Cobol Tutorial

Edited Pictures

Introduction

Aims	In a business-programming environment, the ability to print reports is an important property for a programming language. COBOL allows programmers to write to the printer, either directly or through an intermediate print file.

But there would be little point in being able to write to the printer, if the output could not be formatted properly. COBOL allows sophisticated formatting of output through its Edited Picture clauses.

This tutorial introduces the additional symbols required for edited pictures and shows how they may be used to format data for output to screen or printer. |
| **Objectives** | By the end of this unit you should -

1. Know what an Edited Picture is.
2. Know and be able to use the different kinds of Edited Picture. |
| **Prerequisites** | Introduction to COBOL

Declaring data in COBOL

Basic Procedure Division Commands

Selection Constructs

Iteration Constructs

Introduction to Sequential files |

Processing Sequential files

To top of page

What is an Edited Picture?

Introduction

Most users of the data produced by COBOL programs are not content with the simple raw data. They often want it presented in a particular way. Some people like to have the thousands, in numeric values, separated by commas, others may want leading zeros suppressed while still others may require that the currency symbol "floats" up against the first non-zero digit. In COBOL these things can be achieved using Edited Pictures.

Original value	00023456.78
With commas inserted	00,023,456.78
Plus zero suppression	23,456.78
Plus floating currency symbol	$23,456.78
With anti-fraud printing	$***23,456.78

Edit Symbols

Edited Pictures, are PICTURE clauses that format data intended for output to screen or printer. To enable the data items to be formatted, COBOL provides additional picture symbols to supplement the basic 9, X, A, V and S.

The additional symbols are referred to as "Edit Symbols," and PICTURE clauses that include edit symbols are called "Edited Pictures".

The term edit is used, because the edit symbols have the effect of changing, or editing, the data inserted into the edited item.

Edited items cannot be used as operands in a computation, but they may be used as the result or destination of a computation (they can be used in items placed to the right of the word GIVING).

Types of editing picture

COBOL permits two basic types of editing picture:

1. **Insertion Editing**
   This type of editing modifies a value by including additional items and has the following sub-categories:
   - Simple Insertion
   - Special Insertion

- Fixed Insertion
- Floating Insertion
2. **Suppression and Replacement Editing**
   This type of editing suppresses and replaces leading zeros and has the following sub-categories:
   - Zero suppression and replacement with spaces
   - Zero suppression and replacement with asterisks

Editing symbols

COBOL permits two basic types of editing picture:>

Edit Symbol	Editing Type
, B 0 /	Simple Insertion
.	Special Insertion
+ - CR DB $	Fixed Insertion
+ - $	Floating Insertion
Z *	Suppression and Replacement

To top of page

Insertion Editing

Introduction

There are four types of Insertion Editing:-

- Simple Insertion
- Special Insertion
- Fixed Insertion
- Floating Insertion

Insertion Editing is so called because the edit symbol is inserted into the data item at the same position it occupies in the picture clause.

Simple Insertion editing

Simple Insertion editing consists of specifying the relevant insertion character(s) in the PICTURE string. When a value is moved into the edited item, the insertion characters are inserted into the item at the position specified in the PICTURE.

The comma, the B, the 0, and the slash (/) are the Simple Insertion editing symbols.

All Simple Insertion symbols count toward the number of characters printed or displayed. For instance, an item described as PIC 99/99/9999 will occupy 10 character positions when printed.

The comma (,) symbol
The comma symbol (,) instructs the computer to insert a comma at the character position where the symbol occurs. The comma counts towards the size of the printed item. The comma symbol cannot be the first symbol in the PICTURE string.

If all characters to the left of the comma are zeros and zero-suppression is called for, the comma is replaced by the replacement symbol (asterisk or space).

The space or blank (B) symbol
A space is inserted where the blank symbol (B) occurs.

The slash and zero symbols (/ and 0)
A slash is inserted where the slash symbol (/) occurs, and a 0 is inserted where the zero symbol (0) occurs.

Simple Insertion examples

In the examples/questions below, see if you can figure out what result will be produced when we move the value in the Sending item to the editied picture in the Receiving item. The description of the Sending item is shown in the Picture column and its current value is shown in the Data column.

Sending item		Receiving item	
Picture	**Data**	**Picture**	**Result**
PIC 999999	123456	PIC 999,999	Click arrow for the answer ⌄
PIC 9(6)	000078	PIC 9(3),9(3)	Click arrow for the answer ⌄
PIC 9(6)	000078	PIC ZZZ,ZZZ	Click arrow for the answer ⌄
PIC 9(6)	000178	PIC ***,***	Click arrow for the answer ⌄
PIC 9(6)	002178	PIC ***,***	Click arrow for the answer ⌄
PIC 9(6)	120183	PIC 99B99B99	Click arrow for the answer ⌄
PIC 9(6)	120183	PIC 99/99/99	Click arrow for the answer ⌄
PIC 9(6)	031245	PIC 990099	Click arrow for the answer ⌄

Special Insertion

The decimal point is the only Special Insertion symbol. A decimal point is inserted in the character position where the symbol occurs.

Notes
When a numeric data-item is moved into an edited data-item containing the decimal point symbol, alignment occurs along the position of the decimal point symbol, with zero-filling and truncation as necessary.

There may be only one decimal point in each edited picture clause.

The decimal point symbol cannot be mixed with either the V (assumed decimal point) or the P (scaling position) symbol.

Special Insertion examples

In the examples/questions below, see if you can figure out what result will be produced when the value in the Sending item is moved to the editied picture in the Receiving item. The description of the Sending item is shown in the Picture column and its current value is shown in the Data column.

Sending item		Receiving item	
Picture	Data	Picture	Result
PIC 999V99	12345	PIC 999.99	Click arrow for the answer ⌄
PIC 999V99	02345	PIC 999.9	Click arrow for the answer ⌄
PIC 999V99	71234	PIC 99.99	Click arrow for the answer ⌄
PIC 9(4)	2456	PIC 999.99	Click arrow for the answer ⌄

Fixed Insertion

The default currency symbol is the dollar sign ($) but it may be changed to a different symbol by the CURRENCY SIGN IS clause, in the SPECIAL-NAMES paragraph, of the CONFIGURATION SECTION, in the ENVIRONMENT DIVISION.

Fixed Insertion editing inserts the symbol at the beginning or end of the edited item.

The Fixed Insertion editing symbols are:

- the plus (+) and minus (-) signs,
- the letters CR and DB representing credit and debit,
- and the currency symbol usually the $ sign.

All symbols count toward the size of the printed item.

Plus and minus symbols
These must appear in the leftmost or rightmost character positions and they count towards the size of the data item. They must be the first or last character in the PICTURE string.

Minus
If the sending item is negative, a minus sign is printed. If the sending item is positive, a space is printed instead. Use this to highlight negative values only.

Plus
If the sending item is negative, a minus in printed and if the sending item is positive, a plus is inserted. Use this to when you always want the sign printed.

CR and DB
CR and DB count towards the data item size and occupy two character positions. They may only appear in the rightmost position. Both are only printed if the sending item is negative. Otherwise two spaces are printed.

The currency symbol (usually $).
The currency symbol must be the leftmost character and it counts towards the size of the item. It may be preceded by a plus or a minus sign.

Fixed Insertion examples

SAQ

Like the previous examples/questions above, see if you can figure out what result will be produced when the value in the Sending item is moved to the editied picture in the Receiving item.

Sending item		Receiving item	
Picture	Data	Picture	Result
PIC S999	-123	PIC -999	Click arrow for the answer ⌄
PIC S999	-123	PIC 999-	Click arrow for the answer ⌄
PIC S999	+123	PIC -999	Click arrow for the answer ⌄
PIC S9(5)	+12345	PIC +9(5)	Click arrow for the answer ⌄
PIC S9(3)	-123	PIC +9(3)	Click arrow for the answer ⌄
PIC S9(3)	-123	PIC 999+	Click arrow for the answer ⌄
PIC S9(4)	+1234	PIC 9(4)CR	Click arrow for the answer ⌄
PIC S9(4)	-1234	PIC 9(4)CR	Click arrow for the answer ⌄
PIC S9(4)	+1234	PIC 9(4)DB	Click arrow for the answer ⌄
PIC S9(4)	-1234	PIC 9(4)DB	Click arrow for the answer ⌄
PIC 9(4)	1234	PIC $99999	Click arrow for the answer ⌄
PIC 9(4)	0000	PIC $ZZZZZ	Click arrow for the answer ⌄

Floating Insertion

The problem with using the fixed insertion symbols is that they can be somewhat unsightly. Values like $0045,345.56 or -0012 are more acceptably presented as $45,345.56 and -12.

What makes these formats more presentable is that the leading zeros have been suppressed and the editing symbol has been "floated" up against the first non-zero digit. In COBOL this is achieved using Floating Insertion.

Floating Insertion suppresses leading zeros, and "floats" the insertion symbol up against the first non-zero digit.

The Floating Insertion symbols are;

- The plus and minus signs
- and the currency symbol.

Every floating symbol counts toward the size of the printed item.

Except for the left-most one, which is always printed, each Floating Insertion symbol is a placeholder that may be replaced by a digit. Accordingly, there will always be at least one symbol printed, even though this may be at the cost of truncating the number (see the fourth row in the example below.)

Floating Insertion examples

Like the previous examples/questions above, see if you can figure out what result will be produced when the value in the Sending item is moved to the editied picture in the Receiving item.

Sending item		Receiving item	
Picture	Data	Picture	Result

PIC 9(4)	0000	PIC $$,$$9.99	Click arrow for the answer ⌄
PIC 9(4)	0080	PIC $$,$$9.00	Click arrow for the answer ⌄
PIC 9(4)	0128	PIC $$,$$9.99	Click arrow for the answer ⌄
PIC 9(5)	57397	PIC $$,$$9	Click arrow for the answer ⌄
PIC S9(4)	-0005	PIC ++++9	Click arrow for the answer ⌄
PIC S9(4)	+0080	PIC ++++9	Click arrow for the answer ⌄
PIC S9(4)	-0080	PIC - - - - 9	Click arrow for the answer ⌄
PIC S9(5)	+71234	PIC - - - - 9	Click arrow for the answer ⌄

To top of page

Suppression and Replacement Editing

Introduction

Suppression and replacement editing is used to remove leading zeroes from the value to be edited. There are two varieties of suppression and replacement editing-

- Suppression of leading zeros and replacement with spaces
- Suppression of leading zeros and replacement with asterisks

Notes
The characters **Z** and ***** are the suppression symbols.

Using Z in an editing picture, instructs the computer to suppress a leading zero in that character position and replace it with a space.

Using an * in an editing picture, instructs the computer to suppress a leading zero in that character position and replace it with an *.

If all the character positions in a data item are Z editing symbols and the sending item is 0 then only spaces will be printed.

If a Z or * is used, the picture clause symbol 9, cannot appear to the left of it.

Suppression and Replacement editing examples

Sending item		Receiving item	
Picture	**Data**	**Picture**	**Result**
PIC 9(5)	12345	PIC ZZ,999	Click arrow for the answer ⌄
PIC 9(5)	01234	PIC ZZ,999	Click arrow for the answer ⌄
PIC 9(5)	00123	PIC ZZ,999	Click arrow for the answer ⌄
PIC 9(5)	00012	PIC ZZ,999	Click arrow for the answer ⌄
PIC 9(5)	05678	PIC **,**9	Click arrow for the answer ⌄
PIC 9(5)	00567	PIC **,**9	Click arrow for the answer ⌄
PIC 9(5)	00000	PIC **,***	Click arrow for the answer ⌄
PIC 9(5)V99	00043.45	PIC $**,**9.99	Click arrow for the answer ⌄

Picture string restrictions

Some combinations of picture symbols are not permitted. The table below shows the combination of symbols that is allowed.

Character	May be followed by
P	P B 0 / , + - CR DB 9 V
B	P B 0 / , . + - CR DB 9 V
0	P B 0 / , . + - CR DB 9 V
/	P B 0 / , . + - CR DB 9 V
,	P B 0 / , . + - CR DB 9 V
.	B 0 / , . + - CR DB 9
+	P B 0 / , . + $ 9 V
-	P B 0 / , . - $ 9 V
CR or DB	Nothing at all
$	P B 0 / , . + - CR DB $ 9 V
9	P B 0 / , . + - CR DB 9 V
V	B 0 / , + - CR DB 9

To top of page

Copyright Notice

These COBOL course materials are the copyright property of Michael Coughlan.

(c) Michael Coughlan

Last updated : March 1999
e-mail : CSISwebeditor@ul.ie