---

# Class java.net.URLStreamHandler

```
java.lang.Object
   |
   +----java.net.URLStreamHandler
```

---

public class **URLStreamHandler**
extends Object

Abstract class for URL stream openers. Subclasses of this class know how to create streams for particular protocol types.

**Version:**
1.10, 08/21/95
**Author:**
James Gosling

---

# Constructor Index

o **URLStreamHandler**()

# Method Index

o **openConnection**(URL)
Opens an input stream to the object referenced by the URL.
o **parseURL**(URL, String, int, int)
This method is called to parse the string spec into URL u.
o **toExternalForm**(URL)
Reverses the parsing of the URL.

# Constructors

o **URLStreamHandler**

```
public URLStreamHandler()
```

# Methods

o **openConnection**

```
protected abstract URLConnection openConnection(URL u) throws IOException
```

Opens an input stream to the object referenced by the URL. This method should be overridden by a subclass.

**Parameters:**

u – the URL that this connects to

o **parseURL**

```
protected void parseURL(URL u,
                        String spec,
                        int start,
                        int limit)
```

This method is called to parse the string spec into URL u. If there is any inherited context then it has already been copied into u. The parameters start and limit refer to the range of characters in spec that should be parsed. The default method uses parsing rules that match the http spec, which most URL protocol families follow. If you are writing a protocol handler that has a different syntax, then this routine should be overridden.

**Parameters:**

u – the URL to receive the result of parsing the spec

spec – the URL string to parse

start – the character position to start parsing at. This is just past the ':' (if there is one)

limit – the character position to stop parsing at. This is the end of the string or the position of the "#" character if present (the "#" reference syntax is protocol independant).

o **toExternalForm**

```
protected String toExternalForm(URL u)
```

Reverses the parsing of the URL. This should probably be overridden if you override parseURL().

**Parameters:**

u – the URL

**Returns:**

the textual representation of the fully qualified URL (ie. after the context and canonicalization have been applied).

---