REALTEXT™ AUTHORING GUIDE

RealSystem™ G2

Revision Date: December 21, 1998

RealNetworks, Inc.
1111 Third Avenue, Suite 2900
Seattle, WA 98101 USA

http://www.real.com

CONTENTS

RELEASE UPDATES

If you are new to RealText, please start with "Introduction" on page 2. If you have worked with RealText using a version of this manual published before December 21, 1998, this section gives you a quick look at the changes included in this version of the manual.

Revisions Included December 21, 1998

New Broadcast Files

The RealText broadcast application files included in the `utils` directory of the HTML version of this manual have been updated. Be sure to use the new files if broadcasting RealText.

> **Additional Information**
>
> To get the latest broadcast application, download the bundled HTML version of this manual from **http://service.real.com/help/library/encoders.html**.

Recommendations for Scrollrate and Crawlrate

See the information on the `<window>` tag `scrollrate` and `crawlrate` attributes on page 12 for tips on setting these rates.

Time Tag Tips

A few tips on using `<time.../>` tags have been added.

Updated Font Information

Information about the `<font>` tag on page 18 has been updated. The end of that section contains examples of the `<font>` tag in use.

INTRODUCTION

Welcome to RealText, a RealSystem G2 product for streaming text from files or live sources. With RealText you can create presentations consisting of text alone, or combine text with other media to create, for example, closed-captioned video. This guide tells you how to use the RealText mark-up language to format streaming text for playback in RealPlayer.

Tip

Download the HTML version of this guide from **http://service.real.com/help/library/index.html** to get RealText examples you can view with RealPlayer.

Tools for Creating RealText

You need the following tools to create and test your RealText presentation:

- Text Editor

  To create a RealText file, you can use any word processor, text editor, or XML editor that can save output as plain text.

- RealPlayer G2

  Use RealPlayer G2, available free at **http://www.real.com**, to test your RealText presentation. Other applications may also have RealPlayer G2 features that enable them to receive RealText as well. Note that previous versions of RealPlayer, such as RealPlayer 4.0 and 5.0, cannot display RealText.

- RealServer G2 or a Web server

  RealServer G2 streams RealText presentations to RealPlayer. If you are not operating RealServer yourself, you need to have access to RealServer through, for example, an Internet Service Provider (ISP). Note that previous versions of RealServer cannot stream RealText. A standard Web server can also download a static RealText clip to RealPlayer. It cannot broadcast RealText live, however.

- Broadcast Application

  A broadcast application can capture live text, add RealText mark-up to it, and send it to RealServer. For more information, see Chapter 5 beginning on page 31.

Conventions Used in this Manual

The following table explains the conventions used in this manual.

Notational Conventions

Convention	Meaning
`variables`	Italicized text represents variables. Substitute values appropriate for your situation.
`[options]`	Square brackets indicate optional values you may or may not need to use.
`choice 1|choice 2`	Vertical lines separate values you can choose between.
...	Ellipses indicate nonessential information omitted from the example.

Additional RealSystem Resources

In addition to this manual, you may need the following RealNetworks resources, available at **http://service.real.com/help/library/index.html**:

- *RealSystem G2 Production Guide*

  This manual explains the basics of streaming files with RealSystem. It tells how to calculate bandwidth needs and shows how to put a multimedia presentation together.

- *Embedded RealPlayer Extended Functionality Guide*

  This guide supplements *RealSystem G2 Production Guide*. It explains how to use JavaScript or VBScript to control RealPlayer functions for a presentation embedded in a Web page.

- *RealServer Administration Guide*

  The basic reference for the RealServer administrator, this manual explains how to set up, configure, and run RealServer to stream multimedia. You need this manual only if you are running RealServer yourself.

- RealSystem G2 Software Development Kit (SDK)

  The RealSystem G2 SDK lets you integrate applications with RealSystem. You need the SDK if you want to build a customized RealText broadcast application, for example. A knowledge of programming is required to use the SDK. Download the SDK from **http://www.real.com/devzone/**.

Technical Support

For technical support with RealText, please fill out the form at:

- **http://service.real.com/contact/email.htm**

The information you provide in this form will help technical support personnel to give you a prompt response. For general information about RealNetworks' technical support, visit:

- **http://service.real.com/help/call.html**

Chapter 1

REALTEXT AUTHORING

With RealText you can create streaming text presentations, combining text with other media such as audio or video. You simply create a RealText file with mark-up similar to HTML to describe how and when the text displays. RealServer or a Web server can then stream the text to RealPlayer or Web browsers.

Choosing Window Types

RealText provides a number of window styles that you can choose depending on how you want to display text:

- generic

  A generic window has no preset parameters. You can use it to create any RealText display allowed by the RealText mark-up. You can display and erase lines of text, scroll text through the window, or have text crawl from side to side, for example.

- ScrollingNews

  A ScrollingNews window is preset to have text scroll from the bottom of the window to the top at a set rate for the entire presentation. The text does not crawl from side to side, though.

- TickerTape

  Text in a TickerTape window crawls from the right side of the window to the left. It can also loop back around to the right. It does not scroll up or down, however. Text displays next to the window's top or bottom edge.

- Marquee

  The Marquee window is like the TickerTape in that text crawls from right to left and can loop. It is different in that text is centered vertically within the window.

- TelePrompter

  A TelePrompter window behaves like a generic window except that text arriving at the bottom edge of the window causes the text above it to move up just enough to display the new line.

Using RealText Mark-Up

The RealText mark-up language is similar to HTML. If you are familiar with HTML, you will pick up RealText quickly. The mark-up has some important differences from HTML, though. Keep the following points in mind when writing a RealText file:

- Use lowercase characters for RealText tags and attributes. This keeps your RealText file compliant with the SMIL mark-up language and its XML parent language.
- A tag that does not have a corresponding end tag (for example, the `<ul>` tag has the end tag `</ul>`), closes with a forward slash. For example:

  ```
  <br/>
  ```
- Attribute values must be enclosed in double quotation marks.
- Save your RealText file with the file extension .rt. Do not include spaces in the file name. For example, you can have the file `my_realtext.rt` but not the file `my realtext.rt`.
- Use codes to include angle brackets, ampersands, or nonbreaking spaces as RealText display characters. See “Coded Characters” on page 25.
- As in HTML, you can add a comment to a RealText file like the following. Note that the comment tag does not need to close with a slash.

  ```
  <!-- This is a comment -->
  ```

Additional Information

To learn more about XML, the parent language for RealText, visit **http://www.w3.org/XML**.

Creating a RealText Presentation

The following steps describe how to create a static RealText file. The rest of this manual describes the RealText mark-up in detail.

➤ **To Create a RealText File:**

1. Open a new file in a text editor or XML editor. At the top of the file, add the `<window>` tag with necessary options. Add the `</window>` tag at the bottom:

   ```
   <window...options...>
   ...All Other Mark-up Goes Between These Tags...
   </window>
   ```

   Additional Information

   See "Window Tag Attributes" on page 10.

2. Between the `<window>` and `</window>` tags, add the text that will display in RealPlayer. Format the text with the text tags:

   ```
   <window>
   Mary had a little lamb,
   <br/><time begin="3"/>little lamb,
   <br/><time begin="6"/>little lamb,
   <br/><time begin="9"/>Mary had a little lamb
   <br/><time begin="12"/>whose fleece was white as snow.
   </window>
   ```

   Additional Information

   See "Text Tags" on page 14.

3. Save the file as plain text, using the .rt extension to mark the file as a RealText document. On your local machine, open the RealText file with RealPlayer to test the presentation.

   Tip

   Playing a RealText file on your local machine indicates whether the RealText mark-up is correct. However, it does not guarantee that the file will stream across a network well. Be sure to test that the presentation streams correctly from RealServer.

4. To combine RealText with another file, create a SMIL file that controls the overall presentation. For example, the SMIL file can list a RealText file and video file played together:

```
<smil>
  <body>
    <par>
      <textstream src="rtsp://realserver.company.com/mary.rt"/>
      <video src="rtsp://realserver.company.com/mary.rm"/>
    </par>
  </body>
</smil>
```

Additional Information

See *RealSystem G2 Production Guide* available at **http://service.real.com/help/library/encoders.html** for information about creating a SMIL file.

5. Move your files to RealServer. If you are using an Internet Service Provider, for example, contact the ISP's RealServer administrator for instructions on doing this.
6. In your Web page, add a hypertext link to the SMIL file. Or, if the RealText file is the only file in your presentation, simply link to that file.

   Additional Information

   RealSystem G2 Production Guide available at **http://service.real.com/help/library/encoders.html.html** explains the options for linking your Web page to your presentation. It also explains how to play your RealText presentation in your Web page rather than in RealPlayer.
7. Test the presentation by clicking the hyperlink in your Web page. This launches RealPlayer, which displays the streaming text with the properties you defined through the RealText mark-up.

Notes on Text Streaming

- RealText consumes minimal bandwidth, typically less than 1 Kbps. RealText presentations are therefore easily accessible to users with slow network connections. When combining RealText with another media type, structure the presentation so that RealText has approximately 1 Kbps of available bandwidth.

Additional Information

See *RealSystem G2 Production Guide* available at **http://service.real.com/help/library/encoders.html** for more information about bandwidth consumed by video or audio that accompanies RealText.

- Although RealServer G2 provides reliable streaming, packets occasionally may be lost. If a block of text does not get through, RealPlayer displays the following to indicate missing text:

  ...

Chapter 2

WINDOW TAG ATTRIBUTES

The `<window>` and `</window>` tags that begin and end a RealText file, respectively, determine the window type and set attributes such as window height and width. This chapter explains the options you can set.

> **Note**
>
> When you are familiar with the attributes, you can refer to "Summary of Window Tag Attributes" on page 37.

Specifying Window Attributes

You specify attributes within the `<window>` tag, much as you specify HTML table attributes within the HTML `<TABLE>` tag. For example:

```
<window type="tickertape" duration="2:05:00.0" underline_hyperlinks="false">
```

No attributes are required for the `<window>` tag, however. If you do not specify an attribute, the attribute's default value applies. When you include an attribute, enclose its value in double quotation marks.

type="*window type*"

This attribute defines the RealText window type:

- generic
- tickertape
- marquee
- scrollingnews
- teleprompter

The default is `type="generic"`.

Additional Information

For descriptions of the window types, see “Choosing Window Types” on page 5.

duration=“*dd:hh:mm:ss.xyz*”

The duration attribute specifies the time, relative to the start of the presentation, that this RealText stream stops playing. The default is 60 seconds. The RealPlayer timing slider is keyed to this value, which is in 24-hour format, where `dd` is days, `hh` is hours, `mm` is minutes, `ss` is seconds, `x` is tenths of seconds, `y` is hundredths of seconds, and `z` is milliseconds.

Only the `ss` value is required. When the time value does not include a decimal point, the last field is read as the seconds. For example, `2:05` means 2 minutes and 5 seconds, whereas `2:05:00` means 2 hours and 5 minutes. You can also specify just the seconds. For example, both of the following values end the text stream 2 hours and 5 minutes after the stream begins:

```
duration=”2:05:00.0”
duration=“7500”
```

Tip

Set a high duration when you start building a RealText presentation. Set the final duration time when you have finished defining the mark-up.

width=“*pixels*”

The `width` attribute determines the window width in pixels. The default is 500 for TickerTape and Marquee windows, 320 for other window types. SMIL layout tags can specify a playback region width that overrides the width set here. If word wrap is on, the line length for wrapping corresponds to this width value, not the actual region width. Text centering, however, corresponds to the actual region width set in the SMIL file.

Additional Information

See *RealSystem G2 Production Guide* available at **http://service.real.com/help/library/encoders.html** for information about SMIL.

height=“*pixels*”

The `height` attribute sets the window height in pixels. The default is 30 for TickerTape and Marquee windows, 180 for other window types. SMIL layout tags can specify a window height that overrides the height set here. This typically does not adversely affect the presentation, however.

bgcolor="*color*"

This attribute determines the window's background color. The default is black for TickerTape windows and white for all other window types. See "Colors" on page 24 for information on valid colors.

scrollrate="*pixels per second*"

The `scrollrate` attribute sets the number of pixels per second that the text moves vertically. It has no effect on TickerTape and Marquee windows. The default is 10 for ScrollingNews windows and 0 for all other window types.

> **Tip**
>
> For best results, use a `scrollrate` under 30. (Best values are 25, 20, 10, 8, 5, 4, 2, and 1.) For rates faster than 30, use multiples of 20 or 25, such as 40, 50, 60, 75, 80, and so on.

crawlrate="*pixels per second*"

The `crawlrate` attribute specifies the number of pixels per second that the text moves horizontally. The default is 20 for TickerTape and Marquee windows, 0 for other window types.

> **Tip**
>
> For best results, use a `crawlrate` under 30. (Best values are 25, 20, 10, 8, 5, 4, 2, and 1.) For rates faster than 30, use multiples of 20 or 25, such as 40, 50, 60, 75, 80, and so on.

link="*color*"

This attribute sets the color of hyperlinks within the text. The default is blue. See "Colors" on page 24 for other color options.

underline_hyperlinks="true|false"

This attribute determines whether hyperlinks are underlined. The default is true.

wordwrap="true|false"

This attribute, which defaults to true, specifies whether word wrap is performed. When word wrap is on, text lines longer than the specified window width wrap to the following line. If it is off, long lines are truncated by the window border. This attribute has no effect for windows that have horizontal text motion, such as the TickerTape.

loop="true|false"

This attribute is available only in TickerTape and Marquee windows, where it defaults to true. When set to true, this attribute tells RealPlayer to buffer all text and redisplay ("loop") it if and when the stream runs dry, which occurs when the text has moved out of the window and no new text has arrived. If the text has looped and new text arrives, the new text displays as soon as the old text has moved out of the window. The new text then becomes part of the loop.

extraspaces="use|ignore"

The default value "`use`" for this variable makes RealText recognize all blank spaces between mark-up tags. If three spaces occur between two words in the RealText file, for example, RealPlayer displays all three spaces. It also recognizes carriage returns and tabs.

If you specify `extraspaces="ignore"`, RealPlayer treats spaces, tabs, line feeds, and carriage returns as does a Web browser, except when they are between the `<pre>`...`</pre>` tags described on page 18. When spaces or carriage returns occur contiguously in the text, RealPlayer interprets them as a single space, no matter how many of them are present. So in this case the three spaces display as one space in RealPlayer. It treats each tab as a single space, though.

Chapter 3

TEXT TAGS

RealText provides many mark-up tags that define how the streaming text looks and operates. A tag's default value applies if you do not specify a tag value. You can place mark-up tags anywhere on a line.

Note

When you are familiar with the tags, you can refer to "Summary of Text Tags" on page 39.

Time and Position Tags

The following tags affect when and where the text appears within the window.

<time begin="*dd:hh:mm:ss.xyz*"/>
<time end="*dd:hh:mm:ss.xyz*"/>

The `<time/>` tags control the RealText presentation timeline by determining when a text component appears and disappears, respectively, relative to the start of the presentation. They are meant primarily for window text that does not scroll or crawl. If you do not specify begin times, RealPlayer displays all text as quickly as it can.

The `<time/>` tag values are in 24-hour format, where `dd` is days, `hh` is hours, `mm` is minutes, `ss` is seconds, `x` is tenths of seconds, `y` is hundredths of seconds, and `z` is milliseconds.Only the `ss` field is required. When the time value does not include a decimal point, the last field is read as the seconds. For example, `1:30` means 1 minute and 30 seconds, whereas `1:30:00` means 1 hour and 30 minutes. Note that all the following values are equivalent. They all start the text component 90 minutes after the stream begins:

```
<time begin="1:30:00.0"/>
<time begin="90:00"/>
<time begin="5400"/>
```

Text with an end value is erased when the specified end value is reached. Otherwise it stays active until the presentation ends or the entire window is erased with <clear/>. Note that you can combine the begin and end attributes in a single <time/> tag as shown here:

```
<time begin="23" end="55.5"/>This text displays 23 seconds into the presentation
and disappears at 55.5 seconds.
```

All text following a <time/> tag has the specified begin and end values until new values are given. Once you specify an end time for a text component, you must specify an end time for all following components. For example, the following text would not display properly:

```
<time begin="23" end="55.5"/>Display at 23 seconds in.
<time begin="56"/>Display at 56 seconds in.
```

Because the second line does not include an end value, the previous end value of 55.5 still applies. The second line cannot be displayed because its begin time is later than its end time.

Here are some times on using <time/> tags:

- The <time/> tags are not necessary in a window with a non-zero scroll rate or crawl rate unless you want text to become visible after it has moved into the window, or to disappear before it moves out of the window.
- To freeze text on the screen after the <window> tag's duration has elapsed, do not set an end time. Or, have the end time exceed the window's duration:

  ```
  <window duration="30">
    ...(some text elements)...
    <time begin="25" end="31"/>Text that stays frozen onscreen.
  </window>
  ```

- To replace a line of text with a new line every few seconds (as in video subtitles), do not use end times. For each new line of text, set the appropriate begin time followed by a <clear/> tag.

<clear/>

This tag clears the existing text buffers to remove all text from the window. The text that follows this tag is then displayed starting at the window's normal starting point.

In a window that does not scroll or crawl, you can add <clear/> after <time begin="..."/> to erase existing text when new text arrives. For example, you would specify the following to clear old text and display "Hello!" at 3 minutes into the stream:

```
<time begin="3:00"/><clear/>Hello!
```

However, a <clear/> tag does not remove text that has an end time that has not yet elapsed. Consider this example:

```
<time begin="5"/>They all lived happily.
<time begin="10" time end="20"/>And so our story ends.
<time begin="15"/><clear/>Goodbye!
```

The second line of text is set to end at 20 seconds. The <clear/> tag, which begins at 15 seconds, does not clear this line because the line's end time has not elapsed. The <clear/> tag removes the first line of text, though, which has no end time.

<pos x="*pixels*"/>
<pos y="*pixels*"/>

These tags position the text horizontally and vertically, respectively. The <pos y/> tag moves the upper, left corner of the subsequent text block the specified number of pixels down from the window's top edge. The <pos x/> tag indents the text block the specified number of pixels in addition to the two-pixel default padding that applies to all text blocks. You can combine both tags in a single tag like this:

```
<pos x="10" y="55"/>
```

Note

These tags work only if scroll rate and crawl rate are both 0 (zero).

<tu [color="*color*"]>...</tu>
<tl [color="*color*"]>...</tl>

These tags function only with TickerTape windows. They display the enclosed text at the window's upper (<tu>) or lower (<tl>) edge. When a tag specifies a color with the color option, the color applies to text enclosed by all subsequent tags of that type until another tag of that type changes the color. However, color specified for <tu> elements does not affect color for <tl> elements, and vice versa.

Additional Information

Refer to "Colors" on page 24 for a list of available colors.

Layout Tags

Much as in HTML, the following tags let you define the layout of RealText in the RealPlayer window.

<p>...</p>

Adds space between text. In TickerTape and Marquee windows, it moves the "cursor" to the right edge of the window. In all other window types, the `<p>` and `</p>` each cause the next text to display two lines down.

<br/>

Adds space between text. In TickerTape and Marquee windows, it moves the "cursor" to the right edge of the window. In all other window types, this tag causes the text that follows to display on the next line.

<ol>...</ol>

For compatibility with HTML lists. Text between these tags is indented, but not numbered.

<ul>...</ul>

For compatibility with HTML lists. Text between these tags is indented, but not bulleted.

<li>...</li>

For compatibility with HTML lists. Acts like a `<br/>` tag.

<hr/>

For compatibility with HTML horizontal rules. Acts as two `<br/>` tags.

<center>...</center>

Used to center the enclosed text. Text is centered according to the actual window width, which may differ from the width attribute. These tags behave the same as HTML centering tags. The `<center>` tag forces a line break if and only if a line break caused by a tag such as `<br/>`, `<p>`, or `<hr/>` does not immediately precede it. The `</center>` tag always causes a line break.

> **Note**
>
> RealText does not center text until it has determined the line length. In rare instances, one streamed packet may contain the first part of the line while another packet received several seconds later contains the end of the line. In this case, the first part displays flush left, and the entire line is centered and redisplayed when the second packet arrives.

<pre>...</pre>

Works the same as in HTML. Text tagged with <pre> uses the Courier font at the current size. For example, a preceding <font size="+1"> makes the preformatted text one size larger than the default font size. Line breaks, spaces, and tabs are preserved, with tabs defaulting to 64 pixels for 16 point text (the normal point size). Tab spaces are determined by dividing the text height by 2, then multiplying by 8.

Additional Information

For information on text heights, see the "Font Sizes" table on page 18. See also the <window> tag attribute extraspaces="use|ignore" on page 13.

Appearance Tags

You can use the following set of tags to change the appearance of text.

<b>...</b>

These tags display the enclosed text **bolded**.

<i>...</i>

These tags display the enclosed text *italicized*.

<s>...</s>

These tags ~~strike through~~ the enclosed text.

<u>...</u>

These tags display the enclosed text underlined.

<font attribute="*value*">...</font>

The <font> tag lets you specify text characteristics. Like the HTML <FONT> tag, it uses an end tag, </font>, and can contain several attributes. For example:

<font size="+4" face="courier">...*text*...</font>

RealText supports the following <font> attributes:

- bgcolor="*color*"

  Use this <font> tag attribute to set the text background color. The default is bgcolor="transparent".

Additional Information

See "Colors" on page 24.

- charset="*character set*"

  With this <font> tag attribute you can control the character set used to display the text:

  - "us-ascii"–Default character set used with most RealText fonts.
  - "x-sjis"–Character set for Kanji and the Osaka font.
  - "gb2312"–Character set for Simplified Chinese.
  - "big5"–Character set for Traditional Chinese.

  Note

  Not all computers are capable of displaying all character sets. If a computer does not recognize the specified character set, it displays the text in its default character set. The result is typically unreadable.

  Warning

  RealText always defaults to the us-ascii character set regardless of the default character set of the machine receiving the stream. You must explicitly specify a different character set to stream RealText to machines that do not render us-ascii.

- color="*color*"

  This <font> tag attribute lets you control the font color. It supports all color values available in HTML. For TickerTape windows, however, this attribute has no effect. The <tu> and <tl> tags set the TickerTape text colors.

- face="*font name*"

  This <font> tag attribute controls the text font. Use a font name from the "Windows Font Name" column of the following table, which lists fonts using the us-ascii character set. If you specify no font, RealText uses Times New Roman. It also uses this font if the user's machine does not have the specified font. In this case, however, the text still uses the specified font's letter spacing, so display results will vary.

RealText Font Support for us-ascii Character Set

Windows Font Name	Macintosh Mapping	UNIX Mapping
Algerian	Courier	(tbd)
Arial	Helvetica	Helvetica
Arial Black	Helvetica	(tbd)
Arial Narrow	Helvetica	(tbd)
Arial Rounded Mt Bold	Helvetica	(tbd)
Book Antiqua	Helvetica	(tbd)
Bookman Old Style	Helvetica	(tbd)
Braggadocio	Helvetica	(tbd)
Britannic Bold	Helvetica	(tbd)
Brush Script	Times	(tbd)
Century Gothic	Helvetica	(tbd)
Century Schoolbook	Helvetica	(tbd)
Colonna Mt	Times	(tbd)
Comic Sans Ms	Times	(tbd)
Courier New	Courier	Courier
Desdemona	Helvetica	(tbd)
Fixedsys	Courier	(tbd)
Footlight Mt Light	Helvetica	(tbd)
Garamond	Times	(tbd)
Haettenschweiler	Helvetica	(tbd)
Helvetica (Arial is used if Helvetica is not found.)	Helvetica	Helvetica
Impact	Helvetica	(tbd)
Kino Mt	Times	(tbd)
Matura Mt Script Capitals	Times	(tbd)
Modern	Helvetica	(tbd)
Ms Dialog	Times	(tbd)
Ms Dialog Light	Times	(tbd)
Ms Linedraw	Helvetica	(tbd)

(Table Page 1 of 2)

RealText Font Support for us-ascii Character Set (continued)

Windows Font Name	Macintosh Mapping	UNIX Mapping
Ms Sans Serif	Helvetica	(tbd)
Ms Serif	Helvetica	(tbd)
Ms Systemex	Times	(tbd)
Playbill	Times	(tbd)
Small Fonts	Times	(tbd)
System	Geneva	Fixed
Terminal	Geneva	(tbd)
Times New Roman	Times	Times
Verdana	Helvetica	(tbd)
Wide Latin	Helvetica	(tbd)

(Table Page 2 of 2)

Note

Macintosh and UNIX mappings subject to change before the final release of RealPlayer G2 on those platforms.

RealText also supports the following font faces that use character sets other than us-ascii.

RealText Font Support for Additional Character Sets

Font Name	Characters	RealText Font Face Tag	charset
Osaka	Kanji	<font face="Osaka">	x-sjis
’ 宋体	Simplified Chinese	<font face="'ËÎÌå"> (The face name displays as gibberish without the gb2312 character set.)	gb2312
細明體	Traditional Chinese	<font face="²Ó©úÅé"> (The face name displays as gibberish without the big5 character set.)	big5

- size="n"

This <font> tag attribute lets you control the font size. You can use relative sizes or absolute sizes as shown in the table below. This table also lists the height in pixels for each size. The pixel sizes are for reference only. You cannot specify a pixel size directly in RealText.

Font Sizes

Relative Size	Absolute Size	Pixel Size Reference
-2	1	12 pixels
-1	2	14 pixels
+0 (default)	3	16 pixels
+1	4	20 pixels
+2	5	24 pixels
+3	6	36 pixels
+4	7	48 pixels

Note

You can also specify relative sizes smaller than -2 or larger than +4, but they are treated as -2 and +4, respectively.

Keep in mind that the RealText `<font>` tag works like the HTML `<FONT>` tag. How you nest tags, which attributes you include, and where you place `</font>` tags affects the outcome. For example, compare these RealText samples (**bolding** used for emphasis only):

```
Start with normal text.
<font color=“red”>Make text red.
<font size=”+1”>Make red text one size larger.
</font>Turn off larger size for red text.
</font>Turn off color.
```

```
Start with normal text.
<font color=“red”>Make text red.
</font>Turn off color.
<font size=”+1”>Make text one size larger.
</font>Turn off larger size.
```

```
Start with normal text.
<font color=“red” size=”+2”>Make text red and two sizes larger.
<font color=“blue” size=”+1”>Make text blue and one size smaller.
</font></font><font size=+1”>Turn off color but keep text the same size.
</font>Reduce text to normal.
```

Command Tags

The following tags let you embed RealPlayer commands in your presentation or modify the default streaming behavior.

<a href="*URL*" [target="_player"]>...</a>

This tag makes the enclosed text a hyperlink to the specified URL, which should begin with a protocol designation such as `http://` or `rtsp://`. For static files, you can also specify URLs relative to the location of the RealText source file. For example, the link `<a href="more.htm">...</a>` opens the file `more.htm` in the same directory as the RealText file. Relative links follow the HTML directory syntax.

Additional Information

SMIL files can also define hypertext links that may override the link you set here. For more information, see the SMIL chapter in *RealSystem G2 Production Guide*.

You can also include the `target="_player"` attribute to launch the new stream in the current RealPlayer window. If you do not use the target attribute or you specify `target="_browser"`, the linked URL opens in a Web browser window.

Note

The link text is the color specified in the link attribute of the `<window>` tag. The link is underlined unless the `<window>` tag includes `underline_hyperlinks="false"`.

<a href="mailto:*address*">...</a>

This tag turns the enclosed text into an e-mail hyperlink. When the viewer clicks the link, RealText passes the e-mail address to the viewer's browser. Use an address in the standard form, such as `name@company.com`. If the browser is configured for e-mail, the e-mail client opens a new message with the defined address in the "to" line.

<a href="*command*" target="_player">...</a>

This tag makes the enclosed text a hyperlink that, when clicked, executes a RealPlayer command. The commands are case-sensitive and must be enclosed in double quotes. The `target="_player"` attribute is required. The following command instructs RealPlayer to seek to the specified time in the current text stream:

```
<a href="command:seek(time)" target="_player">
```

For example, the following instructs RealPlayer to seek to 1:35.4 in the stream:

```
<a href=”command:seek(1:35.4)” target=”_player”>
```

When clicked, the following link causes RealPlayer to pause the stream:

```
<a href=”command:pause()” target=”_player”>
```

When clicked, the following link causes RealPlayer to begin or resume playing the stream:

```
<a href=”command:play()” target=”_player”>
```

Note

The link text is the color specified in the link attribute of the <window> tag. The link is underlined unless the <window> tag includes underline_hyperlinks=”false”.

<required>...</required>

Use these tags to enclose text that must be delivered to RealPlayer under any circumstance. During extremely adverse network conditions, RealSystem will halt the presentation if necessary rather than drop the text. You can use these tags sparingly, though, because RealSystem ensures that very little data loss occurs in transmission.

Colors

For the RealText color options, you can use any colors available through the HTML <FONT COLOR> tag. This includes Red/Green/Blue hexadecimal values (#RRGGBB), as well as these predefined color names, listed here with their corresponding hexadecimal values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuchsia (#FF00FF)	red (#FF0000)	maroon (#800000)
lime (#00FF00)	olive (#808000)	green (#008000)	purple (#800080)
aqua (#00FFFF)	teal (#008080)	blue (#0000FF)	navy (#000080)

Transparency

You can also use ”transparent” as a color. For example, <font bgcolor=”transparent”> means that each following word does not have a rectangle drawn behind it. This lets you draw text on top of previous text (using the <pos/> tags) without “erasing” the previous text.

Note

Transparency is not currently supported as a window background color.

Coded Characters

The following table lists the character codes you can include in a RealText source file. Codes begin with an ampersand ("&") and end with a semicolon (";"). RealText interprets these characters the same way as popular Web browsers.

RealText Coded Character Set

Code	Displays as
&lt;	<
&gt;	>
&amp;	&
&nbsp;	(nonbreaking space)

&#32;
to
&#255;
	ISO Latin-1 characters. For a list of these characters, visit the W3C reference at **http://www.w3.org/MarkUp/html-spec/html-spec_13.html**.

For example, the following RealText source text:

```
This is a bold tag: "&lt;b&gt;".
```

is displayed in a RealText window as:

```
This is a bold tag: "<b>".
```

Chapter 4

EXAMPLES

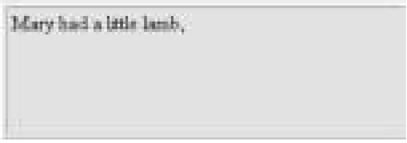
This following examples show how to create and tag the supported RealText window types.

Generic Window

The following example illustrates a RealText source document and resulting display for a generic RealText window. This is the RealText source file (extension .rt):

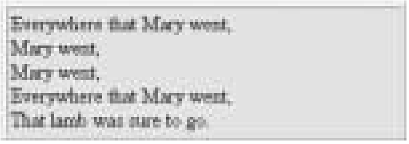
```
<window duration=”30” bgcolor=”yellow”>
Mary had a little lamb,
<br/><time begin=”3”/>little lamb,
<br/><time begin=”6”/>little lamb.
<br/><time begin=”9”/>Mary had a little lamb,
<br/><time begin=”12”/>whose fleece was white as snow.
<br/><time begin=”15”/><clear/>Everywhere that Mary went,
<br/><time begin=”18”/>Mary went,
<br/><time begin=”21”/>Mary went,
<br/><time begin=”24”/>Everywhere that Mary went,
<br/><time begin=”27”/>That lamb was sure to go.
</window>
```

When RealPlayer processes this file, it displays only the first line of the text from zero to three seconds into the stream:



Mary had a little lamb,

Every three seconds after the first line displays, a new line appears as specified by <time begin=”*seconds*”/>. At 15 seconds, <clear/> clears the displayed text and resets the text “cursor” to the top, left-hand corner of the window. When the stream finishes, all lines of text following the last `<clear/>` tag appear in the window:



Everywhere that Mary went,
Mary went,
Mary went,
Everywhere that Mary went,
That lamb was sure to go.

Note the following about this example:

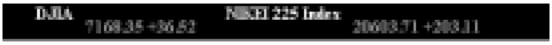
- Because it was not specified in the `<window>` tag, word wrapping defaults to true. However, word wrapping is not necessary because `<br/>` tags force line breaks.
- `<time/>` tags need not appear after a `<br/>` tag. They can appear anywhere in the text.
- The example could have used `<time end="`*`time`*`"/>` tags to make individual lines of text disappear before the `<clear/>` tag cleared all the lines.

TickerTape Window

The following example shows the RealText source document and resulting display for a TickerTape window. This is the RealText (`.rt`) source file:

```
<window type="tickertape" duration="1:00" width="350" loop="true"
underline_hyperlinks="false" link="white">
<br/><b>
<tu><a href="http://www.dowjones.com/">DJIA</a></tu>
<tl>7168.35 +36.52 </tl>
<tu>NIKEI 225 Index</tu>
<tl>20603.71 +203.11</tl>
</b>
</window>
```

This source file produces the following window in RealPlayer.



DJIA 7168.35 +36.52 NIKEI 225 Index 20603.71 +203.11

Note the following about this example:

- The text crawls from right to left at 20 pixels per second, the default crawlrate for a TickerTape window.
- The `<b>` tag at the start **bolds** all following text.

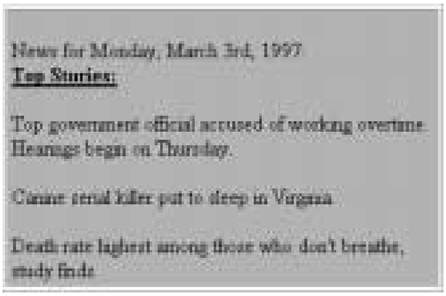
- `<a href=”http://www.dowjones.com/”>DJIA</a>` makes `DJIA` a hyperlink that, when clicked, opens a browser to **http://www.dowjones.com/**.
- `DJIA` is not underlined because `underline_hyperlinks=”false”` is declared in the `<window>` tag. It is drawn in white because `link=”white”` is also in the `<window>` tag.
- The attribute `loop=”true”` in the `<window>` tag means the text loops around and comes back in from the right side of the window as soon as the last character of the text has moved completely out of the window. It is not necessary to specify this attribute explicitly, because in TickerTape windows `loop=”true”` is the default.
- The `<br/>` tag that comes before the first text item forces the text that follows to start just past the window’s right edge. Any break or paragraph tag inside TickerTape text causes the text that follows to start at the right edge. If the `<br/>` tag were absent, the data would appear starting at the window’s left edge.

Scrolling News Window

The following example demonstrates a ScrollingNews window. This is the RealText (.rt) source file:

```
<window type=”scrollingnews” duration=”10” bgcolor=”aqua”>
<br/><br/>News for Monday, March 3rd, 1997:
<br/><b><u>Top Stories:</u></b>
<p>Top government official accused of working overtime.
Hearings begin on Thursday.</p>
<time begin=”1”/>Canine serial killer put to sleep in Virginia.
<p>Death rate highest among those who don't breathe, study finds.</p>
</window>
```

This source file produces the following window in RealPlayer.



News for Monday, March 3rd, 1997

<u>Top Stories:</u>

Top government official accused of working overtime.
Hearings begin on Thursday.

Canine serial killer put to sleep in Virginia.

Death rate highest among those who don't breathe,
study finds.

The example above shows what the text box looks like 1.5 seconds into the presentation. The text window and text appear at the 0,0 (top left) coordinates on the screen. At one second into the stream, the second and third items appear as specified in the `<time begin=”1”/>` tag.

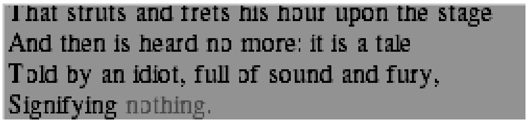
Note that the second item becomes visible after it scrolls into the window. This demonstrates how you can control the visibility of text with the `<time/>` tag. Without the `<time/>` tag, the text would have appeared before one second and would have scrolled up from the bottom of the window.

TelePrompter Window

The following example demonstrates a TelePrompter window. This is the RealText (.rt) source file:

```
<window type=”teleprompter” height=”60” duration=”25”
bgcolor=”lime” wordwrap=”false”>
Out, out, brief candle!
<br/><time begin=”3.5”/>Life's but a walking shadow, a poor player
<br/><time begin=”7”/>That struts
<time begin=”8”/>and frets
<time begin=”9”/>his hour upon the stage
<br/><time begin=”12”/>And then is heard no more:
<time begin=”15”/>it is a tale
<br/><time begin=”16”/>Told by an idiot,
<time begin=”17.5”/>full of sound and fury,
<br/><time begin=”20”/>Signifying
<time begin=”22”/><font color=”red”>nothing.</font>
</window>
```

When the window fills with text and a new line appears, all lines scroll up to make room for the new line. The following illustrates the window when the presentation ends.



That struts and frets his hour upon the stage
And then is heard no more: it is a tale
Told by an idiot, full of sound and fury,
Signifying nothing.

Note the following about TelePrompter windows:

- The `wordwrap` attribute can be `true` or `false`.
- The `scrollrate` and `crawlrate` attributes are ignored.

- You can use a `<clear/>` tag to clear the window and start the next line at the window's upper, left-hand corner.
- Use `<time begin/>` tags at the start of each line and do not let word wrapping cause too many line breaks between `<time/>` tags.
- Multiple lines of text with the same begin time cause the preceding text to move up until all new lines appear at the bottom of the window.

Chapter 5

REALTEXT BROADCAST

Because RealServer can also broadcast RealText live, RealText is an ideal solution for delivering live stock market information or breaking news, for example. Because it is a true streaming server, RealServer streams the data as it comes in, instead of downloading blocks of data to users' computers for later display. This chapter explains how to use the RealText broadcast application.

Additional Information

RealSystem G2 Production Guide provides an overview of broadcasting. For more on configuring a broadcast, see *RealServer Administration Guide*. Both are available at **http://service.real.com/help/library/encoders.html**.

Broadcast Application

You can broadcast RealText to multiple viewers with an application included in the utils directory of the HTML version of this manual. To get this application, download the bundled HTML version of this manual from http://service.real.com/help/library/encoders.html.

The broadcast application runs on any Windows 32-bit operating system. Its C++ source code is publicly available, though, and you can compile it on a different operating system as noted in "Developing a Custom RealText Broadcast Application" on page 36.

The RealText broadcast application connects to RealServer G2 and polls a specified directory for an updated RealText file. When it finds an updated file, it sends the file to RealServer, which broadcasts the file contents to the connected RealPlayers.

Tip

Available at **http://www.real.com/products/tools/**, RealNetworks tools let you broadcast RealAudio and

RealVideo as well. For information on broadcasting RealPix, download *RealPix Authoring Guide* from **http://service.real.com/help/library/encoders.html**.

Installing the RealText Broadcast Application

The RealText broadcast application comprises the **`exlvtxt2.exe`** executable file and a few DLLs. These files must reside together, but you can move them from the `utils` directory to any directory on the RealText broadcast machine. You can broadcast on the RealServer G2 machine or any Windows 32-bit machine with a network connection to RealServer G2. RealText broadcast requires minimal system resources, but RealNetworks recommends using a Pentium 133 MHz or faster processor with at least 32 Megabytes of memory.

Tip

The RealText broadcast application consumes unused CPU for monitoring purposes but releases CPU as other processes need it. On an otherwise idle machine, CPU use for RealText broadcast may seem high, but the actual CPU requirement for broadcasting is small.

Creating a Broadcast RealText Clip

Create a RealText clip for broadcast as a simple text file. Use the RealText mark-up tags described in "Text Tags" beginning on page 14 to format the display text. The file should not have `<window>` and `</window>` tags. You set window attributes such as type, width, and height when you start the broadcast application. RealServer sends these attributes to each RealPlayer when it connects to the broadcast.

Tips for Creating Broadcast RealText Files

Here are some pointers on using RealText mark-up when broadcasting a file:

- Start a new input file with the `<clear/>` tag described on page 15 to clear the existing text in the RealPlayer window. Do not use a `<clear/>` tag in the middle of an input file, though. In a broadcast, a `<clear/>` tag does not need to follow a `<time begin.../>` tag.
- Avoid using `<time.../>` tags to specify when text appears. Timing elements specify offsets from the start of the broadcast, not from the receipt of the

RealText update that includes the elements. Using timing tags can therefore cause unexpected results.

- Keep RealText broadcast files small and broadcast them in a "real time" manner. Do not send large files and use RealText timing tags to delay when the text appears in the RealPlayer window.

Setting the Broadcast URL

Consult with the RealServer administrator to determine the URL for the RealText broadcast. If you are linking a Web page directly to the RealText broadcast, the URL may look like this:

```
<a href="http://realserver.company.com/ramgen/encoder/media/news.rt">...</a>
```

This URL includes two virtual directories. The `ramgen` virtual directory makes RealServer launch RealPlayer without a separate RAM file. The `encoder` virtual directory specifies a broadcast coming in on a certain port of RealServer rather than an actual file existing on a file system. The RealServer administrator sets up these virtual directories.

If you use a SMIL file, the Web page URL to the file may look like this:

```
<a href="http://realserver.company.com/ramgen/media/daily.smi">...</a>
```

Within the SMIL file, you'll have an RTSP link to the RealText broadcast:

```
<textstream src="rtsp://realserver.company.com/encoder/media/news.rt"/>
```

Additional Information

For more on using SMIL and specifying URLs for media clips, see *RealSystem G2 Production Guide*, available at **http://service.real.com/help/library/encoders.html**.

Using SMIL, you can embed a RealText broadcast in a multiclip presentation. You might create a RealVideo clip region and a RealText region that displays breaking news, for example. The SMIL file uses the broadcast URL for the text and a standard URL for the on-demand RealVideo clip. Each person who views the presentation sees the video from its normal beginning, but joins the RealText broadcast in progress.

Starting the RealText Broadcast Application

To run the RealText broadcast application, start the Windows DOS prompt and move to the directory that holds the application. Then start the application as shown in this example:

```
c:\RealText\Broadcast> exlvtxt2.exe
```

The application prompts you for a number of parameter values that the following table describes. Enter all values without quotation marks.

RealText Broadcast Application Start-up Parameters

Parameter	Value
host	The network address of the RealServer G2 machine that will stream the RealText clip. This can be a DNS name such as `realserver.company.com` or an IP address such as `204.71.154.5`.
port	The port on the RealServer machine that receives the RealText data from the broadcast application. The RealServer administrator determines which port is used.
username	A user name set by the RealServer administrator. The broadcast application must supply this user name to RealServer G2 to connect to it.
password	The password for the user name. The application must supply this to connect to RealServer G2. The RealServer administrator sets the password.
stream file name	The "file name" for the RealText live stream. This file name is used in the RealText hypertext link within the SMIL file or Web page. It does not have to be the same as the input file, but it should end with the `.rt` extension.
input file	Name of the RealText file on the broadcast machine to monitor. If the file is not in the same directory as the RealText broadcast application, give the full path as in `c:\RealText\Broadcast\source\news.rt`.
polling period	Frequency in seconds that the broadcast application checks for an update to the input file. The default is 1 second.
window type	Window type as listed in `type="window type"` on page 10. The generic window type is the default.

(Table Page 1 of 2)

RealText Broadcast Application Start-up Parameters (continued)

Parameter	Value
use defaults?	Whether to use the defaults for the chosen window type. Press **Return** to use all default values as described in "Window Tag Attributes", starting on page 10. Enter `n` to have the application step through the window values. For example, the application asks for a pixel value for the window width. If you selected a generic window, you can use the default value of 320 or enter a different pixel width. There is no duration time for live RealText.
Average bit rate	Estimated average bit rate for the live RealText stream in bits per second. The default is 1000 (1 Kbps). When the text stream is part of a multiclip presentation, RealPlayer uses this estimate to determine if it has enough bandwidth to play the presentation.

(Table Page 2 of 2)

Moving a RealText File to the Broadcast Directory

Once started, the broadcast application checks the designated directory with the specified polling frequency for an updated version of the RealText input file. You can place a new file in the directory manually or use any automated method, such as a database query, to create the file. All file updates must use the same input file name specified during application start-up.

The RealText broadcast application resends prior data every three seconds when there is no new data. If you update the input file at 5 and 7 minutes into the broadcast, for example, a viewer joining the broadcast at the 6-minute mark receives the data originally sent at the 5-minute mark within three seconds. That viewer's RealPlayer then ignores all further resends and next displays the file update broadcast at the 7-minute mark.

Stopping the Broadcast Application

To stop the RealText broadcast stream, update the RealText input file so that it has a dollar sign ($) as its first character. Text can follow the dollar sign, but it is ignored:

```
[file start]
$ ignored text
[file end]
```

This file is not broadcast, but causes RealServer to terminate the broadcast stream. It also shuts down the broadcast application console window on the broadcast machine.

Warning

Do not stop the RealText broadcast by pressing **Ctrl+C** from the broadcast application console. This terminates the window but will not properly shut down the RealText stream on RealServer.

Developing a Custom RealText Broadcast Application

The RealSystem G2 SDK, which you can download from **http://www.real.com/devzone**, includes the C++ source code for the RealText broadcast application. Subject to the SDK license agreement, you can customize the broadcast application to do any of the following:

- Compile the RealText broadcast application to run on a different operating system.
- Create a new broadcast application that adds RealText mark-up to a live text stream and sends the data to RealServer.
- Integrate RealText broadcast features into another application.

Closed-Captioning

For closed-captioning, use a device such as TextGrabber from SunBelt Industries (**http://www.sunbeltindustries.com**). This device supplies text from a TV signal through an RS232 interface. Your customized RealText broadcast application can receive the text, format it with RealText mark-up, and send it to RealServer for broadcast.

Appendix A

SUMMARY OF WINDOW TAG ATTRIBUTES

Attribute	Function	Default	Notes
bgcolor="color"	Sets the window's background color.	black for TickerTape, white for others	Refer to "Colors" on page 24.
crawlrate="pixels per second"	Sets the pixels per second that the text moves horizontally.	20 for TickerTape and Marquee, 0 for others	None.
duration="dd:hh:mm:ss.xyz"	Specifies time that presentation stops.	60 seconds	Only seconds ("ss") are required.
extraspaces="use|ignore"	Recognizes or ignores extra spaces in text.	use	ignore value not used with <pre>...</pre>
height="pixels"	Sets the window height in pixels.	30 for TickerTape and Marquee, 180 for others	SMIL can override window height setting.
link="color"	Sets the color of hyperlinks.	blue	Refer to "Colors" on page 24.
loop="true|false"	Determines whether looping occurs.	true for TickerTape and Marquee, false for others	Used only for TickerTape and Marquee.
scrollrate="pixels per second"	Sets the pixels per second that the text moves vertically.	10 for ScrollingNews, 0 for others	No effect on TickerTape or Marquee.
type="window type"	Sets the window type.	generic	Other values are tickertape, marquee, scrollingnews, and teleprompter.
underline_hyperlinks="true|false"	Determines whether hyperlinks are underlined.	true	None.

(Table Page 1 of 2)

Attribute	Function	Default	Notes
`width="pixels"`	Determines the window width in pixels.	`500` for TickerTape and Marquee, `320` for others	SMIL can override window width setting.
`wordwrap="true|false"`	Specifies whether word wrap is performed.	`true`	No effect on TickerTape or Marquee.

(Table Page 2 of 2)

Appendix B

SUMMARY OF TEXT TAGS

Tag	Attributes	Function	Notes
<a href=“command” target=“_player”>…</a>	(none)	Creates hyperlink that executes a command.	Target required.
<a href=“mailto:address”>…</a>	(none)	Opens e-mail message.	Browser must be configured for e-mail.
<a href=“URL” [target=“_player”]>…</a>	target= ”_player”	Creates hyperlink to specified URL.	URL opens in browser if target not set to player.
<b>…</b>	(none)	**Bolds** the enclosed text.	Compatible with HTML.
<br/>	(none)	Adds space between text.	In TickerTape and Marquee, moves “cursor” to the window's right edge.
<center>…</center>	(none)	Centers the enclosed text.	Compatible with HTML.
<clear/>	(none)	Clears all text from the window.	Following text is displayed at the window's normal starting point.
<font attribute=“value”>…</font>	bgcolor	Sets the background color.	See “Colors” on page 24.
	charset	Specifies character set used to display text.	Default is “us-ascii”.
	color	Controls font color except for TickerTape window.	Supports all color values available in HTML. See also <tl> and <tu>.
	face	Sets the text face.	Default is “Times New Roman”.
	size	Sets relative font size.	Values are: -2, -1, +0, +1, +2, +3, +4; or 1, 2, 3, 4, 5, 6, 7.

(Table Page 1 of 2)

Tag	Attributes	Function	Notes
<hr/>	(none)	Acts like two <br/> tags.	For HTML compatibility.
<i>...</i>	(none)	*Italicizes* the enclosed text.	Compatible with HTML.
<li>...</li>	(none)	Acts like a <br/> tag.	For HTML compatibility. No numbers or bullets.
<ol>...</ol>	(none)	Indents text.	For HTML compatibility. No numbers, however.
<p>...</p>	(none)	Adds space between text.	In TickerTape and Marquee, moves “cursor” to the window's right edge.
<pos x=“pixels”/>	(none)	Positions text horizontally.	Applies only if scrollrate and crawlrate are 0 (zero).
<pos y=“pixels”/>	(none)	Positions text vertically.	Applies only if scrollrate and crawlrate are 0 (zero).
<pre>...</pre>	(none)	Preserves tabs, spaces, and breaks.	Uses Courier face.
<required>...</required>	(none)	Ensures that text is delivered.	Presentation will halt if needed until text gets through.
<s>...</s>	(none)	~~Strikes through~~ the enclosed text.	Compatible with HTML.
<time begin=“dd:hh:mm:ss.xyz”/>	(none)	Sets time when text appears.	Only seconds (“ss”) are required.
<time end=“dd:hh:mm:ss.xyz”/>	(none)	Sets time when text disappears.	Only seconds (“ss”) are required.
<tl [color=“color”]>...</tl>	color	Displays text at lower edge in TickerTape.	Color affects only the following <tl> elements. See also “Colors” on page 24.
<tu [color=“color”]>...</tu>	color	Displays text at upper edge in TickerTape.	Color affects only the following <tu> elements. See also “Colors” on page 24.
<u>...</u>	(none)	Underlines the enclosed text.	Compatible with HTML.
<ul>...</ul>	(none)	Indents text.	For HTML compatibility. No bullets, however.

(Table Page 2 of 2)