

# Contents

## **Text Layout Flash Component Overview**

About the Text Layout component .....	1
Installing the Text Layout component .....	1
Adding the Text Layout component to a document .....	3
Understanding fonts .....	5
Specifying read-write policy .....	8
Creating and entering text .....	9
Formatting text .....	10
Formatting paragraphs .....	14
Using tabs and indents .....	17
Advanced typography .....	18
Laying out text with containers .....	22
Inserting links .....	24
Combining text and graphics .....	25

# Text Layout Flash Component Overview

## About the Text Layout component

The Text Layout component is an Adobe® Flash® CS4 Professional component, using the Text Layout Framework and the new text engine in Adobe Flash Player 10, which delivers advanced, easy-to-integrate typographic and text layout features for rich, sophisticated, and innovative typography on the web. Together with the new text engine in Flash Player 10 and AIR 1.5, the Text Layout component delivers multi-lingual, print-quality typography for the web, including support for:

- Bidirectional text, vertical text and over 30 writing scripts including Arabic, Hebrew, Chinese, Japanese, Korean, Thai, Lao, Vietnamese, Indic, Dravidian and others.
- Selection, editing and flowing text across multiple columns and around inline images. Vertical text, Tate-Chu-Yoko (horizontal within vertical text) and justifier for East Asian typography.
- Rich typographical controls, including kerning, ligatures, typographic case, digit case, digit width and discretionary hyphens.
- Cut, copy, paste, undo and standard keyboard and mouse gestures for editing.
- Rich developer APIs to manipulate text content, layout, markup and customize the TextLayout component.
- The Text Layout Flash Component extension provides a user interface (the Text Layout panel) that you can use to visually layout text and access advanced typographic features.

For a complete list of features and additional information regarding this beta release, see the release notes at: [www.adobe.com/go/textlayout\\_releasenotes](http://www.adobe.com/go/textlayout_releasenotes)

## Installing the Text Layout component

### System requirements

System requirements for the Text layout component are the same as those for Adobe Flash CS4 Professional and Flash Player 10. Please refer to the release notes for further information on the Text Layout component's system requirements: [www.adobe.com/go/textlayout\\_releasenotes](http://www.adobe.com/go/textlayout_releasenotes)

### Pre-installation tasks

#### Read the release notes

Please read the Text Layout component release notes for additional information on any installation or operational issues. You can view the release notes at: [www.adobe.com/go/textlayout\\_releasenotes](http://www.adobe.com/go/textlayout_releasenotes)

#### Install Flash Player 10.0.12.36

To use the Text Layout component, you must install Flash Player 10.0.12.36. This update of Adobe Flash Player 10 replaces the Debug and Release versions of Flash Player 10 browser plugins and standalone players that were included in the initial release of Flash CS4 Professional (player version 10.0.2.54).

To install Flash Player 10.0.12.36, see: [www.adobe.com/go/b62ce659](http://www.adobe.com/go/b62ce659)

## Turn on ClearType

To use the Text layout component on Windows, you must enable ClearType®. ClearType is a font rendering technology in Windows XP and Windows Vista that smooths the edges of diagonal lines and makes text more readable.

### ClearType and Windows XP

- 1 Right click on the Windows XP desktop and select Properties.
- 2 Select the Appearance tab and click the Effects button
- 3 Select the checkbox “Use the following method to smooth edges of screen fonts,” then select ‘ClearType’ from the pop-up menu.
- 4 Click OK to close the Effects dialog, and click Apply to complete the process.

### ClearType and Windows Vista

- 1 Open Appearance Settings dialog box by clicking the Start button, clicking Control Panel, clicking Appearance and Personalization, clicking Personalization, and then clicking Window Color and Appearance. If the Appearance Settings dialog box is not displayed, at the bottom of the page, click Open classic appearance properties.
- 2 Click Effects.
- 3 In the Effects dialog box, select Use the Following Method to Smooth Edges of Screen Fonts.
- 4 Click ClearType in the list, and then click OK.

*Note:* Whether you select Standard or ClearType from the list, you must have a video card and monitor that support a color setting of at least 256 colors. You’ll get the best results with High color (24-bit) or highest color (32-bit) support. You can change color quality in Display Settings in Control Panel.

## Install OpenType fonts

The Text Layout component includes a set of OpenType fonts that allow you to test advanced typographic features. To install the included OpenType fonts see the font installation instructions at:

[www.adobe.com/type/browser/fontinstall/instructions\\_english.html](http://www.adobe.com/type/browser/fontinstall/instructions_english.html).

## Install the Text Layout component

You can use Extension Manager CS4 to install the Text Layout component in Adobe Flash CS4 Professional. You cannot use it to install the Text Layout component extension in earlier versions of Flash applications.

*Note:* When installing on Windows Vista you must have administrative privileges to install the Text Layout component.

- 1 Perform the pre-installation tasks. For more information, see “Pre-installation tasks” on page 1.
- 2 If you haven’t already done so, download the Text Layout Flash Component ZIP file. Save the ZIP file on your computer’s hard disk and take note of its location.
- 3 Uncompress the ZIP file.
- 4 Double-click the Text Layout Flash Component.mxp extension file in Windows® Explorer (Windows) or in the Finder (Macintosh®)
- 5 Read the Adobe extensions disclaimer and any third-party extension license.  
Select Accept to continue with the installation, or select Decline to cancel the installation.
- 6 If you have installed another version of the extension or have another extension with the same name, the Extension Manager asks if you want to disable or overwrite it.

Select Yes to replace the previously installed extension with the new extension, or No to cancel the installation and leave the existing extension in place.

- 7 When asked if you want to replace one or more existing files, select Yes or Yes To All to accept the versions included in the extension. Select No to keep the current version of the file.

If you click Cancel, the installation stops and no files are installed or replaced.

If the Extension Manager doesn't encounter any problems, it informs you that the extension has been successfully installed.

## Adding the Text Layout component to a document

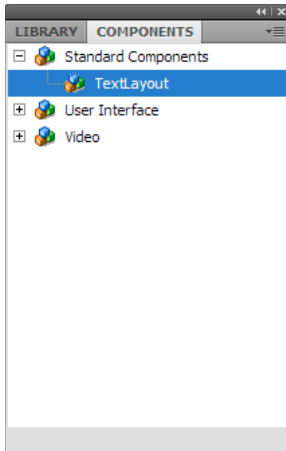
When you drag a FLA-based component from the Components panel to the Stage, Flash imports an editable movie clip to the library. After a component has been imported to the library, you can drag instances of it to the Stage from either the Library panel or the Components panel.

### Add the Text Layout component during authoring

You can add a component to a document by dragging it from the Components panel. You can set properties for each instance of a component in the Properties inspector.

- 1 Select Window > Components.
- 2 Either double-click the Text Layout component in the Components panel or drag the component to the Stage. Placing the Text Layout component on the Stage also adds it to the library (Window > Library).

**Note:** If you don't want an instance of the Text Layout component on the Stage, you can drag it directly to the library.



*TextLayout component in the Components panel*

- 3 If the Properties Inspector is not visible, select Window > Properties.
- 4 In the Properties Inspector, enter an instance name for the Text Layout component instance. For example:  
`myTextLayout`

If you will be using ActionScript 3 code to interact with the component, you can refer to the component by using its instance name.

- 5 Change the position and size of the component as desired by editing the values for the x-axis (X:) and y-axis (Y:) and the width (W:) and height (H:).
- 6 Select Window > Other Panels > Text Layout to display the Text Layout panel.

The Text Layout panel lets you add and format content with the Text Layout component.

***Note:** Once you have initiated the Text Layout panel using the Window > Other Panels > Text Layout menu command, you can click the Edit TextLayout button located in the Component Inspector (Window > Component Inspector) to display the Text Layout panel.*

- 7 Click Send to Stage. This updates the currently active Text Layout component on the Stage with any formatted text you've entered into the Text Layout panel text window. For more information, see "[Updating and saving text from a Flash document](#)" on page 9.
- 8 Save the Flash document containing the Text Layout component.
- 9 Select Control > Test Movie or press Control+Enter to compile the document and see the results of your work in a SWF file.

## Dynamically create a Text Layout component

You can dynamically create an instance of the Text Layout component by adding it to the Flash library, and then using ActionScript® to instantiate the Text Layout component class to create an instance of the Text Layout component.

- 1 Select Window > Components.
- 2 Either double-click the Text Layout component in the Components panel or drag the component to the Stage.

Placing the component on the Stage also adds the component to the library (Window > Library). Once a component is stored in the library, it can be removed from the Stage, and accessed using ActionScript.

- 3 Select the component on the Stage, and delete it.
- 4 Open the Actions panel (Windows > Actions), and enter the following code:


```
import flash.display.DisplayObject;
import fl.controls.TextLayout;
/* textLayout_flash is the base class for the Text Layout component. If you bring up the
properties on the TextLayout Component in the library, you will see this.*/
var myTextComponent:TextLayout;
// Instantiate the textLayout_flash class to create a TextLayout Component.
myTextComponent = new TextLayout();
// Add the new component to the Stage.
var myTextComponentOnStage:DisplayObject = addChild(myTextComponent);
// Get the Text Flow from the TextLayout Component.
var myTextFlow:Object = myTextComponent.textFlow;
// Change the color of the text to red.
myTextFlow.color = 0xFF0000;
```

## Delete the Text Layout component from a document

To delete a Text Layout component instance from the Stage while authoring, simply select it and press the Delete key. This will remove the instance from the Stage but does not remove the component from your application.

To delete a component from your Flash document after you've placed it on the Stage or in the library, you must delete it and its associated assets from the library. It isn't enough to delete the component from the Stage. If you don't remove it from the library, it will be included in your application when you compile it.

- 1 In the Library panel (Window > Library), select the symbol for the component.

- 2 Click the Delete button  at the bottom of the Library panel, or select Delete from the Library panel menu.  
Repeat these steps to delete any assets associated with the component.

## Understanding fonts



A *font* is a complete set of characters—letters, numbers, and symbols—that share a common weight, width, and style, such as Adobe Garamond Bold Italic.

*Typefaces* (often called *type families* or *font families*) are collections of fonts that share an overall appearance, and are designed to be used together, such as Adobe Garamond.

A *type style* is an individual font in a font family. Typically, the *Regular* (also referred to as *Roman* or *Plain*—the actual name varies from family to family) member of a font family is the base font, which may include other type styles such as Bold, Semibold, Italic, and Bold Italic.

## OpenType fonts

OpenType fonts use a single font file for both Windows® and Macintosh® computers, so you can move files from one platform to another without worrying about font substitution and other problems that cause text to reflow. They may include a number of features, such as swashes and discretionary ligatures, that aren't available in PostScript and TrueType fonts.

 *OpenType fonts display the  icon.*

When working with an OpenType font, if it has the necessary features, you can automatically substitute alternate glyphs, such as ligatures, small capitals, fractions, and old style proportional figures, in your text.

OpenType fonts may include an expanded character set and layout features to provide richer linguistic support and advanced typographic control. OpenType fonts from Adobe that include support for central European (CE) languages include the word “Pro,” as part of the font name in application font menus. OpenType fonts that don't contain central European language support are labeled “Standard,” and have an “Std” suffix. All OpenType fonts can also be installed and used alongside PostScript Type 1 and TrueType fonts.

In order to use advanced typographical features and provide international support for script-based languages such as Arabic and Hebrew, the Text Layout component requires fonts that support OpenType Layout tables. On Windows computers, you can use the Microsoft Font properties extension to see if fonts contain the OpenType Layout tables. For information on the Microsoft Font properties extension, see:  
[www.microsoft.com/typography/TrueTypeProperty21.mspx](http://www.microsoft.com/typography/TrueTypeProperty21.mspx)

**Note:** Both OpenType (.otf) and TrueType (.ttf) font files can contain the OpenType Tables required for advanced typographic features and international language support.

For more information on OpenType fonts, see: [www.adobe.com/go/opentype](http://www.adobe.com/go/opentype)

## Ligatures

*Ligatures* are typographic alternatives that combine two or more letter combinations into a single glyph. Many fonts include ligatures for standard letter pairs such as fi, fl, ff, ffi, and ffl. In addition, some fonts include discretionary ligatures for letter pairs such as ct, st, and ft. Although the letters in ligatures appear to be joined, they are fully editable.

## Embedded and device fonts

Flash Player 10 supports a new embedded font format called DefineFont4 that is required by the Text Layout component for high-end typography and international support. The Text Layout component lets you embed fonts using the new DefineFont4 tag. You will need to follow the instructions below to embed fonts when using the Text Layout component.


When you embed fonts, Flash stores all of the font information in the SWF file so the font is displayed properly even if it's not installed on the user's computer. If you use a font in your FLA file that isn't installed on a user's system, and you don't embed the font in the SWF file, Flash Player automatically selects a substitute font to use instead.

### Embed DefineFont4 fonts with the Text Layout component

Embedding a font makes its complete set of characters—letters, numbers, and symbols—available to Flash Player.

- 1 Create a new ActionScript 3.0 document.
- 2 Either double-click the Text Layout component in the Components panel or drag the component to the Stage.
- 3 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 4 Click the Embed Flash Fonts button in the lower right of the Text Layout panel to display the Embed Flash Fonts dialog box.
- 5 In the Embed Flash Fonts dialog box, select the font you want to embed from the Available Fonts pop-up menu, and click the Plus (+) button to add the font to the list of embedded fonts.

You can embed as many fonts as you need.

- 6 (Optional) To remove a font, select it in the list of Embedded Fonts, and click the Trash Can  button.
- 7 Click close when you have embedded the necessary font(s) for your Flash document.

### Device fonts

You can use generic device fonts as an alternative to embedding fonts. Flash Player supports six generic device fonts, `_sans` (similar to Helvetica™ or Arial™), `_serif` (similar to Times New Roman™), and `_typewriter` (similar to Courier). The remaining three device fonts are for use with Japanese: `_gothic`, `_gothicMono`, and `_mincho` (all three of these device fonts names are displayed in Japanese). When you specify one of these fonts and then export the document, Flash Player uses the font on the user's computer that most closely resembles the generic device font.

Because device fonts are not embedded, they result in a smaller SWF file. They can also result in more legible text at small point sizes (below 10 points). However, if a user's computer does not have an installed font corresponding to the device font, text may look different than expected.


## Dynamically load fonts from SWF files

A key feature of ActionScript 3.0 SWF files is that you can dynamically load the SWF files as you need them. This allows you to create a SWF containing fonts, and dynamically load the SWF to provide embedded fonts to the Text Layout component.

- 1 In Flash CS4 Professional, create a new ActionScript 3.0 document.
- 2 Double-click the Text Layout component in the Components panel to add it to the Stage.
- 3 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 4 Click the Embed Flash Fonts button in the lower right of the Text Layout panel to display the Embed Flash Fonts dialog box.

- 5 In the Embed Flash Fonts dialog box, select the font you want to embed from the Available Fonts pop-up menu, and click the Plus (+) button to add the font to the list of embedded fonts.

You can embed as many fonts as you need.

- 6 (Optional) To remove a font, select it in the list of Embedded Fonts, and click the Trash Can  button.
- 7 Click close when you have embedded the necessary fonts for your Flash document.
- 8 Select Window > Actions to open the Actions Panel.
- 9 Add code to register the fonts you embedded when the FontSWF is loaded.

```
import flash.text.Font;
/*You will use the name of the font that was embedded and DF4. For example, in the example
below Arial Black was embedded. Notice that underscores are used. These are added to the
font name when it is embedded to create the font class name. If you embedded something like
Times New Roman, you would use Times_New_Roman_DF4.*/
Font.registerFont(Arial_Black_DF4);
```

- 10 Save the Flash document (FLA) in the same folder in which you will create the SWF using the Text Layout component, and give it a specific name. For example: EmbeddedFonts.flas
- 11 Delete the Text layout component from the library (Window > Library) prior to publishing your SWF file. This removes the component from the SWF, leaving only the embedded fonts, and resulting in a smaller SWF file.
- 12 Select File > Publish to publish the SWF file. It will be created in the folder in which you saved the FLA in Step 10.
- 13 Place the following ActionScript code in the Flash document using the Text Layout component:

```
import flash.events.IOErrorEvent;
var fontUrl:URLRequest;
loadEmbeddedFonts();
function loadEmbeddedFonts():void
{
    var loader:Loader = new Loader();
    loader.contentLoaderInfo.addEventListener(Event.INIT, fontLoaded);
    loader.contentLoaderInfo.addEventListener(IOErrorEvent.IO_ERROR, showError);
    fontUrl = new URLRequest("EmbeddedFonts.swf");
    loader.load(fontUrl);
}
// called on INIT event when embedded fonts SWF is initialized.
function fontLoaded(e:Event):void
{
    var fonts:Array;
    fonts = Font.enumerateFonts(false); // false -> embedded only.
    for each (var font:Font in fonts)
        trace(font.fontName);
}
function showError(e:Event):void
{
    trace("Error opening: " + fontUrl.url);
}
```

The above code will load embedded fonts from the first SWF you created—labeled EmbeddedFonts.swf—into the file using the Text Layout component.

To confirm the font is properly embedded, and being dynamically loaded from the first SWF file into the second, select a font that does not exist on another computer and run the SWF on that computer. If the fonts look correct the embedding is working correctly. Embedded fonts will appear crisp and clear, while device fonts are usually more ragged looking, with less clear definition.

## Anti-aliasing text

Anti-aliasing makes the edges of text look smoother on screen. The anti-aliasing options are particularly effective for rendering smaller font sizes. When anti-aliasing is enabled, all text in the current selection is affected. Anti-aliasing operates with text of all point sizes in the same way. The Flash text rendering engine provides clear, high-quality text rendering in Flash (FLA) documents and published SWF files.

*Note: When using device fonts, anti-aliasing may be performed by the operating system. For example, ClearType on Microsoft Windows.*

The Text Layout component provides two anti-aliasing options:

**CFF (Compact Font Format)** This is a new feature of Flash Player 10. This type of font rendering makes text more legible, and permits higher-quality display of fonts at small sizes. This setting only applies to embedded fonts, and can only be viewed in a SWF file.

If you choose CFF anti-aliasing, you can optionally choose to use font hinting. Font hinting adjusts the character outlines so that they rasterize better on the pixel grid. At low resolution and small point sizes, hinting helps produce text that is more clear and legible to read.

**Normal** This is the type of anti-aliasing used by Flash Player 7 and earlier versions.

To specify the type of anti-aliasing for your Flash document:

- 1 Select the text that you want to apply anti-aliasing to; anti-aliasing applies to the currently selected text.
- 2 In the Antialias panel, click the Antialias pop-up menu and choose CFF or Normal.
- 3 (Optional) If you choose CFF anti-aliasing, specify whether or not the Flash document should use font hinting. The font hinting options are:

**Horizontal Stem** Fits strong horizontal stems to the pixel grid for improved readability.

**None** No hinting is applied. Horizontal stems in the glyphs are not forced to the pixel grid. This is the recommended setting for animation or for large font sizes.

## Specifying read-write policy

You can specify the read-write policy for text you create using the Text Layout component. Read-write policy lets you specify whether the text in the published SWF file is read-only, selectable, or editable.

- 1 Open a Flash ActionScript 3.0 document containing a Text Layout component instance, or create a new document with a Text Layout component.
- 2 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 3 In the Data Source tab, click the Read Policy pop-up menu, and select one of the following options:

**ReadSelect** specifies that text can be selected and copied using the operating system's copy command.

**ReadWrite** Specifies that text can be selected and edited.

**ReadOnly** Specifies that text is only readable in the published SWF file. You cannot select or edit the text.

## Creating and entering text

All text in the Text Layout component resides inside containers (Containers are similar to text frames in InDesign.). Containers can be moved, resized, and their properties changed.

To add text to the Text Layout component:

- 1 In the Data Source tab, click Get from Stage to load any text already in the component on Stage into the Text Layout panel's text window.

*Note:* This step serves as a precaution to prevent losing your work.

- 2 Type text directly into the Text Layout panel text window, or copy-and-paste text from other documents using the operating system's copy-and-paste commands.
- 3 Format the text using the controls located in the Character, Paragraph, and other Text Layout component panels.
- 4 Click Send to Stage. This updates the currently active Text Layout component on the Stage with any formatted text you've entered into the Text Layout panel text window.

## Updating and saving text from a Flash document

Once you create and format text using the Text Layout panel, you must populate the current Flash document on Stage with your text layout. You can also open an existing Flash document containing an existing text layout, populate the Text Layout panel with that text, and continue editing and formatting the text.

*Note:* When you save or retrieve text from the Text Layout component to the Stage, you are transferring the TextLayout XML to the `xmlText` property of the currently active Text Layout component on the Stage. To see the changes you must click on the Stage.

### Saving text to a Flash document

- 1 Enter and format text directly in the Text Layout panel.
- 2 Click Send to Stage. This updates the currently active Text Layout component on the Stage with any formatted text you've entered into the Text Layout panel text window.
- 3 Save the Flash document containing the Text Layout component.

### Updating text from an existing Flash document

- 1 Open an existing Flash ActionScript 3.0 document containing a Text Layout component instance.
- 2 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 3 Click Get to Stage to populate the Text Layout panel with the existing text layout.

## Saving and importing TextLayout XML

When you save text to, or retrieve text from the Text Layout component to the Stage, you are transferring TextLayout XML to the `xmlText` property of the currently active Text Layout component on the Stage. You can save or import TextLayout XML for any ActionScript 3.0 document.

You can save the TextLayout XML both to archive your text layout, as well as to reuse text layout formats between Flash documents.

### Saving TextLayout XML

- 1 Open an ActionScript 3.0 document containing a Text Layout component instance.

- 2 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 3 Click Get to Stage to populate the Text Layout panel with any existing text layout.
- 4 Click the Markup button located in the Data Source tab. This will display the Markup panel, which contains the TextLayout XML for the active Flash document and Text Layout component.
- 5 From the Markup panel's text window, copy the TextLayout XML and paste it into a text editor. Save the file using UTF-8 encoding with an .xml extension. For example: TextLayout.xml

### Importing TextLayout XML

- 1 Open a Flash ActionScript 3.0 document containing a Text Layout component instance, or create a new document with a Text Layout component.
- 2 Select Window > Other Panels > Text Layout to display the Text Layout panel.
- 3 Click the Markup button located in the Data Source tab. This will display the Markup panel, which contains the TextLayout XML for the active Flash document and Text Layout component.
- 4 Open the XML file containing the TextLayout XML you want to import into the current Flash document. Copy the TextLayout XML into the Markup panel's text window, overwriting any existing TextLayout XML, and click Import.

The Text Layout defined by the XML will populate the Text Layout panel. You can now further modify the content if needed, and publish it to the Stage.

## Formatting text

You format characters using the Character panel, which located in the upper, right-hand side of the Text Layout panel.

*Note: Fonts in the Text Layout component do not necessarily match the fonts displayed in the Flash CS4 Professional font menu. The Flash font menu displays formatted, PostScript font names, while the Text Layout component displays the font platform names used to code in ActionScript.*

### Specify a font family and style

- 1 Select the text you want to change. If you don't select any text, the font applies to new text you create.
- 2 In the Character panel, choose a font family from the Font pop-up menu.
- 3 Choose either Embedded CFF or Device from the Lookup pop-up menu.

Flash Player looks for device or embedded fonts with which to render text. If a device font is specified, Flash Player looks for the font on the playback system. Specifying an embedded font causes Flash Player to look for an embedded font with the specified name in the SWF file. Only embedded CFF (Compact Font Format) fonts work with this setting. If the specified font is not found, a fallback device font is used. For more information, see [“Embedded and device fonts”](#) on page 6.

- 4 To apply a font style such as bold, italic, or underline, select the corresponding font style button.

### Specify a font size

Font size in the Text Layout component is measured in pixels. You can specify any typeface size from 1 to 720 pixels in 1-pixel increments.

- 1 Select the text you want to change. If you don't select any text, the font size applies to new text you create.

- 2 In the Character panel set the Size option.

## Apply bold, italic, underline, or strike through formatting to text

- 1 Select the text whose format you want to change.
- 2 In the Character panel, click the appropriate formatting button:

**Bold** **T** Applies bold formatting to the selected text.

**Italic** *T* Applies italic formatting to the selected text.

**Underline** T Applies underline formatting to the selected text.

**Strike-through** ~~T~~ Applies strike-through formatting to the selected text.

## Create superscript or subscript text

Superscript and subscript refer to numbers that are positioned slightly higher or slightly lower than the text on the main line. For example, a footnote or endnote reference number is an example of superscript text, and a scientific formula might use subscript text.

- 1 Select the text you want to change to superscript or subscript.
- 2 In the Character panel, click the appropriate superscript <sup>T</sup> or subscript <sub>T</sub> formatting button.


When you choose Superscript or Subscript, a predefined baseline shift value and type size are applied to the selected text. The values applied are percentages of the current font size and leading value.

## Use tate-chu-yoko rotation

Typically you use text rotation in combination with Asian text where characters must be rotated to display properly in vertical layout. Referred to as *tate-chu-yoko* (also called *kumimoji* and *renmoji*) is a block of horizontal type laid out within vertical type lines. Using tate-chu-yoko makes it easier to read half-width characters such as numbers, dates, and short foreign words in vertical text.



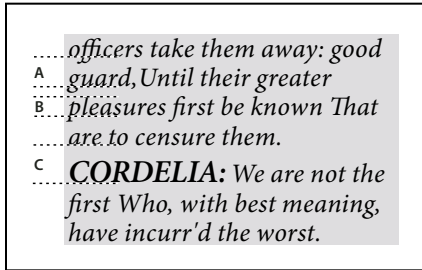
*Tate-chu-yoko text rotation*

- 1 Select the text you want to rotate.
- 2 Click the *tate-chu-yoko* button  located in the Character panel to rotate the text.

## Specify the leading value

The vertical space between lines of type is called *leading*. Leading (also known as line spacing or line height) is measured between the baselines of consecutive lines of text. The default leading value is 120% of the font size (for example, 12-pixels of leading for 10-pixel type). The leading can be specified as an absolute value in pixels or a percentage of the font size.

Leading is a character attribute, which means that you can apply more than one leading value within the same paragraph. The highest leading value of a character sets the leading for that line.



Leading using Roman text

A. Leading B. Text height C. Larger text size of one word increases leading for that one line.

Leading is also affected by Leading Basis and Leading Direction settings located in the Paragraph panel. By default, Leading Basis and Leading Direction are set to AUTO, which changes the behavior based on the justification rule—Roman or East Asian—applied to the paragraph. When using Roman justification the leading direction is up (e.g. leading is applied above the line); when using East Asian justification the leading direction is down (e.g. leading is applied below the line).

To learn more about Leading Basis and Leading Direction, see [“Specify leading direction for Asian languages”](#) on page 15 and [“Understanding leading basis”](#) on page 15.

- 1 Select the text you want to change. If you don’t select any text, the leading value applies to new text you create.
- 2 In the Character panel, enter a value in the Leading text field.

## Shift the baseline

Use Baseline Shift to move selected characters up or down relative to the baseline of the surrounding text. Shifting the baseline is especially useful when you’re hand-setting fractions or adjusting the position of a *picture* font such as a dingbat or ornament.

- 1 Select the characters or type objects you want to change. If you don’t select any text, the shift applies to new text you create.
- 2 In the Advanced Character panel, set the Baseline Shift option. Positive values move the character’s baseline below the baseline of the rest of the line; negative values move it above the baseline.

## Kern and track

*Kerning* is the process of adding or subtracting space between specific pairs of characters. *Tracking* is the process of loosening or tightening the spacing between the characters in selected text or an entire block of text.

The Text Layout Component automatically kerns characters using the kerning information built into the font (which is present in most fonts). Kerning contains information about the spacing of specific pairs of letters. Some common kern pairs are: Lt, P, To, Try, Ta, Tu, Te, Ty, Wa, WA, We, Wo, Ya, and Yo. Kerning is on by default so that specific combinations are automatically kerned when you import or type text.

## Specify kerning options

- 1 Select the text that you want to adjust.
- 2 In the Character panel, set the Kerning option:
  - On** To use a font’s built-in kerning information select On.

**Off** To turn off kerning, set the Kerning option to Off.

**Auto** To use a font’s built-in kerning information select Auto. Selecting Auto specifies that the Text Layout Component uses the kerning information built into the font for Latin (or Roman) characters, and that it will apply kerning to Asian characters that have kerning information built into them. Asian characters which do not use kerning are Kanji, Hiragana, and Katakana.

## Adjust tracking

- 1 Select the range of characters that you want to adjust.
- 2 In the Character panel, set the Tracking option for left or right tracking depending on the direction of the text you are adjusting. For left-to-right writing systems, specify a value Track Left; for right-to-left writing systems, specify a value Track Right.

## Change capitalization styles

Capitalization style (also called typographic case) lets you specify typographic glyphs for specific fonts.

- 1 Select the characters whose capitalization style you want to change.
- 2 Choose one of the following in the Case pop-up menu:

**Title** Specifies that uppercase characters use title glyphs on output. Use this setting to capitalize the first letter of each word in a heading or section title.

**Caps** Specifies that spacing is adjusted for uppercase characters. Apply this to uppercase letters to use the additional tracking (also referred to as “all caps spacing adjustment” that is built into certain fonts).

**Small Caps** Specifies that uppercase characters use small caps glyphs on output. Small caps are uppercase (capital) letters that are about the size of normal lowercase letters in any given typeface. Small caps are less intrusive when all uppercase appear within normal text, or can be used for special emphasis.

When you format text as small caps, the Text Layout component automatically uses the small-cap characters designed as part of the font, if available. Otherwise, the Text Layout component synthesizes the small caps using scaled-down versions of the regular capital letters.

500 BC to AD 700

500 BC to AD 700

*Before (top) and after (bottom) setting BC and AD in small caps to complement old-style numerals and surrounding text*

**Upper** Changes all characters to uppercase. Uppercase (also referred to as *all capitals* or *all caps*) is usually used for emphasis. It is commonly seen in the titles of book covers, chapter or section titles, and other areas where emphasis is required.

**Lower** Changes all characters to lowercase.

**Caps & Small Caps** Specifies a mix of caps and small caps where lowercase characters use small caps glyphs. Small caps are uppercase (capital) letters that are about the size of normal lowercase letters in any given typeface.

# TEXT LAYOUT





*Caps & Small Caps capitalizes the first letter of all words, and uses small capitals for the remaining letters.*

## Formatting paragraphs

### Align or justify text

Text can be aligned with one or both edges of a container. Text is said to be justified when it is aligned with both edges. You can choose to justify all text in a paragraph excluding the last line (Justify Left or Justify Right), or you can justify text in a paragraph including the last line (Justify All).

The Text Layout component also provides controls to specify Start and End Alignment. Choosing Align Start aligns the text to the left for paragraphs that have their alignment set to left-to-right, and right if paragraph alignment is set right-to-left. Choosing left alignment always keeps text left aligned even if the paragraph direction is set right-to-left. These options let you set the line to a reasonable default independent of paragraph direction.

- 1 Insert the cursor in the paragraph you want to change, or select all text in the Text window to change all of the paragraphs.
- 2 In the Paragraph panel, click one of the Alignment buttons—Align Left , Align Center , Align Right , or Justify .
- 3 (Optional) Specify Start **S** or End Alignment **E**. Choosing Align Start aligns the text to the left for paragraphs that have their direction set left-to-right, and right if paragraph direction is set right-to-left.
- 4 (Optional) You can choose to justify all text in a paragraph excluding the last line (Justify Left or Justify Right), or you can justify text in a paragraph including the last line (Justify All).

In narrow columns, a single word can occasionally appear by itself on a line. If the paragraph is set to full justification, a single word on a line may appear to be too stretched out. Instead of leaving such words fully justified, you can center them or align them to the left or right margins.

### Adjust indentation and margins

You can adjust both the indentation of the first line in a paragraph, and the left and right margins of any paragraph.

- 1 Insert the cursor in the paragraph you want to change, or select all text in the Text window to change all of the paragraphs. If you don't insert the cursor in a paragraph, the setting applies to new text you create.
- 2 In the Paragraph panel, specify a value in pixels for Indentation, Left Margin, and Right Margin.

### Adjust paragraph spacing

- 1 Insert the cursor in the paragraph you want to change, or select all of the paragraphs you want to adjust. If you don't insert the cursor in a paragraph or select, the setting applies to new text you create.
- 2 In the Paragraph panel, specify values in pixels for Space Before and Space After. You can specify from 1 to 1,000 pixels in 1-pixel increments.

Unlike traditional page layout applications, the vertical space specified between paragraphs collapses when values overlap. For example, if you have two paragraphs, Para1 followed by Para2, and Para1 has 12 pixels of space after it (Space After), and Para2 has 24 pixels of space before it (Space Before), the Text Layout component produces 24 points between the paragraphs, not 36.

*Note: If a paragraph begins at the top of a column, no extra space is added before the paragraph. In this case, you can use the first baseline offset options of the paragraph.*

## Specify line breaks for Asian text

You can specify paragraph level justification rules for Japanese kinsoku characters. Kinsoku Shori are rules for how to wrap Japanese text. Certain characters in Japanese should not come at either the beginning or end of a sentence, and some characters should never be split across two lines. For example, periods and closing parentheses are not permitted at the start of a line.

- ❖ In the Paragraph panel, choose East Asian from the Just Rule pop-up menu.

Choosing East Asian uses what is commonly referred to as “push in” justification, which moves characters up to the previous line to prevent prohibited characters from ending or beginning a line.

## Specify leading direction for Asian languages

Leading direction determines how line height is measured. For Japanese character sets, line height is the distance of a line's baseline from the previous line's baseline. For Roman and Middle Eastern languages, line height is the distance of a line's baseline from the next line's baseline.

When specifying settings for this option, consider the justification rule you choose in [“Specify line breaks for Asian text”](#) on page 15. Typically you can leave this set to the default setting of Auto.

- ❖ In the Paragraph panel, choose the appropriate leading direction setting for your language from the Leading Direction pop-up menu:

**Down** Use down for Japanese and other Asian languages using the East Asian justification rule.

When using the East Asian justification rule, various attributes and settings are automatically applied to the text. The text uses *mojikumi* (text composition) justification rules rather than spacing justification rules, and it uses Japanese style leading, which applies bottom-down leading versus top-up leading used in Roman text.

**Up** Use Up for Roman and Middle Eastern languages using the Roman justification rule.

**Auto** Auto—the default setting—is interpreted as Up when the Just Rule pop-up menu is set to Roman, and Down when paragraph Just Rule is set to East Asian.

## Understanding leading basis

Leading basis is a paragraph property that determines the baselines of consecutive lines separated by the distance of each other's line height. The way in which leading basis is measured differs between Roman and East Asian character sets.

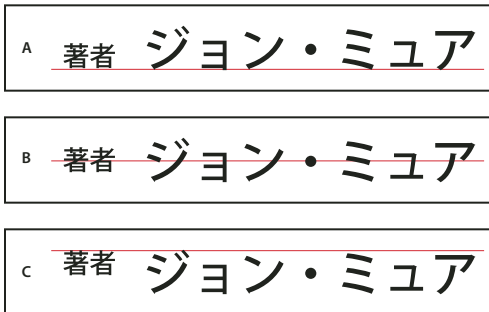
### Roman characters and leading basis

When setting leading basis for Roman text, if there are two consecutive lines of text in a paragraph the line height refers to the distance between their Roman baselines.

### Asian characters and leading basis

When working with Asian characters, leading basis determines the alignment of the characters (referred to as *mojisoro*). When a line of Asian text contains different sizes of characters, you can specify how to align text to the largest characters in the line: to the top, center, or bottom of the embox (right, center, and left for vertical frames), to the Roman baseline.

**Note:** An ideogram is a picture or symbol used in a system of writing to represent a thing or an idea. The characters used in Japanese and other Asian languages are ideograms, which is where the term “ideographic” used to describe the leading basis options is derived.



Asian character alignment options

A. Ideographic bottom B. Ideographic center C. Ideographic top

### Specify leading basis for Roman or Asian characters

❖ In the Paragraph panel, choose the appropriate leading basis for your language from the Leading Basis pop-up menu:

**Auto (default setting)** Interprets leading basis as Roman when the Just Rule (paragraph justification) option is set to Roman, and as Ideographic-Top when paragraph justification is set to East Asian.

**Roman** Aligns the small characters in a line to the large character.

**Ascent** Specifies an ascent baseline.

**Descent** Specifies a descent baseline.

**Ideographic Top** Aligns the small characters in a line to the specified position of the large character’s embox.

**Ideographic Center** Aligns the small characters in a line to the specified position of the large character’s embox.

**Ideographic Bottom** Aligns the small characters in a line to the specified position of the large character’s embox.

### Specify text direction for Arabic and Hebrew

To enter Arabic or Hebrew text, you must change the text direction from “Left to Right” to “Right to Left.”

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

يولد جميع الناس أحرارًا متساوين في الكرامة والحقوق. وقد وهبوا عقلاً وضميرًا وعليهم أن يعامل بعضهم بعضًا بروح الإخاء.

*Roman text with left-to-right direction (top) and Arabic text with right-to-left direction (bottom)*

It is important to understand and differentiate between the paragraph direction from paragraph alignment. Paragraph direction controls the order, inside the paragraph, of multidirectional text. Alignment controls the placement of the whole line in reference to the left and right boundaries of the text frame. Paragraph direction and alignment are independent. If your text is mainly Arabic or Hebrew, use a “Right to Left” paragraph, otherwise, use the “Left to Right” option for Roman text. For additional information on text direction within containers, see “Text flow” on page 24.

❖ In the Paragraph panel, choose “Right to Left” from the Direction pop-up menu.

## Line breaks

You can prevent words from breaking at the end of lines—for example, proper names or words that could be misread when hyphenated. You can also keep multiple words or groups of words together—for example, clusters of initials and a last name.

- 1 Select the characters you want to prevent from breaking.
- 2 Choose No Break from the Break option in the Advanced Character panel.

*Note: If you apply the No Break option to too many adjacent characters, the text may wrap in the middle of a word. However, if you apply the No Break option to more than a single line of text, none of the text will appear.*

## Using tabs and indents

Use tabs when you want columns of words or numbers to be aligned. Tab settings allow you to create aligned text very rapidly.

### Set tab stops

To create tabs in the Text Layout component you create a tab stop by clicking in the ruler and adjusting the tab type in the tabs panel. The Tab panel lets you specify the type of tab to use. If there is no selected tab, the Tab Type pop-up is used to set the default tab type for subsequently created tabs.

**Start** Sets the start position of text that will then run to the right as you enter text.

**Center** Centers a line of text at the point at which you've set the tab stop.

**End** Sets the right end of the text. As you enter text it moves to the left.

**Align** Lets you specify a character—such as a decimal point—to be used for alignment.

## Remove tab stops

- ❖ To remove a tab stop, drag the tab stop icon off the tab ruler. Note that this doesn't remove any tab characters you've typed in your text, though it does make them behave differently (because you've taken away their tab stop).

## Set indents

Indentation determines the distance of the paragraph from either the left or the right margin. Within the margins, you can increase or decrease the indentation of a paragraph or group of paragraphs.

Use the indent markers ▶ in the Ruler to adjust the indentation of a paragraph. The top marker controls the indentation of the first line of text in a paragraph; the bottom marker controls the indentation for the remaining lines of text in the paragraph.

- ❖ Drag the top marker to indent the first line of text. Drag the bottom marker to move both markers and indent the entire paragraph.

# Advanced typography

The following topics address typographic options found in the Advanced Character panel.

## Digit case and width

When working with OpenType fonts that offer both lining and oldstyle numerical figures, you can specify which numeral style to apply and whether to use proportional or tabular numerals using the Digit Case and Digit Width options found in the Advanced Character panel.

Oldstyle figures are a type of numeral which approximates lowercase letterforms by having an x-height and varying ascenders and descenders. They are considerably different from the more common “lining” (or “aligning”) figures which are all-cap height and typically monospaced in text faces so that they line up vertically on charts. Oldstyle figures have what is considered to be a traditional, classic look. They are only available for certain typefaces, sometimes as the regular numerals in a font, but more often within a supplementary or expert font. The figures are proportionately spaced, eliminating the white spaces that result from monospaced lining figures, especially around the numeral one.

0123456789\$£€¥

*Oldstyle figures approximate lowercase letterforms by having an x-height as well as varying ascenders and descenders.*

0123456789\$£€¥

*Lining (or aligning) figures imitate caps in that they all align on the baseline and the cap height.*

Oldstyle figures are most often used within text. Unlike lining figures, they blend in without disturbing the visual line of the body copy. They also work well in headlines since they're not as intrusive as lining figures. Many typographic designers prefer them for most uses except charts and tables.

## Use oldstyle or lining numerals

- 1 Select the numeric figures whose digit case you want to change.

- 2 If not already open, expand the Advanced Character panel.
- 3 To specify lining or oldstyle figures, click the Digit Case pop-up menu and choose Lining or Oldstyle.

### Use proportional or tabular numerals

Tabular numerals are numeric characters where each numeral has the same total character width (the width of the numeral itself plus the white space on both sides). Tabular spacing (also referred to as monospacing) allows numerals to align vertically in tables, financial statements and other columns of figures. Tabular figures are usually lining figures, meaning that they sit on the baseline and have the same height as the capital letters.

Display typefaces usually contain proportional figures. The total character widths of these figures are based on the width of the numeral itself plus a small amount of white space around it. For example, an 8 takes up more width than a 1. Proportional figures can be either lining or old style.

Proportional figures don't align vertically and are unsuitable for use in tables, charts, or other vertical columns.

14,817,611	14,817,611
16,905,620	16,905,620

*Proportional numerals (left) and tabular numerals (right). Note that the tabular numerals each line up with one another, as do the commas used to show place value.*

- 1 Select the numeric figures whose digit width you want to change.
- 2 If not already open, expand the Advanced Character panel.
- 3 To specify tabular or proportional figures, click the Digit Width pop-up menu and choose Tabular or Proportional.

### Understanding dominant and alignment baseline

In addition to their default baselines, some fonts include definitions for the relative positions of other types of baselines. For example, you may use a font whose primary baseline is Roman, but which also includes baseline information for the ideographic top, center, and bottom baselines used in Japanese, Chinese, and Korean ideographic scripts. If you used this Roman font with text whose dominant baseline was, for instance, an ideographic center baseline, the font would align itself with the dominant run of text by aligning its ideographic center baseline with that of the dominant font family's baseline.

When you select a dominant baseline for a given block of text, all fonts reference that selection's baselines for their relative positions, no matter what their type, size, or default baseline. In the illustration below, the text in the top of the illustration is aligned to a Roman baseline, and the text on the bottom is aligned on Ideographic Center.

KATAKANA	片仮名
KATAKANA	片仮名

*Text aligned to a Roman baseline (top) and to an Ideographic Center baseline (bottom)*

Similar to the Leading Basis setting in the Paragraph panel, the Dominant Baseline and Alignment Baseline controls let you specify baseline alignment for fonts that you explicitly select. These controls are useful for aligning baselines of mixed language fonts—for example, Roman and Japanese characters—and for aligning graphic elements inserted within a line of text, such as an inline icon.

*Note: For most situations, you will want to specify Roman when authoring using Roman characters, and Ideographic Center when authoring using Japanese or other Asian languages that use Ideographic baselines.*

### Specify the dominant baseline

The Dominant Baseline control lets you specify the dominant (or primary) baseline for text that you explicitly select (as opposed to Leading Basis, which determines the baseline alignment of an entire paragraph).

- 1 Select the text whose dominant baseline you want to define.
- 2 In the Advanced Character panel, choose the baseline for your text or graphic element from the Dominant Baseline pop-up menu:

**Roman** For text, the font and point size of the text determine this value. For graphic elements the bottom of the image is used.

**Ascent** Specifies an ascent baseline. For text, the font and point size of the text determine this value. For graphic elements the top of the image is used.

**Descent** Specifies a descent baseline. For text, the font and point size of the text determine this value. For graphic elements the bottom of the image is used.

**Ideographic Top** Aligns the small characters in a line to the specified position of the large character's embbox.

**Ideographic Center** Aligns the small characters in a line to the specified position of the large character's embbox.

**Ideographic Bottom** Aligns the small characters in a line to the specified position of the large character's embbox.

### Specify the alignment baseline

Using Alignment Baseline, you can specify a different baseline for text or a graphic image within a paragraph. For example, if you insert an icon into a line of text, you can specify an alignment using the top or bottom of the image relative to the baseline used to align the text.

- ❖ In the Advanced Character panel, choose the appropriate leading basis for your language or graphic element from the Leading Alignment pop-up menu:

**Roman** For text, the font and point size of the text determine this value. For graphic elements the bottom of the image is used.

**Ascent** Specifies an ascent baseline. For text, the font and point size of the text determine this value. For graphic elements the top of the image is used.

**Descent** Specifies a descent baseline. For text, the font and point size of the text determine this value. For graphic elements the bottom of the image is used.

**Ideographic Top** Aligns the small characters in a line to the specified position of the large character's embbox.

**Ideographic Center** Aligns the small characters in a line to the specified position of the large character's embbox.

**Ideographic Bottom** Aligns the small characters in a line to the specified position of the large character's embbox.

**Use Dominant** Specifies that the Alignment Baseline use the Dominant Baseline setting.

## Ligatures

Ligatures are typographic replacement characters for certain letter pairs, such as “fi” and “fl,” when they are available in a given font. Ligatures usually replace consecutive characters sharing common components, and are part of a more general class of glyphs called *contextual forms* where the specific shape of a letter depends on context such as surrounding letters or proximity to the end of a line.

- A. fi ff fj ft Th ct st sp AA CA ET FT OO
- B. fi ff fj ft Th ct st sp AA CA ET FT OO
- C. fi ff fj ft Th ct st sp AA CA ET FT OO
- D. fi ff fj ft Th ct st sp AA CA ET FT OO
- E. fi ff fj ft Th ct st sp AA CA ET FT OO

*Ligature combinations for Hypathia Sans Pro*

A. No ligatures applied B. Minimum ligatures C. Common ligatures D. Uncommon ligatures E. Exotic ligatures



*A script-based font with no ligatures (top) and common ligatures (bottom)*

With OpenType fonts, when you choose Common from the Advanced Character panel, the Text Layout component produces any standard ligature defined in the font, as determined by the font designer. However, some fonts include more ornate, optional ligatures, which can be produced when you choose the Uncommon or Exotic Ligatures option.

Ligatures can be used with both Latin and Non-Latin character sets.

- 1 Select the characters you want to apply ligatures to.
- 2 In the Advanced Character panel, click the Ligatures pop-up menu and choose from the following options:

**None** Specifies no ligatures.

**Minimum** Specifies minimum ligatures.

**Common** Specifies common—or standard—ligatures (the default).

**Uncommon** Specifies uncommon—or discretionary—ligatures.

**Exotic** Specifies exotic—or historical—ligatures. Exotic ligatures are included in only a few font families.

**Note:** *Ligatures are cumulative, so selecting Exotic applies the ligatures found in the Minimum, Common, and Uncommon categories in addition to those found in the exotic category.*

## Text rotation

You rotate individual glyphs in 90 degree increments. Specifying rotation for fonts which do not contain vertical layout information can give undesirable results. Fonts that contain a vmtx (vertical metrics table) or VORG table, such as the Japanese font MS Mincho, work correctly because the tables supply the data that the Flash Text Engine requires for correct layout. Fonts such as Verdana, which do not contain the necessary information, do not.

- 1 Select the characters you want to rotate.
- 2 In the Advanced Character panel, click the Rotation pop-up menu and choose 0, 90, 180, 270 degrees or Auto.  
Auto specifies a 90 degree counter clockwise rotation for full width and wide glyphs only, as determined by the Unicode properties of the glyph. This value is typically used with Asian text to rotate only those glyphs that require rotation. This rotation is applied only in vertical text to return full width and wide characters to a vertical orientation without affecting other characters.

## Specify the opacity of characters.

You can adjust the opacity of characters by adjusting the Alpha setting.

- 1 Select the characters or a graphic image whose opacity you want to modify.
- 2 In the Advanced Character panel, click the Alpha text field, and enter a value between 0% and 100%, where 0% is completely transparent, and 100% is completely opaque.

The alpha value you specify with the Text Layout component combines multiplicatively with any alpha value already set on the graphic itself or on the text line.





## Laying out text with containers

All text you create in the Text Layout component resides in a container. The *container* is the display object for the text you create, and determines text alignment, the width and height of a text block, and the number of columns in the layout. When you place an instance of the Text Layout component on the stage, a container is automatically created with the sample text “TextLayout Sample Text.”

Use the Container panel to change settings such as the number of columns in the container, the vertical alignment of text within the container, or the inset spacing, which is the distance of the margins between the text and the frame.

## Change text alignment


You can align or distribute lines of text in a container along its vertical axis to help keep type vertically consistent among containers and their columns. You can align text to the top, center, or bottom of the container.

- ❖ In the Vertical Justification section of the Container panel, choose one of the following options:
  - To vertically align text down from the top of the container, choose Top . (This is the default setting.)
  - To center lines of text in the container, choose Center .
  - To vertically align lines of text up from the bottom of the container, choose Bottom .
  - To evenly distribute lines of text vertically between the top and bottom of the container, choose Justify .

## Add columns to a container

You can create multi-column layouts within a container.

- 1 In the Container panel, change the Columns option from the default of Auto to Count, and specify the number of columns you want to use in your layout.

 You can also specify a column width, and the Text Layout component will calculate how many columns can fit inside the container.

- 2 In the Gap text field, specify the spacing (or *gutter*) between each column.
- 3 Use the Geometry controls to specify the inset spacing (also known as *padding*), which is the distance of the margins between the text and the container.

## Understanding first line offset

Using the First Line Offset control, you can specify the alignment of the first line of text with the top of the text container. For example, you can make text fall a specific distance below the top of the container.

First line offset is typically referred to as *first baseline offset* when working with Roman characters, where the baseline is an imaginary line on which the majority of the characters in a typeface rest. When working in the Text Layout component, baseline can refer to any of the following depending on the language being used: Roman, Ascent, Descent, Ideographic Top, Ideographic Center, and Ideographic Bottom

### First line offset options

The first line offset options are:

**Auto** Aligns the top of the line—using the tallest glyph—with the top of the container, regardless of the First Line Offset Basis setting. This is the default setting.

**Ascent** The distance between the top inset of the text container and the baseline of the first line of text—as specified using the First Line Offset Basis setting—is the height of the tallest glyph in the font (typically the “d” character in Roman fonts).

**Line Height** The distance between the top inset of the text container and the baseline of the first line of text—as specified using the First Line Offset Basis setting—is the Line Height (leading) of the line.

**pt** Specify the distance in pixels between the baseline of the first line of text and the top inset of the frame.

### First baseline offset basis

As with other options relating to baseline and leading, you can specify a different baseline for text.

**Note:** For most situations, you will want to specify Roman when authoring using Roman characters, and Ideographic Center for when authoring using Japanese and other Asian languages that use Ideographic baselines.

- ❖ In the Advanced Character panel, choose the appropriate baseline basis for your language from the First Line Offset Basis pop-up menu:

**Roman** For text, the font and point size of the text determine this value. For graphic elements the bottom of the image is used.

**Ascent** Specifies an ascent baseline. For text, the font and point size of the text determine this value.

**Descent** Specifies a descent baseline. For text, the font and point size of the text determine this value.

**Ideographic Top** Specifies an ideographic top baseline.

**Ideographic Center** Specifies an ideographic center baseline.

**Ideographic Bottom** Specifies an ideographic bottom baseline.

## Text flow

The Flow panel lets you determine the orientation and direction of text within a container. The text flow options allow you to specify text flow for different languages. For example, a right-to-left horizontal flow for Middle Eastern languages such as Arabic, or a vertical flow for Japanese or Chinese.

The Flow panel options are:

**Orientation** Lets you choose either a horizontal text flow as used by most Western and Middle Eastern languages, or a vertical text flow for Japanese or Chinese.

**Direction** Lets you specify either a left-to-right or right-to-left text direction. Left-to-right is the default setting, and is used for most languages. Right-to-left is used for Middle Eastern languages such as Arabic and Hebrew, and languages based on Arabic script such as Farsi or Urdu.

*Note: When applied at the paragraph level, direction controls the left-to-right or right-to-left text direction, and the indents and punctuation used by the paragraph. When applied to a container, direction controls the column direction. Paragraphs in the container inherit the direction attribute.*

**Line Breaks** You can specify either Hard Breaks Only, which requires that you press Enter or Return to break a line of text, or Auto Line Wrap, which automatically wraps the line of text when it reaches the margin.

**V. Scroll or H. Scroll** Specifies that vertical or horizontal scrolling is enabled. You can choose to turn scrolling on, off, or have it automatically enabled if the text overflows the container.

## Inserting links

You can create link elements within text.

- 1 Select the text or graphic you want to use to create a link.

*Note: If you select a portion of an existing link, the selected area becomes a brand new link. The selection will no longer be part of the original link.*

- 2 Click the Links tab located in the lower left of the Text Layout panel.
- 3 Type the path and filename of the document in the Link URL text field. To link to a document in your site, enter a document-relative or site root-relative path. To link to a document outside your site, enter an absolute path including the protocol (such as http://). Only http: and relative paths from the SWF are supported.
- 4 From the Link Target pop-up menu, select a location in which to open the document:

**\_blank** Loads the linked document in a new, unnamed browser window.

**\_parent** Loads the linked document in the parent frame or parent window of the frame that contains the link. If the frame containing the link is not nested, then the linked document loads in the full browser window.

**\_self** Loads the linked document in the same frame or window as the link. This target is the default, so you usually don't have to specify it.


**\_top** Loads the linked document in the full browser window, thereby removing all frames.

- 5 Click Apply Link to apply the link.

## Combining text and graphics

You can insert inline graphics within text. Supported image formats are: SWF, PNG, and JPEG.

- 1 Place the insertion point within the text where you want to position the graphic.
- 2 Click the Graphics tab located in the lower left of the Text Layout panel.
- 3 Type the path and filename of the image file in the Image URL text field. To link to an image within your site, enter a document-relative or site root-relative path. To link to an image outside your site, enter an absolute path including the protocol (such as `http://`). Only `http:` and relative paths from the SWF are supported.
- 4 Specify the Width and Height of the frame for the image within the container.

 *If you leave the Width and Height fields blank, the Text Layout component uses the native size of the image.*

- 5 (Optional) Click the Status button located in the Graphics tab to view information about the image and both its native and imported width and height.