Java Platform 1.2
Beta 4

Class java.net.JarURLConnection

```
java.lang.Object
   |
   +--java.net.URLConnection
          |
          +--java.net.JarURLConnection
```

public abstract class **JarURLConnection**
extends URLConnection

A URL Connection to a Java ARchive (JAR) file or an entry in a JAR file.

The syntax of a JAR URL is:

```
 jar:<url>!/{entry}
```

for example:

```
jar:http://www.foo.com/bar/baz.jar!/COM/foo/Quux.class
```

Jar URLs should be used to either refer to single JAR entries or as base URLs, to refer to JAR files as codebases, or relative URLs. The example above is a full JAR URL, which refers to a JAR entry. If the entry name is omitted, the URL refers to the whole JAR file: `jar:http://www.foo.com/bar/baz.jar!/`

Users should cast the generic URLConnection to a JarURLConnection when they know that the URL they created is a JAR URL, and they need JAR-specific functionality. For example: `URL url = new URL("jar:file:/home/duke/duke.jar!/"); JarURLConnection jarConnection = (JarURLConnection)url.openConnection(); Manifest manifest = jarConnection.getManifest();`

Examples:

A Jar entry
     `jar:http://www.foo.com/bar/baz.jar!/COM/foo/Quux.class`
A Jar file
     `jar:http://www.foo.com/bar/baz.jar!/`
A Jar directory
     `jar:http://www.foo.com/bar/baz.jar!/COM/foo/`

`!/` is refered to as the *separator*.

When constructing a JAR url, the following rules apply:

- if there is no context URL and the specification passed to the URL constructor doesn't contains a separator, the URL is considered to refer to a JarFile.
- if there is a context URL, the context URL is assumed to refer to a JAR file or a Jar directory.
- if the specification begins with a '/', the Jar directory is ignored, and the spec is considered to be at the root of the Jar file.

  Examples:

  context: **jar:http://www.foo.com/bar/jar.jar!/**, spec:**baz/entry.txt**

```
         url:jar:http://www.foo.com/bar/baz/jar.jar!/baz/entry.txt
context: jar:http://www.foo.com/bar/jar.jar!/baz, spec:entry.txt
         url:jar:http://www.foo.com/bar/baz/jar.jar!/baz/entry.txt
context: jar:http://www.foo.com/bar/jar.jar!/baz, spec:/entry.txt
         url:jar:http://www.foo.com/bar/baz/jar.jar!/entry.txt
```

Since:
> JDK1.2

See Also:
> `URL`, `URLConnection`, `JarFile`, `JarInputStream`, `Manifest`, `ZipEntry`

Inner classes inherited from class java.net.**URLConnection**
URLConnection.Callback

Field Summary

protected **URLConnection**	**jarFileURLConnection** The connection to the JAR file URL, if the connection has been initiated.

Fields inherited from class java.net.**URLConnection**
allowUserInteraction , **connected** , **doInput** , **doOutput** , **ifModifiedSince** , **url** , **useCaches**

Constructor Summary

JarURLConnection(**URL** url)

Method Summary

Attributes	**getAttributes**() Return the Attributes object for this connection if the URL for it points to a JAR file entry, null otherwise.
Certificate[]	**getCertificates**() /** Return the Certificate object for this connection if the URL for it points to a JAR file entry, null otherwise.
String	**getEntryName**() Return the entry name for this connection.
JarEntry	**getJarEntry**() Return the JAR entry object for this connection, if any.
abstract **JarFile**	**getJarFile**() Return the JAR file for this connection.
URL	**getJarFileURL**() Returns the URL for the Jar file for this connection.
Attributes	**getMainAttributes**() Returns the main Attributes for the JAR file for this connection.
Manifest	**getManifest**() Returns the Manifest for this connection, or null if none.

Methods inherited from class java.net.**URLConnection**

Field Detail

jarFileURLConnection

`protected` **`URLConnection`** `jarFileURLConnection`

> The connection to the JAR file URL, if the connection has been initiated. This should be set by connect.

Constructor Detail

JarURLConnection

```
protected JarURLConnection(URL url)
                    throws MalformedURLException
```

Method Detail

getJarFileURL

`public` **`URL`** `getJarFileURL()`

> Returns the URL for the Jar file for this connection.
>
> **Returns:**
>> the URL for the Jar file for this connection.

---

getEntryName

`public` **`String`** `getEntryName()`

> Return the entry name for this connection. This method returns null if the JAR file URL corresponding to this connection points to a JAR file and not a JAR file entry.
>
> **Returns:**
>> the entry name for this connection, if any.

---

getJarFile

```
public abstract JarFile getJarFile()
                          throws IOException
```

Return the JAR file for this connection. The returned object is not modifiable, and will throw UnsupportedOperationException if the caller attempts to modify it.

Returns:
the JAR file for this connection. If the connection is a connection to an entry of a JAR file, the JAR file object is returned

Throws:
IOException - if an IOException occurs while trying to connect to the JAR file for this connection.

See Also:
URLConnection.connect()

---

getManifest

```
public Manifest getManifest()
                      throws IOException
```

Returns the Manifest for this connection, or null if none. The returned object is not modifiable, and will throw UnsupportedOperationException if the caller attempts to modify it.

Returns:
the manifest object corresponding to the JAR file object for this connection.

Throws:
IOException - if getting the JAR file for this connection causes an IOException to be trown.

See Also:
getJarFile()

---

getJarEntry

```
public JarEntry getJarEntry()
                      throws IOException
```

Return the JAR entry object for this connection, if any. This method returns null if the JAR file URL corresponding to this connection points to a JAR file and not a JAR file entry. The returned object is not modifiable, and will throw UnsupportedOperationException if the caller attempts to modify it.

Returns:
the JAR entry object for this connection, or null if the JAR URL for this connection points to a JAR file.

Throws:
IOException - if getting the JAR file for this connection causes an IOException to be trown.

See Also:
getJarFile(), getJarEntry()

---

getAttributes

```
public Attributes getAttributes()
                        throws IOException
```

Return the Attributes object for this connection if the URL for it points to a JAR file entry, null otherwise.

Returns:
the Attributes object for this connection if the URL for it points to a JAR file entry, null otherwise.
Throws:
IOException - if getting the JAR entry causes an IOException to be thrown.
See Also:
getJarEntry()

getMainAttributes

```
public Attributes getMainAttributes()
                            throws IOException
```

Returns the main Attributes for the JAR file for this connection.

Returns:
the main Attributes for the JAR file for this connection.
Throws:
IOException - if getting the manifest causes an IOException to be thrown.
See Also:
getJarFile(), getManifest()

getCertificates

```
public Certificate[] getCertificates()
                            throws IOException
```

/ Return the Certificate object for this connection if the URL for it points to a JAR file entry, null otherwise. This method can only be called once the connection has been completely verified by reading from the input stream until the end of the stream has been reached. Otherwise, this method will return `null`**

Returns:
the Certificate object for this connection if the URL for it points to a JAR file entry, null otherwise.
Throws:
IOException - if getting the JAR entry causes an IOException to be thrown.
See Also:
getJarEntry()