

Tabs and Boxes

This chapter covers the following topics:

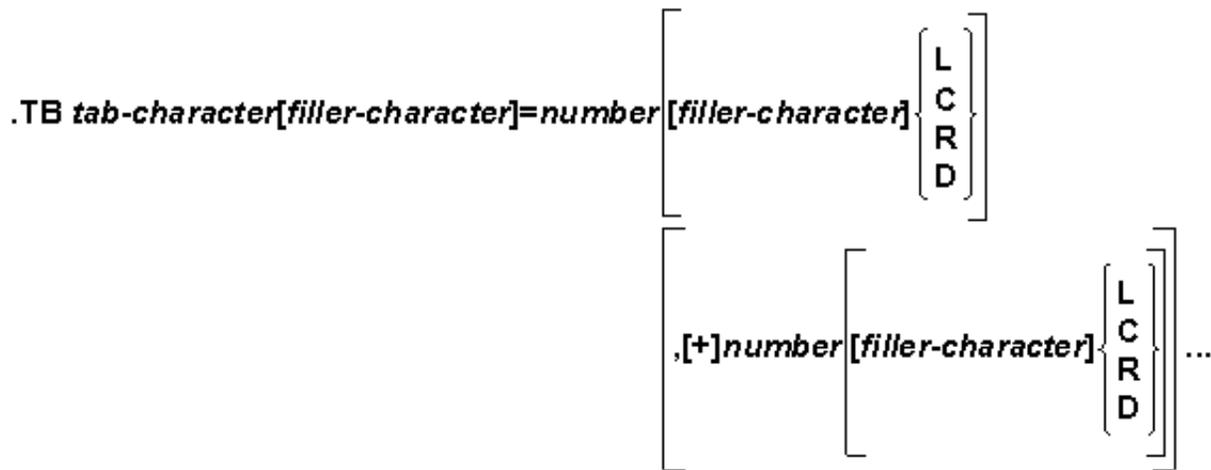
- Tabs
- Boxes

Tabs

When you use tabs to create tables, it is recommended that you switch filling off (.FI OFF), so that the columns in your table are vertically straight.

When filling is switched on, Con-form first tabulates your input and then executes the .FI ON instruction. This may lead to undesired results.

.TB - Set Tab Stops



.TB OFF

This instruction is used to define the position for each tab stop. In its basic form, you have to define a tab character and the tab position.

For example, to define the asterisk (*) as the tab character and the tab position as 10 columns to the right of the current left margin, you must specify:

```
.TB *=10
```

The tab character is the character you enter in your source text in order to move the text which is preceded by it to the defined tab position. You can define any character except a blank space as the tab character. If you want to define the instruction separator character (initially, this is the semicolon) as the tab character, you must repeat it (i.e. you must specify ";;").

Note:

Each tab stop occupies at least one space.

The tab position defines the column where the tab stop is to be set. The position of a tab stop is influenced by the current left margin which has been defined with the .LM instruction. For example, if you defined the left margin as .LM 10, your text starts in column 11. Thus, if you define a tab stop as 10 columns to the right of the current left margin, the tabulated text starts in column 21.

Note:

If the left margin is shifted using the .TI or .OF instruction, all tab positions are shifted accordingly.

If you want to define several tab stops, you must separate the column numbers by commas; you must not include spaces. For example:

```
.TB *=10,20,30,40
```

Each column number except the first can be preceded by a plus (+) sign. This denotes that the tab stop is to be set *n* columns to the right of the previous tab stop. The following two examples are therefore equivalent:

```
.TB *=5,10,18,23,30,40,50,65
```

```
.TB *=5,+5,+8,+5,+7,+10,+10,+15
```

You can define a maximum of 20 tab stops.

Each time you define new tab stops using the .TB instruction, any tab stops which have been defined previously are deleted. The TAB key on your keyboard is not used by Con-form.

Tab Stop Alignment

To align a tab stop, you specify one of the following letters after the column number:

L	Left-justified alignment
C	Centered alignment
R	Right-justified alignment
D	Decimal alignment

Left-justified Alignment

To create a left-justified tab stop, you can specify the letter L after the column number. For example:

```
.TB *=10L
```

The text is aligned so that it starts in the specified column.

Note:

If you do not specify a letter, all tab stops are left-justified.

Centered Alignment

To create a centered tab stop, you must specify the letter C after the column number. For example:

```
.TB *=30C
```

The text is aligned so that its center coincides with the specified column.

Right-justified Alignment

To create a right-justified tab stop, you must specify the letter R after the column number. For example:

```
.TB *=50R
```

The text is aligned so that it ends in the specified column.

Decimal Alignment

To create a decimal tab stop, you must specify the letter D after the column number. For example:

```
.TB *=25D
```

The text is aligned so that the decimal character coincides with the specified column. If your text does not contain a decimal character, one is assumed directly to the right of your text.

Initially, the decimal character is the period (.). You can define a different decimal character. For example, to define the comma as the new decimal character, you specify:

```
.OP DEC=','
```

Since the comma is used by certain Con-form instructions to separate parameters, it is important that you enclose it in apostrophes as shown above. If you want to define, for example, the slash (/), you need not use the apostrophes.

Filler Character

Filler characters appear whenever a tab occurs. You can define any character except the equal sign (=) as the filler character. If you want to define the instruction separator character (initially, this is the semicolon) as the filler character, you must repeat it (i.e. you must specify ";;"). When you do not specify a filler character, spaces are used.

If you want to define the period (.) as the filler character, you specify it before the equal sign:

```
.TB *. =10
```

For example, you can enter the following:

```
.FI OFF
.TB *. =15
Athens*07:48
Frankfurt*09:05
London*10:34
```

The above source text causes the following formatted output:

```
Athens.....07:48
Frankfurt.....09:05
London.....10:34
```

When you use one of the letters that is used to define the type of tab stop alignment (L, C, R or D), you can also define the filler character after the equal sign. In this case, you must specify the filler character between the column number and the letter. You can then define different filler characters for each tab stop. For example:

```
.TB *=10-L,30.C,50_R
```

The filler characters always appear *before* the defined tab stop.

Note:

When you have defined filler characters before and after the equal sign, the character before the equal sign is only considered if a column number (after the equal sign) is not followed by a filler character.

Canceling the Tab Character

If you want to print the character which has been defined as the tab character in your formatted output, you can cancel the tab character as follows:

```
.TB OFF
```

You can also define a different tab character. For example, if the current tab character is the asterisk and you want to use the hash (#), you specify:

```
.TB #=10
```

Example

This example is divided into three parts. The first part illustrates how to create tables using tabs and how to align them decimally. The second part shows that the tab character is not ignored by the .IC instruction. The third part shows how to use tabs with the .OF instruction which sets the left margin for all lines except the first line.

Source Text

```
.LM 0;.RM 60
.FI OFF
.SL 1
.** The following example shows how to create a simple table.
.** The last two columns are aligned decimally
.TB *=10,30D,40D
2 doz*Eggs*2.39*4.78
3 pints*Milk*.99*2.97
1 piece*Butter*2.29*2.29
2 lbs*Meat*15.99*31.98
.SL 1
Total***42.02
.SL 2
.** The following example shows that the tab character is not ignored
.** by the .IC instruction.
Breaks are caused by the following instructions:
.TB #=10,20,30,40
```

```
.IC ON
.BF#.EF#.IP#.OF#.SL
.BX#.EM#.LM#.PH#.TI
.CE#.FI#.NL#.PS#.UL
.CH#.IL#.NP#.RA
.IC OFF
.SL 2
** The following example shows how to use tabs with .OF
.FI ON;.JU ON
.TB *=15
.OF 15
Offsetting:*The tab stop has been set so that the remaining text in the
first line begins in the same column as the text that has been offset.
```

Formatted Output

2 doz	Eggs	2.39	4.78
3 pints	Milk	.99	2.97
1 piece	Butter	2.29	2.29
2 lbs	Meat	15.99	31.98
Total			42.02

Breaks are caused by the following instructions:

```
.BF      .EF      .IP      .OF      .SL
.BX      .EM      .LM      .PH      .TI
.CE      .FI      .NL      .PS      .UL
.CH      .IL      .NP      .RA
```

Offsetting: The tab stop has been set so that the remaining text in the first line begins in the same column as the text that has been offset.

Boxes

You can create either empty boxes or boxes that are filled with text. You can also create tables using boxes.

.BX - Draw a Box

```
.BX [-]number [, [+]number]... , [+]number
.BX
.BX OFF
.BX CANCEL
```

To draw a box, you must proceed as follows:

1. You must first define the positions of the vertical lines. For example, if you want to draw a box starting in column 10 and ending in column 55, you specify:

```
.BX 10,55
```

This instruction draws the first horizontal line of the box. The numbers you entered determine the columns in which the horizontal line is to start and end, and the positions of the vertical box lines.

Note:

When the first parameter is 0, the left vertical box line is not drawn.

You must specify at least two parameters with the .BX instruction. You can specify a maximum of 20 parameters.

Each column number except the first can be preceded by a plus sign (+). This denotes that the vertical box line is to be set *n* columns to the right of the previous line. The following two examples are therefore equivalent:

```
.BX 10,55
```

```
.BX 10,+45
```

The column positions specified with the .BX instruction (unlike the .TB instruction) are absolute positions. They are not influenced by the current position of the left margin.

2. Next, you must specify the height of the box. The number of lines within the box is determined by the text you enter now and/or the number of blank lines you specify. For example, if you want to create an empty box, you specify the number of blank lines:

```
.IL 5
```

This instruction causes the vertical box lines to be drawn at the specified positions. There is no text between both vertical lines.

3. Finally, to draw the final horizontal box line and to return to normal text processing, you specify:

.BX OFF

The instructions in steps 1 to 3 cause the following formatted output:

```
+-----+
!                                     !
!                                     !
!                                     !
!                                     !
!                                     !
+-----+
```

Note:

Some devices use the exclamation mark (!) instead of the vertical bar. The exclamation mark has also been used in all these examples.

Horizontal Box Lines

To draw several horizontal lines within the box, you must use the .BX instruction without parameters as illustrated below:

```
.BX 10,55
.IL 1
.BX
.IL 1
.BX
.IL 1
.BX OFF
```

The above instructions cause the following formatted output:

```
+-----+
!                                     !
+-----+
!                                     !
+-----+
!                                     !
+-----+
```

Grid

To draw a grid with more than two vertical lines, you must specify more than two parameters with the first .BX instruction. The remaining .BX instructions (which have no parameters) draw lines at the specified positions. For example:

```
.BX 10,15,20,25,30,35,40
.IL 1
.BX
.IL 1
.BX
.IL 1
.BX OFF
```

The above instructions cause the following formatted output:

```
+-----+
!   !   !   !   !   !   !
+-----+
!   !   !   !   !   !   !
+-----+
!   !   !   !   !   !   !
+-----+
```

Superimposed Boxes

To draw a series of superimposed boxes with vertical lines at different positions, you must repeat the `.BX` instruction with the new parameters. If the minus sign is used with a parameter, the vertical line is not printed at that column (this is explained in detail later in this chapter). For example:

```
.BX -10,25,35,-50
.IL 1
.BX 10,20,30,40,50
.IL 3
.BX 10,50
.IL 1
.BX OFF
```

The above instructions cause the following formatted output:

```
-----
                !           !
-----
!           !           !           !           !
!           !           !           !           !
!           !           !           !           !
+-----+
!                                           !
+-----+
```

Boxes Containing Text

When you want your text contained in a box, you must make sure that the parameter for the right margin (defined with `.RM`) is lower than the parameter for the right vertical line. The number of lines within the box is determined by the text you enter after the `.BX` instruction. For example:

```
.FI ON;.JU ON
.BX 10,55
.LM 10;.RM 54
```

The parameter for the right margin must be lower than the parameter for the right vertical line. Otherwise, the right border of the box does not appear correctly.

```
.BX OFF
```

The above instructions cause the following formatted output:

```
+-----+
!The parameter for the right margin must be!
!lower than the parameter for the right!
!vertical line. Otherwise, the right border!
!of the box does not appear correctly.      !
+-----+
```

If you prefer the text with a jagged edge, you must make sure that justification has been switched off. However, you must not switch filling off (.FI ON;JU OFF).

```
+-----+
!When filling is switched off, the text appears in the same way as you
!entered it, i.e. the specifications for the right margin are
!not considered.                                     !
+-----+
```

It is your responsibility to ensure that your text fills the box in the required manner. You must always specify the appropriate left and right margins for the text that is to appear within a box.

```
+-----+
This!example shows what happens, if you
do not ensure that the correct margins
are specified.                                     !
+-----+
```

You can define the left and right margins so that blank spaces appear before and after a text line, and you can insert blank lines before and after the text. For example:

```
.FI ON;.JU ON
.BX 10,55
.LM 12;.RM 52
.IL 1
The left and right margins of the text are specified so that 2 blank spaces
appear before and after each text line in the box.
Furthermore, one blank line appears before and after the text.
.IL 1
.BX OFF
```

The above instructions cause the following formatted output:

```
+-----+
!
! The left and right margins of the text !
! are specified so that 2 blank spaces !
! appear before and after each text line !
! in the box. Furthermore, one blank line !
! appears before and after the text.     !
!
+-----+
```

Boxes without a Left Side

To create a box without a left side, you must insert a minus sign (-) before the parameter for the first column. For example:

```
.BX -10,50
```

The following examples illustrate how this feature can be used:

```
.FI ON;.JU ON
.LM 10;.RM 50
.BX -10,51
```

The values for the left vertical line and the left margin are the same.

```
.BX OFF
.IL 1
.BX -11,51
```

The value for the left vertical line has been increased by one.

Now the box starts in the same column as the text.

```
.BX OFF
.IL 1
.BX -15,51
```

The value for the left vertical line is more than the value for the left margin.

```
.BX OFF
```

The above instructions cause the following formatted output:

```
-----
The values for the left vertical line!
and the left margin are the same.    !
-----
```

```
-----
The value for the left vertical line has!
been increased by one. Now the box!
starts in the same column as the text. !
-----
```

```
-----
The value for the left vertical line is!
more than the value for the left margin.!
-----
```

Boxes without a Right Side

To create a box without a right side, you must insert a minus sign (-) before the parameter for the last column. For example:

```
.BX 10,-54
```

The following examples illustrate how this feature can be used:

```
.FI ON;.JU ON
.LM 10;.RM 54
.BX 10,-54
```

The values for the right vertical line and the right margin are the same.

```
.BX OFF
.IL 1
.BX 10,-60
```

The value for the right vertical line is more than the value for the right margin.

```
.BX OFF
.IL 1
.BX 10,-45
```

The value for the right vertical line is less than the value for the right margin.

```
.BX OFF
```

The above instructions cause the following formatted output:

```
+-----+
!The values for the right vertical line and
!the right margin are the same.
+-----+

+-----+
!The value for the right vertical line is
!more than the value for the right margin.
+-----+

+-----+
!The value for the right vertical line is
!less than the value for the right margin.
+-----+
```

Boxes without a Bottom

To create a box without a bottom, you must use the `.BX CANCEL` instruction instead of `.BX OFF`. In this case, the final horizontal box line is not drawn. For example:

```
.BX 10,55
.II 2
.BX CANCEL
```

The above instructions cause the following formatted output:

```
+-----+
!                                     !
!                                     !
```

You can also draw boxes without a bottom and without a left or right side. For example:

```
.FI ON;.JU ON
.LM 12;.RM 30
.BX 10,-33
.II 1
This box has no right side and no bottom.
.II 1
.BX CANCEL
```

The above instructions cause the following formatted output:

```
+-----+
!
! This box has no
! right side and no
! bottom.
!
```

Note:

It is not possible to create a box without a top.

Tables

To create a table, you define the tab stops using the `.TB` instruction and the box lines using the `.BX` instruction. For example:

```
.LM 0;.RM 65
.TB #=10,22
.BX 0,9,21,-67
Keyword#Value#Description
.BX
BIN#number#Select a paper tray
BRN#ON/OFF#Suppress breaks in line filling
BXH#character#Define character for horizontal box lines
BXV#character#Define character for vertical box lines
CHA#NUM/UNN#Chapter numbering
.BX CANCEL
```

The above instructions cause the following formatted output:

```
-----
Keyword ! Value      ! Description
-----
BIN      ! number    ! Select a paper tray
BRN      ! ON/OFF    ! Suppress breaks in line filling
BXH      ! character  ! Define character for horizontal box lines
BXV      ! character  ! Define character for vertical box lines
CHA      ! NUM/UNN   ! Chapter numbering
```

In the following example, the `.OF` instruction has been used to indent the text of the third column, i.e. it sets the left margin for all text following the first line.

```
.LM 2;.RM 58
.FI ON;.JU OFF
.BX 1,13,23,60
.TB #=12,22
.OF +22
Chapter 6#.OP HYP#Define language code
.BX;.OF +22
#.OP HYB#Number of characters before the hyphen
.BX;.OF +22
#.OP HYA#Number of characters after the hyphen
.BX;.OF +22
#.HP#Hyphenated words at the end of a source text line are not
hyphenated in the formatted output.
.BX;.OF +22
Chapter 7#.TB#Define positions of tab stops
.BX;.OF +22
#.BX#Draw a box
.BX;.OF +22
#.OP BXH#Define character for the horizontal box line
.BX;.OF +22
#.OP BXV#Define character for the vertical box line
.BX OFF
```

The above instructions cause the following formatted output:

```

+-----+
! Chapter 6 ! .OP HYP ! Define language code          !
+-----+
!           ! .OP HYB ! Number of characters before the  !
!           !           ! hyphen                          !
+-----+
!           ! .OP HYA ! Number of characters after the      !
!           !           ! hyphen                          !
+-----+
!           ! .HP     ! Hyphenated words at the end of a      !
!           !           ! source text line are not            !
!           !           ! hyphenated in the formatted        !
!           !           ! output.                             !
+-----+
! Chapter 7 ! .TB     ! Define positions of tab stops        !
+-----+
!           ! .BX     ! Draw a box                           !
+-----+
!           ! .OP BXH ! Define character for the            !
!           !           ! horizontal box line                 !
+-----+
!           ! .OP BXV ! Define character for the vertical   !
!           !           ! box line                            !
+-----+

```

.OP BXH - Horizontal Box Line

```
.OP BXH=character
```

The initial character for the horizontal box line is the hyphen (-). However, you can specify any character you want. For example, to specify the asterisk (*) as the new character, you specify:

```
.OP BXH=*
```

When you specify a different character for the horizontal box line, the character for the box corners (+) is no longer in effect.

.OP BXV - Vertical Box Line

```
.OP BXV=character
```

The initial character for the vertical box line is the vertical bar (|). However, you can specify any character you want. For example, to specify the asterisk (*) as the new character, you specify:

```
.OP BXV=*
```

Note:

Some devices use the exclamation mark (!) instead of the vertical bar.

The following example illustrates how to create a box where the horizontal and vertical lines are made up of asterisks:

```
.OP BXH=*
.OP BXV=*
.BX 10,55
.II 2
.BX OFF
```

The above instructions cause the following formatted output:

```
*****
*                                     *
*                                     *
*****
```