Internal Memo

To	CedarInterest	Date	September 15, 1982
From	Bill Paxton	Location	PARC/ISL
Subject	Fancy tabs	File	TiogaTabs.tioga

XEROX

Release as [Indigo]<Tioga>Documentation>TiogaTabs.tioga
Last edited by Paxton, September 15, 1982 7:33 am

Abstract This is a specification for fancy tabs in Tioga.

Introduction

Tabs are not a substitute for general alignment facilites, but they are relatively easy to implement and can handle a substantial set of formatting problems. The general facilities to support tables and other structured objects won't be available until '83, so I propose to implement "fancy tabs" now rather than waiting. These will allow both a choice of alignment and a fill pattern for each tab setting. The alignments will include flush left, flush right, centered, and character aligned. The fill choices will provide for leaders and rules in the tab gap. This functionality is essentially that of BravoX with a few generalizations.

Specification

Following the usual Tioga convention, the formatting information for the tab will be contained in the style. The tab characters in the document will not have any special looks; they simply serve to separate sections of text.

The style determines where the tab stops are located and the properties of each stop. The locations are given in the style as a distance which is relative to either the left margin or the left of the page or viewer. The properties govern alignment and fill.

Locations for tab stops

The first tab in a line of text will be interpreted according to the style for the first tab stop, the second tab according to the second tab stop, etc. This may sound obvious, but it isn't. It is unlike the common interpretation which, like a typewriter, finds the next tab stop after the current location. Our interpretation is superior because it allows the user to know the mapping from tab character to tab stop independent of the width of the text to the left of the tab.

The minimum width for a tab is one space in the current font. You are given control in the style over what to do if there is not enough room available for the tab (i.e., the location on the line where the tab is encountered doesn't leave room for a space before the desired tab stop). The default is to treat the tab like a (fixed width) space. However, there is also an option to treat it like a carriage return to cause a line break.

The line break option is useful in situations such as formats for descriptions in which the word or phrase being described appears at the start of a node, followed by a tab and the defining text. The tab stop is located at the same distance relative to the left margin as the hanging indent for the other lines of the paragraph. For short items, the text after the tab will start on the first line with the same indent as the following lines. For long items in which the initial word or phrase extends beyond the amount of hanging indent, the tab will cause a line break so that the description will start on the next line.

Alignment for tabs

Tab stops can have either flush left, flush right, centered, or character alignment. For a flush left tab, line formatting continues from the location of the stop.

For the other cases, the system first finds the block of text up to the next tab, carriage return, or end of node. This block is then shifted left from the stop until the desired alignment is achieved. If the block of text is too big to be aligned while still leaving a gap of at least the minimum size, the tab is treated like a fixed width space or a carriage return as described above.

For flush right, the desired alignment is to have the right of the text at the stop location.

For centered, it is to have the center of the text at the stop.

For character alignment, the part of the block up to but not including the specified character is set flush right to the stop with the rest of the block immediately following. A typical use of character alignment is to provide decimally aligned numbers.

Fill for tabs

You may specify in the style what fill, if any, is to appear in the gap left for the tab. The choices for fill are as follows:

blank put nothing in the gap.

leaders repeat a specified character to fill the gap.

rules draw one or more horizontal lines in the gap.

For leaders, you may specify both the character to repeat and the spacing between repetitions. You may also choose either centered or congruent placement of the characters. The former means that the sequence of characters will be centered in the tab gap. The latter means that the sequence will be begin at a location on the page or viewer with a horizontal coordinate equal to zero modulo the spacing so that different lines of leaders will be aligned independent of their start or end positions.

For rules, you may specify both the thickness of the line and its vertical offset from the baseline (positive or negative). The rules are drawn the full width of the gap created for the tab.

Looks associated with tab stops

Tables often use special formatting such as putting the first column in a variable pitch text font, others in a fixed pitch font for numbers, and perhaps the final column in bold face to highlight various totals. By defining default looks for the various tab stops, we can get this effect without having to add any explicit looks to the entries. The default looks for a given tab stop are automatically applied to the text following the tab up to the next tab or the end of the line. For example, if a tab stop had default look "b", then the text in the column defined by that stop would automatically appear in bold face.

Default tab stops

We still need to define what happens if you encounter n tabs in a line when the style only defines n-1 stops. This is taken care of by having "default tab stops" which are defined similarly to normal stops except that their location is given as a distance relative to the previous stop. For example, if the style specifies 1 inch default tab stops, then stops will be located at 1 inch intervals after the last explicitly defined stop.

Style language for tabs

This section gives a semiformal description of the language to be used in styles dealing with tab stops. In the following definitions, [ ]'s enclose optional items, < >'s enclose names of defined items, and |'s separate alternatives.

Setting a tab stop

To create a tab stop as part of a style rule, include a statement of the form

<statement> = <location> <tabSpec> tabStop

where

<location> = a number such as 2.5 in or 32 pt, and

<tabSpec> = [<align>] [<pattern>] [<pastSpec>] [<looksSpec>]

The <align> tells how the text following the tab is to be aligned relative to the tab location. The default for <align> is flushLeft.

The <pattern> tells how to fill the gap left for the tab. The default for <pattern> is blank.

The <pastSpec> tells what to do if you are already beyond the tab stop in the line. The default is to treat the tab like a fixed width space.

<looksSpec> = <string> looks

The <string> contains the (lowercase) letters for the default looks.

Alignments

<align> = flushLeft | flushRight | centered | <char> aligned

<char> = a single character string such as "."

Fill

<pattern> = blank | <leaders> | <rule> rule | { <rules> } rules

<leaders> = <char> [<dist>] [centered | congruent] leaders

The <char> is the character to be repeated in the tab gap. The <dist> is the distance between the characters. If <dist> is omitted, the system will pick something pleasing such as twice the character width. Congruent leaders are the default.

<rule> = <weight> <dist>

The <weight> is the thickness of the horizontal line. The <dist> is the vertical distance from the baseline to the bottom of the rule and may be either positive or negative.

What to do if past the tab stop

<pastSpec> = breakIfPast | spaceIfPast

The former means treat the tab like a carriage return and the latter means treat it like a space.

Miscellaneous commands

Setting default tab stops

To set the default tab stops, use a statement of the form

<statement> = <dist> <tabSpec> defaultTabStops

To read the current specification for default tab stops, use the following statement to push a full description on the JaM stack

<statement> = the defaultTabStops

Clearing stops

The command clearTabStops will remove all tab stops. This should be used before you start making new settings.

Fixed or relative tab stops

To control the interpretation of tab locations, use either of these statements:

<statement> = relative tabStops | fixed tabStops

Relative means that tab locations are measured from the left margin. Fixed means that they are measured from the left edge of the page or viewer.

Reading tab stops for debugging purposes

You can read the current tab settings by using the following statement.

<statement> = <exec> forEachTabStop

For each tab stop, this pushes a full description on the JaM stack and then executes <exec>.