Preserving Mainframe Focus

Data Correlation and Documentation

DCD IV Quick Start Guide

03-15-2015

Release 4.2

Marble Computer, Inc.
6416 Via De Albur Court
Suite 100
El Paso, TX 79912
Tel: (800) 252-1400
Fax: (915) 845-7918
support@marblecomputer.com
www.marblecomputer.com

DCD IV provides fast ANALYSIS on COBOL Programs

The following are nine prime examples of how DCD IV can make a major difference in the maintenance of COBOL programs.

Analysis on a Single COBOL program

1. A Cobol Narrative alongside each data field in the Data Division
2. An Alternate Compile Listing that eliminates most back and forth tracing
3. Unused Procedure Division Code
4. Unused Data Division Records and Data Names

Analysis on Multiple Programs

5. COPY Member Analysis
6. CALL Program Analysis
7. One Record Analysis across multiple programs
8. Matching One Record Analysis across multiple programs

JCL Analysis

9. Different JCL reports on DSNAME Analysis, DDNAME Analysis and more

1. A Cobol Narrative alongside each data field in the Data Division

From Data Division of Alternate Compile Listing:

From Alternate Compiler Data Division:

Seq #	Data Field	Narrative

```
777  05  MR-ACCT-NBR      >  IF TR-ACCT-NBR NOT = # (1651,1744)
                              MOVE # TO TR-ACCT-NBR @487 (1342,1442)
                              MOVE '42997' TO # (1163)
                              MOVE TR-ACCT-NBR @487 TO # (1772)
                              MOVE ZEROS TO # (1166,1282)
```

Notes: When **#** is used in narrative it refers to the name (**Data Field**) shown on the left (**MR-ACCT-NBR**).
The numbers in parenthesis or after an @ character refer to compiler numbers (See **Seq #** above).

Finding out how one field is used throughout the COBOL program.

Without DCD IV, a programmer needs to view the COBOL cross reference map and find the field **MR-ACCT-NBR** in the Cross-reference of data names as follows:
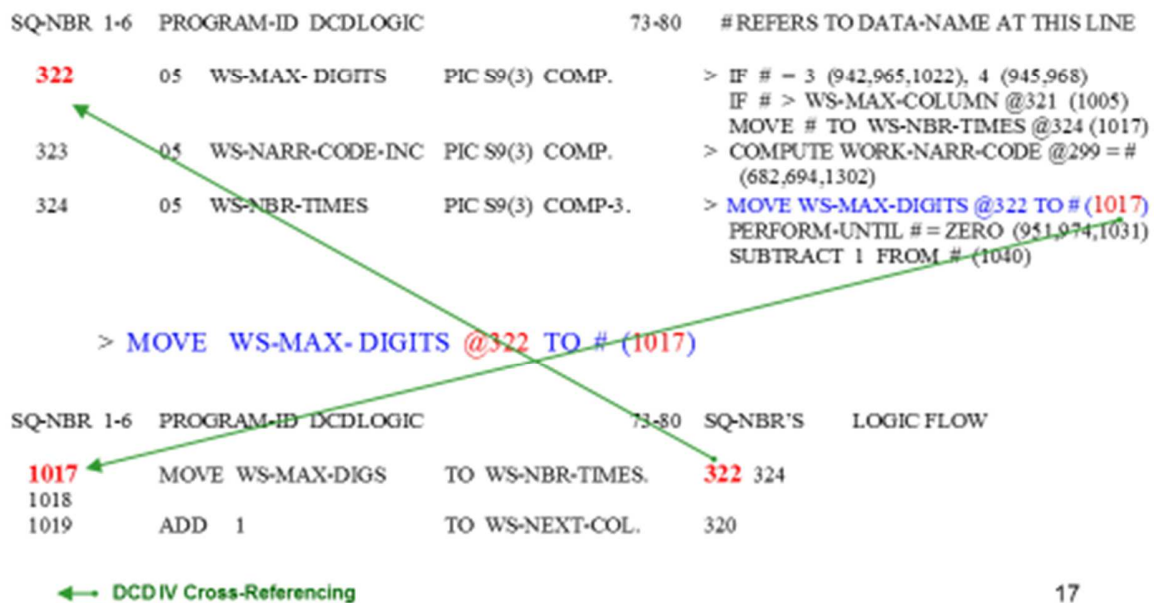
```
777  MR-ACCT-NBR. . . . . . . M1163 M1166 M1282 1342 1442 1651 1744 1772
```

Then you can visit each of the eight (8) references for **MR-ACCT-NBR** in the Procedure Division and develop mentally the same **Narrative** developed above **AND** then either remember it or develop it in written format as shown above.

The results are always the same. The time frame in using **DCD IV** is about 10 seconds by visiting this name in the Data Division of the Alternate Compile Listing. The time frame for not using **DCD IV** is at a minimum of about 5 or 6 minutes after you visit all 8 references and jot down some way of remembering what is happening and then analyzing the output.

2. An Alternate Compile Listing that eliminates most back and forth tracing.

Alternate Compile Listing

Tracing between DIVISIONs

```
SQ-NBR 1-6   PROGRAM-ID DCDLOGIC              73-80    # REFERS TO DATA-NAME AT THIS LINE

  322      05   WS-MAX- DIGITS    PIC S9(3) COMP.      >  IF # = 3 (942,965,1022), 4 (945,968)
                                                          IF # > WS-MAX-COLUMN @321 (1005)
                                                          MOVE # TO WS-NBR-TIMES @324 (1017)
  323      05   WS-NARR-CODE-INC  PIC S9(3) COMP.      >  COMPUTE WORK-NARR-CODE @299 = #
                                                            (682,694,1302)
  324      05   WS-NBR-TIMES      PIC S9(3) COMP-3.    >  MOVE WS-MAX-DIGITS @322 TO # (1017)
                                                          PERFORM-UNTIL # = ZERO (951,974,1031)
                                                          SUBTRACT 1 FROM # (1040)

           >  MOVE  WS-MAX- DIGITS  @322  TO  #  (1017)

SQ-NBR 1-6   PROGRAM-ID DCDLOGIC              73-80   SQ-NBR'S        LOGIC FLOW

  1017      MOVE WS-MAX-DIGS      TO WS-NBR-TIMES.     322  324
  1018
  1019      ADD  1               TO WS-NEXT-COL.       320
```

◄─► DCD IV Cross-Referencing   17

When viewing the Cobol program in ISPF one of the most constructive activities done is to spend time by viewing the logic from one place at a time in the Procedure Division, and then as needed gathering information about data-name usage and how it is used at other places, to know where to visit next in the Procedure Division.

The way Alternate Compile Listing is presented makes it easy to support this back and forth tracing.

Taking a moment to follow the above tracing image gives a better understanding of what using the 'Alternate Compile Listing' can do for your program analysis.

TIME SAVINGS

Without Alternate Compile Listing

- Look at one line of the Procedure Division
- Find wanted name in Cross Reference Map
- Make a list of all REFERENCES for this name
- Visit all references except the line just done
- Remember or write down what happens

Approximate Time: 5-10 minutes

With Alternate Compile Listing

- Look at one line of the Procedure Division
- Use number on right to find Data-name with Narrative

Approximate Time: 10 seconds

19

3. Unused Procedure Division Code

```
Unused Procedure Names
                              Unused  SECTIONS
  Seq#   Section/Paragraph Name         Message or Warning
  574  H-ROUTINE-TO-TEST-BACKUP         SECTION not externally referenced
                              Unused  PARAGRAPHS
  Seq#   Section/Paragraph Name         Message or Warning
  575  H010-ROUTINE-TO-SHOW-CURRENT   PARAGRAPH not externally referenced
  581  H999-EXIT                       PARAGRAPH not externally referenced
```

Finding DEAD Procedure Division code is available to help clean up and make a COBOL program easier to follow.

4. Unused Data Division Records and Data Names

Unused Program Parts

```
10/11/2008 07:46    Unreferenced PARAGRAPHs and SECTIONs for ANAR        Page  28

   There are no unreferenced PARAGRAPHs or SECTIONs


10/11/2008         Unused 01 Records and 77 Entries for Prog: ANAR        PAGE  29

   Prog#      Edit#     Lvl      Field Name

   00092      00092      01   WS-BEGINS-HERE
   00216      COPY       01   PRINT-PASSED-RECORD


10/11/2008                     Unused Data Names for Prog: ANAR           PAGE  30

   Prog#      Edit#     Lvl      Field Name

   00054****  00054 01   NR-RECORD********************** contains unused fields **
   00058      00058      10   NR-REST-OF-NARR
                              Indirectly Changed @54 Used @54                    <
   00060      00060      05   NR-COUNT-OF-INDIRECTS
                              Indirectly Changed @54 Used @54                    <

   00068****  00068 01   N4-RECORD********************** contains unused fields **
   00069      00069      05   N4-CMPLR-NBR
                              Indirectly Changed @68                             <
   00070      00070      05   N4-NARRATIVE                                      26
                              Indirectly Changed @68                             <
```

A listing of unused 01 records and unused data names may be used to help clean up a COBOL program.

5. COPY Member Analysis (*2 sequences available*)

COPY in MEMBER sequence

COPY ANALYSIS		MEMBER	SEQUENCE		01/01/2001	PAGE	1
MEMBER	**PROGRAM**	**SEQ-#**	**LEVEL**	**IDENTIFIER**	**SECTION**		
BANRCOPY	DCDLOGIC	0061	01	BC-BANNER-FIELDS	WORKING-STORAGE SECTION		
CORTRAN	DCDLOGIC	0110	01	COB-REC	WORKING-STORAGE SECTION		
CPYREC	DCDLOGIC	0079	**01**	**COB-REC**	WORKING-STORAGE SECTION		
HC02IF	DCDLOGIC	0134		**030-COPY-HERE**	**PROCEDURE DIVISION**		
INFOTYP7	DCDLOGIC	0012	**FD**	**FILE-1**	FILE SECTION		
TRANCPY	PAYMONTH	0024	01	TEST-COPY-REC	WORKING-STORAGE SECTION		
	PAYMONTH	0036	01	TEST-COPY-REC-2	WORKING-STORAGE SECTION		
WTRAST	DCDLOGIC	0008		SOURCE-COMPUTER	ENVIRONMENT DIVISION		

41

Following COPY members across a system of programs is made easy.

6. CALL Program Analysis   (*2 sequences available*)

CALL in PROGRAM sequence

CALL ANALYSIS	PROGRAM-ID / CALL-NAME SEQUENCE	01/01/2001	PAGE 1

PROGRAM-ID	PARAGRAPH-NAME	SEQ-#	CALL / ENTRY STATEMENT	PARAMETERS
PAYMONTH	A070-COMPUTE-DEDUCTIONS	0169	CALL "PAYDEDUC" USING	PN-RECORD PN-WEEKLY-DEDUCTION-1 PN-WEEKLY-DEDUCTION-2
		0126	PROCEDURE DIVISION ENTRY (PAYMONTH)	
PAYROLL	A010-VERIFY-EMPLOYEE-NBR	0146	CALL "EMPVERFY" USING	WS-SYSTEM-DATE PT-EMPLOYEE-NBR WS-VERIFY-CODE
	B010-COMPUTE-FED-TAX	0026	CALL "FEDTAX" USING	WS-SYSTEM-DATE PN-RECORD
	A070-COMPUTE-DEDUCTIONS	0184	CALL "PAYDEDUC" USING	PN-RECORD PN-WEEKLY-DEDUCTION-1 PN-WEEKLY-DEDUCTION-2
		0124	PROCEDURE DIVISION ENTRY (PAYROLL)	

42

Understanding how programs CALL each other is made simple.

System Record Analysis

01/01/2001 11:30	SYSTEM RECORD ANALYSIS - DATA NAME SUMMARY						PAGE 3

FROM-TO PROGRAM	01 REC NAME	SEQ-#	LV	DATA NAME	PICTURE	DIRECT FLAGS SET/USE/TEST	INDIRECT FLAGS
91-121 PAYMONTH	PN-RECORD	0087	05	PN-CURRENT-DATA		U	SI UI
PAYMONTH	PT-RECORD	0043	05	PT-CURRENT-DATA			UI
PAYROLL	PN-RECORD	0085	05	PN-CURRENT-DATA			SI UI TI
91-96 PAYMONTH	PN-RECORD	0088	10	PN-CURR-HR-RATE	PIC S9(4)V99	S	SI UI
PAYMONTH	PT-RECORD	0044	10	PT-CURR-HR-RATE	PIC S9(4)V99		UI
PAYROLL	PN-RECORD	0086	10	PN-CURR-HR-RATE	PIC S9(4)V99	T	SI UI
126-131 PAYMONTH	PN-RECORD	0098	10	PN-WEEKLY-GROSS	PIC S9(4)V99	S U T	SI UI
PAYMONTH	PT-RECORD	0054	10	PT-WEEKLY-GROSS	PIC S9(4)V99		UI
PAYROLL	PN-RECORD	0096	10	PN-WEEKLY-GROSS	PIC S9(4)V99	S U T	SI UI
132-137 PAYMONTH	PN-RECORD	0099	10	PN-WEEK-FED-TAX	PIC S9(4)V99	S	SI UI
PAYMONTH	PT-RECORD	0055	10	PT-WEEK-FED-TAX	PIC S9(4)V99		UI
PAYROLL	PN-RECORD	0097	10	PN-WEEK-FED-TAX	PIC S9(4)V99	S U	SI UI

40

Select a commonly used COPY member and/or selected records and trace these records through the whole system showing how each field is used directly and indirectly through every program at one glance.

This capability is especially helpful when enlarging the size or changing the data type of the field. **DCD IV** finds the field – code relationships regardless of the data name.

8. Matching One Record Analysis across multiple programs

DATA ANALYSIS report

```
                                SYSTEM DATA-NAME CROSS REFERENCE
PROGRAM    SECTION         SEQ-# LV DATA-NAME  NARRATIVE       01/01/2001    PAGE   05

PAYROLL    WORKING-STORAGE 0118 88 WS-EMPLOYEE-NBR-INVALID                   VALUE 'X'
                                     IF # (150)

PAYROLL    WORKING-STORAGE 0114 88 WS-END-OF-FILE                           VALUE 'Y'
                                     PERFORM-UNTIL # (137)

PAYMONTH   WORKING-STORAGE 0115 05 WS-END-OF-FILE-SW     PIC X              VALUE SPACE
                                     MOVE 'Y' TO # (135,202)

PAYROLL    WORKING-STORAGE 0113 05 WS-END-OF-FILE-SW     PIC X              VALUE SPACE
                                     MOVE 'Y' TO # (324)

PAYMONTH   WORKING-STORAGE 0110 05 WS-FICA-AMOUNT        PIC S9(4)  COMP-3  VALUE +0
                                     COMPUTE # = .0715 (173), 3131.70 (179),
                                        PN-WEEKLY-GROSS @89 (173),
                                        WS-FICA-YTD-TOTAL @109 (179)
                                     COMPUTE WS-FICA-YTD-TOTAL @111 = # (160)
                                     MOVE # TO PN-WEEKLY-FICA @89 (160)

PAYROLL    WORKING-STORAGE 0108 05 WS-FICA-AMOUNT        PIC S9(4)  COMP-3  VALUE +0
                                     COMPUTE WS-FICA-YTD-TOTAL @109 = # (175)
                                     MOVE # TO PN-WEEKLY-FICA @89 (181)
                                     MOVE ZEROS TO # (171)

                                                                            43
```

The Data Analysis report may be on every field in multiple programs or reduced to just those included within the previous System Record Analysis Report.

9. Different JCL reports on DSNAME Analysis, DDNAME Analysis and more

JCL DSNAME Report

```
DATA SET ANALYSIS ** 01/01/2001 *************************************** PAGE   1
```

DSNAME	PROGRAM	PROC	STEP-NBR	STEP-NAME	DDNAME	DISPOSITION	
&&PASSFILE	DCDJCL	DCDJCL	2	JCLRPT	PROFILE	OLD	DELETE
	MBRFETCH	DCDJCL	1	MBR	OUTSET	MOD	PASS
ALT.DSN.FOR.APPL	DCDMAIN	LIBACL	2	DCD	COBOLIN	OLD	DELETE
		PANACL	2	DCD	COBOLIN	OLD	DELETE
	LIBRARAN	LIBACL	1	LIB	OSJOB	NEW	PASS
	PAN#1	PANACL	1	PAN	PAN002	NEW	PASS
CARDIN	DCDJCL	DCDJCL	2	JCLRPT	SYSIN		
DUMMY	DCDJCL	DCDJCL	2	JCLRPT	SORTMESS		
LIBRARAN.SOURCE	LIBRARAN	LIBACL	1	LIB	MASTER	SHR	
PANVALET.SOURCE	PAN#1	PANACL	1	PAN	PAN001	SHR	
SYSOUT	DCDJCL	DCDJCL	2	JCLRPT	PRINT		
			2	JCLRPT	PRTMSG		
	DCDMAIN	COMACL	1	DCD	PRINT		

47

Various JCL reports are available to help understand a system of programs.