

---

In this appendix:

- *The Applet Tag*

# B

## *HTML Markup For Applets*

### *B.1 The Applet Tag*

The introduction of Java created the need for additional HTML tags. In the alpha release of Java, the HotJava browser used the `<APP>` tag to include applets within HTML files. However, `<APP>` was unacceptable to the standards committee because it could have an infinite number of parameters. It was replaced by the `<APPLET>` tag, used in conjunction with the `<PARAM>` tag. Apparently, the standards folks did not like the `<APPLET>` tag either, so you can expect it to be replaced eventually, although at this point, there is no agreement about its successor, and it is highly unlikely that any production browser would stop supporting `<APPLET>`.

The syntax of the `<APPLET>` tag is shown below; the order of the parameters does not matter:

```
<APPLET
  [ALIGN = alignment]
  [ALT = alternate-text]
  CODE = applet-filename or OBJECT = serialized-applet
  [CODEBASE = applet-directory-url]
  [ARCHIVE = filename.zip/filename.jar]
  HEIGHT = applet-pixel-height
  [HSPACE = horizontal-pixel-margin]
  [MAYSCRIPT = true/false]
  [NAME = applet-name]
  [VSPACE = vertical-pixel-margin]
  WIDTH = applet-pixel-width
>
<PARAM NAME=parameter1 VALUE=value1>
<PARAM NAME=parameter2 VALUE=value2>
<PARAM NAME=parameter3 VALUE=value3>
...
[alternate-html]
</APPLET>
```

**<APPLET>**

The **<APPLET>** tag specifies where and how to display an applet within the HTML document. If the browser does not understand the **<APPLET>** and **<PARAM>** tags, it displays the `alternate-html`. (It displays the `alternate-html` because it doesn't understand the surrounding tags and ignores them. There's no magic to the `alternate-html` itself.) If a browser does understand **<APPLET>** but cannot run Java (for example, a browser on Windows 3.1) or Java has been disabled, the browser displays the `alternate-html` or the `alternate-text` specified by the optional `ALT` parameter. The `CODE`, `WIDTH`, and `HEIGHT` parameters are required. Parameters within the **<APPLET>** tag are separated by spaces, not by commas.

**</APPLET>**

Closes the **<APPLET>** tag. Anything prior to **</APPLET>** is considered `alternate-html` if it is not a **<PARAM>** tag. The `alternate-html` is displayed when Java is disabled, when Java cannot be run in the current browser, or when the browser does not understand the **<APPLET>** tag.

The following parameters may appear inside the **<APPLET>** tag.

**ALIGN**

`alignment`, optional. Specifies the applet's alignment on the Web page. Valid values are: `left`, `right`, `top`, `texttop`, `middle`, `absmiddle`, `baseline`, `bottom`, `absbottom`. Default: `left`. The alignment values have the same meanings as they do in the **<IMG>** tag.

**ALT**

`alternate-text`, optional. The alternate text is displayed when the browser understands the **<APPLET>** tag but is incapable of executing applets, either because Java is disabled or not supported on the platform. Support of this tag is browser dependent; most browsers just display the `alternate-html` since that is not restricted to text.

**ARCHIVE**

`filename.zip/filename.jar`, optional. Points to a comma-separated list of uncompressed ZIP or JAR files that contain one or more Java classes. Each file is downloaded once to the user's disk and searched for the class named in the `CODE` parameter, and any helper classes required to execute that class. JAR files may be signed to grant additional access. (JAR files are Java archives, a new archive format defined in Java 1.1. JAR files support features like digital signatures and compression. While they are not yet in wide use, they should become an important way of distributing sets of Java classes.)

**CODE**

`applet-filename`. This parameter or the `OBJECT` parameter is required. Name of applet `.class` file. The `.class` extension is not required in the `<APPLET>` tag but is required in the class's actual filename. The filename has to be a quoted string only if it includes whitespace.

**CODEBASE**

`applet-directory-url`, optional. Relative or absolute URL specifying the directory in which to locate the `.class` file or ZIP archive for the applet. Default: html directory.

**HEIGHT**

`applet-pixel-height`, required. Initial height of applet in pixels. Many browsers do not allow applets to change their height.

**HSPACE**

`horizontal-pixel-margin`, optional. Horizontal margin left and right of the applet, in pixels.

**MAYSCRIPT**

Required for applets that wish to use LiveConnect and the `netscape.javascript` classes to interact with JavaScript. Set to `true` to communicate with JavaScript. Set to `false`, or omit this parameter to disable communication with JavaScript. Both Java and JavaScript must be enabled in the browser.

**NAME**

`applet-name`, optional. Allows simultaneously running applets to communicate by this name. Default: the applet's class name.

**OBJECT**

`serialized-applet`. This parameter or the `CODE` parameter is required. Name of applet saved to a file as a serialized object. When loaded, `init()` is not called again but `start()` is. Parameters for running the applet are taken from this `<APPLET>` tag, not the original.

**VSPACE**

`vertical-pixel-margin`, optional. Vertical margin above and below the applet, in pixels.

**WIDTH**

`applet-pixel-width`, required. Initial width of applet in pixels. Many browsers do not allow applets to change their width.

The `<PARAM>` tag may appear between the `<APPLET>` and `</APPLET>` tags:

### <PARAM>

The <PARAM> tag allows the HTML author to provide run-time parameters to the applet as a series of NAME and VALUE pairs. The NAME is case insensitive, a String. See Chapter 14, *And Then There Were Applets* for a discussion of how to read parameters in an applet. Quotes are required around the parameter name or its value if there are any embedded spaces. There can be an infinite number of <PARAM> tags, and they all must appear between <APPLET> and </APPLET>

The special parameter name CABBASE is used for sending CAB files with Internet Explorer 3.0. CAB files are similar to ZIP files but are compressed into a CABinet file and can store audio and image files, in addition to classes. (For a full explanation see: <http://207.68.137.43/workshop/java/overview.htm>.) When .class files are placed within a CAB file, they are decompressed at the local end. Here's an example:

```
<APPLET CODE="oreilly.class" WIDTH=400 HEIGHT=400>
<PARAM NAME="cabbase" VALUE="ora.cab">
</APPLET>
```

The special parameter name ARCHIVES is reserved for sending JAR files. JAR files can also be specified using the ARCHIVES parameter to the <APPLET> tag.\* Here's an example:

```
<APPLET CODE="oreilly.class" WIDTH=400 HEIGHT=400>
<PARAM NAME="archives" VALUE="ora.jar">
</APPLET>
```

---

\* For a full explanation see <http://www.javasoft.com/products/JDK/1.1/docs/guide/jar/index.html>.