

Con-form Instructions

This chapter explains how to specify Con-form instructions in your source text. It also explains how to suppress the recognition of Con-form instructions.

The following topics are covered below:

- Instruction Syntax
 - Option Syntax
 - Suppressing the Recognition of Instructions
-

Instruction Syntax

Each description of a Con-form instruction is preceded by a diagram which shows all the possibilities for specifying the instruction in question. For example:

<pre>.RA <i>number</i> .RA ON .RA OFF</pre>

Each instruction is indicated in upper-case letters and must be specified as shown in the above diagram. Each parameter for which you must substitute a number, character or text is indicated in lower-case italics.

Note:

All instructions are indicated in upper-case letters. However, you can specify instructions and parameters in any combination of upper- and lower-case letters.

The following symbols are used within the syntax descriptions:

[]	Elements contained within square brackets are optional.
{ }	Elements contained within braces indicate that one (that is, one only) of the elements must be specified.
...	A term preceding an ellipsis may be repeated.

The Period

You must begin each Con-form instruction with a period (.). For example:

```
.BR
```

As a rule, you begin an instruction in the very first column of a line. However, it is possible to mix text and instruction in a line. See the description of the instruction separator character for further information.

When you specify more than one period, the subsequent input is not interpreted as a Con-form instruction.

The Parameters

With certain instructions, you must specify one or more parameters.

When an instruction requires a parameter, you must insert a space character between the instruction and the parameter. For example:

```
.FI ON
```

When an instruction requires more than one parameter, you must insert a comma (,) between the parameters. For example:

```
.BX 10,55
```

The Instruction Separator Character

When you define several instructions in the same line, you must insert the instruction separator character between the instructions (initially, the instruction separator character is the semicolon). For example:

```
.FI ON;.JU ON
```

You can also mix text and instructions in a line. The instruction separator character followed by a period (;.) always indicates that the subsequent input is an instruction (you must *not* include a blank between the instruction separator character and the period). For example:

```
This is text;.IL;This is more text;.IL;And this is also text
```

```
.CE ON;Centered Heading;.CE OFF;.IL;Text below the heading.
```

Caution:

The instruction separator character is only recognized, when it is followed by a period. When you specify a blank between the instruction separator character and the period, the subsequent input is not interpreted as an instruction.

.OP CSE - A Different Instruction Separator Character

```
.OP CSE=character
```

Initially, the instruction separator character is the semicolon. However, you can define a different character. For example, to define the colon (:) as the instruction separator character, you must specify:

```
.OP CSE=:
```

Option Syntax

.OP - Option

```
.OP keyword=value[,keyword=value]...
```

Options are special instructions that determine several operating characteristics of Con-form. When necessary, you can redefine the initial value of an option.

You must insert a space character between .OP and the keyword. You must insert an equal sign between the keyword and the value. You must not insert blanks before and after the equal sign.

The following example shows how to specify an option. In this case, CSE is the keyword and the colon (:) is the value.

```
.OP CSE=:
```

You can combine several options in the same line. To do so, you must insert a comma between the options. The following example shows how to define three different options in the same line:

```
.OP CSE=: ,END=% ,ESC=/'
```

When a character is already used for a different purpose (for example, the comma is used to separate the options of the .OP instruction), it is important that you enclose the value in apostrophes. For example:

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

The options are described in the different sections of this documentation which deal with the topic in question.

Suppressing the Recognition of Instructions

.IC - Ignore Instructions

```
.IC ON  
.IC OFF
```

Since Con-form interprets a period in the first column of a line as the instruction identifier, you must not use this character in the first column of your text. However, you can suppress the recognition of instructions. This enables you to use the period in the first column and to output Con-form instructions in the formatted text.

To suppress the recognition of instructions, you must specify the following instruction:

```
.IC ON
```

All information after this instruction will also appear in the formatted version. To switch this feature off, so that Con-form instructions are executed again, you must specify the following instruction beginning in the *first* column of a line (this corresponds to the initial setting):

```
.IC OFF
```