([URL](http://docs.google.com/java/net/URL.html) context, [String](http://docs.google.com/java/lang/String.html) spec, [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html) handler)            Creates a URL by parsing the given spec with the specified handler within a specified context. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/net/URL.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this URL for equality with another object. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAuthority**](http://docs.google.com/java/net/URL.html#getAuthority())()            Gets the authority part of this URL. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getContent**](http://docs.google.com/java/net/URL.html#getContent())()            Gets the contents of this URL. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getContent**](http://docs.google.com/java/net/URL.html#getContent(java.lang.Class%5B%5D))([Class](http://docs.google.com/java/lang/Class.html)[] classes)            Gets the contents of this URL. |
| int | [**getDefaultPort**](http://docs.google.com/java/net/URL.html#getDefaultPort())()            Gets the default port number of the protocol associated with this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getFile**](http://docs.google.com/java/net/URL.html#getFile())()            Gets the file name of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getHost**](http://docs.google.com/java/net/URL.html#getHost())()            Gets the host name of this URL, if applicable. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPath**](http://docs.google.com/java/net/URL.html#getPath())()            Gets the path part of this URL. |
| int | [**getPort**](http://docs.google.com/java/net/URL.html#getPort())()            Gets the port number of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getProtocol**](http://docs.google.com/java/net/URL.html#getProtocol())()            Gets the protocol name of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getQuery**](http://docs.google.com/java/net/URL.html#getQuery())()            Gets the query part of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRef**](http://docs.google.com/java/net/URL.html#getRef())()            Gets the anchor (also known as the "reference") of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUserInfo**](http://docs.google.com/java/net/URL.html#getUserInfo())()            Gets the userInfo part of this URL. |
| int | [**hashCode**](http://docs.google.com/java/net/URL.html#hashCode())()            Creates an integer suitable for hash table indexing. |
| [URLConnection](http://docs.google.com/java/net/URLConnection.html) | [**openConnection**](http://docs.google.com/java/net/URL.html#openConnection())()            Returns a URLConnection object that represents a connection to the remote object referred to by the URL. |
| [URLConnection](http://docs.google.com/java/net/URLConnection.html) | [**openConnection**](http://docs.google.com/java/net/URL.html#openConnection(java.net.Proxy))([Proxy](http://docs.google.com/java/net/Proxy.html) proxy)            Same as openConnection(), except that the connection will be made through the specified proxy; Protocol handlers that do not support proxing will ignore the proxy parameter and make a normal connection. |
| [InputStream](http://docs.google.com/java/io/InputStream.html) | [**openStream**](http://docs.google.com/java/net/URL.html#openStream())()            Opens a connection to this URL and returns an InputStream for reading from that connection. |
| boolean | [**sameFile**](http://docs.google.com/java/net/URL.html#sameFile(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) other)            Compares two URLs, excluding the fragment component. |
| protected  void | [**set**](http://docs.google.com/java/net/URL.html#set(java.lang.String,%20java.lang.String,%20int,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) host, int port, [String](http://docs.google.com/java/lang/String.html) file, [String](http://docs.google.com/java/lang/String.html) ref)            Sets the fields of the URL. |
| protected  void | [**set**](http://docs.google.com/java/net/URL.html#set(java.lang.String,%20java.lang.String,%20int,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) protocol, [String](http://docs.google.com/java/lang/String.html) host, int port, [String](http://docs.google.com/java/lang/String.html) authority, [String](http://docs.google.com/java/lang/String.html) userInfo, [String](http://docs.google.com/java/lang/String.html) path, [String](http://docs.google.com/java/lang/String.html) query, [String](http://docs.google.com/java/lang/String.html) ref)            Sets the specified 8 fields of the URL. |
| static void | [**setURLStreamHandlerFactory**](http://docs.google.com/java/net/URL.html#setURLStreamHandlerFactory(java.net.URLStreamHandlerFactory))([URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) fac)            Sets an application's URLStreamHandlerFactory. |
| [String](http://docs.google.com/java/lang/String.html) | [**toExternalForm**](http://docs.google.com/java/net/URL.html#toExternalForm())()            Constructs a string representation of this URL. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/net/URL.html#toString())()            Constructs a string representation of this URL. |
| [URI](http://docs.google.com/java/net/URI.html) | [**toURI**](http://docs.google.com/java/net/URL.html#toURI())()            Returns a [URI](http://docs.google.com/java/net/URI.html) equivalent to this URL. |

| **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** |
| --- |

### URL

public **URL**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) host,  
 int port,  
 [String](http://docs.google.com/java/lang/String.html) file)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL object from the specified protocol, host, port number, and file.

host can be expressed as a host name or a literal IP address. If IPv6 literal address is used, it should be enclosed in square brackets ('[' and ']'), as specified by [RFC 2732](http://www.ietf.org/rfc/rfc2732.txt); However, the literal IPv6 address format defined in [*RFC 2373: IP Version 6 Addressing Architecture*](http://www.ietf.org/rfc/rfc2373.txt) is also accepted.

Specifying a port number of -1 indicates that the URL should use the default port for the protocol.

If this is the first URL object being created with the specified protocol, a *stream protocol handler* object, an instance of class URLStreamHandler, is created for that protocol:

1. If the application has previously set up an instance of URLStreamHandlerFactory as the stream handler factory, then the createURLStreamHandler method of that instance is called with the protocol string as an argument to create the stream protocol handler.
2. If no URLStreamHandlerFactory has yet been set up, or if the factory's createURLStreamHandler method returns null, then the constructor finds the value of the system property: java.protocol.handler.pkgs  
    If the value of that system property is not null, it is interpreted as a list of packages separated by a vertical slash character '|'. The constructor tries to load the class named: <*package*>.<*protocol*>.Handler  
    where <*package*> is replaced by the name of the package and <*protocol*> is replaced by the name of the protocol. If this class does not exist, or if the class exists but it is not a subclass of URLStreamHandler, then the next package in the list is tried.
3. If the previous step fails to find a protocol handler, then the constructor tries to load from a system default package. <*system default package*>.<*protocol*>.Handler  
    If this class does not exist, or if the class exists but it is not a subclass of URLStreamHandler, then a MalformedURLException is thrown.

Protocol handlers for the following protocols are guaranteed to exist on the search path :-

http, https, ftp, file, and jar

Protocol handlers for additional protocols may also be available.

No validation of the inputs is performed by this constructor.

**Parameters:**protocol - the name of the protocol to use.host - the name of the host.port - the port number on the host.file - the file on the host **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if an unknown protocol is specified.**See Also:**[System.getProperty(java.lang.String)](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)), [setURLStreamHandlerFactory( java.net.URLStreamHandlerFactory)](http://docs.google.com/java/net/URL.html#setURLStreamHandlerFactory(java.net.URLStreamHandlerFactory)), [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html), [URLStreamHandlerFactory.createURLStreamHandler( java.lang.String)](http://docs.google.com/java/net/URLStreamHandlerFactory.html#createURLStreamHandler(java.lang.String))

### URL

public **URL**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) host,  
 [String](http://docs.google.com/java/lang/String.html) file)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL from the specified protocol name, host name, and file name. The default port for the specified protocol is used.

This method is equivalent to calling the four-argument constructor with the arguments being protocol, host, -1, and file. No validation of the inputs is performed by this constructor.

**Parameters:**protocol - the name of the protocol to use.host - the name of the host.file - the file on the host. **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if an unknown protocol is specified.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

### URL

public **URL**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) host,  
 int port,  
 [String](http://docs.google.com/java/lang/String.html) file,  
 [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html) handler)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL object from the specified protocol, host, port number, file, and handler. Specifying a port number of -1 indicates that the URL should use the default port for the protocol. Specifying a handler of null indicates that the URL should use a default stream handler for the protocol, as outlined for: java.net.URL#URL(java.lang.String, java.lang.String, int, java.lang.String)

If the handler is not null and there is a security manager, the security manager's checkPermission method is called with a NetPermission("specifyStreamHandler") permission. This may result in a SecurityException. No validation of the inputs is performed by this constructor.

**Parameters:**protocol - the name of the protocol to use.host - the name of the host.port - the port number on the host.file - the file on the hosthandler - the stream handler for the URL. **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if an unknown protocol is specified. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow specifying a stream handler explicitly.**See Also:**[System.getProperty(java.lang.String)](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)), [setURLStreamHandlerFactory( java.net.URLStreamHandlerFactory)](http://docs.google.com/java/net/URL.html#setURLStreamHandlerFactory(java.net.URLStreamHandlerFactory)), [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html), [URLStreamHandlerFactory.createURLStreamHandler( java.lang.String)](http://docs.google.com/java/net/URLStreamHandlerFactory.html#createURLStreamHandler(java.lang.String)), [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [NetPermission](http://docs.google.com/java/net/NetPermission.html)

### URL

public **URL**([String](http://docs.google.com/java/lang/String.html) spec)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL object from the String representation.

This constructor is equivalent to a call to the two-argument constructor with a null first argument.

**Parameters:**spec - the String to parse as a URL. **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - If the string specifies an unknown protocol.**See Also:**[URL(java.net.URL, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.net.URL,%20java.lang.String))

### URL

public **URL**([URL](http://docs.google.com/java/net/URL.html) context,  
 [String](http://docs.google.com/java/lang/String.html) spec)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL by parsing the given spec within a specified context. The new URL is created from the given context URL and the spec argument as described in RFC2396 "Uniform Resource Identifiers : Generic \* Syntax" :

<scheme>://<authority><path>?<query>#<fragment>

The reference is parsed into the scheme, authority, path, query and fragment parts. If the path component is empty and the scheme, authority, and query components are undefined, then the new URL is a reference to the current document. Otherwise, the fragment and query parts present in the spec are used in the new URL.

If the scheme component is defined in the given spec and does not match the scheme of the context, then the new URL is created as an absolute URL based on the spec alone. Otherwise the scheme component is inherited from the context URL.

If the authority component is present in the spec then the spec is treated as absolute and the spec authority and path will replace the context authority and path. If the authority component is absent in the spec then the authority of the new URL will be inherited from the context.

If the spec's path component begins with a slash character "/" then the path is treated as absolute and the spec path replaces the context path.

Otherwise, the path is treated as a relative path and is appended to the context path, as described in RFC2396. Also, in this case, the path is canonicalized through the removal of directory changes made by occurences of ".." and ".".

For a more detailed description of URL parsing, refer to RFC2396.

**Parameters:**context - the context in which to parse the specification.spec - the String to parse as a URL. **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if no protocol is specified, or an unknown protocol is found.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html), [URLStreamHandler.parseURL(java.net.URL, java.lang.String, int, int)](http://docs.google.com/java/net/URLStreamHandler.html#parseURL(java.net.URL,%20java.lang.String,%20int,%20int))

### URL

public **URL**([URL](http://docs.google.com/java/net/URL.html) context,  
 [String](http://docs.google.com/java/lang/String.html) spec,  
 [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html) handler)  
 throws [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html)

Creates a URL by parsing the given spec with the specified handler within a specified context. If the handler is null, the parsing occurs as with the two argument constructor.

**Parameters:**context - the context in which to parse the specification.spec - the String to parse as a URL.handler - the stream handler for the URL. **Throws:** [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if no protocol is specified, or an unknown protocol is found. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow specifying a stream handler.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLStreamHandler](http://docs.google.com/java/net/URLStreamHandler.html), [URLStreamHandler.parseURL(java.net.URL, java.lang.String, int, int)](http://docs.google.com/java/net/URLStreamHandler.html#parseURL(java.net.URL,%20java.lang.String,%20int,%20int))

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

### set

protected void **set**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) host,  
 int port,  
 [String](http://docs.google.com/java/lang/String.html) file,  
 [String](http://docs.google.com/java/lang/String.html) ref)

Sets the fields of the URL. This is not a public method so that only URLStreamHandlers can modify URL fields. URLs are otherwise constant.

**Parameters:**protocol - the name of the protocol to usehost - the name of the hostport - the port number on the hostfile - the file on the hostref - the internal reference in the URL

### set

protected void **set**([String](http://docs.google.com/java/lang/String.html) protocol,  
 [String](http://docs.google.com/java/lang/String.html) host,  
 int port,  
 [String](http://docs.google.com/java/lang/String.html) authority,  
 [String](http://docs.google.com/java/lang/String.html) userInfo,  
 [String](http://docs.google.com/java/lang/String.html) path,  
 [String](http://docs.google.com/java/lang/String.html) query,  
 [String](http://docs.google.com/java/lang/String.html) ref)

Sets the specified 8 fields of the URL. This is not a public method so that only URLStreamHandlers can modify URL fields. URLs are otherwise constant.

**Parameters:**protocol - the name of the protocol to usehost - the name of the hostport - the port number on the hostauthority - the authority part for the urluserInfo - the username and passwordpath - the file on the hostref - the internal reference in the URLquery - the query part of this URL**Since:** 1.3

### getQuery

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

Gets the query part of this URL.

**Returns:**the query part of this URL, or null if one does not exist**Since:** 1.3

### getPath

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

Gets the path part of this URL.

**Returns:**the path part of this URL, or an empty string if one does not exist**Since:** 1.3

### getUserInfo

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

Gets the userInfo part of this URL.

**Returns:**the userInfo part of this URL, or null if one does not exist**Since:** 1.3

### getAuthority

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

Gets the authority part of this URL.

**Returns:**the authority part of this URL**Since:** 1.3

### getPort

public int **getPort**()

Gets the port number of this URL.

**Returns:**the port number, or -1 if the port is not set

### getDefaultPort

public int **getDefaultPort**()

Gets the default port number of the protocol associated with this URL. If the URL scheme or the URLStreamHandler for the URL do not define a default port number, then -1 is returned.

**Returns:**the port number**Since:** 1.4

### getProtocol

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

Gets the protocol name of this URL.

**Returns:**the protocol of this URL.

### getHost

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

Gets the host name of this URL, if applicable. The format of the host conforms to RFC 2732, i.e. for a literal IPv6 address, this method will return the IPv6 address enclosed in square brackets ('[' and ']').

**Returns:**the host name of this URL.

### getFile

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

Gets the file name of this URL. The returned file portion will be the same as getPath(), plus the concatenation of the value of getQuery(), if any. If there is no query portion, this method and getPath() will return identical results.

**Returns:**the file name of this URL, or an empty string if one does not exist

### getRef

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

Gets the anchor (also known as the "reference") of this URL.

**Returns:**the anchor (also known as the "reference") of this URL, or null if one does not exist

### equals

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

Compares this URL for equality with another object.

If the given object is not a URL then this method immediately returns false.

Two URL objects are equal if they have the same protocol, reference equivalent hosts, have the same port number on the host, and the same file and fragment of the file.

Two hosts are considered equivalent if both host names can be resolved into the same IP addresses; else if either host name can't be resolved, the host names must be equal without regard to case; or both host names equal to null.

Since hosts comparison requires name resolution, this operation is a blocking operation.

Note: The defined behavior for equals is known to be inconsistent with virtual hosting in HTTP.

**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 URL to compare against. **Returns:**true if the objects are the same; false otherwise.**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**()

Creates an integer suitable for hash table indexing.

The hash code is based upon all the URL components relevant for URL comparison. As such, this operation is a blocking operation.

**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 for this URL.**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)

### sameFile

public boolean **sameFile**([URL](http://docs.google.com/java/net/URL.html) other)

Compares two URLs, excluding the fragment component.

Returns true if this URL and the other argument are equal without taking the fragment component into consideration.

**Parameters:**other - the URL to compare against. **Returns:**true if they reference the same remote object; false otherwise.

### toString

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

Constructs a string representation of this URL. The string is created by calling the toExternalForm method of the stream protocol handler for this object.

**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.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLStreamHandler.toExternalForm(java.net.URL)](http://docs.google.com/java/net/URLStreamHandler.html#toExternalForm(java.net.URL))

### toExternalForm

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

Constructs a string representation of this URL. The string is created by calling the toExternalForm method of the stream protocol handler for this object.

**Returns:**a string representation of this object.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLStreamHandler.toExternalForm(java.net.URL)](http://docs.google.com/java/net/URLStreamHandler.html#toExternalForm(java.net.URL))

### toURI

public [URI](http://docs.google.com/java/net/URI.html) **toURI**()  
 throws [URISyntaxException](http://docs.google.com/java/net/URISyntaxException.html)

Returns a [URI](http://docs.google.com/java/net/URI.html) equivalent to this URL. This method functions in the same way as new URI (this.toString()).

Note, any URL instance that complies with RFC 2396 can be converted to a URI. However, some URLs that are not strictly in compliance can not be converted to a URI.

**Returns:**a URI instance equivalent to this URL. **Throws:** [URISyntaxException](http://docs.google.com/java/net/URISyntaxException.html) - if this URL is not formatted strictly according to to RFC2396 and cannot be converted to a URI.**Since:** 1.5

### openConnection

public [URLConnection](http://docs.google.com/java/net/URLConnection.html) **openConnection**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns a URLConnection object that represents a connection to the remote object referred to by the URL.

A new connection is opened every time by calling the openConnection method of the protocol handler for this URL.

If for the URL's protocol (such as HTTP or JAR), there exists a public, specialized URLConnection subclass belonging to one of the following packages or one of their subpackages: java.lang, java.io, java.util, java.net, the connection returned will be of that subclass. For example, for HTTP an HttpURLConnection will be returned, and for JAR a JarURLConnection will be returned.

**Returns:**a URLConnection to the URL. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLConnection](http://docs.google.com/java/net/URLConnection.html), [URLStreamHandler.openConnection(java.net.URL)](http://docs.google.com/java/net/URLStreamHandler.html#openConnection(java.net.URL))

### openConnection

public [URLConnection](http://docs.google.com/java/net/URLConnection.html) **openConnection**([Proxy](http://docs.google.com/java/net/Proxy.html) proxy)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Same as openConnection(), except that the connection will be made through the specified proxy; Protocol handlers that do not support proxing will ignore the proxy parameter and make a normal connection. Calling this method preempts the system's default ProxySelector settings.

**Parameters:**proxy - the Proxy through which this connection will be made. If direct connection is desired, Proxy.NO\_PROXY should be specified. **Returns:**a URLConnection to the URL. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager is present and the caller doesn't have permission to connect to the proxy. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - will be thrown if proxy is null, or proxy has the wrong type [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the subclass that implements the protocol handler doesn't support this method.**Since:** 1.5 **See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLConnection](http://docs.google.com/java/net/URLConnection.html), [URLStreamHandler.openConnection(java.net.URL, java.net.Proxy)](http://docs.google.com/java/net/URLStreamHandler.html#openConnection(java.net.URL,%20java.net.Proxy))

### openStream

public final [InputStream](http://docs.google.com/java/io/InputStream.html) **openStream**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Opens a connection to this URL and returns an InputStream for reading from that connection. This method is a shorthand for:

openConnection().getInputStream()

**Returns:**an input stream for reading from the URL connection. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs.**See Also:**[openConnection()](http://docs.google.com/java/net/URL.html#openConnection()), [URLConnection.getInputStream()](http://docs.google.com/java/net/URLConnection.html#getInputStream())

### getContent

public final [Object](http://docs.google.com/java/lang/Object.html) **getContent**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Gets the contents of this URL. This method is a shorthand for:

openConnection().getContent()

**Returns:**the contents of this URL. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs.**See Also:**[URLConnection.getContent()](http://docs.google.com/java/net/URLConnection.html#getContent())

### getContent

public final [Object](http://docs.google.com/java/lang/Object.html) **getContent**([Class](http://docs.google.com/java/lang/Class.html)[] classes)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Gets the contents of this URL. This method is a shorthand for:

openConnection().getContent(Class[])

**Parameters:**classes - an array of Java types **Returns:**the content object of this URL that is the first match of the types specified in the classes array. null if none of the requested types are supported. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O exception occurs.**Since:** 1.3 **See Also:**[URLConnection.getContent(Class[])](http://docs.google.com/java/net/URLConnection.html#getContent(java.lang.Class%5B%5D))

### setURLStreamHandlerFactory

public static void **setURLStreamHandlerFactory**([URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) fac)

Sets an application's URLStreamHandlerFactory. This method can be called at most once in a given Java Virtual Machine.

The URLStreamHandlerFactory instance is used to construct a stream protocol handler from a protocol name.

If there is a security manager, this method first calls the security manager's checkSetFactory method to ensure the operation is allowed. This could result in a SecurityException.

**Parameters:**fac - the desired factory. **Throws:** [Error](http://docs.google.com/java/lang/Error.html) - if the application has already set a factory. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkSetFactory method doesn't allow the operation.**See Also:**[URL(java.lang.String, java.lang.String, int, java.lang.String)](http://docs.google.com/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String)), [URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html), [SecurityManager.checkSetFactory()](http://docs.google.com/java/lang/SecurityManager.html#checkSetFactory())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/URL.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/URISyntaxException.html)   [**NEXT CLASS**](http://docs.google.com/java/net/URLClassLoader.html) | [**FRAMES**](http://docs.google.com/index.html?java/net/URL.html)    [**NO FRAMES**](http://docs.google.com/URL.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#lnxbz9) |

[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).