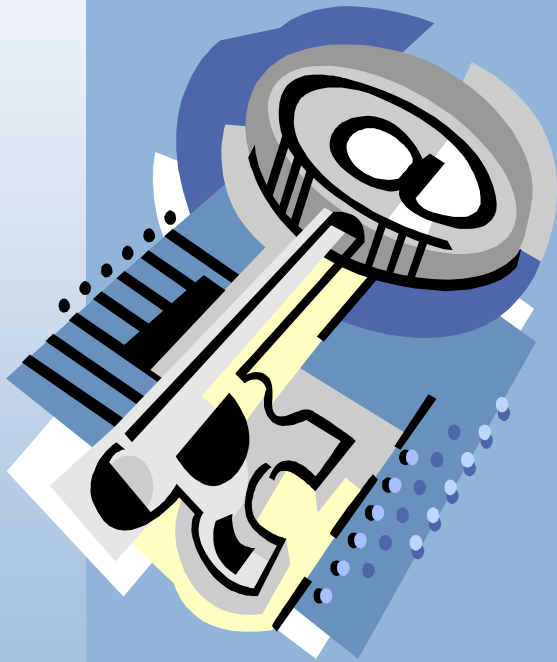


COBOL for the 21st Century

11th edition

John Wiley & Sons, Inc.



Chapter 6

Moving Data, Printing Information, and Displaying Output Interactively

Chapter Objectives

To familiarize you with

- Options of MOVE statement
- Rules for moving fields and literals
- Printing edit symbols
- Designing and printing reports

MOVE Statement

FORMAT 1

MOVE identifier-1 TO identifier-2

- Copies contents of identifier-1 To identifier-2
- identifier-1 is sending field
- identifier-2 is receiving field

MOVE Statement Example

Example

Move Qty-In To Qty-Out

- Contents of Qty-Out replaced by contents of Qty-In
- If Qty-In = 253, contents of Qty-Out = 253 after MOVE executed
- Qty-In remains unchanged

MOVE Statement

FORMAT 2

MOVE literal-1 TO identifier-2

- Value of literal-1 copied to identifier-2
- Data type of literal and identifier should match
 - Move numeric literals to numeric fields
 - Move nonnumeric literals to alphanumeric fields

MOVE Statement Examples

Examples

05 Qty-Out Pic 999.

05 Color-Out Pic X(4).

Move 100 To Qty-Out

Move 'Blue' To Color-Out

- Contents of Qty-Out replaced with 100
- Contents of Color-Out replaced with 'Blue'

Multiple Moves

Full Format

MOVE { identifier-1 } TO identifier-2 ...
 { literal-1 }

Move 0 To Total-1, Total-2, Total-3

- Move may have multiple receiving fields
- Contents of all three total fields set to 0

Moving Figurative Constants

- ZEROS may be moved to numeric or alphanumeric field
 - Moves 0 to every position of receiving field
- SPACES moved only to alphanumeric field
 - Moves space or blank to every position of receiving field

Numeric MOVE

- Numeric field or literal is moved to numeric receiving field
- When PIC clauses of both fields identical, sending and receiving fields same after MOVE
- Previous contents of receiving field are replaced by MOVE

Numeric MOVE Rules

- Decimal alignment always maintained
 - If 86.52 is moved to a field
 - 86 always treated as integer part of number
 - .52 always treated as decimal part of number
- Digits in integer part moved right to left starting at decimal point
- Digits in decimal part moved left to right starting at decimal point

Numeric MOVE Example

- Operation: Move Amt-1 To Amt-2

<u>Amt-1</u>		<u>Amt-2</u>	
Picture	99V99	Picture	99V99
Contents	12,34	Contents	67,89

- Digits in integer part moved so that
 - 2 replaces 7, 1 replaces 6
- Digits in decimal part moved so that
 - 3 replaces 8, 4 replaces 9

Numeric MOVE Example

- Operation: Move Amt-1 To Amt-2

<u>Amt-1</u>	<u>Amt-2</u>
Picture 99V99	Picture 99V99
Contents 12^34	Contents 67^89

- After MOVE, contents of Amt-2 = 12^34

Numeric MOVE Rules

- If receiving field has more integer positions than sending field
 - Unfilled high-order (leftmost) integer positions filled with zeros
- If receiving field has more decimal positions than sending field
 - Unfilled low-order (rightmost) decimal positions filled with zeros

Numeric MOVE Example

- Operation: Move Amt-3 To Amt-4

Amt-3
Picture 9V9
Contents 3_^4

Amt-4
Picture 99V99
Contents 56_^78

- Digits in integer part moved so that
 - 3 replaces 6, 0 replaces 5
- Digits in decimal part moved so that
 - 4 replaces 7, 0 replaces 8

Numeric MOVE Example

- Operation: Move Amt-3 To Amt-4

Amt-3
Picture 9V9
Contents 3_^4

Amt-4
Picture 99V99
Contents 56_^78

- After MOVE, contents of Amt-4 = 03_^40

Numeric MOVE Rules

- If receiving field has fewer integer positions than sending field
 - High-order (leftmost) digits truncated
- If receiving field has fewer decimal positions than sending field
 - Low-order (rightmost) digits truncated

Numeric MOVE Example

- Operation: Move Amt-5 To Amt-6

<u>Amt-5</u>		<u>Amt-6</u>	
Picture	99V99	Picture	9V9
Contents	12 _^ 34	Contents	5 _^ 6

- Digits in integer part moved so that
2 replaces 5, 1 not moved
- Digits in decimal part moved so that
3 replaces 6, 4 not moved

Numeric MOVE Example

- Operation: Move Amt-5 To Amt-6

<u>Amt-5</u>		<u>Amt-6</u>	
Picture	99V99	Picture	9V9
Contents	12 _^ 34	Contents	5 _^ 6

- After MOVE, contents of Amt-6 = 2_^3

Nonnumeric MOVE

- Sending field is alphanumeric field or nonnumeric literal
- Receiving field is alphanumeric field

Rules for Nonnumeric MOVE

- Characters moved from left to right
- If receiving field longer, low-order (rightmost) positions replaced with spaces
- If receiving field shorter, low-order characters in sending field truncated

Nonnumeric MOVE Example

- Operation: Move Code-1 To Code-2

<u>Code-1</u>	<u>Code-2</u>
Picture X(3)	Picture X(6)
Contents abc	Contents mnopqr

- Characters abc replace mno
- Remaining three positions in Code-2 replaced with blanks

Nonnumeric MOVE Example

- Operation: Move Code-3 To Code-4

<u>Code-3</u>		<u>Code-4</u>	
Picture	X(5)	Picture	X(3)
Contents	vwx yz	Contents	efg

- Characters vwx replace efg
- y and z are not moved since all positions in Code-4 are filled

Group Moves

- When receiving field is group item, alphanumeric MOVE rules followed
- If subfields are numeric, invalid data may be stored in subfields

Group Move Example

- Operation: Move "ABCD" To WS-Code 01 WS-Code.
05 WS-Part-1 Pic XX.
05 WS-Part-2 Pic 99.
- After MOVE, value of WS-Part-1 is AB, WS-Part-2 is CD
- Causes error if WS-Part-2 then used in arithmetic operation

MOVE Operations

- Avoid using sending field with different data type than receiving field
- Move numeric fields, numeric literals or ZEROS to numeric fields
- Move alphanumeric fields, nonnumeric literals, ZEROS or SPACES to alphanumeric fields

Disk Files vs Output for Users

- Conciseness and efficiency important for data stored in disk files
- Clarity and readability important for output read by people
 - Displayed on screen
 - Printed in reports

Output for Users

- Edit symbols used to make data fields more readable
 - Display 001495 as \$14.95
- Use spacing between lines and page margins for readability
- Space fields across page or screen
- Heading and footing lines added to improve clarity of output

Editing Functions

- Report-item - elementary item that contains edit symbols in its PIC clause
- Editing occurs when elementary item moved to report-item
- PIC clause of report-item determines editing performed

Decimal Points

- Implied decimal points (V) do not print
- Move field with V in Picture to report-item field with actual decimal point for printing or displaying

Decimal Points

- Operation: Move In-Amt To Out-Amt

Field	Picture	Contents	Positions
In-Amt	99V99	12^34	4
Out-Amt	99.99	12.34	5

- Actual decimal point
 - Replaces implied one
 - Uses a storage position

Suppressing Leading Zeros

- Zeros in leftmost position of number usually suppressed
 - Value of number not affected
 - 00025 and 25 have same value
 - Improves readability
- Edit symbol Z in PIC clause suppresses leading zeros, replaces with blanks

Suppressing Leading Zeros

- Use Z for each position where leading zero should be replaced by blank

Examples

In-Qty
Pic 999
018
003
100
000

Out-Qty
Pic ZZZ
ø18
øø3
100
øøø

{ 100 has no leading
zeros so zeros not
suppressed

ø denotes a blank (space)

Suppressing Leading Zeros

- To suppress only some leading zeros
 - Use Z's and 9's in same report-item
- Z's must precede 9's

Examples

In-Qty
Pic 999
003
000

Out-Qty
Pic ZZ9
~~00~~3
~~00~~0

Zero Suppression and Decimals

- Zeros to right of decimal point are significant so they are not suppressed

Examples

In-Cost
Pic 99V99

05^08

00^02

00^00

Out-Cost
Pic ZZ.99

~~0~~5.08

~~00~~.02

~~00~~.00

Dollar Signs and Commas

- \$ and , often used in PICTURE clause with Z and decimal point
- Comma suppressed if only zeros precede it

Examples

<u>In-Cost</u>	<u>Out-Cost</u>
Pic 9(4)V99	Pic \$Z,ZZZ.99
1027^40	\$1,027.40
0051^06	\$ 000 51.06
0000^04	\$ 0000 .04

Asterisks

- Check protection symbol
- Used in place of Z to replace leading zeros with * instead of space
- Prevents blanks from being filled in with numbers to change amount of check
 - \$ 51.06 could be changed to \$3,951.06
 - \$***51.06 not easily changed

Asterisks

Examples

WS-Chk-Amt

Pic 9(4)V99

2345_15

0072_08

0000_06

Out-Chk-Amt

Pic \$*,***.99

\$2,345.15

\$***72.08

\$*****.06

Plus or Minus Signs

- PIC clause with 9's is unsigned field
 - Value assumed to be positive
- Use S with 9's in PIC clause of sending field to define signed field
 - Use PIC of S9(3) to store -425
- Report-item field must include + or - editing symbol to print or display a sign

Displaying Minus or Plus Sign

- Use - (minus) symbol to display minus sign if sending field is negative
 - Displays - for negative values
 - Displays blank (no sign) for positive or unsigned values
- Use + (plus) symbol to display sign for both positive and negative values
 - Displays - for negative values
 - Displays + for positive or unsigned values

Displaying Minus or Plus Sign

Examples

Assume a PICTURE of S9(3) for WS-Amt

<u>WS-Amt</u>	<u>Out-Amt</u> <u>PICTURE</u>	<u>Contents</u>
007 ⁻	-ZZZ	- 00 7
218 ⁺	ZZZ-	218 0
082 ⁺	+ZZ9	+ 0 82
030 ⁻	\$ZZZ+	\$ 0 30-

Debit and Credit Symbols

- For accounting applications replace - (minus) sign with
 - DB to debit an account when negative
 - CR to credit an account when negative
- Always specified to right of report-item
- Print only when value moved to report item is negative

Debit and Credit Symbols

Examples

Assume a PICTURE of S9(3) for WS-Amt

<u>WS-Amt</u>	<u>Out-Amt</u> <u>PICTURE</u>	<u>Contents</u>
123 ⁻	999CR	123CR
123 ⁻	999DB	123DB
123 ⁺	999CR	123 CR
123 ⁺	999DB	123 DB

Spaces, Zeros, or Slashes

- B (space or blank), 0 (zero) or / (slash) may be inserted in report-item PIC as separator
- Used to edit either nonnumeric or numeric fields

Spaces, Zeros, or Slashes

Examples

Assume

PIC 9(3) for WS-Amt

PIC X(3) for WS-Name

	<u>Contents</u>	<u>PICTURE</u>	<u>Report-Item</u> <u>Contents</u>
WS-Amt	528	99BB9	528
WS-Amt	218	999,000	218,000
WS-Name	RAY	XBXBX	RAY
WS-Name	JAN	X/X/X	J/A/N

Editing and Arithmetic Operations

- Editing performed when
 - Sending field moved to report-item
 - Result of arithmetic operations stored in report-item
- PICTURE of report-item determines type of editing

Editing and Arithmetic Operations

- All items used in arithmetic operation must be numeric
- Error if report-item used in calculation of result
- This statement invalid if Total-Out is report-item
 Add WS-Total To Total-Out
- Report-item may be used after GIVING

Editing with Floating Strings

- Used to place \$, + or - directly preceding first significant digit
- For example, 000516 becomes \$5.16 instead of \$ 5.16
- In edit string, replace all Z's with floating character (\$, + or -)
- Suppresses leading zeros, commas

Editing with Floating Strings

Examples

Assume a PICTURE of S9(4)V99 for WS-Amt

<u>WS-Amt</u>	<u>Out-Amt</u>	<u>Contents</u>
0002 _^ 29 ⁻	\$\$,\$\$\$\$.99	bbb \$2.29-
0036 _^ 62 ⁺	++,+++\$.99	bbb +36.62
0007 _^ 05 ⁻	---99.99	bb -07.05

BLANK WHEN ZERO Option

- To print spaces when sending field is all zeros rather than \$.00 or -0
- For the report-item

05 Qty-Out Pic +++
Blank When Zero.

035 will print as +35

000 will print as ~~000~~ (all spaces)

Defining Print Records

- Define each type of output line as separate 01-level record in WORKING-STORAGE
- May include 01-level records for heading, detail, total, footing lines, etc.
- Establishes separate storage area for each record
- All constants and blanks may be preassigned with VALUE clauses

WRITE ... FROM Statement

- To transfer data from storage to print area and then print
- Replaces MOVE and WRITE

Write Print-Rec From Heading-1

instead of

Move Heading-1 to Print-Rec
Write Print-Rec

ADVANCING Option

- Single spacing for WRITE is default
- Use AFTER or BEFORE ADVANCING to obtain any number of blank lines between each print line

WRITE Statement

FORMAT

WRITE record-name-1 [FROM identifier-1]
{ AFTER } ADVANCING { integer-1 } { LINE }
{ BEFORE } { identifier-2 } { LINES }

- integer-1 or identifier-2 must be non-negative integer value
- AFTER ADVANCING prints line after paper is spaced
- BEFORE ADVANCING prints line before spacing occurs

WRITE ... AFTER ADVANCING

- Write Print-Rec From Heading-Line-1 After Advancing 2 Lines
 - Causes paper to advance two lines
 - Heading-Line-1 printed on second line
 - Creates double spacing - one blank line followed by printed line

WRITE ... BEFORE ADVANCING

- Write Print-Rec From Heading-Line-1 Before Advancing 2 Lines
 - Heading-Line-1 printed first, then spaces two lines
 - Results in overprinting if first line already contains printed output
 - As a rule, use either BEFORE or AFTER in program but not both

PAGE option

- Word PAGE after ADVANCING causes paper to advance to next page
- To advance paper to top of new page and print a heading, code

Write Print-Rec From Heading-Line-1
After Advancing Page

End-of-Page Control

- When page is full, need to advance to next page and print headings again
- Define `WORKING-STORAGE` field for line counter
- Initialize line counter to zero and increment it each time a line is printed
- When line counter equals or exceeds line limit for page, advance to next page

Printing Page Numbers

- Define `WORKING-STORAGE` field for page number
- Initialize and then increment page number each time new page begins
- Move page number to report-item in heading line before printing it at top of a new page

Date with Four-Digit Year

- Newer compilers provide access to full four-digit year
- Use intrinsic function CURRENT-DATE with MOVE in place of ACCEPT
 - Change Run-Year in WS-Date to Pic 9(4)

Move Function Current-Date To WS-Date

- If date is January 25, 2003 then 2003 will be stored in Run-Year

Report Design Guidelines

1. Include heading to identify report
2. Include date, page number
3. Include column headings to identify fields printed
4. Place most significant fields where they are most visible
5. Edit numeric fields for readability

Report Design Guidelines

6. Include totals at end of page or report
7. Use *'s to identify level of total

EXAMPLE

Dept Total is \$33,266.25*

Final total is \$167,267.53**

8. Include page footings at end of each page, report footings at end of report

Chapter Summary

- Numeric MOVE Rules
 - Decimal alignment maintained
 - Integer and decimal portion digits moved from decimal point out
 - Nonfilled high-, low-order positions replaced with zeros
 - Truncation of high-, low-order digits occurs if receiving field not large enough

Chapter Summary

- Nonnumeric MOVE Rules
 - Movement from left to right
 - Low-order nonfilled positions replaced with spaces
 - Low-order characters truncated if receiving field not large enough

Chapter Summary

- Format of receiving field determines type of MOVE operation
- See Table 6.3 in text for summary of edit symbols

Chapter Summary

- Rules for Printing Output
 - Use Printer Spacing Chart to determine print positions to be used
 - Define all printed output in `WORKING-STORAGE` so `VALUE` clause can be used
 - Use appropriate editing symbols in report-items

Chapter Summary

- WRITE statement clauses
 - Move records to print area in FILE SECTION with WRITE ... FROM
 - Use BEFORE or AFTER ADVANCING to control spacing between lines
 - Use AFTER ADVANCING PAGE to go to beginning of next page