

The_Factory_pattern_in_MIDP_2.0

The MIDP 2.0 Generic Connection Framework uses an advanced implementation of the Factory design pattern as its foundation. As the name suggests, the Generic Connection Framework is designed to handle all kinds of connections. In MIDP 1.0 the framework was limited to handling HTTP connections, but MIDP 2.0 has added HTTPS, comm connections, sockets, and more.

Note that because the Generic Connection Framework implements the Factory pattern, all connections can be handled with the Connection interface. The abstractions are done by extending interfaces. Connector is a factory class that has a set of open() methods for creating connections. The open() method takes a string as a parameter. The parameter form is:

```
{scheme}:{target} [{params}]
```

{scheme} is the name of a protocol such as HTTP, the {target} is normally some kind of network address, and {params} are formed as a series of equates of the form ";param=value".

Connections in MIDP

Creating connections in MIDP is very simple. All you have to do is to call the open() method with the right string and you have the connection. For example, you can open an HTTP connection with just one line of code:

```
Connection con = Connector.open("http://www.mypage.com");
```

When the string parameter is a normal HTTP address, the Connector opens an HTTP connection, which can be handled as a Connection and can be converted to an HttpURLConnection like so:

```
HttpURLConnection httpcon = (HttpURLConnection)con;
```

Fetching a file is easy!

After you've established an HTTP connection you're ready to use it. In the following code sample, I've used the Generic Connection Framework to fetch an XML file from a Web server.

```
HttpURLConnection con = null;
InputStream is = null;
String xml = new String();
OutputStream out = null;
try {
    = (HttpURLConnection)Connector.open(this.url);
    setRequestMethod(HttpURLConnection.GET);
    setRequestProperty("Connection", "close");

    // The call to openInputStream() opens the connection
    is = con.openInputStream();

    // Read the XML file
    ByteArrayOutputStream bas = new ByteArrayOutputStream();
    int ch;
    while ((ch = is.read()) != -1) {
        write(ch); bas.
    }
}
```

The_Factory_pattern_in_MIDP_2.0

```
// The
    = ba.toString();

} catch (Exception e) {
    printStackTrace();
} finally {
try {
if (null!=out) out.close();
if (null!= is) is.close();
    close();    con.
} catch (Exception ex) {
    printStackTrace();
}
}
```

First, I called the Connector open() method with the specific Web address, then set the request method and the Connection property. To actually open the connection I called openInputStream(). Having opened stream to the content (in this case an XML file) I then used the ByteArrayOutputStream to read it.