

12.2 The contentType and pageEncoding Attributes

The `contentType` attribute sets the `Content-Type` response header, indicating the MIME type of the document being sent to the client. For more information on MIME types, see [Table 7.1](#) (Common MIME Types) in [Section 7.2](#) (Understanding HTTP 1.1 Response Headers).

Use of the `contentType` attribute takes one of the following two forms.

```
<%@ page contentType="MIME-Type" %>
<%@ page contentType="MIME-Type; charset=Character-Set" %>
```

For example, the directive

```
<%@ page contentType="application/vnd.ms-excel" %>
```

has the same basic effect as the scriptlet

```
<% response.setContentType("application/vnd.ms-excel"); %>
```

The first difference between the two forms is that `response.setContentType` uses explicit Java code (an approach some developers try to avoid), whereas the `page` directive uses only JSP syntax. The second difference is that directives are parsed specially; they don't directly become `_jspService` code at the location at which they appear. This means that `response.setContentType` can be invoked conditionally whereas the `page` directive cannot be. Setting the content type conditionally is useful when the same content can be displayed in different forms—for an example, see the next section (Conditionally Generating Excel Spreadsheets).

Unlike regular servlets, for which the default MIME type is `text/plain`, the default for JSP pages is `text/html` (with a default character set of `ISO-8859-1`). Thus, JSP pages that output HTML in a Latin character set need not use `contentType` at all. If you want to change both the content type and the character set, you can do the following.

```
<%@ page contentType="someMimeType; charset=someCharacterSet" %>
```

However, if you only want to change the character set, it is simpler to use the `pageEncoding` attribute. For example, Japanese JSP pages might use the following.

```
<%@ page pageEncoding="Shift_JIS" %>
```

Generating Excel Spreadsheets

[Listing 12.2](#) shows a JSP page that uses the `contentType` attribute and tab-separated data to generate Excel output. Note that the `page` directive and comment are at the bottom so that the carriage returns at the ends of the lines don't show up in the Excel document. (Note: JSP does not ignore white space—JSP usually generates HTML in which most white space is ignored by the browser, but JSP itself maintains the white space and sends it to the client.) [Figure 12-4](#) shows the result in Internet Explorer on a system that has Microsoft Office installed.

Listing 12.2 Excel.jsp

First	Last	Email Address
Marty	Hall	hall@coreservlets.com
Larry	Brown	brown@coreservlets.com

```
Steve    Balmer    balmer@ibm.com
Scott    McNealy   mcnealy@microsoft.com
<%@ page contentType="application/vnd.ms-excel" %>
<%-- There are tabs, not spaces, between columns. --%>
```

Figure 12-4. Excel document ([Excel.jsp](#)) in Internet Explorer.

The screenshot shows a Microsoft Internet Explorer window displaying an Excel spreadsheet. The title bar reads "http://localhost/page-directive/Excel.jsp - Microsoft Internet Explorer". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Go To, Favorites, Help. The toolbar has various icons for file operations. The address bar shows the URL "http://localhost/page-directive/Excel.jsp". The spreadsheet is titled "First" and contains the following data:

	A	B	C	D	E	F	G
1	First	Last	Email Address				
2	Marty	Hall	hall@coreservlets.com				
3	Larry	Brown	brown@coreservlets.com				
4	Steve	Balmer	balmer@ibm.com				
5	Scott	McNealy	mcnealy@microsoft.com				
6							

The status bar at the bottom shows "Unknown Zone" and navigation buttons for the browser.

[[Team LiB](#)]

[[PREVIOUS](#)] [NEXT](#)]