PRINCE SOFTWARE

Translate/RW

If you have programs containing Report Writer features, Translate/RW can convert this code to native COBOL. Designed for efficiency, accuracy and simplicity, Translate/RW automatically translates the Report Writer features of your old COBOL programs — which the newer versions of COBOL do not recognize — with code that they do. Your programs are made compatible with the latest COBOL versions without hours of manual translation or the risk of human error.

Two Ways to Use Translate/RW

You can use Translate/RW in either of two ways, whichever best meets your needs:

- To translate your existing COBOL Report Writer programs to native COBOL, easing the upgrade to COBOL for MVS or COBOL for OS/390.
- As a pre-compiler, so you can continue using COBOL Report Writer in your new COBOL environment. (In this capacity, Translate/RW is invoked by a JCL procedure, which executes both Translate/RW and the new COBOL compiler.)

No Run-time Support Libraries Needed

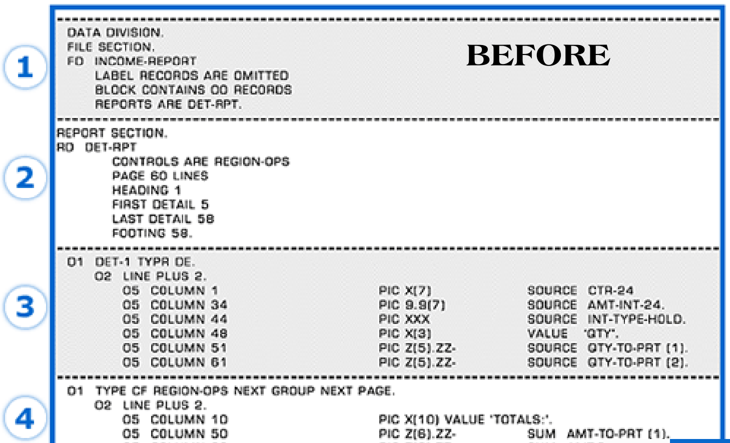
Translate/RW converts all Report Writer code in a simple, one-step procedure. It requires no additional calls to assembler modules and no special run-time library — so you save effort, time and money.

The program generated by Translate/RW is independent, stand-alone, native COBOL.

The resulting code is structured, affording readability and ease of maintenance.

On the following page are typical Report Writer features and the corresponding COBOL code that would result from translation.

Translate/RW



```
DATA DIVISION.
FILE SECTION.
FD  INCOME-REPORT
    LABEL RECORDS ARE OMITTED
    BLOCK CONTAINS 00 RECORDS
    REPORTS ARE DET-RPT.
REPORT SECTION.
RD  DET-RPT
        CONTROLS ARE REGION-OPS
        PAGE 60 LINES
        HEADING 1
        FIRST DETAIL 5
        LAST DETAIL 58
        FOOTING 58.
 01  DET-1 TYPR DE.
     02  LINE PLUS 2.
         05  COLUMN 1          PIC X(7)          SOURCE  CTR-24
         05  COLUMN 34         PIC 9.9(7)        SOURCE  AMT-INT-24.
         05  COLUMN 44         PIC XXX           SOURCE  INT-TYPE-HOLD.
         05  COLUMN 48         PIC X(3)          VALUE   'QTY'.
         05  COLUMN 51         PIC Z(5).ZZ-      SOURCE  QTY-TO-PRT (1).
         05  COLUMN 61         PIC Z(5).ZZ-      SOURCE  QTY-TO-PRT (2).
 01  TYPE CF REGION-OPS NEXT GROUP NEXT PAGE.
     02  LINE PLUS 2.
         05  COLUMN 10         PIC X(10) VALUE 'TOTALS:'.
         05  COLUMN 50         PIC Z(6).ZZ-      SUM  AMT-TO-PRT (1).
         05  COLUMN 60         PIC Z(6).ZZ-      SUM  AMT-T
 PROCEDURE DIVISION.
      INITIATE DET-RPT.
      PERFORM REPORT-BODY.
 REPORT-BODY.
      PERFORM READ-OPS THRU END-READ-OPS
      GENERATE DET-1.
```

1 The REPORTS ARE clause is removed from report FDs and a record declaration of the correct size is created.

3a A WORKING-STORAGE record is generated for each report line. VALUE items are represented by FILLERs with the VALUE clause; SOURCE and SUM items are represented by fields with the correct PICTURE clause.

4a WORKING-STORAGE areas are created to hold the totals for SUM counters.

5a INITIATE and TERMINATE statements in the PROCEDURE DIVISION are changed to PERFORM statements.

6a GENERATE statements in the PROCEDURE DIVISION are changed to perform statements.

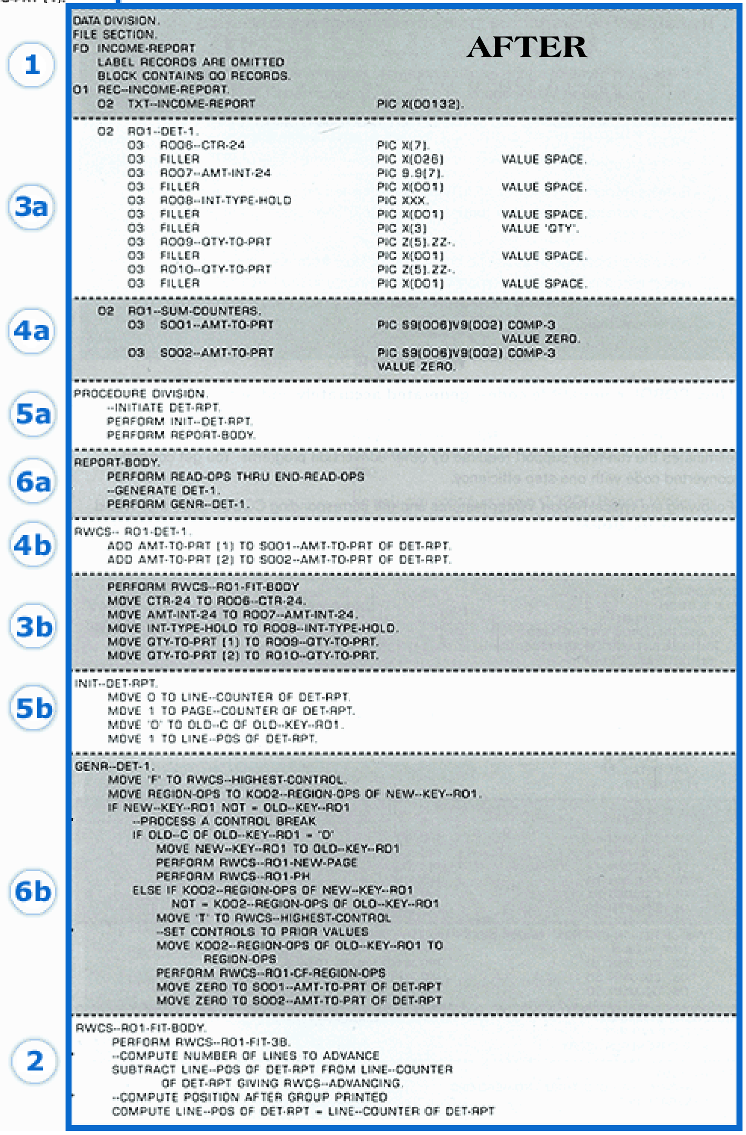
4b Logic is generated to increment SUM counters at the appropriate times.

3b Logic is generated to fill in the SOURCE and SUM fields in the report lines prior to printing the report file.

5b The generated procedures for INITIATE and TERMINATE handle the housekeeping for the beginning and the end of a report.

6b The generated procedure for a GENERATE statement checks for control breaks before presenting DETAIL groups. When a control break is found, the proper CONTROL FOOTING and CONTROL HEADING groups are presented.

2 Logic is generated to place each report group in the correct position on a page.



```
DATA DIVISION.
FILE SECTION.
FD  INCOME-REPORT
    LABEL RECORDS ARE OMITTED
    BLOCK CONTAINS 00 RECORDS.
01  REC--INCOME-REPORT.
    02  TXT--INCOME-REPORT              PIC X(00132).
    02  RO1--DET-1.
        03  ROO6--CTR-24                PIC X(7).
        03  FILLER                      PIC X(026)        VALUE SPACE.
        03  ROO7--AMT-INT-24            PIC 9.9(7).
        03  FILLER                      PIC X(001)        VALUE SPACE.
        03  ROO8--INT-TYPE-HOLD         PIC XXX.
        03  FILLER                      PIC X(001)        VALUE SPACE.
        03  FILLER                      PIC X(3)          VALUE 'QTY'.
        03  ROO9--QTY-TO-PRT            PIC Z(5).ZZ-.
        03  FILLER                      PIC X(001)        VALUE SPACE.
        03  RO10--QTY-TO-PRT            PIC Z(5).ZZ-.
        03  FILLER                      PIC X(001)        VALUE SPACE.
    02  RO1--SUM-COUNTERS.
        03  SOO1--AMT-TO-PRT            PIC S9(006)V9(002) COMP-3
                                                          VALUE ZERO.
        03  SOO2--AMT-TO-PRT            PIC S9(006)V9(002) COMP-3
                                        VALUE ZERO.
PROCEDURE DIVISION.
     --INITIATE DET-RPT.
     PERFORM INIT--DET-RPT.
     PERFORM REPORT-BODY.
REPORT-BODY.
     PERFORM READ-OPS THRU END-READ-OPS
     --GENERATE DET-1.
     PERFORM GENR--DET-1.
RWCS-- RO1-DET-1.
     ADD AMT-TO-PRT (1) TO SOO1--AMT-TO-PRT OF DET-RPT.
     ADD AMT-TO-PRT (2) TO SOO2--AMT-TO-PRT OF DET-RPT.
     PERFORM RWCS--RO1-FIT-BODY
     MOVE CTR-24 TO ROO6--CTR-24.
     MOVE AMT-INT-24 TO ROO7--AMT-INT-24.
     MOVE INT-TYPE-HOLD TO ROO8--INT-TYPE-HOLD.
     MOVE QTY-TO-PRT (1) TO ROO9--QTY-TO-PRT.
     MOVE QTY-TO-PRT (2) TO RO10--QTY-TO-PRT.
INIT--DET-RPT.
     MOVE O TO LINE--COUNTER OF DET-RPT.
     MOVE 1 TO PAGE--COUNTER OF DET-RPT.
     MOVE 'O' TO OLD--C OF OLD--KEY--RO1.
     MOVE 1 TO LINE--POS OF DET-RPT.
GENR--DET-1.
     MOVE 'F' TO RWCS--HIGHEST-CONTROL.
     MOVE REGION-OPS TO KOO2--REGION-OPS OF NEW--KEY--RO1.
     IF NEW--KEY--RO1 NOT = OLD--KEY--RO1
        --PROCESS A CONTROL BREAK
        IF OLD--C OF OLD--KEY--RO1 = 'O'
           MOVE NEW--KEY--RO1 TO OLD--KEY--RO1
           PERFORM RWCS--RO1-NEW-PAGE
           PERFORM RWCS--RO1-PH
        ELSE IF KOO2--REGION-OPS OF NEW--KEY--RO1
              NOT = KOO2--REGION-OPS OF OLD--KEY--RO1
           MOVE 'T' TO RWCS--HIGHEST-CONTROL
           --SET CONTROLS TO PRIOR VALUES
           MOVE KOO2--REGION-OPS OF OLD--KEY--RO1 TO
                 REGION-OPS
           PERFORM RWCS--RO1-CF-REGION-OPS
           MOVE ZERO TO SOO1--AMT-TO-PRT OF DET-RPT
           MOVE ZERO TO SOO2--AMT-TO-PRT OF DET-RPT
RWCS--RO1-FIT-BODY.
     PERFORM RWCS--RO1-FIT-3B.
     --COMPUTE NUMBER OF LINES TO ADVANCE
     SUBTRACT LINE--POS OF DET-RPT FROM LINE--COUNTER
            OF DET-RPT GIVING RWCS--ADVANCING.
     --COMPUTE POSITION AFTER GROUP PRINTED
     COMPUTE LINE--POS OF DET-RPT = LINE--COUNTER OF DET-RPT
```