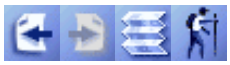
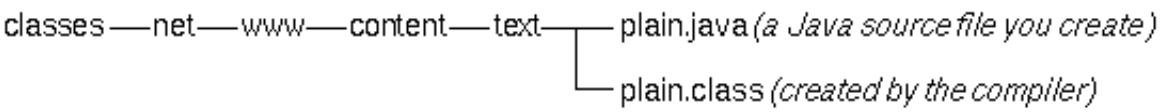


The "text/plain" Content Handler

Note: This page reflects the 1.0 Alpha 3 release of HotJava and Java. It will be updated for the Beta release of HotJava.

By following the steps on this page, you can create and use a content handler. Content handlers are Java programs that HotJava loads when it needs to interpret a particular MIME type/subtype combination. Some examples of MIME type/subtype combinations are text/plain, text/richtext, image/gif, and so on. (You can download the official list of MIME types.) When you've followed all the steps, your directory (folder) structure should look something like this:



Create directories

Create the directory structure shown in the figure above:
`classes/net/www/content/text`.

By convention, the directory structure starts with a directory named `classes`. The next three directories *must* be named as shown (`net/www/content`). The final directory must have the same name as the MIME type of the data. In this case, this is "text". In the next step, you'll create a Java source file named after the MIME subtype of the data ("plain").

Create a Java source file for the content type

In the `text` directory, create a file named plain.java with these contents. This file defines a class named plain that's a subclass of ContentHandler (a class in the net.www.html package). The name of each ContentHandler subclass must match the name of the MIME subtype it handles [or Generic if it handles all subtypes of the type? (not implemented yet)].

Compile the Java source file

Compile the content handler using the Java compiler.

If compilation succeeds, the compiler creates a file named `plain.class`. If compilation fails, make sure you typed in and named the program exactly as shown above.

Set the CLASSPATH environment variable

Set the CLASSPATH environment variable to include the classes directory, so that HotJava can find the content handler.

Turn off any special mailcap handling of text/plain

If you've used a mailcap file to specify special handling of the text/plain MIME type/subtype, you need to turn off that special handling, so that HotJava will use your content handler instead.

Start HotJava

Even if HotJava is already running, you need to quit and restart it so that it can read the new value of the CLASSPATH environment variable and (if appropriate) reread the mailcap file.

Load the content handler

The content handler is automatically loaded the first time you visit a plain/text document.

Load the content handler by viewing a plain text document, such as this one. You should see the contents of the file in the browser, followed by this text:

```
-- This closing message brought to you by your plain/text
content handler. To remove this content handler, delete the
net.www.content.text.plain class from your class path.
```

Now what?

Now you can either:

- Return to the *Trail Map* to get an overview of the trails you can follow.
- Learn more about writing protocol handlers by going to the *Extending HotJava* trail.

---

plain.java

```
/*
 * Copyright (c) 1994 Sun Microsystems, Inc. All Rights Reserved.
 *
 * Permission to use, copy, modify, and distribute this software
 * and its documentation for NON-COMMERCIAL purposes and without
 * fee is hereby granted provided that this copyright notice
 * appears in all copies. Please refer to the file "copyright.html"
 * for further important copyright and licensing information.
 *
 * SUN MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF
 * THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
 * TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
 * PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SUN SHALL NOT BE LIABLE FOR
 * ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR
 * DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
 */
/*
 * Plain text file handler
 */
package net.www.content.text;
import net.www.html.ContentHandler;
import net.www.html.URL;
import java.io.InputStream;
import awt.GifImage;

public class plain extends ContentHandler {
    public Object getContent(InputStream is, URL u) {
        StringBuffer sb = new StringBuffer();
        int c;

        while ((c = is.read()) >= 0) {
            sb.appendChar(c);
        }
        sb.append("\n\n-- This closing message brought to you by your plain/text\n");
        sb.append("content handler. To remove this content handler, delete the\n");
        sb.append("net.www.content.text.plain class from your class path.\n");
        is.close();
        return sb.toString();
    }
}
```

Return to Lesson

Platform–Specific Details: Setting the CLASSPATH Environment Variable

UNIX:
The method of setting environment variables depends on your shell. For C shell, you can enter something like the following at the shell prompt, or put it in your .cshrc file:

```
setenv CLASSPATH .:/home/kwalrath/test/classes
```

DOS shell (Windows 95/NT):

```
#ifdef COMMENT
HELP!
#endif
```

Return to Lesson

Return to Lesson

Platform–Specific Details: Compiling a Java Source File

UNIX:

```
javac plain.java
```

DOS shell (Windows 95/NT):

```
javac plain.java
```

Return to Lesson

Return to Lesson

Platform–Specific Details: Changing the Mailcap Specification of "plain/text"

UNIX:

Generally, per–user configuration of mailcap is done in a file named `~/.mailcap`. If you have no such file, contact your system administrator to find out where your mailcap information is.

1. In your mailcap file, find the line beginning with "text/plain". For example:

   ```
   text/plain; /usr/openwin/bin/textedit %s
   ```

2. Comment out the line (using the '#' character) or delete it. For example:

   ```
   #text/plain; /usr/openwin/bin/textedit %s
   ```

Return to Lesson