

RealText Authoring Guide
RealSystem G2 BETA 1 Release



Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RealNetworks, Inc.

©RealNetworks, Inc.

RealAudio and RealVideo are registered trademarks of RealNetworks, Inc.

The Real logo, RealServer, RealPlayer, RealText, RealPix, RealPublisher and RealSystem are trademarks of RealNetworks, Inc.

ACELP-NET codec used under license from Université de Sherbrooke. Sipro Lab Télécom, Inc. Copyright ©1994-1997. All rights reserved.

Apple, Macintosh, and Power Macintosh are registered trademarks of Apple Computer, Inc.

Macromedia is a registered trademark and Flash and Shockwave are trademarks of Macromedia, Inc.

Microsoft, MS-DOS, Windows, and Windows NT are registered trademarks and ActiveX is a trademark of Microsoft Corporation.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

Pentium is a registered trademark and MMX is a trademark of Intel Corporation.

Other product and corporate names may be trademarks or registered trademarks of other companies. They are used for explanation only, with no intent to infringe.

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

<http://www.real.com>

Contents

Introduction	1
Tools for Creating RealText.....	1
Additional RealSystem Resources.....	2
Conventions Used in this Manual.....	2
RealText Authoring.....	3
Choosing Window Types	3
Using RealText Mark-Up.....	4
Creating a RealText Presentation.....	4
Broadcasting RealText Live	6
Notes on Text Streaming	7
<window> Tag Attributes	8
Text Tags	11
Time and Position Tags	11
Layout Tags	13
Appearance Tags.....	15
Command Tags	17
Colors	19
Coded Characters	19
Examples	21
Generic Window	21
TickerTape Window	22
Scrolling News Window.....	23
TelePrompter Window	24
Summary of <window> Tag Attributes.....	25
Summary of Text Tags	26

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

The HTML version of this guide, available at <http://www.real.com>, contains 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

RealServer streams your RealText presentation 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.

- Broadcast Application

To broadcast text live, you integrate an application that captures the live text feed, adds RealText mark-up to it, and sends it to RealServer. You need the RealSystem G2 Software Development Kit (SDK) to do this.

Additional RealSystem Resources

In addition to this manual, you may need the following RealNetworks resources, available at <http://www.real.com>:

- *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.

- *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 the RealSystem. You need the SDK and its documentation if you want to broadcast RealText live to create, for example, a real-time stock ticker. A knowledge of programming is required to use the SDK.

Conventions Used in this Manual

The following table explains the conventions used in this manual.

Notational Conventions	
Convention	Meaning
<i>variables</i>	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.

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 can then stream the text to RealPlayer or Web browsers.

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.

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:

- Although RealPlayer can read RealText tags in any case, RealNetworks recommends that you make tags lowercase. 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 `` tag has the end tag ``), closes with a forward slash. For example:
`
`

Although tags without a closing slash still work in RealText, you will not be able to edit the RealText file in an XML editor without a Document Type Definition (DTD).

- 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 19.
- As in HTML, you can add a comment to a RealText file like this:
`<!-- 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 8.

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 11.

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://www.real.com> 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://www.real.com> 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.

Broadcasting RealText Live

To broadcast RealText live, you do not create a RealText file. Instead, you build a broadcast application to add RealText mark-up to the live text. Your application then feeds the text and mark-up to the broadcast library on RealServer. The documentation for the RealSystem G2 Software Development Kit (SDK) explains how to create a broadcast application and integrate it with RealServer. You can get the SDK at <http://www.real.com>.

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://www.real.com>** 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:

...

<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. 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.

Note

When you are familiar with the attributes, you can refer to “Summary of <window> Tag Attributes” on page 25.

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 3.

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://www.real.com> 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 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 19 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.

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.

link=“color”

This attribute sets the color of hyperlinks within the text. The default is blue. See “Colors” on page 19 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 text that loops.

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. RealPlayer treats spaces, tabs, line feeds, and carriage returns as does a Web browser. It ignores these elements when they are between tags (except the <pre> and </pre> tags). When they occur between text elements, it interprets them as a single space, no matter how many of them are actually between the text elements.

Note

When you are familiar with the tags, you can refer to “Summary of Text Tags” on page 26.

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.

Note

The `<time/>` tags are not necessary in a window with a non-zero scrollrate or crawlrate unless you want text to become visible after it has moved into the window, or to disappear before it moves out of the window.

`<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 scrollrate and crawlrate 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 19 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.

**
**

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.

...

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

...

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

...

For compatibility with HTML lists. Acts like a
 tag.

<hr/>

For compatibility with HTML horizontal rules. Acts as two
 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
, <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 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 15.

Appearance Tags

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

...

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

<i>...</i>

These tags display the enclosed text *italicized*.

<u>...</u>

These tags display the enclosed text underlined.

...

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

```
<font size="+4" face="courier">
```

RealText supports the following attributes:

- size="n"

This 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 points for each size. The point sizes are for reference only. You cannot specify a point size directly in RealText.

Font Sizes

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

Note

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

- `color="color"`

This `` 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.

- `bgcolor="color"`

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

Additional Information

See "Colors" on page 19.

- `face="face"`

With this `` tag attribute you can control the text face. Available faces are the following:

- "Times" – This is the default face. It is equivalent to Times New Roman on Windows.
 - "Arial" or "Helvetica" – You can specify either of these faces to display text in Arial on Windows, Helvetica on UNIX, and Helvetica on the Macintosh.
 - "System" or "Geneva" – You can specify either of these faces to display text in System on Windows, Fixed on UNIX, and Geneva on the Macintosh.
 - "Courier" – Monospace font on all platforms.
 - "Osaka" – This is valid only when used with ``.
 - "ËÏÏâ" – This is valid only when used with ``. ("ËÏÏâ" may appear as gibberish if you are not viewing this HTML document with the gb2312 character set.)
 - "zÓ©úÅé" – This is valid only when used with ``. ("zÓ©úÅé" may appear as gibberish if you are not viewing this HTML document with the big5 character set.)
- `charset="character set"`

With this `` tag attribute you can control the character set used to display the text. Available character sets are the following:

- "us-ascii" – This is the default character set.

- "x-sjis"–You can use this character set only with ``.
- "gb2312" (Chinese)–You can use this character set only with ``. ("ËÏà" may appear as gibberish if you are not viewing this HTML document with the gb2312 character set.)
- "big5" (Traditional Chinese)–You can use this character set only with ``. ("²ÔúÅé" may appear as gibberish if you are not viewing this HTML document with the big5 character set.)

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 unreadably in its default character set.

Command Tags

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

`...`

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 `...` 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 section 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"`.

...

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.

...

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 `` tag. This includes Red/Green/Blue hexadecimal values (`#RRGGBB`), as well as these predefined color names, listed here with their corresponding hexadecimal values:

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

Transparency

You can also use "transparent" as a color. For example, `` 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
<code>&lt;</code>	<
<code>&gt;</code>	>
<code>&amp;</code>	&
<code>&nbsp;</code>	(nonbreaking space)
<code>&#32;</code> to <code>&#255;</code>	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: "".

is displayed in a RealText window as:

This is a bold tag: "".

Examples

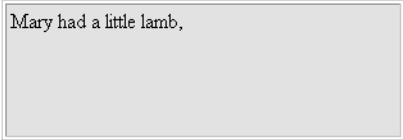
The following examples show how to create and tag the supported RealText window types.

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):

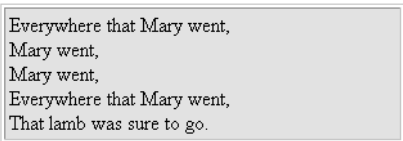
```
<window duration="30" bgcolor="yellow">
Mary had a little lamb,
<br/><time begin="3"/>little lamb,
<br/><time begin="6"/>little lamb.
<br/><time begin="9"/>Mary had a little lamb,
<br/><time begin="12"/>whose fleece was white as snow.
<br/><time begin="15"/><clear/>Everywhere that Mary went,
<br/><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
 tags force line breaks.
- <time/> tags need not appear after a
 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.



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 tag at the start **bolds** all following text.
- DJIA 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.

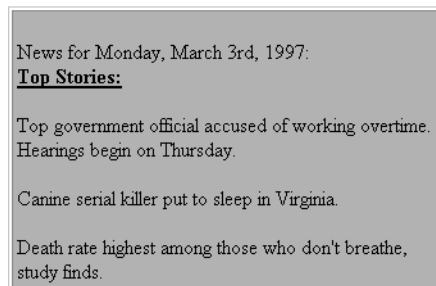
- The `
` tag that comes before the first text item forces the text that follows to start just past the window's right edge. Any break or paragraph tag inside TickerTape text causes the text that follows to start at the right edge. If the `
` tag were absent, the data would appear starting at the window's 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 the RealPlayer.



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/>`

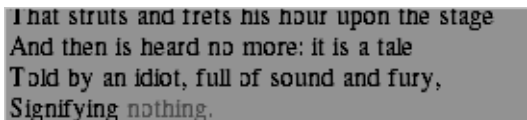
tag. Without the `<time/>` tag, the text would have appeared before one second and would have scrolled up from the bottom of the window.

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.

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 19.
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.
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 19.
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 <u>underlined</u> .	true	None.
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.

Summary of Text Tags

Tag	Attributes	Function	Notes
<code>...</code>	(none)	Creates hyperlink that executes a command.	Target required.
<code>...</code>	(none)	Opens e-mail message.	Browser must be configured for e-mail.
<code>...</code>	target= " <code>_player</code> "	Creates hyperlink to specified URL.	URL opens in browser if target not set to player.
<code>...</code>	(none)	Bolds the enclosed text.	Compatible with HTML.
<code>
</code>	(none)	Adds space between text.	In TickerTape and Marquee, moves "cursor" to the window's right edge.
<code><center>...</center></code>	(none)	Centers the enclosed text.	Compatible with HTML.
<code><clear/></code>	(none)	Clears all text from the window.	Following text is displayed at the window's normal starting point.
<code>...</code>	size	Sets relative font size.	Values are: -2, -1, +0, +1, +2, +3, +4; or 1, 2, 3, 4, 5, 6, 7
	color	Controls font color except for TickerTape window.	Supports all color values available in HTML. See also <code><tl></code> and <code><tu></code> .
	bgcolor	Sets the background color.	See "Colors" on page 19.
	charset	Controls the character set used to display text.	Default is "us-ascii".
	face	Sets the text face.	Default is "Times".
<code><hr/></code>	(none)	Acts like two <code>
</code> tags.	For HTML compatibility.
<code><i>...</i></code>	(none)	<i>Italicizes</i> the enclosed text.	Compatible with HTML.
<code>...</code>	(none)	Acts like a <code>
</code> tag.	For HTML compatibility. No numbers or bullets.
<code>...</code>	(none)	Indents text.	For HTML compatibility. No numbers, however.

(Table Page 1 of 2)

Tag	Attributes	Function	Notes
<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.
<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 19.
<tu [color=“color”]>...</tu>	color	Displays text at upper edge in TickerTape.	Color affects only the following <tu> elements. See also “Colors” on page 19.
<u>...</u>	(none)	<u>Underlines</u> the enclosed text	Compatible with HTML.
...	(none)	Indents text.	For HTML compatibility. No bullets, however.

(Table Page 2 of 2)