

DCD IIITM

Data Correlation and Documentation System

For the **IBM OS** Environment

User's Manual

DCD III

Data Correlation and Documentation System

User's Manual

TABLE OF CONTENTS

New Release Highlights.....	I
Alternate Compile Listing Facility.....	A-1
Tracing & Analysis and Other Reports	B-1
Computer Aided Software Engineering (CASE) Files.....	C-1
JCL PROC Analysis Reports Facility.....	D-1
Error Messages.....	E-1
Installation Guide.....	F-1
Index.....	G-1
SPF Panels.....	H-1

INTRODUCTION

DCD III (Data Correlation and Documentation System) is a maintenance tool for use in the ongoing maintenance of COBOL programs.

DCD III provides complete documentation of a COBOL program automatically and now provides Tracing & Analysis.

The Alternate Compile Listing Facility of DCD III provides an alternative to using the standard compile listing of the COBOL compiler.

The Tracing & Analysis and Other Reports Facility of DCD III provides both Tracing within a program AND a method of analyzing the use of data throughout the program on multiple programs at a one time.

DCD III can produce work files, which may be used in re-engineering Computer Aided Software Engineering (CASE) projects.

The JCL PROC Analysis Reports Facility of DCD III provides a method for producing reports on JCL PROCs.

DCD III supports ANSI standards for COBOL for 1962, 1968, 1974, and 1985.

DCD III is supported for the following computers:

- IBM/OS -all mainframe operating systems

- Unisys - Unisys 1100/2200 all mainframe operating systems

- IBM/PC – Windows

DCD III supports commonly used software packages that interface with COBOL such as the commonly used CA-Optimizer. See “Other Features Supported” in the Alternate Compile Listing Facility section for a full list of the features supported by DCD III.

Copyright August 1, 2008 by MARBLE Computer, Inc. All rights reserved.

The software described in this manual is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

IBM is a trademark of the International Business Machines Corporation.

CA-Optimizer is a software package marketed by Computer Associates International, Inc.

COBOL/Aid is a software package marketed by Compuware Corporation.

Panvalet is a software package marketed by Computer Associates International, Inc.

Librarian is a software package marketed by Computer Associates International, Inc.

Because of work processing constraints, examples shown in this document may not be identical to actual results obtained.

MARBLE Computer, Inc. would like to thank the many people who have contributed their assistance, suggestions, and guidance in the development of DCD III and for providing feedback for continued improvement of the software and the documentation in this manual.

DCD III has a client base of over 600 mainframe installations.

MARBLE Computer, Inc.

P.O. Box 920692

El Paso, Texas 79902

Phone: 800-252-1400 / 915-845-0963

Fax: 915-845-7918

E-Mail Address: info@marblecomputer.com

Web Site: www.marblecomputer.com

DCD III

New Release Highlights

(See the end of this section for the most current release highlights.)

New Release Highlights

Release 1.1

April 1, 1989

This release of DCD III has more changes within it than all prior releases of DCD II combined. In essence, the entire package has been completely rewritten. Listed below are a few of the major changes and the overall effect that they have on the new package.

1. The JCL it takes to execute DCD III has been simplified and reduced. For example:

25 DCD WORK FILES have been reduced to 6

10 DCD PRINT FILES have been reduced to 1

The CPU savings here in allocation and de-allocation are substantial.

2. The narrative documentation printed alongside the Data Division on the Source Listing has the most significant change of this new release. In prior releases of DCD II, this narrative was in English. The new narrative is now in COBOL, making de-bugging of the source code much faster. Further, the narrative is more concise. For example, what used to be lumped together under the phrase:

(LOGIC OPERATOR)

will now print in one of these formats:

IF # < 01 (885)

IF # = S-NEXT-NARRATIVE-CODE @309 (909)

IF # NOT > 25 (892)

IF NOT # (664)

IF # IS NUMERIC (871,879)

IF WS-MAX-PD-DIGITS @322 > # (1005)

PERFORM-UNTIL # = 0 (951,974,1013)

or in another more appropriate format.

The narrative now relates directly to the exact Procedure Division Statement.

3. The amount of paper produced by DCD III has been substantially reduced. The DCD III Source Listing and Data and Procedure Division Cross Reference Reports combined are now about the same size as a compiler listing with the same reports. Prior to this release, the DCD II listing was twice the size of the compiler listing.

New Release Highlights
Release 1.1 (Continued)
April 1, 1989

4. The Data Division and the Procedure Division Cross Reference Reports are now condensed rather than expanded. The result is a 70% savings in paper over a compiler cross reference and a 90% savings in paper over the old DCD II Expanded Cross Reference. The reason this can be done is due to the new and more complete narrative in COBOL in the Data Division.
5. The Data Analysis Report (now the System Data Name Cross Reference) is now in a condensed format which saves paper when printing this report.
6. DCD III has been separated into three distinct facilities:
 - The output from the program side, now referred to as the Alternate Compile Listing Facility, is designed to more closely resemble the COBOL compiler output.
 - Most reports previously available on the system side are now available under the Other COBOL Reports Facility.
 - The Data Set Analysis Reports formerly on the system side, have been moved to the JCL PROC Analysis Reports Facility.
7. Five new reports were added in the Alternate Compile Listing Facility:
 - CALL Statements
 - COPY Statements
 - Figurative Constants
 - Literals
 - Special Registers

These reports print one after the other without a page break. In many cases, all five reports will take only a page or two to print out.
8. Several reports that were available on the program side have been moved to the Other COBOL Reports Facility.
9. Producing the Layout Report is now easier and additional options are available to customize the report.
10. The File Analysis Report has been re-named the System Record Analysis Report. Furthermore, where the File Analysis Report produced only direct flags for Set, Used and Tested, the System Record Analysis Report produces both direct and indirect flags.
11. It is now possible to access DCD work files for use in the re-engineering of COBOL programs and systems. This approach to Computer Aided Software Engineering (CASE) gives the Software Engineer a new dimension in the maintenance of COBOL systems.

New Release Highlights
Release 1.1 (Continued)
April 1, 1989

12. The source code of this new release (maintained by MARBLE Computer, Inc.) is top down, structured and easier to maintain, allowing us to better serve our customers when technical problems arise.
13. The error messages have been expanded and completely re-organized.
14. A Call Report Form is now available for FAXing technical problems and questions to MARBLE Computer, Inc.
15. The CLIST Feature available on prior releases of DCD II is no longer supported. Users wishing to maintain it on their own are welcome to do so.
16. The DCD III User's Manual has been completely re-written.

New Release Highlights

Release 1.2

April 1, 1990

This is the second release of DCD III. It contains in excess of 120 modifications for small discrepancies (most reported by our users) and the new enhancements listed below.

1. The System Record Analysis Report is now selectable by COPY member name as well as by record name.
2. The Data Analysis Report may now be limited by the control cards used for the System Record Analysis Report.
3. Further support of IBM's VS COBOL II Release 3 has been added as users begin to migrate to that compiler. Items now included for Release 3 include the following:
 - a. Reading the compile listing of Release 3, which is significantly different than the output of other compilers.
 - b. Handling of BINARY and PACKED-DECIMAL clauses.
 - c. Allowing both single apostrophes and double quotes within the same program.
 - d. Adjusting our scanning of COBOL programs to allow for reference modification now available within Release 3.
 - e. Documenting the new GLOBAL and EXTERNAL clauses available within this new release.
4. Further support of SQL by resolving INCLUDE's as COPY's.
5. For the CA-Optimizer users, continual support has been given to resolving the compiler listing (already formatted by this package) as newer versions are released.

New Release Highlights

Release 1.3

April 1, 1991

The following enhancements have been added to the new release of DCD III:

1. A numeric FROM and TO position relative to the beginning of the record (e.g. 1-10,12-13) is now available to be placed within columns 73-80 of the DATA DIVISION for all the data names found there. Options are available to allow other uses of columns 73-80.
2. Selection is now available on the Verb Analysis Report. Up to ten (10) different verbs may be used at any one time for selection.
3. Further support of IBM's VS COBOL II, Release 3 is included as listed below.
 - Saving formatted information from the Release 3 listing and including that information within the DCD III Source Listing (see R3N options).
 - Handling of EVALUATE/WHEN statements.
 - Document data name references within the parentheses of reference modification.
4. Support is now provided for Release 3 of the CA-Optimizer.
5. Within the CASE and Data Dictionary files:
 - The COPY members of the CASE DATA DIVISION records are now included on the installation tape.
 - Additions to verbs handled are included within the documentation.
 - In the Data Dictionary File, multiple programs may now be processed at one time.
6. An option is now available to distinguish between COBOL 68 and COBOL 74.

New Release Highlights

Release 1.4

April 1, 1992

The following enhancements have been added to the new release of DCD III:

1. A new report 'The CALL Hierarchy Report' has been added with static and dynamic calls handled. Control statements allow deleting common calls and adding non-COBOL programs that call other COBOL programs for keeping the report complete.
2. The JCL Analysis Report (formerly two sequences - DDNAME and DSNAME) now has an additional sequence added for listing formatted DDNAMEs and DSNAMEs in the same order as they are in the JCL. Primary sorts on PROC name and program name are now allowed. Programs with no ddnames may now be listed. DDNAMEs which have DATA, DUMMY, DDNAME=, SYSOUT, or *, may now optionally be excluded.
3. A new JCL Report is now available for reporting JOB names and PROC names from JOBLIB members.
4. A new option (EIB) will invoke the CICS copy member DHFEIBLK into the program for CICS programs that have not been run through the CICS pre-compiler.
5. The DATA Analysis Report has been enhanced as follows:
 - Option D88 lists all values when multiple values are used within one 88 entry.
 - Options (DA1 to DA5 and DAL/NODAL) allow for sorting data names after omitting leading numeric digits. This allows similar data names to be reported together.
6. The LAYOUT report has been enhanced as follows:
 - Headings have been modified for both the LAYOUTs and the Table of Contents for LAYOUTs.
 - The page number has been moved from the bottom to the top of the report.
 - When the LOR option is used:
 - a) SECTION is removed from the report.
 - b) Sequence numbers are removed from the report.
 - c) Member name is put on the Layout Report when the Member Fetcher Proc is used.
 - A date has been added to the report.

New Release Highlights
Release 1.4 (Continued)
April 1, 1992

7. The support of our compile mode has two changes:
 - Support has been added for our handling of a change in the formatting of the CA-Optimizer.
 - Support has been added for our handling of a change in the format of the Release 3 VS COBOL II COBOL compiler.

8. Two items of support were added for VS COBOL II:
 - DAY-OF-WEEK was added as a SPECIAL REGISTER.
 - The use of no data name at the group level is now handled.

New Release Highlights

Release 1.5

April 1, 1993

1. ISPF Panels are now available for the execution of DCD III.
2. A new option NIS (Narrative In Sequence) has been added to allow a different sort sequence for the Data Division narrative. The current default for the narrative uses the COBOL verb (ADD, IF, MOVE etc.) as the primary sort. When this option is used, the Narrative uses the ascending Procedure Division sequence number as the primary sort. This new option affects three DCD reports:
 - Source Listing of the Alternate Compile Listing Facility
 - Data Name Cross Reference of the Other COBOL Reports Facility
 - LAYOUTs of the Other COBOL Reports Facility
3. Beginning support is provided for IBM's new compiler 'IBM SAA AD / CYCLE COBOL / 370'. Continuing changes have also been made for release 3 of VS COBOL II, especially in COMPILE mode of the UCLF.
4. For those who use CA-Optimizer with VS COBOL II, we have made changes to keep pace with modifications so that DCD reads CA-Optimizer listings correctly.
5. Small problems in the handling of SQL code (EXEC SQL) have corrected.

New Release Highlights

Release 1.6

April 1, 1994

1. DCD III now handles COPY members within COPY members. Both COPY reports (one in UCLF and one in OCRF) show the nested level of the COPY member.
2. All host variables with SQL SEARCH PREDICATEs are now fully documented instead of just the first variable.
3. Additions and corrections to our new ISPF panels:
 - Headings have been changed to conform with requests from users and present a more SPF look.
 - The COLORs have been changed to conform more with typical usage.
 - Corrections for problems reported within the first year's usage have been made.
4. Release 4 of VS COBOL II is now supported. Earlier releases may not fully support this newer release.
5. Group usage POINTER and INDEX is now supported.
6. Enhancements have been made to support COPY REPLACING pseudo- text as it is being used for advanced levels of prefix substitution within COBOL programs.
7. A new option has been added to the LAYOUTs when documenting COPY members or documenting individual records not directly coming from a COBOL program.

New Release Highlights
Release 1.7
April 1, 1995

The following enhancements have been added to this release of DCD III:

1. SPF panels have been modified to allow for COBOL 370.
2. Other enhancements and corrections have been made to the SPF panels initially released by MARBLE two years ago.
3. Handling of the COBOL verb EVALUATE has been made to allow for NUMERIC and NOT NUMERIC.
4. FROM and TO positions of fields within 01 records used in three DCD III reports,
 - a. Alternate Compile Listing
 - b. LAYOUTs
 - c. System Record Analysishave been enhanced to allow for two extra digits as compilers are now allowing individual record sizes greater than 6 digits.
5. Enhancements have been made to both correct problems and to conform to newer compiler enforcement of USAGE when the usage is specified both on a group field and an elementary field within that group.

New Release Highlights

Release 2.1

June 1, 1996

The following features and enhancements are included in this new release of DCD III.

1. Tracing of Literals for user selected fields is now available with this release. Literals are traced through several (17) levels of MOVES & SETs for user selected fields. Three different reports show possible Literals:
 - for each field traced
 - showing where the literal came from
 - showing the possible paths to arrive at a fieldSamples of these reports are provided in Other COBOL Reports Facility pages.
2. For VS COBOL II COBOL user's (Release 3 or higher) and COBOL 370 & higher users, a new DCD III option (**PMO**) is added for COMPILE MODE to trap HEX OFFSETS from the compiler HEXLOC cross reference map and then match merge these LOCATIONS into the DCD III UCLF source listing. These locations are inserted alongside each Procedure Division source code statement.
 - As we anticipate the option to be heavily used, (from numerous user comments), this option is defaulted, too. It may be turned off.
3. Two new options are available with the CALL Analysis Report and two new options are available with the COPY Analysis Report for individually producing the reports in just PROGRAM sequence or in just COPY or CALL sequence. Previously, when producing these reports, both sequences of either report came out together.
4. Two forms of Qualification of Paragraphs are now with this release handled by DCD III. They are:
 - a) GO TO DUPLICATE-PARAGRAPH-NAME will go to the paragraph name within the section which contains this GO TO statement.
 - b) GO TO DUPLICATE-PARAGRAPH-NAME OF SECTION-WITHIN-PROG will go to the duplicate named paragraph within the section specified.
5. Initial handling of verbs using COBOL 370 Intrinsic Functions is now provided. Advanced handling of Intrinsic Functions is now under development.
6. SYNC is now handled on an 01 record.

New Release Highlights

Release 2.2

May 1, 1997

The following features and enhancements are included in this new release of DCD III.

1. Modifications are included to insure correct handling of our software into and beyond the **Year 2000**.
 - The control password is changed from a two digit year to a four digit year (from 99 to 9999).
 - The reports listing have all been changed from a two digit year to a four digit year (from mm/dd/**YY** to mm/dd/**YYYY**).
 - CASE generated work files and internal debugging files of DCD III have been modified to incorporate a new 10- character date with 4 -digit year in their header record.
 - The code has been tested to insure that execution will continue normally after December 31, 1999.
2. DCD III is modified to accept the new **IBM COBOL for MVS and VM** in Compile Mode of the Alternate Compile Listing.
3. A method of handling PREFIXs with COPY REPLACING with the use of colons or parentheses by the newest IBM compilers, has been added by user request and to conform with IBM documentation.
 - An example follows:
COPY PAYCOPY REPLACING ==:TAG:== BY ==PAYROLL==
where the COPY member contains entries similar to:
01 :TAG:..
05 :TAG:-WEEKPIC X(02).
05 :TAG:-DAYPIC X(02).
4. DATA DIVISION literals are now omitted in LITERALS report of the UCLF listing.
 - This will shorten the UCLF listing, saving paper.
 - An option is present to allow a user to re-include them.
 - Documentation is included at the top of the UCLF listing explaining the change.
5. One enhancement and one correction are made in the handling of the SPF option for COBOL 370.
 - The correction now insures an initial default when SPF brings up that panel.
 - The COBOL 370 option now includes the new COBOL FOR MVS AND VM compiler.
6. A new password is provided. It must be used to conform with modification #1 above.

New Release Highlights

Release 2.3

April 1, 1998

The following features and enhancements are included in this new release of DCD III.

1. The newest IBM compiler '**IBM COBOL for OS/390 and VM 2.1.0**' is supported with this release of DCD III. If you install this newest COBOL compiler and immediately experience problems with the COMPILE mode of DCD III, installing this new release will eliminate this problem.
2. Two changes have been made to DCD III's narrative as follows:
 - a) Always in Upper Case, DCD III's narrative is now in **lower case** unless changed back through a DCD option.
 - b) The use of REFERENCE MODIFICATION in the COBOL program is now shown directly in the DCD narrative as a colon (:) placed next to the operand it is used with.
3. A new report is added to the UCLF (main side) of DCD III. This new report is a '**Condensed VERB Report**' allowing the COBOL program to find infrequently used verbs quickly or to look for possible errors by quickly identifying the use of PERFORM separate from PERFORM THRU's.
4. An option has been added to **allow HEX numbers showing FROM and TO positions** in column 73-80 in place of DECIMAL numbers.
5. A brand new feature has been added for those users doing **Year 2000 Conversions**. (See enclosed brochure.)
 - a) For finding those fields for Year 2000 Conversion, a brand new powerful SELECTION language (free format in a COBOL like SYNTAX) has been added.
 - b) Immediately for those fields selected, a report is produced showing DCD III's unique COBOL narrative right under the selected field.
 - c) Overlapping fields (and all REDEFINED fields) relating to the selected field are shown in one of two sequences underneath the selected field.
 - d) COBOL narrative for each Overlapping or Redefined field is shown underneath each overlapping field.
 - e) Nine options are available for controlling this powerful new report.Our technical staff is prepared to answer questions concerning this new feature.
6. Enhancements have been made for **PANVALET and LIBRARIAN** users:
 - a) Multiple PAN LIBs are now allowed for PANVALET users.
 - b) A different way of selecting Librarian modules is provided to prevent security violations in packages such as RACF, ACF2, TOP-SECRET, and others.

(Continued)

New Release Highlights
Release 2.3 (Continued)
April 1, 1998

7. Advanced handling of **Intrinsic Functions** in COBOL is now handled.
8. Cross referencing is now provided for the COBOL clause (DEPENDING ON Data-Name) providing a reference to assist in locating the Data-Name when it is separate from the table.
9. Changes in the minimum amount of REGION used in SORT packages such as SYNC-SORT have been addressed by upping the REGION used for both SORTREG and overall REGION within all DCD III JCL. Also, it is now possible to externally specify REGION when using the DCD III SPF facility.
10. As always, minor fixes reported during the last year have been added to this new release.

New Release Highlights

Release 2.4

March 1, 1999

A major new feature, Tracing and Analysis is added to DCD III with this release. Parts of this new feature were released last year in release 2.3.

This year, **Full Tracing** and more has been added as shown below.

The bottom line for this new feature is that a programmer can start on a Modification to a COBOL program and within a quick and simple DCD III run, produce a report showing every single data name that will need to be considered in making any changes for any data field in the program.

For user's who have not completed their **Year 2000 conversion**, the following is available:

- An option which uses the above Tracing & Analysis Feature to **automatically trace through the program and build an easy to read set of CHANGE LINES** which the user may use to **VERIFY** that all changes previously done for **Y2K conversion ARE CORRECT!**

If the user has any code still to convert, **MARBLE** offers **Code Changing Software** which works in conjunction with the above **CHANGE LINE** feature to actually change **ALL** source code and **COPY** members.

Full features of 'Tracing and Analysis'

1. **Full SELECTION language allowing:**
 - Simple selection where the user knows which field to select.
 - A complete selection language allowing the user a way to select by supplying various criteria.
2. **Inclusion of all Overlapping (REDEFINES, Groups, 88's, & other fields) associated with the selected name.**
3. **Tracing through infinite levels to isolate every field that can be tied to the original field or fields selected.**
4. **Complete & Understood Narrative for every field to show all related Procedure Division activity. This Narrative shows the way through each deeper level of tracing.**
5. **Bypassing of some fields where the user determines that a particular trace is not wanted. This allows a user to end a trace through that field.**

Additionally, for any **Year 2000 work** not done, **CHANGE** lines can easily and automatically be built to allow the user to verify that his or her **Year 2000 changes** are accurate.

Release 2.5

May 1, 2000

The following features and enhancements are included in this new release of DCD III

1. Several enhancements are made to our new Tracing & Analysis feature. Three specific areas are listed here:
 - a) Control statement for SELECTs may now be entered without the **SELECT IF NAME** = prefix. (The user now enters only the name he or she is selecting.)
 - b) DDnames now use the **SELECT** and **BYPASS** names for easier recognition
 - c) Several areas of the report have been modified including changes in how we report errors and wording changes for easier report understanding.
2. Several SQL enhancements are added for fuller DB2 processing:
 - a) Modifications were done for the correct processing of intrinsic functions such as CURRENT DATE and CURRENT TIMESTAMP within EXEC SQL statements.
 - b) Additional Host variables within WHERE clauses for SQL CREATE _VIEW SELECT_sub_clauses, UPDATE, and DELETE verbs are now handled.
 - c) Additional Host variables within VALUE clauses for the SQL INSERT verb are now handled.
3. Intrinsic FUNCTION handling has been expanded to handle FUNCTIONs within FUNCTIONs.

New Release Highlights
Release 2.6
April 1, 2001

The following features and enhancements are included in this new release of DCD III.

1. More enhancements are made to our newest Tracing and Analysis Feature.
 - a) Internal Table sizes used in Tracing & Analysis have been increased.
 - b) If any of these newer table sizes are exceeded, the run will now continue to end to allow a user to see a partial run.
 - c) More attention has been given to error messages to make them more readable.
 - d) A problem in looping during infinite tracing levels has been fixed.
 - e) A new PARM field has been included to allow the user to limit the maximum trace levels used.
2. Reference Modification in complex AND/OR statements is now allowed for. The Procedure Division Narrative for this has been corrected to document all data names addressed.
3. Special Register 'LENGTH OF' is now processed by DCD III.
4. A special PARM option has been added to invoke older COBOL COPY member handling where the user expects the 01 record name preceding the COPY on the same line to overlap the 01 record name within the COPY member.
5. Documentation on use of the IBM Compiler TITLE statement in Compile mode is included.
6. A problem has been addressed where continuous page feeding occurred when the LNCNT option is set to zero.

New Release Highlights
Release 2.7
April 1, 2002

The following features and enhancements are included in this new release of DCDIII

1. Support in this release is provided for IBM's newest COBOL Compiler "IBM ENTERPRISE COBOL FOR Z/OS AND OS/390" which runs under the new Z/OS operating system.
 - For clients running the Alternate Compile Listing in *Compile Mode (the normal way of producing compile reports)*, this new release will need to be installed to allow us to read this listing and make DCDIII continue to work.
2. A new report is available to produce an INCLUDE report (similar to COPYreport) for all INCLUDEs found in CICS, DL1, and SQL programs in the Alternate Compile Listing
 - For CICS, DL1 or DB2 programs, this report will be produced if INCLUDEs are Found and the COPY parm option is specified.
3. A new report is available and is automatically turned on with the use of SECTIONs to show SECTIONs stand alone without paragraphs included. This report will follow the PROCEDURE DIVISION cross-reference report in the Alternate Compile Listing.
4. DCD III options are more cleanly documented. Default options are now always shown on the parm page indicating clearly that an option is on.
 - Other supporting documentation on the options page promotes user use of DCD III's newer Tracing and Analysis feature.
5. A reference Modification bug within DCD III that caused DCD III to stop running has been fixed.
6. Parm option VS2 is now the default option specifying use of a newer COBOL compiler.
7. A CLASS test has been added for DBCS.
8. The SET verb has been modified to allow for PROCEDURE-POINTER and FUNCTION- POINTER.
9. New RESERVED words have been accounted for.
10. Two new Special Registers (XML-CODE and XML-EVENT) were added.
11. USAGE clauses now account for NATIONAL, FUNCTION-POINTER and PROCEDURE-POINTER.

New Release Highlights

Release 3.1

April 1, 2003

The following features and enhancements are included in this new release of DCD III.

1. **Indirect References** for all data fields have been added in the main source code report of the Alternate Compile Listing.

- Previous releases have shown narrative for each field that is **directly** related to the data field by a direct PROCEDURE DIVISION statement.
- **Indirect references** to this data field via PROCEDURE DIVISION statements to **REDEFINES**, group fields or other fields with overlapping fields were ignored and left for the customer to search out on their own.
- In this release, **Indirect references** are now documented on the Alternate Compile Listing for most fields and on a separate report for large group fields with so many references as to make it not practical to report in the main report. *(If the indirect references are moved to the separate report, documentation on the main report will state that this is being done.)*
- The indirect references will be shown immediately after all direct reference Narrative is complete. A sample of both direct and indirect together is shown Below with the Indirect references in bold font:

Move TR-CUST-ACCT @722 to # (1356,1482,1941)

If # = SR-CUST-ACCT @569 (1275,1518)

Indirectly Changed @895,914 Tested @892,914 Used @892,895,917

- By design, when there is no direct narrative for a field, then indirect references will not be reported unless a special DCD option is invoked. Activity for 88 level Condition names will be shown with the data field that the 88 levels belong to. FILLERs will show no indirect references.
2. The use of double parenthesis is now handled within the use of Reference Modification.
 3. Error messages have been corrected to not flag a missing period when an END-EXEC precedes an 01 level.
 4. Region size and Sort Region Size have been increased to accommodate changes in newer IBM operating systems.
 5. Additional work files have been removed from the JCL that were used for Year 2000 work.

New Release Highlights

Release 3.2

June 1, 2004

The following features and enhancements are included in this new release of DCD III.

1. This version is modified so that all DCD III modules are AMODE 31 compliant to work easily with IBM's newest Enterprise COBOL compiler.
2. The CALL report for listing DYNAMIC CALLs has been modified within the 'Alternate Compile Listing'. Previous versions showed only a Data Division Name within the normal CALL report. This enhanced report will show both the Data Division VALUE and any MOVE literals into the field used for the DYNAMIC CALL to allow for a fuller understanding of how CALLs are used.
3. In the last release, excessive **Indirect References*** were moved to a separate report. In this release, the user has the option to leave all the Indirect References in the main report of the 'Alternate Compile Listing'.
4. As of this release, the user may qualify which **Indirect references*** are included. Just **Changed** references, just **Indirectly Tested**, or just **Indirectly Used**, or any combination of the three. Prior to this release qualification was not allowed.

* **Indirect References** were put into DCD III one year ago in release 3.1

5. A problem occurring when qualification of an higher level name existed, but was invalid for use and DCD III did not put out a message has been resolved and a proper error message is now issued.
6. An error in compile mode where a particular formatting of a PROCESS card ahead of the COBOL program causing an abend has been fixed.
7. A problem where the Verb Analysis was intermittently causing an abend has been fixed.

New Release Highlights

Release 3.3

May 15, 2005

The following features and enhancements are included in this new release of DCD III.

1. The formatting of entries in the CALL report in the Alternate Compile Listing has been modified so that additional Dynamic CALL information added last year, along with prior CALL information and with CANCELs added this year are presented in groups related to each CALL.
2. The CANCEL verb has been added to the CALL report in both the Alternate Compile Listing and in the Tracing, Analysis & Other Reports.
3. Processing of SQL (DB2) statements in COBOL has been modified to insure that the ending END-EXEC will always stop our scanning. In rare instances, the END-EXEC was not caught.
4. Within commented out EXEC CICS statements, not all data division names were always fully documented. This has been fixed.
5. The normal 'End of Member Fetcher' message has to been corrected to remove part of another message that should not be there.
6. Minor corrections to last year's release 3.2 (a major release where all our programs had to be modified) have been made to insure error free processing including the following:
 - a) Two different intermittent OC7s, one for the DATA option and another sorting Indirect references have been removed.
 - b) In COMPILE Mode, several minor errors have been corrected.
 - c) Extra beginning blank lines in the CALL report have been removed.
 - d) The incorrect formatting of Narrative of literals used in a CALL statement alongside a MOVE with a LENGTH OF clause has been fixed.
 - e) The reporting of incorrect TIME alongside the DATA in some of our reports has been fixed.

New Release Highlights

Release 3.4

March 1, 2006

The following features and enhancements are included in this new release of DCD III.

Three new reports are available in this release of DCD III to assist in the cleaning up of all COBOL programs. They are:

- A report showing all unused 01 records in the program.
- A report showing unused Data Names **that are not** embedded in a COPY member. If there are several data names together where there are group names involved, the largest group field will always be shown in place of showing individual data names.
- A report showing unused Paragraph and Section Names. This report may be used to identify code that is not be accessed. Special attention is used to showing oddities in performs and ENTRY's into the program in relation individually and together with both SECTIONS and with PARAGRAPHS.

Five discrepancies are fixed in this release reported over the last year.

- A fix was included to correct SYNTAX when more than 125 operands were used on one side on an IF statement.
- When error message ACLY6C01-D was issued for a very large program, an OC4 resulted.
- An OC4 occurred in selected sites, when DSN=SYS1.SCEERUN was not used and concatenated ahead of the DCD load library.
- An OC7 resulted when option IREFS was used and there were not data names in the program.
- Infrequently DCD III issued ACLLRFA1-C messages when there were 3 levels of qualification.

Release 3.5
August 1, 2007

The following enhancements are included in this new release of DCD III.

1. When using the DCDJCL feature and also using the ISPF feature to select libraries, concatenated libraries are now processed correctly.
2. In isolated cases where an DB2 'INCLUDE SQLCA' needs to be included to resolve missing data name messages and is not, DCD III now removes these warning message from appearing in the DCD III error message reports.
3. An occasional OC7 popping up when including the new 'Unused Data Name Report' added last year, has been fixed.
4. An incorrect error message, associated with the new reports added last year, associated with a table being exceeded, has been corrected.
5. An error in how we process COPY replacing pseudo-text, where the phrase `==(*)== BY =====` is used allows the replaced data-name to slide left backwards into the preceding level number has been fixed.
6. A double qualification problem within one IF similar to the following:

IF FIELD-1 OF QUAL-FIELD = 'A'
 OR FIELD-2 OF QUAL-FIELD = 'B'

where some references were lost and/or gave wrong references, has been corrected.

7. The sub-heading in the 'Alternate Compile Listing, labeled 'Literals', has been changed to a more correct label of 'Procedure Division Literals'.
8. When new DDNAME DCDWK08 is missing, an appropriate message is issued, rather than having DCD III issue an abend.
9. A correction was made in the new report showing unreferenced Paragraphs and Sections to include EXEC CICS references to Paragraphs.
10. The user manual now correctly reflects the new DDnames SELECT and BYPASS used for Tracing and Analysis.
11. References in the User's manual to DCD III being compatible with Code Changer software for Field Expansion in Release 3.2, released three years ago have all been removed. Parties interested in Field Expansion software should contact Marble Computer, Inc.

New Release Highlights

Release 3.6

October 1, 2008

The following enhancements are included in this new release of DCD III.

1. A new release of DCD III may now be downloaded from the web rather than obtaining a physical cartridge from Marble Computer.
2. All **600+ Error Messages** produced by DCD III, most dating back to 1989 when DCD III was rewritten into its current format have been rebuilt to a new format making it easier and quicker to identify these messages. Some messages no longer used have been removed. These error messages may be found in the Error Message section of this manual.
3. The entering of the number of **WHOLE-DIGITS** and **DECIMAL-DIGITS** within Tracing and Analysis has been modified to allow either 1 or 2 digits after the operator that follows either keyword mentioned above, rather than always entering 2 digits. This eliminates annoying error messages and simplifies use.
4. SQL processing has been corrected where one type of handling for the **SQL word COUNT** was ignored.
5. Intermittent handling of the **OPTIONAL** clause with COBOL FD SELECTs has been corrected.
6. A correction has been fixed to treat **INCLUDE** members in the same manner as **COPY** members are treated for the purpose of excluding these data names from showing within the **Unused Data Names report**.
7. A correction has been added to fix a problem where an **AND** following a **WHEN** within an **EVALUATE** statement was incorrectly handled and showed up in the narrative as a subscript for another name.

Release 3.6

DCD III

Alternate Compile Listing

Facility

DCD III

Alternate Compile Listing Facility

TABLE OF CONTENTS

Use of This Section / List of Exhibits	A-4
Choice of DCD III PROCs	A-5
Compile Mode Consideration	A-6
Other Features Supported	A-10
Specifying PARM Options	A-14
PARM Options.....	A-15
Corresponding PROC Symbolics	A-19
JCL Examples	A-20
JCL to Execute DCD III Without Using PROCs.....	A-23
PROCs Provided With the DCD III System	A-25
Introduction to the Alternate Compile Listing Facility	A-30
Overview of Alternate Compile Listing Facility Reports.....	A-31
Use of DCD III PARM Options.....	A-44
Use of DCD III PROCs and Symbolics	A-47
Executing DCD III When Using a PROC.....	A-49
Unused 01 Records and Unused Data Names.....	A-51
Unused Paragraphs and Sections.....	A-52

Use of This Section

This section is designed for everyday use of the Alternate Compile Listing Facility. Topics like, “Introduction to the Alternate Compile Listing Facility”, “Overview of Alternate Compile Listing Facility Reports” and others which are designed to be read only once or twice by the programmer are moved to the end of this section, while topics such as “Choice of DCD III PROCs” and “PARM Options” are moved toward the front of this section.

When using this section for the first time, read the following topics first:

1. Introduction to the Alternate Compile Listing Facility
2. Overview of Alternate Compile Listing Facility Reports
3. Use of DCD III PARM Options
4. Use of DCD III PROCs and Symbolics

These four topics are together within this section. See the Table of Contents.

After reading and reviewing the above topics, continue using this section, starting at the top of the Table of Contents.

List of Exhibits

1.	CA1 Report Example.....	A-13
2.	CA2 Report Example.....	A-13
3.	Compile Mode PROC - COMACL.....	A-26
4.	Independent Mode PROC - DCDACL	A-27
5.	Librarian Independent Mode PROC -LIBACL	A-28
6.	Panvalet Independent Mode PROC - PANACL.....	A-29
7.	Source Listing Report - Data Division	A-32
8.	Source Listing Report - Procedure Division.....	A-34
9.	Condensed VERB Report	A-35
10.	CALL and COPY Statements Reports.....	A-36
11.	Figurative Constants Report	A-37
12.	Literals Report	A-38
13.	Special Registers Report.....	A-39
14.	Data Division Condensed Cross Reference	A-41
15.	Procedure Division Condensed Cross Reference	A-43
16.	Overflow Report for Indirect References	A-43
17.	DCDACL PROC.....	A-50

Choice of DCD III PROCs

Four PROCs are available to choose from when invoking the Alternate Compile Listing Facility. They are:

1. **LIBACL**
2. **PANACL**
3. **COMACL**
4. **DCDACL**

The first and second PROCs are made available for running the Alternate Compile Listing Facility when the source code resides on Librarian or Panvalet files respectively and the user is not using the Compile Mode. (See next paragraph for Compile Mode.) (See “Other Features Supported” for Librarian and Panvalet support.)

The third PROC COMACL is made available for running the Alternate Compile Listing Facility in conjunction with and following the COBOL compile step. The advantage to using this PROC in conjunction with the COBOL compile step is that **one listing** is returned to the COBOL programmer. (See “Compile Mode Considerations” for Compile Mode Support.)

The last PROC DCDACL is made available for running the Alternate Compile Listing Facility when not using one of the above methods. The COBOL source program is provided to the PROC by means of a COBOLIN DD. If COPY members are present within the program (*and a COPYLIB DD is not already provided within the DCDACL PROC*) a COPYLIB DD must be provided to point to the appropriate PDS containing COPY members. This must be the same PDS(s) provided within the SYSLIB DD card within the PROC used to compile the programs. Symbolic parameters may be added to the PROC to override already established defaults. See “Specifying PARM Options” for a list of the symbolic parameters and their uses. The following example is provided:

```
//STEP1 EXEC DCDACL,UNREF=NO
//DCD.COBLIN DD DSN=USER.PROGRAM.LIB(PROG301),DISP=SHR
//DCD.COPYLIB DD DSN=USER.COPY.LIB,DISP=SHR
```

More examples are provided under the heading “JCL Examples” in this section. See the Table of Contents.

For a further description on using DCD III PROCs review the following two headings:

1. **Use of DCD III PARM Options**
2. **Use of DCD III PROCs and Symbolics**

Compile Mode Considerations

The Alternate Compile Listing Facility is a stand-alone system, which gives a complete set of reports for the maintenance of COBOL programs. It is not, however, a COBOL compiler. As such, a means is provided, named Compile Mode, in this Alternate Compile Listing Facility for producing one report from the two steps listed here:

1. The COBOL compile step
2. The Alternate Compile Listing Facility Step

Some advantages of using the Compile Mode in the Alternate Compile Listing Facility are listed here:

1. The programmer can have the advantage of using the Alternate Compile Listing Facility every time he or she does a COBOL compile.
2. Extra paper is not generated. The source listing from the COBOL compiler is removed and replaced with one from DCD III. The DCD III Condensed Cross Reference Reports are much shorter than the compiler cross-reference listings.
3. Information such as PMAP or DMAP reports or similar information on the source listing produced when using software packages such as the CA-Optimizer or COBOL/Aid is not lost, but is kept and listed appropriately when using Compile Mode. (CA-Optimizer and COBOL/Aid are mentioned under the heading “Other Features Supported”.)
4. The running of the Alternate Compile Listing Facility is usually set to stop and return just the compiler listing if E-level errors are found. This is controlled by the JCL COND parameter within the DCD III step.
5. The sequence numbers (generated compiler numbers on the Source Listing Report and referenced in other reports) are guaranteed to be the same on all reports merged together, whether from the COBOL compile, the software package like CA-Optimizer, or the Alternate Compile Listing Facility.

Compile Mode requires the following two steps:

1. Run the compile step with modified JCL for the SYSPRINT DD line and do not allow the use of the TITLE statement within the compile step.
2. Run the Alternate Compile Listing Facility with the Compile Mode option turned on.

The SYSPRINT file (printed listing) produced by the COBOL compiler must be saved (not printed) so that it may be read in and pulled apart by the Alternate Compile Listing Facility.

To do this requires a JCL override for the SYSPRINT file when invoking the compile PROC. When other overrides are present, this override must go in sequence so that the SYSPRINT DD within the compile PROC is actually overridden.

The LRECL and BLKSIZE below is set for 133 and 1330. If using a COBOL 74 compiler, before VS COBOL II, unless an option is set with your compiler to produce a LRECL of 133, then substitute 121 and 1210 for LRECL and BLKSIZE. Because of the way the compiler may be set to run at your installation, it is recommended that the SYSPRINT BLKSIZE **not be made initially larger** than 1330 or 1210. If the BLKSIZE is made larger and problems occur, reduce the BLKSIZE before going further.

DISP=MOD must be used as is shown below.

The following is an example of an overriding JCL statement, which must follow the EXEC compile PROC.

```
//COB.SYSPRINT DD DSN=&&COMPLIST,  
//          DISP=(MOD,PASS),  
//          UNIT=SYSDA,  
//          SPACE=(CYL,(2,2)) ,  
//          DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)
```

Also, the following should be done when running the Compile step:

1. Ensure that the compiler option NONUM option (not NUM) is in effect. NONUM is normally the default.
2. If the LISTER option is available on your compiler, do not turn that option on when using the Alternate Compile Listing Facility.
3. Observe the rules already mentioned above about using a MOD disposition and initially using a block size of 1330 or 1210.
4. Ensure that the compiler options for printing the data name and procedure name cross references are turned off. This will save paper as DCD III has Condensed Cross Reference Reports which are one third the size.

Running the Alternate Compile Listing Facility Step

Use the COMACL PROC for Compile Mode. PROC COMACL has the following differences in it from the other PROCs to accommodate Compile Mode:

1. The PARM option COMPILE is turned on in the PARM field of the PROC.
2. The DDNAME INFILE is provided within the PROC and points back to the saved compiler SYSPRINT file.
3. An additional step is added to this PROC for returning the entire compiler listing if one of the following conditions should occur:
 - a. The Source Listing Report option in the Alternate Compile Listing Facility is not turned on (e.g. SOURCE=NO).
 - b. E-level errors were found in the compiler listing during the compile step. (This is controlled by the COND=parameter.)
 - c. The PARM option STOP is used to bypass the running of the Alternate Compile Listing Facility.
 - d. The Alternate Compile Listing Facility step abends for any reason.
 - e. A trial copy of DCD III has expired.
4. A COMLCNT= symbolic is provided within the PROC for entering the compiler line count that is used in the compile step.
5. A COND CODE of **0001** is normally returned from the first step in the COMACL PROC (not 0, 4, 8, or 12 as from the other PROCs). The second step is normally **not** executed.

When using the COMACL PROC, the DDNAME COBOLIN is not required, as the COBOL program is taken from the SYSPRINT file via the INFILE DD mentioned earlier.

Most COPYs are automatically resolved during the compile step. However, the COPYLIB DD should always be provided for resolving COPY members where the SUPPRESS clause has been used. Also the SRESOLVE PARM option should be used to indicate that DCD III should resolve these members. If SRESOLVE is not used, then a COPYLIB DD is not necessary. (However, in doing this the documentation will be incomplete for references made to those fields within the suppressed COPY statement.)

Running the Alternate Compile Listing Facility Step
(Continued)

In the DCD III step, when reading the SYSPRINT file in and pulling off the COBOL source program from it, DCD III must also determine when EJECTs are present within the source program, so that the EJECTs are present within the Source Listing Report produced by DCD III. To consistently do this and not inadvertently miss any, DCD III needs to know the compiler line count used within the compile step. Sometimes DCD III can pick this up from the compiler listing provided the compiler **statistics** are printed *before* the COBOL source is printed. To ensure accurate finding of all EJECTs use the COMLCNT=symbolic to enter the compiler line count used within the compile step.

Other Features Supported

This heading provides a list of support for those other features both inside and outside of the standard support for the COBOL compiler that the user may have need for. Those features supported by DCD III are:

1. BASIS
2. CA-Optimizer
3. CICS
4. COBOL/Aid
5. COBOL 68 and earlier
6. COBOL 74
7. COBOL 85
8. COBOL 370, COBOL for MVS & VM, COBOL 390 & Enterprise COBOL
9. DB2 (SQL)
10. DL1
11. Librarian
12. Panvalet

1. BASIS

BASIS is an IBM Extension, which is a library system for making changes to a COBOL program, which is for the most part infrequently used. If this feature is used in the Compile Mode of DCD III, nothing needs to be done within DCD III. When using the DCDACL PROC (not Compile Mode) an additional DD line with the name BASIS will need to be used to input the BASIS lines. The rules that apply for the COBOL compiler when using BASIS apply to DCD III.

2. CA-Optimizer

The CA-Optimizer is commercial software package widely used. Information from the DMAP and PMAP of the compiler is put alongside the Source Listing Report in the same area that DCD III narrative is put. To accommodate this situation in DCD III within Compile Mode, several options are available. They are:

- a) Run in Independent Mode. (Do not use Compile Mode.)
- b) Use Compile Mode with no special options and all DCD III narrative will be dropped down one line.
- c) Use PARM option CA1, which will move the CA-Optimizer displacement for the Procedure Division into COBOL positions 73-80 on the Source Listing Report. See Exhibit 1 in this section.
- d) Use PARM option CA2, which is the same as CA1 except that it also (by means of cut and paste) moves CA-Optimizer information for the Data Division into COBOL positions 73-80 on the Source Listing Report. See both Exhibit 1 and Exhibit 2 in this section.
- e) Use PARM option CA3, which will move the CA-Optimizer displacement for the Procedure Division into print positions 128-133. The Data Division CA-Optimizer printed information is left alone.
- f) With release 2.1 or greater of DCD III, omit using CA-Optimizer and let defaulted option PMO, move OFFSETs from the compiler OFFSET map on the Procedure Division listing.

3. CICS

CICS is fully supported by DCD III. Ideally the CICS program should be run through the CICS preprocessor before being brought into DCD III. When using Compile Mode, the program is run through the preprocessor before the compile step is done. If the preprocessor step is done, both CICS commands and the CALL statements generated by CICS are documented. If the CICS program is not run through the preprocessor, then generated CALL statements are not documented by DCD III.

4. COBOL/Aid

COBOL/Aid is a software package which, like the CA-Optimizer, prints DMAP and PMAP information alongside the Source Listing Report. When this package is used, the PARM option COAID must be used. DCD III handles COBOL/Aid in the same manner with the options provided for the CA-Optimizer. See the CA-Optimizer within this heading.

5. COBOL 68 and earlier

PARM option C68 should be used when processing COBOL 68 programs. The only other special support required for 68 and earlier versions of COBOL is when Compile Mode is used and the compiler is version 3 or earlier (easily determined by the fact that the compiler does not generate compiler line numbers for SKIPS and EJECTs). When this is the case, PARM option VR3 should be included in the PARM field.

6. COBOL 74

Use option NOVS2 for COBOL 74.

7. COBOL 85

Include PARM option VS2 to indicate the handling of VS COBOL II. Also, when using Compile Mode, use a LRECL of 133 and a corresponding BLKSIZE when providing JCL for the COB.SYSPRINT override.

8. COBOL 370, COBOL for MVS & VM, COBOL 390, & Enterprise COBOL

Include PARM option VS2, the same as for COBOL 85. When using the ISPF panels, a separate option is included there for COBOL 370. It is not needed in batch submission.

9. **DB2**

DB2 programs contain SQL statements. Ideally, the DB2 program should run through the DB2 preprocessor before being brought into DCD III. When using Compile Mode, the program is run through the preprocessor before the compile step is done. If the preprocessor step is done, then the CALL statements generated by SQL are documented. Host variables within the EXEC SQL statements (names that reference into the COBOL program Data Division) are fully documented by DCD III.

10. **DL1**

DL1 is fully supported by DCD III. Ideally the DL1 program should be run through the DL1 preprocessor before being brought into DCD III. When using Compile Mode, the program is run through the preprocessor before the compile step is done. If the preprocessor step is done, both DL1 commands and the CALL statements generated by DL1 are documented. If the DL1 is not run through the preprocessor, then generated CALL statements are not documented by DCD III.

11. **Librarian**

When INCLUDEs are used (as opposed to COPY statements), they are resolved when the COBOL source code is brought in from Librarian, and therefore are not listed within the COPY Statements Report.

If Compile Mode is used, no other support is required. If Compile Mode is not used, then use the PROC LIBACL which contains a Librarian step ahead of the DCD III step for pulling the COBOL program off the Librarian Data File. See Example 2 under the heading 'Examples' in this section for assistance in using the LIBACL PROC.

12. **Panvalet**

When INCLUDEs are used (as opposed to COPY statements), they are resolved when the COBOL source code is brought in from Panvalet, and therefore not listed within the COPY Statements Report.

If Compile Mode is used, no other support is required. If Compile Mode is not used, then use the PROC PANACL which contains a Panvalet step ahead of the DCD III step for pulling the COBOL program off of the Panvalet Data File. See Example 3 under the heading 'Examples' in this section for assistance in using the PANACL PROC.

DCD III – Alternate Compile Listing Facility

SQ-NBR	1-----6	PROGRAM -ID	DCDLOGIC	01/01/2001	14:56	PAGE	23	73-----80	SQ-NBR=S	LOGIC FLOW
108	032200		OR WK-8-12 IS EQUAL TO 'ID D'						047	
109	032300		MOVE 'I' TO WS-INDENT-SW					000D90	038	
110	032400		ADD 1 TO WS-CNT					000D94	044	
111	032500		GO TO B999-EXIT.					000D9A		COND GO TO 112
112	032600	B999-EXIT.						000D9A		COMES FROM 104,
113	032700	EXIT.						000D9A		RETURN TO 074
114	032800	C-WRITE-READ SECTION.						000DA0		PERFORMED BY 077
115	032900	C100-WRITE.								
116	033000	WRITE DISK-REC FROM WS-WORK-AREA						000DA4	029 045	
117	033100	INVALID KEY DISPLAY 'NO MORE ROOM ON DISK FILE'						000DEA		
118	033200	MOVE 'I' TO WS-EOF-SW						000DFE	040	
119	033300	GO TO C999-EXIT.						000E02		COND GO TO 130
120	033400	C200-READ.						000E06		
121	033500	READ TAPE-FILE INTO WS-WORK-AREA						000E06	022 045	
122	033600	AT END								
123	033700	MOVE 'I' TO WS-EOF-SW						000E3A	040	
124	033800	GO TO C999-EXIT.						000E3E		COND GO TO 130
125	033900	C300-CHK-FOR-INDENT.						000E42		
126	034000	IF WK-8-12 IS EQUAL TO 'INDEN'						-UNCOND-	047	
127	034100	OR WK-8-12 IS EQUAL TO 'ID DI'							047	
128	034200	OR WK-8-12 IS EQUAL TO 'ID D'							047	
129	034300	GO TO C999-EXIT.						000E56		COND GO TO 130
130	034400	C999-EXIT.						000E56		COMES FROM 119,
131	034500	EXIT.						000E56		RETURN TO 077

Exhibit 1

CA1 Report Example

SQ-NBR	1-----6	PROGRAM -ID	DCDLOGIC	01/01/2001	14:56	PAGE	09	73-----80	#	REFERS TO DATA-NAME AT THIS
40	012000	05 WS-EOF-SW	PIC X	VALUE '0'.				BL=4 039	>	MOVE 'I' TO # (118, 123)
41	012100	88 WS-EOF		VALUE '1'.					>	IF # (78)
42	012200	05 WS-CTL-ERROR-SW	PIC X	VALUE '0'.				BL=4 03A	>	MOVE 'I' TO # (88, 92)
43	012300	88 WS-CONTROL-ERROR		VALUE '1'.					>	IF # (71)
44	012400	05 WS-CNT	PIC S9(3)	COMP-3	VALUE ZERO.			BL=4 03B	>	ADD 1 TO # (110)
45	012500	01 WS-WORK-AREA.						BL=4 040	>	IF # > 56 (75)
(99, 121)										READ TAPE-FILE @22 INTO #
(116)										WRITE DISK-REC @29 FROM #
46	012600	05 WK-FILLER	PIC X(7)	VALUE SPACES.				BL=4 040		
47	012700	05 WK-8-12	PIC X(5)	VALUE SPACES.				BL=4 047	>	IF # = 'INDEN' (126) , 'ID DI'
(127) ,										'D D' (128)
48	012800	05 WK-FILLER-2	PIC X(68)	VALUE SPACES.				BL=4 04C		

Exhibit 2

CA2 Report Example

Specifying PARM Options

DCD III is controlled by the use of PARM options included within the PARM field. However, when using DCD III PROCs to execute the Alternate Compile Listing Facility, the entering of PARM fields is done by means of PROC Symbolics.

This heading lists the PARM options along with a description of the use of each option. Also, in the next heading is a list showing corresponding PROC Symbolics for PARM options.

How these PROC Symbolics are entered into the JCL is shown in many ways in the heading “**Examples**” in this section.

A more detailed explanation on the use of options is found in the two headings:

1. **Use of DCD III PARM Options**
2. **Use of DCD III PROCs and Symbolics**

PARM Options

ASKIP NOASKIP	Account for SKIP1, SKIP2, and SKIP3 lines in the DCD III Source Listing Report.
BASIS <u>NOBASIS</u>	This option may be used in Independent Mode of DCD III to specify the use of BASIS cards to modify the COBOL program. When this option is used, an extra DD (BASIS) is required which contains the IBM format BASIS cards.
CA1	For older CA-Optimizer or COBOL/Aid users. Moves Procedure Division Optimizer information (six-digit displacements, “-UNEXEC-” and “-UNCOND-”) into columns 73 through 80 of the Source Listing Report. This allows DCD III narrative to be placed where CA-Optimizer narrative was. (See CA1 Exhibit in “Other Features Supported”.)
CA2	For older CA-Optimizer or COBOL/Aid users. CA2 also automatically invokes CA1. Data Division CA-Optimizer information (base registers and displacements) is moved into columns 73 through 80 of the Source Listing Report, allowing DCDIII narrative to be placed where CA-Optimizer data originally was located. CA-Optimizer input without CA2 in effect causes DCD III correlation data to be placed on the following line. (See CA2 Exhibit in “Other Features Supported”.)
CA3	For older CA-Optimizer or COBOL/Aid users. Moves Procedure Division Optimizer information (six-digit displacement and “-UNEXEC-”) into print positions 126-133 of the print line. This allows DCD III data to be placed where CA-Optimizer information originally was located.
	Notes: 1. When using the CA n options, only use one (CA1, or CA2, or CA3 2. For newer CA-Optimizer, optionally user parm option PMO or R3N.
<u>CALL</u> NOCALL	Produce the CALL Statements Report.
<u>DYCALLS</u> NODYC	Add Dynamic CALL information to the CALL report.
<u>CHECK</u> NOCHECK	Used when COMPILE mode is used. This will remove source code where the sequence numbers are lower than previous sequence numbers. This is necessary when running with some older releases of the CA-Optimizer.
CICS	This option is always turned on. However, for CICS COBOL programs, the CICS program should be run through the CICS pre-processor prior to being run through DCD III.
CLN=nn	Used to specify to DCD III the same number used in the compiler step in the LINECOUNT= option. This option is only used when COMPILE is used as a DCD III option. It is not required for VS COBOL II. If it is not specified and VS2 is not used as a DCD III option, a default will be created which is 3 less than the number used in the LNCNT= PARM field of DCD III. (This option is used by DCD III to determine the presence of page EJECTs when reading in the compiler SYSPRINT file.)

<u>COAID</u> <u>NOCO Aid</u>	This option indicates to DCD III that the compiler listing coming into DCD III was also run through the commercial COBOL/Aid software package. When this option is used, DCD III options CA1 or CA2 or CA3 may also be used.
COMPILE	This option indicates that the Alternate Compile Listing Facility will be run in COMPILE MODE (i.e., in conjunction with the COBOL compiler). See " Compile Mode Considerations " in this section. When not specified the default (INDEP option) will be used.
COF <u>NOCO F</u>	Forces an older compiler standard where COPY statements on the same line with an 01 record with a different 01 record name in the COPY member will use the 01 name within the COPY rather than the 01 record preceding it.
<u>COPY</u> NOCOPY	Produce the COPY Statements Report. For CICS, DL1, or DB2 programs also produce a separate INCLUDE report if INCLUDEs are found.
C68	Use when COBOL 68 is used.
<u>DDC</u> NODDC	Produce the Data Division Condensed Cross Reference Report.
DL1	This option is always turned on. However, for DL1 COBOL programs, the DL1 Programs should be run through the DL1 pre-processor prior to being run through DCD III.
<u>DNF</u> NODNF	Print a warning message for names not found in the Data Division when resolving Procedure Division statements.
EIB	This option may be used to insert COPY member DFHEIBLK into the COBOL program for cross-referencing CICS names EIBDATE, EIBTIME, and other EIB fields. (Do not use this option if COPY member DFHEIBLK is already brought in, as duplicate names will be created.)
<u>ERRORS</u> NOERRORS	NOERRORS may be used to omit the printing of errors. A summary count of all errors is still provided. This does not affect the return code produced by DCD III.
<u>FGCONSTS</u> NOFGCONSTS	Produce the Figurative Constants Report.
<u>FTB</u> FTO NOFTO F16	<p>This option is for printing FROM-TO field positions relative to the 01 record that the field belongs to. FTB prints the FROM-TO field positions in columns 73-80 if these columns are not already used. FTO forces the FROM-TO in 73-80 positions.</p> <p>This option will insert the FROM Field position in columns 1 through 6.</p>

HFT	Use HEXADECIMAL rather than DECIMAL FROM-TO positions in 73-80.
IDDLITS <u>NOIDDLITS</u>	Include Data Division Literals in the LITERALS report.
IEJECT <u>NOIEJECT</u>	Ignore EJECT lines when creating the DCD III Source Listing Report.
INDEP	This option is defaulted to when the COMPILE option is not used. It indicates that DCD III is not being run in conjunction with the COBOL compiler.
<u>IREFS</u> NOIREFS	This option produces Indirect References showing where verb activity resides in REDEFINES, overlapping field positions and group fields for this field.
IRn (n=1-6)	Use IR1 to IR6 to limit Indirect References to following: IR1 (Indirect Changed), IR2 (Indirect Used), IR3 (Indirect Tested), IR4 (Indirect Changed & Used), IR5 (Indirect Used & Tested), IR6 (Indirect Changed & Tested).
IRN	Force Indirect Narrative even if there is no direct narrative.
IRX	Keep Indirect References in main report, eliminating overflow report.
<u>LITERALS</u> NOLITERALS	Produce the Literals Report.
LNCNT=nnn <u>LNCNT=60</u>	Sets the maximum lines per page for all DCD III reports. Up to three digits may be used in specifying LNCNT.
NIS	Re-arrange the SORT sequence of the COBOL narrative produced by DCD into Primary sequence by compiler sequence number rather than a verb name.
NRSPACE <u>NONRSPACE</u>	Inserts a blank line in the Data Division of the Source Listing Report between the last line of narrative for one data name and the next data name line in those cases where the next data name line also has narrative associated with it. Do not use the NRSPACE option if the CA-Optimizer or COBOL/Aid is used.
NUC	Convert all NARRATIVE produced by DCD III to UPPER Case Characters.
<u>PDC</u> NOPDC	Produce the Procedure Division Condensed Cross Reference Report. If SECTIONs are present, also produce a just SECTIONs report.
<u>PMO</u> NOPMO	Only a valid option when COMpile option is used , when R3N option is not used and when the compiler option OFFSET is used . This option is only valid for VS COBOL II, release 3 or higher. When used, HEX OFFSETS are taken from the OFFSET map and placed all the way to the right of the source listing.
QUOTE <u>NOQUOTE</u> DETERMINE	Use double quote in place of single apostrophe as the delimiter for alphanumeric literals. The defaulted DETERMINE option allows using the first quote or apostrophe encountered to determine the value of QUOTE or NOQUOTE as a delimiter.
R3N	This is an option for VS COBOL II, Release 3 or later. When used, the PL and SL columns and any narrative from the right side of the compile listing will be carried through to the DCD III source listing.

<u>RESOLVE</u> NORESOLVE	Expand COPYs encountered in the source program. If COPY members are not resolved numerous warnings may appear for missing data names.
<u>SRESOLVE</u> NOSRESOLVE	Expand COPY ... SUPPRESS statements. Use SRESOLVE if COPY ... SUPPRESS statements are in the code for Compile Mode. Use SRESOLVE in Independent Mode.
<u>SORTREG=nnnnnnn</u> <u>SORTREG=600000</u>	Indicates the SORT region size in bytes. If encountering problems with SORT REGION, increase this amount slightly as needed and ensure that the number in DCB=BUFNO=nn on DCD work files is kept to 5 or smaller. REGION must be more than SORTREG. Keep SORTREG under 1,000,000.
<u>SOURCE</u> NOSOURCE	Print or suppress the Source Listing Report. While running in Compile Mode if NOSOURCE is specified, the compiler source listing is printed.
<u>SPREGS</u> NOSPREGS	Produce the Special Registers Report.
<u>STOP</u> <u>NONSTOP</u>	This option causes DCD III to immediately stop processing and issue a Condition Code of zero. If used in Compile Mode, the COBOL SYSPRINT file will be printed out. (A small amount of Initiator CPU time is invoked when this option is used.)
<u>U01</u> NOU01 UNUSED	Produce a report at the end of the Alternate Compile Listing showing all unused 01records. See Heading 'Unused 01 Records and Unused Data Names'. Use option UNU to turn off all three unused reports (U01,UDN,UPARAS).
<u>UDN</u> NOUDN UNUSED	Produce a report at the end of the Alternate Compile Listing showing all unused data names. See Heading 'Unused 01 Records and Unused Data Names'. Use option UNU to turn off all three unused reports (U01,UDN,UPARAS). This report will be turned off, if there are no data names or if NOIREF is specified.
<u>UPARAS</u> NOUPARAS <u>UPIGNORE</u> NOUPIGNORE UNUSED	Produce a report at the end of the Alternate Compile Listing showing all unused Paragraphs & Sections. See Heading 'Unused Paragraph and Section Names'. The default UPI option IGNORES reporting unused Paragraphs or Sections that are in the range of a PERFORM or PERFORM THRU. Use NOUPI to report these routines as unused. Use option UNU to turn off all three unused reports (U01,UDN,UPARAS).
<u>UNREF</u> NOUNREF	Print or suppress un-referenced data names and paragraph names in the Data Division and Procedure Division Condensed Cross Reference Reports.
<u>VERB</u> NOVERB	Produce Condensed VERB Report.
VR3 <u>NOVR3</u>	Emulate 1974 Version 3 COBOL (and earlier versions), in which compiler line numbers are not generated for SKIPs and EJECTs. For use in Compile Mode.
<u>VS2</u> NOVS2	Use NOVS2 for COBOL older program prior to the 1985 COBOL standard.

Corresponding PROC Symbolics

<u>PARM Option</u>	<u>Corresponding PROC Symbolic</u>	<u>PARM Option</u>	<u>Corresponding PROC Symbolic</u>
ASKIP NOASKIP	default OTHER='NOASK,...'	INDEP	default unless PROC COMACL is used.
BASIS NOBASIS	OTHER='BAS,...' default	IREFS NOIREFS IR1 – IR6 IRN IRX	IREFS=, IREFS=NO, OTHER='IRn' n=1-6 OTHER='IRN' OTHER='IRX'
CA1 CA2 CA3	OTHER='CA1,...' OTHER='CA2,...' OTHER='CA3,...'	LNCNT=nnn	LINECNT=nnn,
CALL NOCALL	CALL= CALL=NO,	NIS NUC	OTHER='NIS,...' OTHER='NUC,...'
DYCALLS NODYCALLS	default OTHER='NODYC'	NRSPACE	OTHER='NRS,...'
CHECK NOCHECK	default OTHER='NOCHE,...'	PDC NOPDC	CPRPCDV=, CPROC DV=NO,
CICS	always on	PMO NOPMO	default OTHER='NOPMO,...'
CLN=nnn	COMLCNT=nnn,	QUOTE NOQUOTE DETERMINE	OTHER='QUO,...' default OTHER='DET,...'
COAID NOCOAID	OTHER='COA,...' default	R3N	OTHER='R3N,...'
COMPILE	default in PROC COMACL	RESOLVE NORESOLVE	RESOLVE=, RESOLVE=NO,
COPY NOCOPY	COPY= COPY=NO,	SRESOLVE NORESOLVE	default OTHER='NOSRE,...'
C68	OTHER='C68,...'	SORTREG=n	SORTREG=nnnnnnnn,
DDC NODCC	CDATDV=, CDATADV=NO,	SOURCE NOSOURCE	SOURCE=, SPREGS=NO,
DL1	always on	SPREGS NOSPREGS	default SPREGS=NO,
DNF NODNF	default OTHER='NODNF,...'	STOP	OTHER='STOP,...'
EIB	OTHER='EIB,...'	UNREF NOUNREF	UNREF=, UNREF=NO,
ERRORS NOERRORS	default OTHER='NOERR,...'	U01,UDN,UPA,UPI UNUSED NOU01 NOUPI	default OTHER='NOUNU' OTHER='NOU01' OTHER='NOUPI'
FGCONSTS NOFGCONSTS	FIGCON=, FIGCON=NO,	VERB NOVERB	default OTHER='NOVER,...'
FTB FTO NOFTO HFT	default OTHER='FTO,...' OTHER='NOFTO,...' OTHER='HFT,...'	VR3 NOVR3	OTHER='VR3,...' default
IDDLITS NOIDDLITS	OTHER='IDD,...' default	VS2 NOVS2	default OTHER='NOVS2,...'
IEJECT NOIEJECT	OTHER='IEJ,...' default		

Note: The PROC symbolic **OTHER** is used to enter one or multiple PARM options for which there are no corresponding symbolics. For example: **OTHER='CA3,NOASK,NOVS2'**

JCL Examples

All the examples within this section use one of four DCD III PROCs. These four PROCs are:

1. COMACL - Compile Mode
2. DCDACL - Independent Mode
3. LIBACL - Independent Mode for Librarian
4. PANACL - Independent Mode for Panvalet

These examples all assume that the PROCs have been set to print all of the reports available within the Alternate Compile Listing Facility. The first ACLF report (the SOURCE listing) accounts for 85% or more of the paper generated. The remaining reports take less than 15% in paper size.

Within each of the examples, capitalized letters should be left coded as they are shown and non-capitalized letters require the user to enter in the names or entries required to complete the JCL.

Example 1

Print all of the Alternate Compile Listing Facility Reports. The input COBOL program is stored on a partitioned data set. The COBOL program does contain COPY statements. DCD III is going to be run in Independent Mode.

```
//step EXEC DCDACL
//DCD.COBOLIN DD DSN=user.source(progname),DISP=SHR
//DCD.COPYLIB DD DSN=user.copylib,DISP=SHR
```

Example 2

Print all of the Alternate Compile Listing Facility Reports. Do not print unreferenced names within the Condensed Cross Reference Reports. The Input COBOL program is stored on a Librarian Data Set. The COBOL program also contains COPY statements. DCD III is going to be run in Independent Mode for Librarian.

```
//step EXEC LIBACL,UNREF=NO,SRCLIB='librarian.dsn'
//LIB.SYSIN DD *
-OPT EXEC
-SEL progname,pswd
-EMOD
-END
/*
//DCD.COPYLIB DD DSN=user.copylib,DISP=SHR
```

Example 3

Print all of the Alternate Compile Listing Facility Reports with the exception of the Data Division Condensed Cross Reference Report. The input COBOL program is stored on a Panvalet Data Set. The COBOL program does not contain COPY statements. (If any COPY statements are found, DCD III is told not to resolve them.) DCD III is going to be run in Independent Mode for Panvalet.

```
//step EXEC PANACL,CDATADV=NO,RESOLVE=NO,
//SRCLIB='panvalet.dsn'
//PAN.SYSIN DD *
++WRITE WORK,progname
/*
```

Example 4

Print The Source Listing Report. Do not print any of the other reports. There are no COPY statements. DCD III is going to be run in Independent Mode. The input COBOL program is stored on a sequential file.

```
//STEP EXEC DCDACL,CALL=NO,COPY=NO,FIGCON=NO,  
// LITERAL=NO,SPREGS=NO,CDATADV=NO,CPROC DV=NO,OTHER='UNU,NOVER'  
//DCD.COBLIN DD DSN=sequential.file.UNIT=disk,  
// VOL=SER=volser,DISP=(OLD,KEEP)
```

Example 5

Print all of the Alternate Compile Listing Facility Reports. Run DCD III in Compile Mode.

```
//step EXEC COBUL,... (or other compile PROC name)  
//COB.SYS PRINT DD DSN=&&COMPLIST,  
// DISP=(MOD,PASS),UNIT=SYSDA,  
// SPACE=(CYL,(1,2)),  
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)  
//COB.SYS IN DD DSN=user.pds (cobolpgm),DISP=SHR  
//step EXEC COMACL
```

Example 6

Do not print the Data Division and Procedure Division Condensed Cross Reference Reports. Run DCD III in Compile Mode. COPY statements using the SUPPRESS clause may be present. If they are found, DCD III will override the SUPPRESS clause and expand the COPY clause.

```
//step EXEC COBUCL,... (or other compile PROC name)  
//COB.SYS PRINT DD DSN=&&COMPLIST,  
// DISP=(MOD,PASS),UNIT=SYSDA,  
// SPACE=(CYL,(1,2)),  
// DCB=(RECFM=FB,LRECL=121,BLKSIZE=1210)  
//COB.SYS IN DD DSN=user.pds(cobolpgm),DISP=SHR  
//step EXEC COMACL,CDATADV=NO,CPROC DV=NO,  
// OTHER=SRE  
//DCD.COPYLIB DD DSN=user.copylib,DISP=SHR
```

JCL to Execute DCD III Without Using PROCs

The JCL necessary to execute the Alternate Compile Listing Facility is listed here and is informational. It is recommended however, that PROCs be used for the actual execution of DCD III. The rest of this manual is designed for the use of one of the four PROCs provided.

```
//DCD      EXEC  PGM=DCDMAIN,REGION=6144K,
//  PARM=(options)
//*
//STEPLIB   DD  DSN=user.loadlib,DISP=SHR
//*
//DCDWK01  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK02  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK03  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK04  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK05  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK06  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK07  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK08  DD  UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//*
//PRINT     DD  SYSOUT=*,DCB=BLKSIZE=133
//*
//SYSOUT    DD  DUMMY=*,DCB=BLKSIZE=121
//SORTMESS  DD  DUMMY=*,DCB=BLKSIZE=133
//SORTLIB   DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01  DD  UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02  DD  UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03  DD  UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//*
//CLEANUP   DD  DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL   DD  DSN=user.pds(DCDCNTRL),DISP=SHR
//*
```

See the next page for possible additional DD cards.

Extra DD statements listed here may be required:

- BASIS - only required if IBM's BASIS is used. See "Other Features Supported". Also, use the BASIS parm option.
- COBOLIN - required if Independent Mode is used. Points to the COBOL program. See "Examples" for use of this DD.
- COPYLIB - required in Independent Mode if COPY statements are present. Required in Compile Mode if COPY statements with the SUPPRESS clause are used and the SRESOLVE option is used. Must point to the PDS where the COPY members are kept. This must be the same PDS(s) pointed to in the SYSLIB statement used by the COBOL compiler.
- INFILE - required if Compile Mode is used. Points to the SYSPRINT file saved from the compile step. See the COMACL PROC for an example of this DD.
- PRTCMLR - required if Compile Mode is used. This is a SYSOUT= file and contains the compiler SYSPRINT file without the compiler source listing. See the COMACL PROC for an example of this DD.

The REGION is set to 6144K. To make smaller, consult the Installation Instructions for guidelines on REGION size.

The DCB=BUFNO is set to 5 for MVS. If increasing, increase REGION size.

See Installation Instructions for information on the CONTROL DD.

PROCs Provided With the DCD III System

Four PROCs are provided. They are listed here and are shown on the next four pages. Not all of these may be installed on your system.

See the heading “Examples” for examples of using these PROCs.

1. **COMACL** - Compile Mode
2. **DCDACL** - Independent Mode
3. **LIBACL** - Independent Mode for Librarian
4. **PANACL** - Independent Mode for Panvalet

```
//COMACL PROC SOURCE=, * PRODUCESOURCE REPORT
//          CALL=,      * PRODUCE   CALL          REPORT
//          COPY=,      * PRODUCE   COPY           REPORT
//          FIGCON=,    * PRODUCE   FIGURATIVE CONSTANT REPORT
//          LITERAL=,   * PRODUCE   LITERAL         REPORT
//          SPREGS=,    * PRODUCE   SPECIAL REGISTER  REPORT
//          CDATADV=,   * PRODUCE   CONDENSED DATA DIV REPORT
//          CPROCDV=,   * PRODUCE   CONDENSED PROC DIV  REPORT
//          RESOLVE=,   * RESOLVE   COPY MEMBERS WHEN FOUND
//          UNREF=,     * INCLUDE   UNREF NAMES IN CROSS REFERENCE
/*  USE    XXXX =NO ABOVE TO TURN OFF (E.G. COPY=NO TO OMIT COPY REPORT)
//          OTHER=';', * USED TO ENTER OTHER PARM OPTIONS
//          COMLCNT=00, * COMPILER LINE COUNT FROM LAST STEP
//          LINECNT=60, * NUMBER OF LINES TO PRINT PER PAGE
//          BUF=5,      * USE BUF=5
//          PRINT='*', * SEND PRINTER OUTPUT TO MSGCLASS
//          REG=6144K,  * REGION SIZE
//          SORTREG=600000, * MAKE 600000 MINIMUM AVAILABLE TO SORT
//          WORK=SYSDA * UNIT=SYSDA,DISK OR OTHER
/*  PROC TO RUN THE 'ALTERNATE COMPILE LISTING' FACILITY OF DCD III
/*  IN COMPILE MODE
/*  MARBLE COMPUTER, INC. 1-800-252-1400 PROC=COMACL  RELEASE 3.6
/*                                     LM080108

//DCD    EXEC PGM=DCDMAIN,REGION=&REG,
//          PARM=(&SOURCE.SOU,&CALL.CAL,&COPY.COP,&FIGCON.FGC,
//          &LITERAL.LIT,&SPREGS.SPR,&CDATADV.DDC,&CPROCDV.PDC,
//          COM,'CLN=&COMLCNT',
//          &RESOLVE.RES,&UNREF.UNR,&OTHER,
//          'SOR=&SORTREG','LNC=&LINECNT')
/*  INSERT STEPLIB HERE IF NECESSARY
//CLEANUP DD DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
/*  INSERT COPYLIB HERE IF NECESSARY
//DCDWK01 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK02 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK03 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK04 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK05 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK06 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK07 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK08 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//INFILE DD DSN=&&COMPLIST,DISP=(OLD,PASS)
//PRINT DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//PRTCMPLR DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTMESS DD DUMMY,DCB=BLKSIZE=133
//SORTWK01 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SYSOUT DD DUMMY,DCB=BLKSIZE=121
/*
//RETCMPLR EXEC PGM=GIVEBACK,COND=((1,EQ,DCD),EVEN)
/*  GIVEBACK IS EXECUTED WHEN:
/*  1. THE SOURCE LISTING IS NOT PRINTED
/*  2. THE DCD STEP IS NOT RUN
//INFILE DD DSN=&&COMPLIST,DISP=(OLD,DELETE)
//PRTFILE DD SYSOUT=&PRINT,DCB=BLKSIZE=133
```

Exhibit 3

Compile Mode PROC – COMACL

```
//DCDA CL  PROC SOURCE=,      *  PRODUCE  SOURCE          REPORT
//          CALL=,           *  PRODUCE  CALL            REPORT
//          COPY=,           *  PRODUCE  COPY             REPORT
//          FIGCON=,         *  PRODUCE  FIGURATIVE CONSTANT REPORT
//          LITERAL=,        *  PRODUCE  LITERAL          REPORT
//          SPREGS=,         *  PRODUCE  SPECIAL REGISTER   REPORT
//          CDATADV=,        *  PRODUCE  CONDENSED DATA DIV REPORT
//          CPROCDV=,        *  PRODUCE  CONDENSED PROC DIV  REPORT
//
//          RESOLVE=,        *  RESOLVE  COPY MEMBERS WHEN FOUND
//          UNREF=,          *  INCLUDE  UNREF NAMES IN CROSS REFERENCE
//
//*  USE XXXX=NO ABOVE TO TURN OFF (E.G. COPY=NO TO OMIT COPY REPORT)
//
//          OTHER=' ',       *  USED TO ENTER OTHER PARM OPTIONS
//          LINECNT=60,      *  NUMBER OF LINES TO PRINT PER PAGE
//          BUF=5,           *  USE BUF=5
//          PRINT='*',       *  SENT PRINTER OUTPUT TO MSGCLASS
//          REG=6144K,       *  REGION SIZE
//          SORTREG=600000,  *  MAKE 600000 MINIMUM AVAILABLE TO SORT
//          WORK=SYSDA      *  UNIT=SYSDA, DISK OR OTHER
//
//*  PROC TO RUN THE 'ALTERNATE COMPILE LISTING' FACILITY OF DCD III
//*  BY ITSELF, WITHOUT MAKING REFERENCE TO THE COBOL COMPILER
//
//*  MARBLE COMPUTER, INC.   1-800-252-1400  PROC=DCDA CL    RELEASE 3.6
//*                                     LM080108
//DCD  EXEC  PGM=DCDMA IN,REGION=&REG,
//      PARM=(&SOURCE.SOU,&CALL.CAL,&COPY.COP,&FIGCON.FGC,
//      &LITERAL.LIT,&SPREGS.SPR,&CDATADV.DDC,&CPROCDV.PDC,
//      &RESOLVE.RES,&UNREF.UNR,&OTHER,
//      'SOR=&SORTREG','LNC=&LINECNT')
//
//*  INSERT STEPLIB HERE IF NECESSARY
//
//CLEANUP DD  DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL DD  DSN=USER.PDS (DCDCNTRL ), DISP=SHR
//
//*  INSERT COPYLIB HERE IF NECESSARY
//
//DCDWK01 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK02 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK03 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK04 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK05 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK06 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK07 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK08 DD  UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//
//PRINT DD  SYSOUT=&PRINT,DCB=BLKSIZE=133
//
//SORTLIB DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTMESS DD  DUMMY,DCB=BLKSIZE=133
//SORTWK01 DD  UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD  UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD  UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SYSOUT DD  DUMMY,DCB=BLKSIZE=121
//
```

Exhibit 4

Independent Mode PROC - DCDACL

```

//LIBACL PROC SOURCE=,      * PRODUCE SOURCE REPORT
//      CALL=,             * PRODUCE CALL REPORT
//      COPY=,             * PRODUCE COPY REPORT
//      FIGCON=,           * PRODUCE FIGURATIVE CONSTANT REPORT
//      LITERAL=,          * PRODUCE LITERAL REPORT
//      SPREGS=,           * PRODUCE SPECIAL REGISTER REPORT
//      CDATADV=,          * PRODUCE CONDENSED DATA DIV REPORT
//      CPROCDV=,          * PRODUCE CONDENSED PROC DIV REPORT
//      RESOLVE=,          * RESOLVE COPY MEMBERS WHEN FOUND
//      UNREF=,            * INCLUDE UNREF NAMES IN CROSS REFERENCE
/* USE XXXX=NO ABOVE TO TURN OFF (E.G. COPY=NO TO OMIT COPY REPORT)
//      OTHER=';',         * USED TO ENTER OTHER PARM OPTIONS
//      LINECNT=60,        * NUMBER OF LINES TO PRINT PER PAGE
//      SRCLIB='LIBRARAN.SOURCE', * NAME OF LIBRARIAN FILE
//      LIBPGM=LIBRARAN,   * EXECUTABLE MEMBER NAME ON LOAD LIB
//      BUF=5,             * USE BUF=5
//      PRINT='*',         * SEND PRINTER OUTPUT TO MSGCLASS
//      REG=6144K,         * REGION SIZE
//      SORTREG=600000,    * MAKE 600000 MINIMUM AVAILABLE TO SORT
//      WORK=SYSDA        * UNIT=SYSDA, DISK OR OTHER
/* PROC TO RUN 'ALTERNATE COMPILE LISTING' FACILITY OF DCD III
/* WITH LIBRARIAN AND NOT WITH THE COBOL COMPILER
/* MARBLE COMPUTER, INC. 1-800-252-1400 PROC=LIBACL RELEASE 3.6
/* LM080108
//LIB EXEC PGM=&LIBPGM,PARM='NJTA,NRJS' ** CHECK LIBRARIAN PARMS
/* INSERT STEPLIB HERE IF NECESSARY
//SYSPRINT DD SYSOUT=&PRINT
//LIST DD SYSOUT=&PRINT
//MASTER DD DSN=&SRCLIB,DISP=SHR
//OSJOB DD DSN=&&TEMP,DCB=BLKSIZE=6160,UNIT=&WORK,
// DISP=(,PASS),SPACE=(CYL,(2,2))
//DCD EXEC PGM=DCDMAIN,REGION=&REG,
// PARM=(&SOURCE.SOU,&CALL.CAL,&COPY.COP,&FIGCON.FGC,
// &LITERAL.LIT,&SPREGS.SPR,&CDATADV.DDC,&CPROCDV.PDC,
// &RESOLVE.RES,&UNREF.UNR,&OTHER,
// 'SOR=&SORTREG','LNC=&LINECNT')
/* INSERT STEPLIB HERE IF NECESSARY
//COBOLIN DD DSN=&&TEMP,DISP=(OLD,DELETE)
//CLEANUP DD DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
/* INSERT COPYLIB HERE IF NECESSARY
//DCDWK01 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK02 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK03 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK04 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK05 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK06 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK07 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK08 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//PRINT DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTMESS DD DUMMY,DCB=BLKSIZE=133
//SORTWK01 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SYSOUT DD DUMMY,DCB=BLKSIZE=121

```

Exhibit 5

Librarian Independent Mode PROC - LIBACL

```

//PANACL PROC SOURCE=, * PROCUCE SOURCE REPORT
// CALL=, * PRODUCE CALL REPORT
// COPY=, * PRODUCE COPY REPORT
// FIGCON=, * PRODUCE FIGURATIVE CONSTANT REPORT
// LITERAL=, * PRODUCE LITERAL REPORT
// SPREGS=, * PRODUCE SPECIAL REGISTER REPORT
// CDATADV=, * PRODUCE CONDENSED DATA DIV REPORT
// CPROCDV=, * PRODUCE CONDENSED PROC DIV REPORT
// RESOLVE=, * RESOLVE COPY MEMBERS WHEN FOUND
// UNREF=, * INCLUDE UNREF NAMES IN CROSS REFERENCE
/* USE XXXX=NO ABOVE TO TURN OFF (E.G. COPY=NO TO OMIT COPY REPORT)
// OTHER=' ', * USED TO ENTER OTHER PARM OPTIONS
// LINECNT=60, * NUMBER OF LINES TO PRINT PER PAGE
// SRCLIB='PANVALET.SOURCE', * NAME OF PANVALET FILE
// BUF=5, * USE BUF=5
// PRINT='*', * SEND PRINTER OUTPUT TO MSGCLASS
// REG=6144K, * REGION SIZE
// SORTREG=600000, * MAKE 600000 MINIMUM AVAILABLE TO SORT
// WORK=SYSDA * UNIT=SYSDA, DISK OR OTHER
/* PROC TO RUN THE 'ALTERNATE COMPILE LISTING' FACILITY OF DCD III
/* WITH PANVALET AND NOT WITH THE COBOL COMPILER
/* MARBLE COMPUTER, INC. 1-800-252-1400 PROC=PANACL RELEASE 3.6
/* LM080108
//PAN EXEC PGM=PAN#1,
/* INSERT STEPLIB HERE IF NECESSARY
//SYSPRINT DD SYSOUT=&PRINT
//PANDD1 DD DSN=&SRCLIB,DISP=SHR
//PANDD2 DD DSN=&&TEMP,DCB=BLKSIZE=6160,UNIT=&WORK,
// DISP=(,PASS),SPACE=(CYL,(2,2))
//DCD EXEC PGM=DCDMAIN,REGION=&REG,
// PARM=(&SOURCE.SOU,&CALL.CAL,&COPY.COP,&FIGCON.FGC,
// &LITERAL.LIT,&SPREGS.SPR,&CDATADV.DDC,&CPROCDV.PDC,
// &RESOLVE.RES,&UNREF.UNR,&OTHER,
// 'SOR=&SORTREG','LNC=&LINECNT')
/* INSERT STEPLIB HERE IF NECESSARY
//COBOLIN DD DSN=&&TEMP,DISP=(OLD,DELETE)
//CLEANUP DD DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
/* INSERT COPYLIB HERE IF NECESSARY
//DCDWK01 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK02 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK03 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK04 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK05 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK06 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK07 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK08 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//PRINT DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTMESS DD DUMMY,DCB=BLKSIZE=133
//SORTWK01 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SYSOUT DD DUMMY,DCB=BLKSIZE=121

```

Exhibit 6

Panvalet Independent Mode PROC - PANACL

Introduction to the Alternate Compile Listing Facility

The Alternate Compile Listing Facility of DCD III has one primary function. That is to provide an alternative listing from the one produced by the COBOL compiler with information provided in a way which will reduce substantially the time spent in maintaining a COBOL program.

The following DCD III reports are available through the Alternate Compile Listing Facility of DCD III for replacing significant parts of the COBOL compiler listings:

1. Source Listing (Replaces compiler source listing)
2. Condensed VERB Report
3. CALL Statements
4. COPY Statements
5. Figurative Constants
6. Literals
7. Special Registers
8. Data Division Condensed Cross Reference
(Replaces compiler data name cross reference)
9. Procedure Division Condensed Cross Reference
(Replaces compiler procedure name cross reference)
10. Unused Paragraphs and Sections
11. Unused 01 Records
12. Unused Data Names
13. Overflow report for Indirect References
(Only exists for fields with multiple lines of Indirect References – Most Indirect References are shown in the Data Divisions)

Some reports produced by the COBOL compiler (e.g. DMAP and PMAP reports) have no corresponding reports produced by the Alternate Compile Listing Facility of DCD III. Whether using commercial packages such as COBOL/Aid, the CA-Optimizer or just running the compiler stand alone, features are available within DCD III to produce one complete source listing for the COBOL programmer to work from.

Samples of the reports produced by the Alternate Compile Listing Facility along with a description of each are provided in this manual. See “Overview of Alternate Compile Listing Facility Reports”.

Overview of Alternate Compile Listing Facility Reports

1. Source Listing

The Alternate Compile Listing Facility Source Listing is the main report. While other DCD III and compiler cross reference reports center around the Source Listing Report, most maintenance of the COBOL program in DCD III is done without consulting cross-reference reports.

Besides holding a listing of the COBOL program, the DCD III Source Listing Report also contains uniquely formatted cross-reference information. The formation of this information is done in one format for the Data Division and is done in another format for the Procedure Division.

DATA DIVISION

In the Data Division part of the listing, formatting is done as follows:

1. A Procedure Division look-alike statement is generated.
2. The data name from the left-hand side of the listing is represented by an # on the right hand side of the listing.
3. The compiler sequence number of the Procedure Division Statement(s) is put in parentheses.

For example, the statement:

```
05 WS-NEXT-COLUMN PIC S9(4) COMP VALUE ZERO.> Add 1 to # (1019)
```

indicates that a Procedure Division statement ADD 1 TO WS-NEXT-COLUMN will be found at compiler line number 1019 within the program.

If another data name is involved in the associated Procedure Division statement, then the compiler (or DCDIII) sequence number of that data name is given following an @ character.

For example:

```
05 WS-NEXT-COLUMN PIC S9(4) COMP VALUE ZERO. > If # = WS-MAX-COLUMN @321 (1005)
```

If the same Procedure Division statement is used more than once, then multiply sequence numbers are provided. For example:

```
05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO. > Move WS-MAX-PD-DIGITS @322 to # (939,962,1017)
```

In some instances, only a part of the Procedure Division statement is provided. For example:

```
05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO. > Perform-Until # = ZERO (951,974,1031)
```

Most instances, however, have the entire COBOL statement from the Procedure Division shown. For example:

```
05 WS-NUMBER-TIMES          PIC S9 (3)  COMP-3 VALUE ZERO.    > Subtract 1 from # (1040)
```

In many cases, where the Procedure Division statements are similar, but relate to different operands, the narrative will not reprint the left most part of the Procedure Division statement that is unchanged between the similar statements. For example:

```
05 WS-MAX-PD-DIGITS        PIC S9 (3)  COMP-3 VALUE +5.      > If # = 3 (942,965,1022,1050), 4 (945, 968,1025, 1064)
                                Move 03 to # (497), 04 to # (501)
```

When no narrative appears along side a Data Division name, this indicates that the data name is unreferenced.

When narrative is present, Indirect References will also be shown for procedure division Activity that is present for redefined fields of this field and other group field and any field that has overlapping field positions of this field. See bolded line just below.

* * *

SQ-NBR	1----	6	PROGRAM-ID	DCDLOGIC	01/01/01	14:56	PAGE	09	73----	80	#	REFERS TO DATA-NAME AT THIS	SQ-NBR
320	014000	05	WS-NEXT-COLUMN	PIC S9 (4)	COMP	VALUE ZERO.						> Add 1 to # (1019) If # > WS-MAX-COLUMN @321 (1005) Move 02 to # (1008) Set # to W34-INDEX @276 (1041) Set W34-INDEX @276 TO # (1020) Indirectly Changed @ 318	
321	014100	05	WS-MAX-COLUMN	PIC S9 (4)	COMP	VALUE +34						> If WS-NEXT-COLUMN @320 > # (1005) If WS-MAX-PD-DIGITS @322 > # (1007) Move 34 to # (893) Indirectly Changed @ 318	
322	014200	05	WS-MAX-PD-DIGITS	PIC S9 (3)	COMP-3	VALUE +5.						If # = 3 (942,965,1022,1050), 4 (945,968, 1025,1064) > If # > WS-MAX COLUMN @321 (1007) Move # to WS-NUMBER-TIMES @324 (939, 962, 1017) Indirectly Changed @ 318	
323	014300	05	WS-NARR-CODE-INC	PIC S9 (3)	COMP-3	VALUE ZERO.						> Compute S-WORK-NARRATIVE-CODE @299 = # (682, 694, 705) Move 03 to # (674) , 06 to # (678) , ZERO to # (670) Indirectly Changed @ 318	
324	014400	05	WS-NUMBER-TIMES	PIC S9 (3)	COMP-3	VALUE ZERO.						> Move WS-MAX-PD-DIGITS @322 to # (939, 962, 1017) Perform-Until # = ZERO (951, 974, 1031) Subtract 1 from # (1040) Indirectly Changed @ 318	
325	014500	05	WS-RECORDS-TO-SORT	PIC S9 (9)	COMP	VALUE ZERO.							

Exhibit 7

Source Listing Report - Data Division

PROCEDURE DIVISION

The cross-reference information for the Procedure Division accommodates two types of information.

a. **SQ-NBR=S**

The first type references back to the Data Division (Procedure Division statements such as MOVE or ADD referencing a data name which resides in the Data Division). For these, DCD III provides a sequence number referencing the line where the data name resides in the Data Division. For example:

```
1017      MOVE WS-MAX-PD-DIGITS  TO WS-NUMBER-TIMES.      322 324
```

The first data name, WS-MAX-PD-DIGITS, is documented by the first number on the right, 322, which points to the Data Division sequence number where WS-MAX-PD-DIGITS resides. The second number, 324, references the Data Division sequence number where WS-NUMBER-TIMES resides. See Exhibit 8.

b. **LOGIC FLOW**

The second type, references transfer of control statements such as GO TO or PERFORM, with references to or from another area of the program or statements which reference an external program. DCD III handles these by providing appropriate narrative on the right and (if not external to the program) by providing a sequence number of where the transfer of control is going to or coming from. An example of a GO TO is provided here:

```
1028      GO TO B999-EXIT.                                GO TO 1044
1044      B999-EXIT.                                       COMES FROM GO TO 1028
```

If the transfer of control statement is conditional on the action of an IF, AT END, or other conditional statement, then the four letters COND will precede the generated narrative. See the GO TO in Exhibit 8.

PERFORMS are handled in the following manner:

1030	PERFORM B500-FLOAT-FL2-IN THRU B599-EXIT.	PERFORM 1036 THRU 1042
	.	
	.	
1036	B500-FLOAT-FL2-IN.	PERFORMED BY 1030
	.	
	.	
1042	B599-EXIT. EXIT.	RETURN TO PERFORM AT 1030

Statements that transfer control to or from another program are documented accordingly with statements such as:

PROCEDURE DIVISION.	BEGIN OF PROGRAM
CALL 'DCDPRINT'.	CALL TO EXTERNAL PROGRAM
GOBACK.	GOBACK

* * *

SQ-NBR 1-----6	PROGRAM-ID	DCDLOGIC	01/01/2001 14:56	PAGE 23	73-----80	SQ-NBR=S	LOGIC FLOW
1017 033400	MOVE	WS-MAX-PD-DIGITS TO	WS-NUMBER-TIMES.			322 324	
1018 033500							
1019 033600	ADD	1	TO WS-NEXT-COLUMN.			320	
1020 033700	SET	W34-INDEX	TO WS-NEXT-COLUMN.			276 320	
1021 033800							
1022 033900	IF	WS-MAX-PD-DIGITS = 3				322	
1023 034000		SET FL2-INDEX TO 3				292	
1024 034100	ELSE						
1025 034200	IF	WS-MAX-PD-DIGITS = 4				322	
1026 034300		SET FL2-INDEX TO 2				292	
1027 034400	ELSE						
1028 034500		GO TO B999-EXIT.					COND GO TO 1044
1029 034600							
1030 034700	PERFORM	B500-FLOAT-FL2-IN THRU B599-EXIT					PERFORM 1036 THRU 1042
1031 034800	UNTIL	WS-NUMBER-TIMES = ZERO.				324	
1032 034900							
1033 035000	B299-EXIT. EXIT.						RETURN TO PERFORM AT 880
1034 035100							
1035 035200							
1036 035300	B500-FLOAT-FL2-IN.						PERFORMED BY 950 973 1030
1037 035400	MOVE	FL2-ALPHA-POS (FL2-INDEX) TO W34-POS (W34-INDEX).				291 292 275 276	
1038 035500	SET	FL2-INDEX UP BY 1.				292	
1039 035600	SET	W34-INDEX UP BY 1.				276	
1040 035700	SUBTRACT	1 FROM WS-NUMBER-TIMES.				324	
1041 035800	SET	WS-NEXT-COLUMN TO W34-INDEX.				320 276	
1042 035900	B599-EXIT. EXIT.						RETURN TO PERFORM AT 973 1030
1043 036000							
1044 036100	B999-EXIT.						COMES FROM GO TO 1028
1045 036200	EXIT.						END OF OUTPUT PROCEDURE FROM 511

Exhibit 8

Source Listing Report - Procedure Division

2. Condensed VERB Report

The Condensed VERB Report shows all references for each VERB in the Procedure Division. Some verbs such as PERFORM are broken up into two categories (e.g. PERFORM & PERFORM THRU)

The Verb sequence numbers are listed in a string of numbers at the right of VERB.

The following sequence is used for this report:

1. VERB
2. Sequence Number

See Exhibit 9 below for an example of the Condensed VERB Report.

VERB REPORT FOR PROG103										
ACCEPT	1203	1569	2045							
ADD	982	995	1256	1290	1337	1498	1499	1503	1602	
CLOSE	1572									
COMPUTE	1545	1547	1549	1551	1623	1625	1627	1629	1660	1783
DISPLAY	2041	2042	2043	2163						
.										
.										
.										
STRING	1802									
SUBTRACT	961	984								
WRITE	1337	1924	1927	2011	2013					

Exhibit 9

Condensed VERB Report

3. CALL Statements

The CALL Statement Report shows all ENTRYs into a program and CALLs out of the program in a concise report provided for quick and easy analysis of CALL activity within a program.

For Dynamic CALL names, up to two extra lines may precede the documentation for this line show any Data Division literal and any immediate 'MOVE literal to Dynamic-Call-Name'.

Each ENTRY or CALL is listed on one line. The number of parameters associated with that ENTRY or CALL is also listed on the same line along with a sequence number for finding the ENTRY or CALL within the program.

The following sequence is used for this report:

1. CALL name (or ENTRY name)
2. Sequence number in the Procedure Division of where the CALL or ENTRY is found.

4. COPY Statements

The COPY Statements Report gives a listing of all COPY members used within the program and for CICS, DL1 & DB2 programs gives a listing of all INCLUDE members.

Each COPY statement is listed on one line. Additional information associated with the COPY statement such as SUPPRESS, the REPLACING clause and the clause OF LIBRARY-NAME have their presence indicated as shown in Exhibit 9. The division where the COPY statement resides is also shown, as is the sequence number of where the COPY statement may be found.

The following sequences are used for this report:

1. COPY text-name
2. Sequence number of where the COPY statement resides

See Exhibit 10 below for an example of CALL & COPY statement formatting.

CALL STATEMENTS

478	PROGRAM	ENTRY	DCDLOGIC	USING	1	PARAMETER
482		CALL	DCDPRINT	USING	2	PARAMETERS
653		CALL	DCDPRINT	USING	2	PARAMETERS
702		CALL	DCDPRINT			
552		CALL	DCDSTAMP			

COPY STATEMENTS

DATA DIVISION	344	COPY	DCDLINK	SUPPRESS		OF LIBRNAME
DATA DIVISION	105	COPY	PARATYP2		REPLACING	
DATA DIVISION	204	COPY	PDNARTBL			

INCLUDE STATEMENTS

DATA DIVISION	356	INCLUDE	PDINCLUDE
DATA DIVISION	219	INCLUDE	SALESTB3

Exhibit 10 CALL and COPY Statements Reports

5. Figurative Constants

The Figurative Constants Report lists occurrences of each Figurative Constant by sequence number of where they occur in the program.

Figurative Constants include the following: ZERO(S) (ES), SPACE(S), HIGH-VALUE(S), LOW-VALUE(S), QUOTE(S) and ALL literal.

For purposes of listing the Figurative Constants, the plural and single form of the word are treated and listed as one. For example, ZERO, ZEROS and ZEROES are listed as ZERO(S) on the report.

ALL literals are listed in the Literals Report, not in the Figurative Constants Report. The Literals Report directly follows the Figurative Constants Report. Any occurrences of the ALL literal are listed first within the Literals Report. See Exhibit 10.

The following sequence is used for this report:

1. Figurative Constants in alphabetic order
2. Sequence number of where the Figurative Constant is used.

See Exhibit 11 for an example of formatting of the Figurative Constants Report.

* * *

FIGURATIVE CONSTANTS

LOW-VALUE(S)	095	097	486	493										
SPACE(S)	098	099	328	330	322	334	336	394	487	494	568	633	645	662
	711	722	759	771	783	800	833	886	1018					
ZERO(S)	101	171	172	173	316	317	318	319	320	323	324	325	630	638
	667	671	681	704	708	715	719	764	776	780	797	830	943	962

Exhibit 11

Figurative Constants Report

6. Literals

The literals Report lists all occurrences of ALL literal, nonnumeric literals within quotes and numeric literals by sequence number of where they occur in the program.

ALL literal items are listed first, putting them closer to the Figurative Constants Report just above or ahead of this report.

Thirty characters are provided for listing the non-numeric literal including the quotes that surround them. If the literal is not over 28 characters, then the entire non-numeric literal is shown. If the literal is over 28 characters, then only the first 26 characters are shown, followed by two periods and the ending quote. See Exhibit 11 for two examples of a literal over 28 characters long.

The following sequence is used for this report:

1. ALL literals are placed ahead of other literals
2. Literal value (for non-numeric literals, the surrounding quotes also figure in the sorting sequence)
3. Sequence number of where the literal is used

See Exhibit 12 for an example of the Literals Report.

* * *											
LITERALS											
ALL '9'	849	850	858	864	1007						
+34	321										
+5	322										
'PF2'	486										
'PF3'	493										
'000 UNUSED ..'	254	256									
'0726 ALTER NNNN TO PROCED..'	1082										
'1'	369	397									
'2'	370										
'3'	371										
'A'	398										
'COND'	1098										
'DCDLOGIC END WS DCDLOGIC'	340										
'E'	457										
'1'	424										
'N'	383	502	871	888	892						
'0'	460										
'Y'	144	147	329	331	333	335	337	353	356	358	361
	375	378	382	386	389	393	404	407	410	419	426
	442	445	472	537	619	742	851	859	893	906	932
01	178	627	896								
1	959	982	1030	1039	1049	1050	1051				
2	957	980	1037								
	953	954	976	977	1033	1034	1061				

Exhibit 12

Literals Report

7. Special Registers

The Special Registers Report lists all occurrences of Special Registers.

Special Registers refer to compiler generated areas that may be used in conjunction with specific COBOL features. The following are examples of Special Registers:

1. CURRENT-DATE
2. DATE
3. DEBUG-ITEM
4. DEBUG-SUB-1
5. LINAGE-COUNTER
6. RETURN-CODE
7. SORT-RETURN
8. TALLY
9. TIME

Only those Special Registers found within the program will be listed in this report. If no Special Registers are used, the report will not appear.

The following sequence is used for this report:

1. Special Registers
2. Sequence number of where the Special Register is used.

See Exhibit 13 for an example of the Special Registers Report.

* * *

SPECIAL REGISTERS

CURRENT-DATE	744
RETURN-CODE	604 843 849 1283
TIME-OF-DAY	746

Exhibit 13

Special Registers Report

8. Data Division Condensed Cross Reference

The Data Division Condensed Cross Reference Report provides an alphabetic listing of the data names within the Data Division.

What is unique about this report is that no references are given to the Procedure Division statements that reference the data name. The only sequence number given points to the field's location within the Data Division. The cross reference given at that location within the Source Listing Report gives in COBOL everything that happens to this field.

Besides listing the data name and the sequence number of its location, four characters are reserved alongside the listing to provide the following information about the data field:

1. The level number (01-49, etc.) associated with this field
2. The letter G for those fields which are GROUP items
3. The letter's INDX for index items
4. The letter's FILE for file names

It is recommended that this report be used in place of the compiler cross reference when using the DCD III Source Listing Report for the following reasons:

1. The Cross Reference Report is a much less used report when using the DCD III Source Listing Report.
2. The narrative alongside each field in the Data Division provides a complete cross reference for that field.
3. The Condensed Cross Reference saves approximately 70% of the paper generated by a compiler cross reference.

The following sequence is used for this report:

1. Data name
2. Sequence number of the data name

See Exhibit 14 for an example of this report.

DCD III – Alternate Compile Listing Facility

01/01/2001 15:26

DATA DIVISION CONDENSED CROSS REFERENCE FOR DCDLOGIC

PAGE 27

05 0316 AL-ALTER-CMPLR-NBR	:	88 0370 LINK-PRINT-JCL-DSNAME	:	88 0120 PAR-2-LAST-TOKEN-IN-PROG
05 0317 AL-ALTER-COLUMN-NBR	:	88 0375 LINK-PRINT-LAYOUTS	:	88 0147 PAR-2-MULTIPLE-OPER...
05 0318 AL-GO-TO-CMPLR-NBR	:	20 0374 LINK-PRINT-LAYOUTS-SW	:	05 0146 PAR-2-MULTIPLE-OPERAND
05 0319 AL-PARA-CMPLR-NBR	:	88 0378 LINK-PRINT-RECORDS05	:	05 0150 PAR-2-NAME-COLUMN-NU
G 01 0092 CAFE-DATA-ENVIRON-NAME	:	20 0377 LINK-PRINT-RECORDS-SW	:	05 0149 PAR-2-NAME-EXP-LINE-NUMBR
01 0089 DCDLOGIC-BEGINS-HERE	:	88 0382 LINK-PRINT-SOURCE-LISTING	:	88 0129 PAR-2-OUTPUT-PROCEDURE
01 0339 DCDLOGIC-ENDS-HERE	:	20 0381 LINK-PRINT-SOURCE-LISTING-SW	:	88 0130 PAR-2-OUTPUT-PROCEDURE-TH
10 0282 FL-ALPHA-POS	:	88 0386 LINK-PRINT-SYS-REC-ANAL	:	05 0112 PAR-2-PARA-OR-SECT
INDX 0283 FL-INDEX	:	20 0385 LINK-PRINT-SYS-REC-ANAL-SW	:	88 0117 PARA-2-PARAGRAPH-NAME
G 05 0281 FL-NUMBER-ALPHA	:	88 0393 LINK-PRINT-UNREF	:	88 0125 PAR-2-PERFORM
05 0280 FL-NUMBER-NUMERIC	:	88 0389 LINK-PRINT-VERB-ANALYSIS	:	88 0126 PAR-2-PERFORM-THRU
G 01 0279 FL2-NUMBER-RECORD	:	20 0388 LINK-PRINT-VERB-ANALYSIS-SW	:	88 0119 PAR-2-PRIOR-TOKEN-OF-PARA-S
10 0291 FL2-ALPHA-POS	:	05 0419 LINK-PROGRAM-NAME	:	88 0138 PAR-2-PROGRAM-ENTRY
INDX 0292 FL2-INDEX	:	05 0458 LINK-QUOTE-VALUE	:	G 01 0106 PAR-2-RANGES-FOR-PARAS
G 05 0290 FL2-NUMBER-ALPHA	:	G 01 0345 LINK-RECORD	:	05 0114 PAR-2-RELEVANT-SECTION
05 0287 FL2-NUMBER-NUMERIC	:	G 05 0438 LINK-RELEASE-DATE	:	88 0118 PAR-2-SECTION-NAME
G 01 0286 FL2-NUMBER-RECORD	:	10 0439 LINK-RELEASE-MONTH	:	05 0152 PAR-2-SOURCE-BEGIN-NBR
05 0288 FL2-NUMBER-ZZ-999	:	10 0441 LINK-RELEASE-YEAR	:	05 0155 PAR-2-SOURCE-END-NBR
10 0348 LINK-100	:	G 15 0350 LINK-REPORT-OPTIONS-1-25	:	05 0108 PAR-2-SOURCE-LINE-NUMBER
G 10 0349 LINK-100-R1	:	05 0423 LINK-RETURN-CODE	:	88 0134 PAR-2-STOP-RUN
G 10 0404 LINK-100-R2	:	05 0427 LINK-SORT-REGION	:	05 0110 PAR-2-TOKEN-COLUMN-NBR
05 0445 LINK-ADDR-OF-DCDPRTLN	:	88 0398 LINK-SYS-REC-REC-ANAL-ALL	:	05 0116 PAR-2-VERB-CODE
88 0417 LINK-BASIS	:	88 0400 LINK-SYS-REC-ANAL-DIR-INDIR	:	05 0162 PAR-3-EXPANDED-LINE-NUMBE
05 0416 LINK-BASIS-SW	:	88 0397 LINK-SYS-REC-ANAL-ONLY-01	:	05 0166 PAR-3-FILLER
88 0414 LINK-COMPILE-MODE	:	88 0401 LINK-SYS-REC-ANAL-ONLY-DIR	:	05 0164 PAR-3-NARRATIVE
05 0413 LINK-COMPILE-MODE-SW	:	20 0396 LINK-SYS-REC-ANAL-SW	:	G 01 0160 PAR-3-NARRATIVE-FOR-SR-LST
88 0456 LINK-COPY-REPLACE-FOUND	:	88 0430 LINK-SYSTEM-EXPIRED	:	FILE 0056 PARA-FILE-2
05 0455 LINK-COPY-REPLACE-FOUND-SW	:	05 0429 LINK-SYSTEM-EXPIRED-SW	:	FILE 0065 PARA-FILE-3
88 0383 LINK-DO-NOT-PRINT-SRC-LIST	:	05 0443 LINK-TIME-OF-DAY	:	01 0098 PARA2-NAME
88 0394 LINK-DO-NOT-PRINT-UNREF	:	88 0433 LINK-TRIAL-TAPE-EXPIRED	:	01 0099 PARA3-NAME

Exhibit 14

Data Division Condensed Cross Reference

9. Procedure Division Condensed Cross Reference

The Procedure Division Condensed Cross Reference Report provides an alphabetic listing of the paragraph/section names within the Procedure Division. If sections are also used, a separate report shows just SECTIONS and is shown at the end of the Paragraph/Section report.

This report is similar in nature to the Data Division Condensed Cross Reference Report in that no references are given to Procedure Division statements which reference the paragraph or section. The only sequence number given points to its location within the Procedure Division. If the paragraph or section is referenced by other Procedure Division statements such as GO TO or PERFORM, narrative will appear alongside the line in the Source Listing Report where the paragraph or section resides. See Exhibit 8 examples of the narrative that appears alongside the Procedure Division lines that contain a paragraph or section (i.e., lines 1033, 1036, 1042, and 1044).

The format of this report, like the Data Division report, has four characters to provide additional information as follows:

1. For paragraph names, these four characters are blank
2. For section names, these four characters contain the letters- SECT

Like the previous report, it is recommended that this report be used over the compiler procedure name report for the same reasons given there, among them a 70% savings in paper.

The following sequence is used for this report:

1. Paragraph/section name
2. Sequence number of the procedure name

See Exhibit 15 for an example of this report.

DCD III – Alternate Compile Listing Facility

01/01/2001 15:26

PROCEDURE DIVISION CONDENSED CROSS REFERENCE FOR DCDLOGIC

PAGE 29

	0470	003-BUILD-ASSIGN-NAMES	:	0640	A300-GO-TO-DEPENDING	:	0930	B140-TEST-FOR-NUM-1-FLOAT
	0494	010-REBUILD-PDNARTBL-COPY	:	0666	A399-EXIT	:	0953	B150-TEST-FOR-NUM-2-FLOAT
	0504	020-SORT-DATA-NAMES	:	0668	A400-PERFORM-TYPE	:	0976	B170-MOVE-IN-NARRATIVE
	0514	090-GOBACK	:	0714	A499-EXIT	:	0983	B180-WRITE-NARR-RECORD
SECT	0517	A-BUILD-WORK-RECORDS	:	0717	A500-ALTER-PARA	:	0987	B1980-RETURN-NEXT-RECD
	0518	A 010-OPEN-PARA-FILE-2	:	0775	A599-EXIT	:	0999	B199-EXIT
	0522	A020-READ-FIRST-RECORD	:	0778	A600-CANCEL-PROGRAM	:	1003	B200-FLOAT-IN-NEXT-NUMB
	0528	A030-BUILD-WORK-RECORDS	:	0792	A699-EXIT	:	1011	B210-SET-UP-FOR-FLOAT
	0533	A040-CLOSE-PARA-FILE-2	:	0794	A700-ALL-OTHERS	:	1033	B299-EXIT
	0537	A050-END-OF-INPUT-PROCEDURE	:	0825	A799-EXIT	:	1036	500-FLOAT-FL2-IN
	0543	A100-BUILD-WORK-RECORDS	:	0827	A999-EXIT	:	1042	B599-EXIT
	0544	A110-GO-TO	:SECT	0828	B-BUILD-NARRATIVE-RECORDS	:	1044	B999-EXIT
	0561	A120-GO-TO-DEPENDING-ON	:	0830	B010-OPEN-FILE-A	:SECT	1047	C-REBUILD-PDNARTBL-COPY
	0567	A130-PERFORM-TYPE	:	0834	B020-RETURN-FIRST-SORT-REC	:	1048	C010-TEST-MAX-OF-3
	0578	A140-ALTER-PARA	:	0850	B030-PROCESS-ALL-RECORDS	:	1062	C020-TEST-MAX-OF-4
	0584	A160-CANCEL-PROGRAM	:	0843	B030-RETURN-SECOND-SORT-REC	:	1076	C999-EXIT
	0590	A170-ALL-OTHERS	:	0862	B040-CLOSE-FILE-3	:SECT	1078	D-CHECK-FOR-INSERT-COND
	0601	A190-READ-NEXT-RECORD	:	0866	B050-END-OUTPUT-PROCEDURE	:	1079	D010-CHECK-LENGTH
	0610	A199-EXIT	:	0870	B100-PROCESS-ALL-RECS	:	1084	D020-INSERT-COND
	0613	A200-GO-TO	:	0907	B120-CHECK-MATCH-TO-PREV-REC	:	1097	D999-EXIT
	0637	A299-EXIT	:	0926	B130-SET-UP-EXP-LINE-NBR	:		

01/01/2001 15:26

SECTIONS ONLY REPORT FOR DCDLOGIC

PAGE 30

SECT	0517	A-BUILD-WORK-RECORDS	:SECT	1047	C-REBUILD-PDNARTBL-COPY
SECT	0828	B-BUILD-NARRATIVE-RECORDS	:SECT	1078	D-CHECK-FOR-INSERT-COND

Exhibit 15

Procedure Division Condensed Cross Reference

01/01/2001 15:26

INDIRECT REFERENCES (See UCL Report for fields not shown here)

PAGE 31

DATA-NAME	SEQ-#	INDIRECT ACCESS FOR THIS FIELD
LINE-28	00025	Indirectly changed@40,60,80,102,103,104,106,107,108,109,111,117,121,125,129,13,133,134,137,138 Tested @ 105,111, Used @ 110,111,117,121,135,129,130,133,137,138

Exhibit 16

Overflow report to Indirect References

Use of DCD III PARM Options

This heading is provided to make the user aware of the way in which PARM fields are used to specify options for DCD III. For a complete list of the options available and more information

on executing the Alternate Compile Facility: finish reviewing this heading, review the heading

“Use of DCD III PROCS and Symbolics,” then consult the Table of Contents and go to the beginning headings in this section.

Options to control execution are provided through the use of PARM fields. The user provides control information and uses the PARM field as the means to pass this information through to the DCD III program.

Two types of control fields may be entered via the PARM field.

1. Keyword options which consist of:
 - a. A leftmost identifying field
 - b. An equals (=) sign
 - c. A value field following the equal sign
2. Non-keyword options which consist of:
 - a. A leftmost identifying field (with or without the two letters NO ahead of it)

Keyword options should be enclosed in single apostrophes as shown here:

`'LNCNT=60'`

There is no reason to enclose non-keyword options in single apostrophes.

The following shows the PARM field for indicating the printing of the Source Listing Report both with and without the prefix NO ahead of it:

`SOURCE`
`NOSOURCE`

When NO precedes the PARM field specified, the option is turned off. When the PARM option is specified without the NO prefix, the option is turned on.

For both non-keyword and keyword PARM fields, the following information applies:

1. Most of the PARM fields are of the non-keyword type.
2. For non-keyword PARM fields and for the leftmost identifying field in a keyword PARM field, the field may be abbreviated to the first 3 characters.

The following examples are provided:

`'LNC=60'`
`SOU`
`NOSOU`

3. When the PARM option is not specified in the PARM field, then an internally specified default is used. This default is provided for every non-keyword and for every keyboard option in this manual by the use of an underline. Two examples are provided:

LNCNT=nn
LNCNT=60

SOURCE
NOSOURCE

The above two examples indicate that the default is to print 60 lines per page and that unless otherwise specified, the Source Listing Report is to be printed.

4. PARM fields must be separated by commas. Also, the entire PARM field should be enclosed by parentheses. An example follows:

```
PARM=(SOU,NOCALL,NOCOPY,'LNCNT=58',LIT)
```

5. PARM fields may be placed in any order within the parentheses.

The PARM field itself is a keyword parameter which follows the PGM=keyword on the JCL EXEC statement. An example follows:

```
//STEP1 EXEC PGM=DCDMAIN,REGION=6144K,  
// PARM=(SOU,CAL,NOLIT,'SOR=600000')
```

The users may execute DCD III programs by providing their own JCL or they may use PROCs (catalogued JCL) which are listed in this manual and should be available for use at your installation.

Sample JCL for invoking DCD III without the use of a PROC is provided in this manual. However, all the examples used in this manual are provided with the assumption that the user is using standard PROCs provided with this software package.

When using DCD III PROCs, PARM fields are entered through the use of PROC Symbolics. Entering the PARM fields through the use of symbolics is done in a different manner from what has been shown here for entering the PARM fields. The use of these symbolics is discussed in the next heading, “Use of DCD III PROCs and Symbolics.”

Use of DCD III PROCs and Symbolics

The execution of DCD III, as with other software, is made easier through the use of cataloged JCL (PROCs).

This heading provides general information about using PROCs to execute DCD III. It also provides general information about the use of PROCs and specifically symbolic parameters.

USE OF SYMBOLICS

The use of symbolics when used in a PROC, is an easy way of providing a substitute value within the cataloged JCL just prior to execution of that JCL.

When a symbolic is used in the JCL, it must be preceded by an ampersand (&). An example follows:

```
//DCDWK01 DD UNIT=&WORK, SPACE= (CYL, (2,2) ), DCB=BUFNO=&BUF
```

The two symbolics are &WORK and &BUF. Prior to execution, these symbolics must be replaced with valid or real values.

Typically, default values for these symbolics are provided within and at the top of the PROC as shown here:

```
//DCDA CL PROC WORK=SYSDA, BUF=5
```

The ampersand (&) is not used when listed at the top of the PROC. It is only used when embedded within the actual JCL where the substitution value will be placed.

See the DCD III PROC DCDA CL provided in Exhibit 15 and identify the use of symbolic parameters within that PROC.

To use symbolics during the execution of the PROC to override the default defined within the PROC, just add the symbolic in the following format after the EXEC procname. An example follows:

```
//STEP1 EXEC DCDA CL,WORK=DISK,BUF=4
```

Look again at the DCDA CL PROC and look at the use of symbolics within the PARM field and also at the corresponding symbolic at the top of the PROC. The following symbolics are listed below:

```
//DCDA CL PROC SOURCE=,  
//          CALL=,  
.  
.  
.  
//          PARM=(&SOURCE.SOU,&CALL.CAL,...)
```

Notice that SOURCE= and CALL= have no value associated with them. When the symbolics &SOURCE along with the terminating period and &CALL along with the terminating period are removed and replaced with no value, the substitution JCL looks as follows:

```
//  PARM=(SOU,CAL,
```

which matches the manner in which PARM fields were shown in the previous heading. To turn an option off with the DCDACL PROC, use a symbolic=NO on the EXEC line. An example follows:

```
//STEP1 EXEC DCDACL,SOURCE=NO
```

The resulting PARM field after replacement of &SOURCE will look like this:

```
//  PARM=(NOSOU,CAL,
```

Most of the options are turned on or off within DCD III PROCs by either providing no value after the symbolic or by using symbolic=NO on the EXEC PROC JCL. An example follows:

```
//DCD EXEC DCDACL,SOURCE=,  
//          CALL=NO,  
//          COPY=,  
//          FIGCON=NO
```

In the above JCL, the Source Listing Report and COPY Statements Report will be produced while the CALL Statements and Figurative Constants Reports will not be produced.

It is important to note that the symbolic used within the PROC and the PARM option are not always the same. For example:

PARM field: FGCONSTS
Symbolic: FIGCON

Also, the PARM field FGCONSTS may be abbreviated to 3 characters, while the symbolic may not be abbreviated and must be used as shown.

Executing DCD III When Using a PROC

To execute DCD III when using a PROC, simply enter an EXEC DCD III PROC line, enter any changes to the PROC by using symbolics and add any indicated extra DD cards into the job stream.

The following is an example of executing the DCDACL PROC to produce all DCD III Alternate Compile Listing Facility Reports available with the exception of the Procedure Division Condensed Cross Reference Report. Also, unreferenced names will be omitted from the Data Division Condensed Cross Reference Report. The DCDACL PROC is shown on the next page.

```
//STEP1   EXEC  DCDACL,CPROCDV=NO,UNREF=NO
//COBOLIN DD    DSN=USER.LIBRARY(PROG831),DISP=SHR
//COPYLIB DD    DSN=USER.COPYLIB,DISP=SHR
```

```

//DCDACL PROC SOURCE=, * PRODUCE SOURCE REPORT
// CALL=, * PRODUCE CALL REPORT
// COPY=, * PRODUCE COPY REPORT
// FIGCON=, * PRODUCE FIGURATIVE CONSTANT REPORT
// LITERAL=, * PRODUCE LITERAL REPORT
// SPREGS=, * PRODUCE SPECIAL REGISTER REPORT
// CDATADV=, * PRODUCE CONDENSED DATA DIV REPORT
// CPROCDV=, * PRODUCE CONDENSED PROC DIV REPORT
/*
// RESOLVE=, * RESOLVE COPY MEMBERS WHEN FOUND
// UNREF=, * INCLUDE UNREF NAMES IN CROSS REFERENCE
/* USE XXXX=NO ABOVE TO TURN OFF (E.G. COPY=NO TO OMIT COPY REPORT)
/*
// OTHER=' ', * USED TO ENTER OTHER PARM OPTIONS
// LINECNT=60, * NUMBER OF LINES TO PRINT PER PAGE
// BUF=5, * USE BUF=5
// PRINT='*', * SENT PRINTER OUTPUT TO MSGCLASS
// REG=6144K, * REGION SIZE
// SORTREG=600000, * MAKE 600000 MINIMUM AVAILABLE TO SORT
// WORK=SYSDA * UNIT=SYSDA, DISK OR OTHER
/* PROC TO RUN THE 'ALTERNATE COMPILE LISTING' FACILITY OF DCD III
/* BY ITSELF, WITHOUT MAKING REFERENCE TO THE COBOL COMPILER
/*
/* MARBLE COMPUTER, INC. 1-800-252-1400 PROC=DCDACL RELEASE 3.6
/* LM080108
//DCD EXEC PGM=DCDMAIN,REGION=&REG,
// PARM=(&SOURCE.SOU,&CALL.CAL,&COPY.COP,&FIGCON.FGC,
// &LITERAL.LIT,&SPREGS.SPR,&CDATADV.DDC,&CPROCDV.PDC,
// &RESOLVE.RES,&UNREF.UNR,&OTHER,
// 'SOR=&SORTREG','LNC=&LINECNT')
/*
/* INSERT STEPLIB HERE IF NECESSARY
/*
//CLEANUP DD DUMMY,DCB=(LRECL=80,BLKSIZE=3120)
//CONTROL DD DSN=USER.PDS (DCDCNTRL ), DISP=SHR
/*
/* INSERT COPYLIB HERE IF NECESSARY
/*
//DCDWK01 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK02 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK03 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK04 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK05 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK06 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK07 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
//DCDWK08 DD UNIT=&WORK,SPACE=(CYL,(2,2)),DCB=BUFNO=&BUF
/*
//PRINT DD SYSOUT=&PRINT,DCB=BLKSIZE=133
/*
//SORTLIBDD DSN=SYS1.SORTLIB,DISP=SHR
//SORTMESS DD DUMMY,DCB=BLKSIZE=133
//SORTWK01 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=&WORK,SPACE=(TRK,(100),,CONTIG)
//SYSOUT DD DUMMY,DCB=BLKSIZE=121
/*

```

Exhibit 17

DCDACL PROC

Unused 01 Records

Note – These 3 new Unused reports are also available on a separate //CLEANUP DD with a LRECL of 80 characters and a BLKSIZE of 3120.

01 Records that are completely unused are listed in this report along with unused 77 level entries.

Redefined 01 records are shown independently from the 01 record that it redefines.

Sometimes a beginning 01 record is not referenced and is used for setting up VALUEs for a redefined 01 record that follows it. If the first 01 record is not referenced and the next redefined 01 record is referenced, the first 01 record is shown as not referenced and then a CAUTION line is printed showing that the next 01 record is used! Conversely, it is possible to have unused 01 records **follow** to add space to a previous table. These records should normally not be deleted.

If the first FD 01 record is not unused, including having no OPEN, READ, CLOSE, etc. to the associated FILE name with the FD, then the 01 record is shown as unused, otherwise it will be treated as used. A second, third or more FD 01 record for the same FD will be shown as unused if none of the fields in the respective record including that 01 are unused.

01 records are shown as unused whether or not they reside in a COPY member.

Unused Data Names

Only unused 02-49 level entries are shown in this report. 01 record names, 77 level entries, and 88 level entries are not shown. FILLERs are not shown as unused. 02-49 level entries underneath a 66 level RENAMEs clause are shown.

If the entire 01 record was shown above as unused, then individual 02-49 entries are not shown.

Also, 02-49 level entries within a COPY member are not shown as unused, as it is expected that most names in a COPY are not necessarily used in any one program. To clean up COPY book records use the System Record Analysis report in the Tracing Analysis & Other Reports section.

When all of the fields within a GROUP field are unused, then the GROUP field will be shown as unused and the individual fields will not be listed. The only time the group field will not be shown is when a larger group field containing this group field is unused.

Caution must be used before removing a field that is shown as unused. The field may be named for documentation where a FILLER could be used and the field could be necessary to proper running of the program. To prevent against removing a field that should be left in place, Indirect References are shown after each field that is shown as unused. The line numbers of where the other fields are that have overlapping field positions to this field are listed. If no Indirect References are listed then the field may be removed. If Indirect References are shown, then they must be examined carefully before removing the listed field.

The 01 record associated with each field listed in this report is shown with its line number to assist in examining a field.

Unused Paragraphs and Sections

Un-referenced Paragraphs and Sections are shown for the entire program, with the exception of those within a DECLARATIVES SECTION. All paragraphs and sections that are present in any Declaratives code are accepted as used and will never be reported as being unused.

The first tag (beyond any Declaratives Section, if present) whether it is a SECTION or a PARAGRAPH name is treated as an implied fall through and is never treated as un-referenced. If the first tag in the program is a SECTION than all paragraphs in that SECTION are treated as a fall through up until the first found STOP RUN, GOBACK, or EXIT, following a coded period within that section. If the first tag in the program is a paragraph name, than all paragraphs are treated as a fall through up until the first found STOP RUN, GOBACK, or EXIT following a coded period or the first found SECTION.

When option UPI (the default) is used to print this report, all paragraphs or sections within a PERFORM range whether it is a perform of a section or a range of paragraphs or sections will not be listed as unused, but will be treated as being used in the range of the PERFORM. To show these paragraphs or sections, use the option NOUPI.

SECTIONS if present and found to be not used, are listed in their own report ahead of PARAGRAPHS.

When a section is shown as being un-referenced, all paragraphs within this section will also be shown as un-referenced within the Paragraph report.

If a SECTION is otherwise un-referenced, but there is either an ENTRY verb into the section or there is a reference to a paragraph within the SECTION, than the SECTION will be listed as un-referenced and an additional line will immediately follow to show the presence of one or more ENTRY verbs and/or the presence of paragraphs being referenced.

If a PARAGRAPH is un-referenced, but there is one or more ENTRY verbs within the paragraph, that the paragraph will be shown as un-referenced, and an additional line will immediately follow to show the presence or one or more ENTRY verbs being present.

Before removing any SECTIONS or PARAGRAPHS, it is up to the user to insure that a fall through does not exist or perhaps the routine's access is commented out and needs to be kept.

If a fall through exists:

If it is intended, then it is recommended that this logic be well documented as to why. If it was not intended, there is the possibility of an existing PERFORM Error and the code should be looked at closely. If the user is unsure whether or not a fall through may exist within the logic, one suggestion is to test for it by adding an additional un-referenced paragraph with a single DISPLAY just ahead of the paragraph or section that shows as being un-referenced. If the DISPLAY is invoked, during any test or production running of the program, then that is proof that a fall through exists and the user is cautioned to look more seriously at why this exists.

If a fall through does not exist, then the paragraph or section with all code may be removed!

Release 3.6

DCD III

Tracing & Analysis and Other Reports

Facility

DCD III

Tracing & Analysis and Other Reports Facility**TABLE OF CONTENTS**

Use of This Section / List of Exhibits	B-4
Choice of DCD III Reports including Tracing & Analysis	B-5
Choice of DCD III PROCs	B-6
Inputting COBOL Programs and COPY Records	B-7
Other Features Supported	B-9
Specifying PARM Options	B-11
PARM Options.....	B-12
Corresponding PROC Symbolics	B-17
Control Statements	
Control Statements for System Record Analysis.....	B-18
Control Statements for Literal Tracing.....	B-21
Control Statements for Layouts	B-22
Control Statements for Verb Analysis	B-23
Control Statements for Using The MBRFETCH PROC	B-24
Control Statements for CALL Hierarchy Report.....	B-27
Control Statements for Tracing & Analysis	B-28
SELECT Control Statements	B-28
BYPASS Control Statements	B-30
JCL Examples	B-32
JCL to Execute DCD III Without Using PROCs.....	B-36
PROCs Provided for Use in This Section	B-38
Efficient Use of Work Space	B-43
Overview of All Reports including Tracing & Analysis	B-45
Data Dictionary Interface File	B-71
WRITPDS and READPDS Options	B-74

Use of This Section

This section is designed for Tracing & Analysis and use in producing other COBOL reports from one or more COBOL programs at one time. Topics that are initially necessary when first starting to use the DCD III software package are shown below and will be in the middle of this section or in another section. Topics that are designed to be referenced quickly in everyday use are found in the beginning of this section for quick reference.

When using this manual for the first time, read the following topics first:

1. Overview of DCD III COBOL Reports
2. Use of DCD III PARM Options (in Section A)
3. Use of DCD III PROCs and Symbolics (in Section A)

The second and third topics are covered in the section “Alternate Compile Listing Facility”. These topics were also topics necessary for using the Alternate Compile Listing Facility part of the DCD III software system.

After reading and reviewing the above topics, continue using this section starting at the top of the Table of Contents.

List of Exhibits

1.	PANCOBOL PROC	B-39
2.	LIBCOBOL PROC	B-40
3.	DCDCOBOL PROC	B-41
4.	MBRFETCH PROC	B-42
5.	CALL Analysis Report - Two Sequences	B-46
6.	CALL Hierarchy Report	B-47
7.	COPY Analysis Report - Member Sequence	B-48
8.	COPY Analysis Report - Program Sequence	B-49
9.	System Data Name Cross Reference	B-50
10.	System Cross Reference for Figurative Constants	B-52
11.	Table of Contents for Layouts	B-53
12.	Layout Report	B-54
13.	System Cross Reference for Literals	B-56
14.	System Paragraph Cross Reference	B-57
15.	01 Record Report - Length Sequence	B-58
16.	01 Record Report - Entry Sequence	B-59
17.	System Cross Reference for Special Registers	B-61
18.	System Record Analysis Report - 01 Record Summary	B-62
19.	System Record Analysis Report - Data Name Summary	B-63
20.	Possible Literals for each Field Traced	B-64
21.	Possible Literals showing Owner Fields	B-65
22.	Possible Literals with Possible Paths	B-66
23.	Verb Analysis Report	B-67
24.	Verb Analysis Report - Summary Page	B-67
25.	Tracing & Analysis Report	B-68

Choice of DCD III Reports

The following is a list and brief description of the COBOL reports and one file which are available by using the PROCs and PARM options shown in this section. All of the reports listed below are available for one or multiple COBOL programs. A more detailed description with sample reports is available in the, “**Overview of DCD III Other COBOL Reports**”, later in this section.

1. **CALL** Analysis Report - a listing of CALLs and ENTRYs in two report sequences
2. **CALL Hierarchy** Report - shows the organization of CALLs in a system
3. **COPY** Analysis Report - a listing of COPY member usage in two report sequences
4. System **Data Name Cross-Reference** - an expanded cross reference of Data Division names
5. **Data Dictionary** Interface File - a file containing useful information for interfacing with a Dictionary or other engineering projects. See the heading, “**Data Dictionary Interface File**”.
6. System Cross Reference for **Figurative Constants** - a listing of the use of figurative constants
7. **Layout** Report - automatic generation of layouts with several options for **custom building**
8. System Cross Reference for **Literals** - (Also, see “**Tracing of Possible Literals**” below.)
9. System **Paragraph** Cross Reference - a cross reference listing of paragraph names
10. **01 Record** Report - a cross reference of 01 records with their computed record length in two report sequences
11. System Cross Reference for **Special Registers** - a listing of the use of Special Registers
12. **System Record Analysis** Report - a listing on one or more selected records with flags showing direct or indirect data movement or comparison on every name within the record
13. **Tracing of Possible Literals by Field** - three different reports showing possible literals and paths used to move a literal to one field – Limited to **one program** at a time.
14. **Tracing & Analysis** – does tracing to find every involved name for selected names. A report is produced showing the original field selected, all overlapping fields which REDEFINE the field, all traced fields (infinite levels), and all Procedure Division references for all fields shown in DCD narrative format. – Limited to **one program** at a time.
15. **Verb Analysis** Report - a listing of the use of Procedure Division verbs with a summary page

Choice of DCD III PROCs

The three PROCs listed below are available to choose from when invoking Other COBOL Reports. A brief description is given on this page of using these 3 PROCs along with the MBRFETCH PROC. More examples are given on the next two pages and under the heading “JCL Examples”, in this section.

1. LIBCOBOL
2. PANCODOL
3. DCDCODOL

LIBCOBOL & PANCODOL

The first and second PROC may be used when the COBOL source code resides on Librarian or Panvalet files respectively.

DCDCODOL

The third PROC is to be used when the COBOL source code resides as member (or members) on a partitioned data set (PDS) or other non-Librarian or non-Panvalet file.

An example is given here using the DCDCODOL PROC:

```
//STEP1 EXEC DCDCODOL,COPY=,DATA=,PARA=
//COBOLIN DD DSN=USER.COBOL.LIB(PROG01),DISP=SHR
//          DD DSN=USER.COBOL.LIB(PROG02),DISP=SHR
//COPYLIB DD DSN=USER.COPY.LIB,DISP=SHR
```

The COBOL source code is brought in through the COBOLIN DD statement. Note that two programs were brought in. A different way of bringing in COBOL programs is through the member fetcher procedure MBRFETCH which is explained below.

COPY members are resolved by pointing the COPYLIB DD statement to the PDS(s) that contain the COPY members. The PDS(s) specified here should be the same PDS(s) specified in the SYSLIB DD in your installation’s COBOL compile PROC.

Symbolic parameters are used to specify options, such as specifying the reports desired.

MBRFETCH

A fourth PROC is available for bringing in multiple programs at once from partitioned data sets. This PROC is:

4. MBRFETCH

When this PROC is used, the DCDCODOL PROC is also used and follows the MBRFETCH PROC.

Inputting COBOL Programs and COPY Records

Several methods may be used for providing input into DCD III. These methods are described below.

The main input to DCD III when producing the reports mentioned in this section is provided through the COBOLIN DD statement. An example follows:

```
//STEP1 EXEC DCDCOBOL,DATA=,RESOLVE=NO
//DCD.COBO LIN DD DSN=USER.COBO.LPDS(PROGRAM1),DISP=SHR
//          DD DSN=USER.COBO.LPDS(PROGRAM2),DISP=SHR
//          DD DSN=USER.COBO.LPDS(PROGRAM3),DISP=SHR
```

In the above example, three COBOL programs are entered through the use of the COBOLIN DD.

When Layout Reports are being produced on just COPY records through the use of the DCD III PARM option (LOR), then the COBOLIN DD statement is used to enter the COPY records. An example follows:

```
//STEP2 EXEC DCDCOBOL,LAYOUT=,OTHER=LOR
//DCD.COBO LIN DD DSN=USER.COPI.LIBRARY(RECORD1),DISP=SHR
//          DD DSN=USER.COPI.LIBRARY(RECORD2),DISP=SHR
```

If COPY members are present when handling COBOL programs, they are resolved by pointing DCD III to user's COBOL COPY libraries. These are the same library (or libraries) pointed to in the SYSLIB DD statement of the user's COBOL compile PROCs. An example follows:

```
//STEP3 EXEC DCDCOBOL,DATA=
//DCD.COBO LIN DD DSN=USER.COBO.LPDS(PROGRAM1),DISP=SHR
//          DD DSN=USER.COBO.LPDS(PROGRAM2),DISP=SHR
//          DD DSN=USER.COBO.LPDS(PROGRAM3),DISP=SHR
//DCD.COPI LIB DD DSN=USER.COPI.LIBRARY,DISP=SHR
```

When the COBOL input is coming from Librarian or Panvalet, then the PROCs LIBCOBOL or PANCOBOL should be used and the COBOLIN DD should be omitted. An example follows:

```
//STEP4 EXEC PANCOBOL,DATA=
//PAN.SYSIN DD *
++WRITE WORK,PROGRAM1
++WRITE WORK,PROGRAM2
++WRITE WORK,PROGRAM3
/*
//DCD.COPI LIB DD DSN=USER.COPI.LIBRARY,DISP=SHR
```

The previous example assumes that there are COBOL COPY members in addition to, or in place of Panvalet (or Librarian) INCLUDEs present in the COBOL source by the use of the COPYLIB statement.

When multiple COBOL programs (or in a separate run, multiple COBOL COPY records) are brought into DCD III from partitioned data sets, a DCD III partitioned data set member fetcher utility may be used ahead of the DCDCOBOL PROC to simplify the process of entering the COBOL programs. Two examples follow:

```
//STEP5 EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCDMBR DD *
INDD=USERNAME=COBOL=MR
//MBR.USERNAME DD DSN=USER.COBOL.PDS,DISP=SHR
//STEP5A EXEC DCDCOBOL,DATA=
//DCD.COBLIN DD DSN=PASSFILE,DISP=(OLD,PASS)
//DCD.COPYLIB DD DSN=USER.COPY.LIBRARY,DISP=SHR
```

The above example pulls off all members that are specifically COBOL programs and specifically whose member name starts with the two characters (MR) and passes these COBOL programs to the DCDCOBOL PROC.

```
//STEP6 EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCDMBR DD *
INDD=USERNAME=COBRECS=TER
//MBR.USERNAME DD DSN=USER.COPY.LIBRARY,DISP=SHR
//STEP6A EXEC DCDCOBOL,LAYOUT=,OTHER=LOR
//DCD.COBLIN DD DSN=PASSFILE,DISP=(OLD,PASS)
```

The above example pulls off all members that are specifically COBOL records that begin with a valid COBOL level number and specifically whose member name starts with the three characters (TER) and passes these COBOL records into DCD III for producing Layout Reports on these records.

Further information on the member fetcher PROC is provided under the heading, “Control Statements for Using the MBRFETCH PROC”.

Other Features Supported

This heading provides a list of support for those other features both inside and outside of the standard support for the COBOL compiler that the user may have a need for. Those features supported by DCD III in this section are:

1. CICS
2. COBOL 68 and earlier
3. COBOL 74
4. COBOL 85
5. COBOL 370, COBOL for MVS & VM, COBOL 390, ENTERPRISE & Z/OS
6. DB2 / SQL
7. DL1
8. Librarian
9. Panvalet

1. CICS

CICS is fully supported by DCD III. Ideally, the CICS program should be run through the CICS pre-processor before being brought into DCD III. If the pre-processor step is done, both CICS commands and the CALL statements generated by CICS are documented. If the CICS program is not run through the pre-processor, then generated CALL statements are not documented by DCD III.

2. COBOL 68 and earlier

PARM option C68 is recommended when all programs being processed are COBOL 68.

3. COBOL 74

Include PARM option NOVS2 for COBOL 74.

4. COBOL 85

Include default PARM option VS2 to indicate the handling of VS COBOL II.

5. COBOL 370, COBOL for MVS & VM, COBOL 390, ENTERPRISE & Z/OS

Include default PARM option VS2, the same as for COBOL 85. When using the ISPF panels, a separate COBOL 370 option is included there for COBOL 370. The separate option is not needed for batch submission.

6. DB2

DB2 programs contain SQL statements. Ideally, the DB2 program should be run through the DB2 pre-processor before being brought into DCD III. If the pre-processor step is done, the CALL statements generated by SQL are documented. Narrative created by the use of Host variables in SQL statements are listed in the System Data Name Cross Reference and the Layout Reports when the LNR option is used.

7. DL1

DL1 is fully supported by DCD III. Ideally the DL1 program should be run through the DL1 pre-processor before being brought into DCD III. If the pre-processor step is done, both DL1 commands and the CALL statements generated by DL1 are documented. If the DL1 program is not run through the pre-processor, then generated CALL statements are not documented by DCD III.

8. Librarian

When INCLUDEs are used (as opposed to COPY statements), they are resolved when the COBOL source code is brought from Librarian. They are *not* listed within the COPY Analysis Report.

The PROC LIBCOBOL, which contains a Librarian step ahead of the DCD III step for pulling the COBOL programs off of the Librarian Data File, should be used. See Example 2 under the heading, “Examples”, in this section for assistance in using the LIBCOBOL PROC.

9. Panvalet

When INCLUDEs are used (as opposed to COPY statements), they are resolved when the COBOL source code is brought from Panvalet. They are *not* listed within the COPY Analysis Report.

The PROC PANCOBOL, which contains a Panvalet step ahead of the DCD III step for pulling the COBOL programs off of the Panvalet Data File, should be used. See Example 1 under the heading “Examples”, in this section for assistance in using the PANCOBOL PROC.

Specifying PARM Options

DCD III is controlled by the use of PARM options included within the PARM field. However, when using DCD III PROCs, the entering of PARM fields is done by the means of PROC Symbolics.

The heading, “PARM Options”, on the next page lists the PARM options along with a description of the use of each option. Also, in the following heading is a list showing corresponding PROC Symbolics for PARM options.

How these PROC Symbolics are entered into the JCL is shown in many ways within the “Examples” heading.

A more detailed explanation on the use of options is found in the two headings shown below within the Alternate Compile Listing Facility section.

1. Use of DCD III PARM Options
2. Use of DCD III PROCs and Symbolics

PARM Options

<u>BEGIN</u> <u>CONT</u>	Options used in conjunction with WRITPDS & READPDS options. CONT option will not initialize internal work files and is used for the 2 nd , 3 rd , or later runs using WRITPDS option.
<u>CAHIER</u> <u>NOCAHIER</u>	Produce the CALL Hierarchy Report. See the heading “ Control Statements for CALL Hierarchy ” within this section.
<u>CALL</u> <u>NOCALL</u> <u>CAP / CAC</u>	Produce the CALL Analysis Report. Two sequences will be produced. The first sequence is Program/CALL sequence. The second is CALL/Program sequence. Use CAP to produce only Program/CALL sequence. Use CAC to produce only CALL/Program sequence.
<u>CICS</u>	This option is always turned on. However, for CICS COBOL programs, the CICS program should be run through the CICS pre-processor prior to being run through DCD III.
<u>COPY</u> <u>NOCOPY</u> <u>COQ / COR</u>	Produce the COPY Analysis Report. Two sequences are produced. The first is Program sequence. The second is Member sequence. Use COQ to produce only Program sequence. Use COR to produce only Member sequence.
<u>COF</u> <u>NOCOF</u>	Forces an older compiler standard where COPY statements on the same line with an 01 record with A different 01 record name in the COPY member will use the 01 name within the COPY rather than the 01 record name preceding it.
<u>C68</u>	Use when COBOL 68 is used.
<u>DATA</u> <u>NODATA</u>	Produce the System Data Name Cross Reference. Individual options shown below may be used to enhance the DATA report.
<u>DASELECT</u>	When invoked, the control statements used for the System Record Analysis will also be used to limit the DATA Analysis Report. (See Control Statements for System Record Analysis). When used without producing the System Record Analysis Report, record names in the control statements may instead be just a group or elementary name.
<u>D88</u> <u>NOD88</u>	This option may be used to individually list out all of the multiple 88 values for each 88 level, when multiple values (e.g. VALUES ARE) are used.
<u>DA1 to DA5</u>	When used, any numeric digits (up to the number specified in this option) which immediately precedes a Data Division name will not be sorted when producing the Data Analysis Report. (As an example, 20ACCT-NAME and 32ACCT-NAME will sort together when option DA2, DA3, DA4 or DA5 is used.) Use only 1 of these options during one run.
<u>DAL</u> <u>NODAL</u>	This option is only effective when one of the options (DA1 to DA5) is used. NODAL removes the leading NODAL selected digits from the data name in the report, while DAL leaves the digits in front of the data name.
<u>DICT</u>	Produce an Interface File for any desired use- including interfacing with a commercial Data Dictionary. (This option has the same effect as invoking the SRA option, i.e. – control statements are required). The output file is on DD DCDDICT and at job end has a record length of 120 and a block size of 3120. See <u>both</u> “Data Dictionary Interface File” and “Control Statements for System Record Analysis” in this section.
<u>DDF</u> <u>CDD</u>	Option DDF is default and OPENs the DCDDICT file as OUTPUT to clear the file for each run. Option CDD does not clear the file for each run.
<u>DIT</u> <u>DRL, DRF or</u> <u>DRC</u>	Produce Direct Tracing of Literals. See “Control Statements for Literal Tracing” in this section for overview of report and for individual reports. Three reports are available. Sub-option <u>DRC</u> is the default and is for printing all three reports.

DL1 This option is always turned on. However, for DL1 COBOL programs, the DL1 programs Should be run through the DL1 pre-processor prior to being run through DCD III.

DNF
NODNF Print a warning message for data names not found in the Data Division when resolving Procedure Division statements.

EIB This option may be used to insert COPY member DFHEIBLK into the COBOL program for cross referencing CICS names EIBDATE, EIBTIME and other EIB fields from that COPY.

ERRORS
NOERRORS NOERRORS may be used to omit the printing of all errors.

FGCONSTS
NOFGCONSTS Produce the System Cross Reference for Figurative Constants.

LAYOUTS
NOLAYOUTS Produce Layout Reports on Data Division records. See the several options that follow for user customizing Layout Reports and see the heading “**Control Statements for Layouts**”.

The following options affect the LAYOUT reports:

LFD Using this option ensures that Layout Reports will be produced on all records within the File Section. This option overrides the default option NOLS1 for records in the File Section. Using this option will automatically invoke or turn on the LAYOUTS option.

LHX Using this option indicates that in addition to decimal from and to record positions, that hexadecimal from and to record positions will be produced underneath the decimal positions. Using this option automatically turns on the LAYOUTS option.

LHR Using this option turns on the LHX option. However, the hexadecimal locations will be printed on the right hand side of the Layout Report rather than underneath the decimal locations. If LNR is used as an option, this option will be treated as LHX.

LK1 Using this option will keep 01 records on the same page. No page break will occur when a new 01 is encountered.

LLA On the printing of the Layout Reports, a plus (+) character is normally used to intersect lines. Use of this option will substitute an asterisk (*) for the plus character.

LLI On the printing of the Layout Reports, a vertical bar (|) is used for vertical lines. Use of this option will substitute the letter I in place of the vertical character.

LOR This option indicates that just 01 records are being brought into DCD III for producing Layout Reports. COBOL programs are not present. When this option is used, the LAYOUTS option is turned on automatically and other report options for producing reports other than Layout Reports are not allowed. When using this option, the COBOL records are brought into the DCD III system by using the COBOLIN DD. When using this option, place a ceiling limit between 10,000 and 12,000 lines per run. (If the Member Fetcher is used with the LOR option, the associated member name will be printed on the Layout Report.) Do not use this option for COPY MEMBERS where pseudo-text is present and the pseudo-text is not in a valid COBOL format.

LSQ This option works in conjunction with the LOR option ONLY. This option generates a sequential number in the left margin for each data name found on the Layout Reports. Otherwise, these numbers are suppressed.

LPV
NOLPV This option indicates that a VALUE clause when present will be printed below the data name in The Layout Reports. This option is turned on automatically when the LAYOUTS option is Turned on. It may be turned off by using NOLPV. Notice – VALUE clauses for 88 level entries are inserted automatically on the same line of the 88, regardless of the option used here.

DCD III – Tracing & Analysis and Other Reports

<u>LPP</u> NOLPP	This option indicates that a PICTURE clause which is designed as NE (Numeric Edited) will be printed below the data name in the Layout Reports. This option is turned on automatically when the LAYOUTS option is turned on. It may be turned off by using NOLPP.
LNR	This option indicates that COBOL narrative is to be printed on the right-hand side of the Layout Report. This option is only valid for single COBOL programs. If LOR is used as an option, this option is turned off. If multiple COBOL programs are inputted, narrative will only appear alongside the first one if this option is used.
LS1 <u>NOLS1</u>	This option (if turned on) indicates that Layout Reports should be produced for records that have no elementary fields (levels 02-49) underneath them. If not turned on, (which is the default) they will be omitted. This option will be turned on automatically if option L77 is used.
<u>LTC</u> NOLTC	This option is turned on automatically if the LAYOUTS option is turned on. It indicates that a Table of Contents for the Layout Reports by record name will be produced ahead of the Layout Reports. Unless the LTS option is used, the Table of Contents will be sorted in alphabetic sequence by record name.
LTS	This option when used turns the LTC option on. However, the records in the Table of Contents will not be sorted. They will be left in sequence as they were inputted into DCD III or as they appear in the COBOL program.
L77	This option when used indicates that Layout Reports should also be produced for 77 level entries.
LITERALS <u>NOLITERALS</u>	Produce the System Cross Reference for Literals.
LNCNT=nn <u>LNCNT=60</u>	Set the maximum lines per page for all DCD III reports.
NIS	Re-arrange the SORT sequence of the COBOL narrative produced by DCD into primary sequence by compiler sequence number, then verb name. Without this option, the primary sequence is verb name.
NUC	Convert all NARRATIVE produced by DCD III to UPPER Case Characters.
PARAGRAPH <u>NOPARAGRAPH</u>	Produce the System Cross Reference.
PRANGES <u>NOPRANGES</u>	Produce a set of ranges of beginning and ending expanded (after Copy members have been resolved) sequence numbers for paragraphs and sections within a COBOL program.
QUOTE <u>NOQUOTE</u> <u>DETERMINE</u>	Use the double quote rather than the single apostrophe as a delimiter for alphanumeric literals. The defaulted DETERMINE option allows using the first quote or apostrophe encountered to determine whether to use option QUOTE or NOQUOTE as a delimiter.
RECORDS <u>NORECORDS</u>	Produce the 01 Records Report.
<u>RESOLVE</u> NORESOLVE	This option indicates DCD III will resolve COPY members by accessing the members on the data set specified on the COPYLIB DD. IF NORESOLVE is used, the COPYLIB DD is not necessary. (COPY members within COPY members are not currently resolved in this section.)

SORTREG=nnnnnnn <u>SORTREG=600000</u>	Indicates the SORT region size in bytes. Do not specify a SORTREG larger than 1,000,000.
SPREGS <u>NOSPREGS</u>	Produce the System Cross Reference for Special Registers Report.
STOP <u>NOSTOP</u>	This option causes DCD III to immediately stop processing and issue a Condition Code of zero. This option may be used when the DCD III step is embedded with other steps in a PROC and is not wanted.
SRANAL <u>NOSRANAL</u>	Produce the System Record Analysis Report. When this option is used control statements must also be used. This option is turned on automatically when the DICT option is used. When using this option see “ Control Statements for System Record Analysis ”.
The following options affect the System Record Analysis Report: <i>Limit System Record Analysis Report as follows:</i>	
RA1	RA1 - Print only as 01 summary report.
<u>RAA</u>	RAA - The default - no limiting.
RAB	RAB - Select records that have any flags turned on.
RAD	RAD - Select records that have any direct flags turned on.
RAS	RAS - Select records if the direct (S)et flag is turned on.
RAT	RAT - Select records if any (S)et flag is turned on.
S03 S15 S50 S3H S1T	Use one of these options to correct error messages DCDUF010-D or DCDUFA02-D where an OCCURS clause in one or more 01 records causes the combined table space needed for this report to be exceeded. (S03) limits all OCCURS to a maximum of 3 TIMES. (S15), a maximum of 15 TIMES. (S50), 50 TIMES. (S3H), 300 TIMES. (S1T), 1000 TIMES. Warning: The smaller the OCCURS (e.g. S03 or S15), the more likely that an existing REDEFINES overlapping the OCCURS will assist in either producing incorrect results and/or causing DCD error messages for invalid DATA DIVISION structures.
SRCOPY <u>NOSRCOPY</u>	Produce the System Record Analysis Report using COPY member names to select which 01 records are selected. (Include option SRDELCOPY if the COPY report is not desired.)
TRACE <u>NOTRACE</u>	Produce the ‘Tracing & Analysis’ report. See ‘ Tracing & Analysis Supporting Options ’ on the next page. Also, see ‘ Control Statements for Tracing & Analysis ’ in this section.
UNREF <u>NOUNREF</u>	Print or suppress un-referenced names in the System Data Name Cross Reference and System Paragraph Cross Reference. Note - when NOUNREF (the default) is used and PARM option D88 is also used, all 88 level names with either multiple 88's or the THRU clause will be listed whether or not they are referenced.
VERB <u>NOVERB</u>	Produce the Verb Analysis Report.
VR3 <u>NOVR3</u>	Causes DCD III to emulate Version 3 and earlier versions of 1974 and 1968 COBOL in which sequence numbers are not generated for SKIPs and EJECTs.
<u>VS2</u> NOVS2	Causes DCD III to scan for newer 1985 VS COBOL II format. Use NOVS2 for older 1974 COBOL.

(Continued..)

WRITPDS
READPDS

Used to store information over time and/or several runs on an internal DCD III Data Dictionary using the WRITPDS option and then retrieving the information for reporting during on run using a READPDS option. Do not use this feature when using the 'Tracing & Analysis' Report. See the heading '**WRITPDS and READPDS Options**' in this section for more information.

Tracing & Analysis Supporting Options

These options are available providing the **TRACE** option is on. See 'Control Statements for Tracing & Analysis' in this section for **SELECTing** and also for **BYPASSing** of Names.

Utility Options

YOP
NOYOP

Print a separate page showing all related TRACE options on the PRINT report.

YTEST
NOYTEST

Used to TEST all Select Statements. The only processing that is done is to test all tracing Select Statements and verify that no errors are present. This option will cause any other report options submitted to be overridden and their respective reports will not be produced.

Directly Selected Name Options

YDNARR
NOYDNARR

Produce Procedure Division Narrative for each directly selected field and list the narrative right underneath this field.

Y2PROG
NOY2P

Sort Direct Names by PROGRAM name so one program's names are done ahead of the next. The secondary sort is Alphabetic by data name. To reverse sequence, use NOY2P option.

Overlapping Option

*(Applies to **REDEFINES**, **GROUP** fields or otherwise Overlapping fields)*

Y2INDIR
NOY2INDIR

For each Direct Name selected, this option selects names for every associated REDEFINES, GROUP field, or otherwise Overlapping field.

Trace Options

(Traces infinite levels deep[up to 999 times] to get all possible fields; see TAC=)

YAP
NOYAP

Traces names selected and using overlapping fields and their record positions also traces through these fields to find all possible names that may be affected.

TAC=nnn

Limits the number of Traces to a specific number such as 010 or 025.

Supporting Options for Overlapping (Y2INDIR) and Trace (YAP) Options

YINARR
NOYINARR

Produce Procedure Division Narrative for every overlapping name selected here and list the narrative right underneath this field.

YIPROG
YIALPHA

Within each Directly Selected Name, overlapping names are sorted in program sequence or the order in which they are listed within the COBOL program. To sort in alphabetic sequence, use YIALPHA option.

YAI
NOYAI

Show the existence of overlapping names even if there are no Procedure Division references to this name. To omit names with no Procedure Division references, use NOYAI option.

Corresponding PROC Symbolics

PARM Option	Corresponding PROC Symbolic	PARM Option	Corresponding PROC Symbolic
CAHIER NOCAHIER	OTHER='CAH,...' default	NIS NUC	OTHER='NIS,...' OTHER='NUC,...'
CALL NOCALL CAC / CAP	CALL=, default OTHER='CAx,...'	PARAGRAPH	PARA=,
CDD	OTHER='CDD,...'	QUOTE NOQUOTE DETERMINE	OTHER='QUO,...' default OTHER='DET,...'
COPY NOCOPY COQ / COR	COPY=, default OTHER='COx,...'	RECORDS NORECORDS	RECORDS=, default
C68	OTHER='C68,...'	RESOLVE NOREOLVE	default RESOLVE=NO,
DATA NODATA D88 / DAn / DAL	DATA=, default OTHER='Dxx,...'	SORTREG=n	SORTREG=nnnnnnnn,
DASELECT	OTHER='DAS,...'	SPREGS NOSPREGS	OTHER='SPR,...' default
DICT	OTHER='DIC,...'	SRANAL NOSRANAL	SRA=, default
DIT	OTHER='DIT=	RA1 / RAA / RAB RAD / RAS / RAT	OTHER='RAX,...' OTHER='RAX,...'
DRL / DRF / <u>DRC</u>	OTHER='DRx,...'	S03 / S15 / S50 / S3H / S1T	OTHER='Snx,...'
EIB	OTHER='EIB,...'	note: n=numeric digit, x=letter or number	
ERRORS	default	SRCOPY	OTHER='SRC,...'
NOERRORS	OTHER='NOERR,...'	TRACE NOTRACE	TRACE=, default
FGCONSTS	OTHER='FGC,...'	TAC=nnn	TRLIMIT=nnn
NOFGCONSTS	default	default	
LAYOUTS	LAYOUT=,	UNREF	OTHER='UNR,...'
NOLAYOUTS	default	VERB	OTHER='VER,...'
LFD / LHR / LHX / LK1	OTHER='Lxx,...'	VR3	OTHER='VR3,...'
LLA / LLI / LNR / LOR	OTHER='Lxx,...'	NOVS2	OTHER='NOVS2,...'
LSQ / LS1 / LTC / LTS	OTHER='Lxx,...'		
L77	OTHER='L77,...'		
NOLPP / NOLPV / NOLTC	OTHER='NOLxx		
	note: x=letter or number		
LITERALS	OTHER='LIT,...'		
NOLITERALS	default		
LNCNT=nnn	LINECNT=nnn,		

Supporting TRACE Options

YTE / YIA	OTHER='Yxx,...'
NOYOP / NOYDN / NOY2P	OTHER='NOYxx,...'
NOY2I / NOYIN / NOYAI	OTHER='NOYxx,...'
NOYAP	OTHER='NOYAP,...'
note: x=letter or number	
WRITPDS / READPDS	OTHER='WRI,...'
WRITPDS / READPDS	OTHER='REA,...'

Note: The PROC symbolic **OTHER** is used to enter one or multiple PARM options for which there are no corresponding symbolics. For example: **OTHER='LNR,NOLPV,NOVS2'**

Control Statements for System Record Analysis

The System Record Analysis Report (& Data Dictionary Interface file) requires control statements to select records to report on. The DD for these control cards is CTLCD SRA. The maximum number of control statements must not exceed 775 in the entire file.

More than one set of reports may be run through during one run of DCD III. Each set of reports should be run on like records which ideally have the same record length and format. The data names do not need the same spelling since fields are matched up by field positions.

To break apart sets of control statements, each set of control statements must begin with a header control statement of the following format:

Column 1 - An asterisk (*) Columns 2 through 72 - Documentation for the user

Within each set of control statements, records must be selected which make up that set. Selecting records requires specifying both program name and record name. Each control statement has two fields: program-id and record name. This control statement uses the following format:

Starting in column 1 - Program-id, *followed by at least one space*
In next available column - Record Name

The field for program-id must contain either a (name) which matches to a COBOL program-id from an Identification Division or the constant -ALL-. If **-ALL-** is used, the record that follows will be pulled for selection from all programs that were entered.

The field for record name must contain either a record name which matches to an 01 record name within the COBOL program or it must be a numeric number for matching to the length of one or more records within a COBOL program. Also, See '**Selecting by COPY Name...**'.

Note - When a number is used, it may be followed by a second number to form a selection range. For example, the control statement (-ALL- 1 7500) without the enclosing parentheses may be used to select all records with an overall record length not greater than 7500 bytes. While the SRA report produced using this range may not be very meaningful, the Data Dictionary file produced may be used for several purposes such as loading data dictionaries or repositories.

An example of control statements follows:

```
//DCD.CTLCD SRA DD *
*THIS IS THE TITLE FOR THE FIRST SET
-ALL- 184
*THIS IS THE TITLE FOR THE SECOND SET
-ALL- MS-RECORD-A
-ALL- MR-RECORD
*SELECTION FOR CERTAIN RECORDS
PAYR001 RECORD-1
PAYR001 RECORD-2
PAYR002 RECORD-1
PAYR003 224
/*
```

In the example shown, the first set of statements selects all records from all programs which have a record length of 184. In case of variable length records, the largest record length applies.

In the second set, the records MS-RECORD-A and MR-RECORD are selected from all programs.

In the last set, various records were individually selected from programs PAYR001 and PAYR002. In program PAYR003, all records are selected that have an overall record length of 224.

A listing of the control statements used will be printed at the beginning of the REPORTS DD print file ahead of all other reports. The System Record Analysis report may not immediately follow these control statements.

Further limiting of the records documented in this report may be accomplished through the use of RAX options. See the heading, “PARM Options”.

Limiting DATA Analysis Report when using with SRA Report

The above control statements are also used for limiting the DATA Analysis Report, *providing* **PARM** option (**DASELECT**) is also used. When used *without* the System Record Analysis Report, the control statement field for the record name may be used to select just a group name or an elementary dataname. Also, see the **UNREF PARM** option for further limiting of the Data Analysis Report.

Selecting by COPY Name for System Record Analysis

Selecting 01 records for use in the System Record Analysis report may be done by using COPY member name. Simply specify the COPY member name and those COPYs which begin in the COPY immediately with an 01 will be selected. (Caution, if the COPY begins with other than an 01, such as an FD, it will be selected.) The end result is a full report on all SET, USED and TESTED activity (direct or indirect) on each field within COPY used throughout a system of COBOL programs.

The PARM option needed to invoke this feature is SRCOPY. The control statements have the following format:

Columns 1-7 - The constant SRCOPY= Columns
Columns 8-15 - The **COPY member name**

The DDNAME for inputting these control statements is the same CTLCDSDRA used for other System Record Analysis control statements. It is possible to combine other control statements for these SRCOPY control statements. When this is done, the other control statements must come before the SRCOPY control statements.

When the SRCOPY option is used, both the COPY Analysis Report and System Record Analysis Report will be printed. The COPY Analysis Report will list all COPY members found in all programs. The System Record Analysis Report will list only those records selected by the control statements submitted with the CTLCDSDRA DDNAME.

See Examples for ‘System Record Analysis Reports’ on *next* page

Example 1:

```
//STEP1 EXEC DCDCOBOL,OTHER='SRCOPY'  
//DCD.COBO LIN DD DSN=user.cobol(prog1),DISP=SHR  
// DD DSN=user.cobol(prog2),DISP=SHR  
//DCD.COPYLIB DD DSN=user.copylib,DISP=SHR  
//DCD.CTLCD SRA DD *  
SRCOPY=MEMBER1  
SRCOPY=MEMB333  
/*
```

This example does not combine statement formats. This example selects all 01 records which have a COPY MEMBER1 or a COPY MEMB333 at the beginning of the record. The 01 record name may come immediately before the COPY or after the first (non-comment) entry within the COPY.

Example 2:

```
//STEP2 EXEC DCDCOBOL,OTHER='SRCOPY'  
//DCD.COBO LIN DD DSN=user.cobol(prog1),DISP=SHR  
// DD DSN=user.cobol(prog2),DISP=SHR  
//DCD.COPYLIB DD DSN=user.copylib,DISP=SHR  
//DCD.CTLCD SRA DD *  
-ALL- USER-RECORD-25  
SRCOPY=MEMB5  
/*
```

This example combines control statement formats by selecting all 01 records for USER-RECORD-25 and also all 01 records which have a COPY MEMB5.

Note - SRCOPY= control statements will be ignored by DCD III if the PARM option SRCOPY is not turned on.

Control Statements for Literal Tracing

The Tracing of Literals reports (*3 individual reports are listed at bottom of page*) require control statements specifying which COBOL data fields are to be traced. Additionally, options DRL, DRF or DRC must be specified stating which reports are to be run. See 'PARM options' in this section or at the bottom of this page for specifying one of these options. Also, this report is formatted for, and limited to only one program at a time. Extra programs will be ignored.

Control statements are specified with the DD name CTLCDDLT. COBOL fields to be traced are entered starting in column 1. Multiple statements must be provided for multiple COBOL fields. If qualification is necessary to clarify a field, enter a Q, one space, then a group or record name as a qualifier control statement ahead of the COBOL field being selected. For instance, if duplicate names ACCT exist in group or record names FIELD-1 and FIELD-2, then qualification control statements for FIELD-1 or FIELD-2 must be used ahead of the field to be selected. To select ACCT OF FIELD-1 enter control statements as follows:

```
//DCD.CTLCDDLT DD *  
Q FIELD-1  
ACCT  
/*
```

If more than one level of qualification is required, enter multiple qualification statements (statements beginning with a Q) ahead of the COBOL field name control statement. Put the 01 Record Name or Group Name having the lowest level number first (01 being lowest) when specifying qualification control statements.

Up to **17** levels of tracing are accomplished by tracking literals through **MOVEs** and **SET 88-level-name TO TRUE** statements. To limit the number of tracing levels to less than 17, modify the PARM field within the DCD step of the DCDSYSTEM PROC to include 'DLT=nn' where nn is a number from 01 to 17.

A maximum of 1000 control statements are allowed. Much less are recommended. One COBOL program at a time may be processed.

PARM options are specified with the OTHER PROC symbolic. A sample is shown:
OTHER='DRC,DLT=06'

Three reports are available:

- | | |
|--|---|
| 1. Possible literals for each field traced | (Use option DRF for just report 1) |
| 2. Possible literals showing owner fields | (Use option DRL for reports 1 & 2) |
| 3. Possible literals showing possible paths | (Use option DRC for all 3 reports) |

Control Statements for Layouts

The running of Layout Reports does not specifically require the use of control statements. However, control statements may be used to limit the selection of records for Layout Reports to selected record names or selected sizes.

Two types of control statements are available, but only one type (not both) may be used within one run. The DD name for these control statements is CTLCDLAY.

One way that control statements may be used is to specify by record name (or group name within an 01 record) the records or groups that Layout Reports are to be run for. (In the case of group names, the field positions in the Layout Reports will be relative to the start of the 01 record as opposed to the start of the group field, however the first field position in the group field may be reset to one by specifying an asterisk (*) after the group field in the control statement.)

The format of this type of control statement is as follows:

Column 1 through ?	-	01 record name (or 02-48 level group name)
Column ?+1	-	SPACE
Column ?+2	-	Optional Asterisk for group names that need to be reset

An example follows:

```
//DCD.CTLCDLAY DD *  
MR-RECORD-1  
MR-RECORD-2  
MR-GROUP-FIELD-3 *  
/*
```

In the above example, the first two control statements indicate selection of two 01 record names. The third control statement indicates the selection of a group field within an 01 record and the from-to positions within the group field will be reset starting at 1 for this group.

Another type of control statement allows selection to be limited by not printing Layout Reports for records smaller than a given record length and/or larger than a given record length. Within these control statements is the ability to print just records which have a specific record length. In the case of variable length records for determining size, the largest record length applies.

The format of this type of control statement is as follows:

Columns 1 through 6	-	The constant (SELECT)
Column 7	-	The character >, < or =
Column 8 through x	-	A numeric number with no punctuation

Only two control statements are allowed. One with SELECT< and one with SELECT>. Both are not required if one is used. If SELECT= is used, then only that control statement is allowed.

An example follows:

```
//DCD.CTLCDLAY DD *  
SELECT>25  
SELECT<1000  
/*
```

In the above example, Layout Reports will not be produced for small records that are 25 characters or under in length. They will also not be printed for records that are 1000 characters or over in length.

Control Statements for Verb Analysis

The running of the Verb Analysis report does not require the use of control statements. However, control statement(s) may be used to allow printing of selected verbs. Up to 10 control statements are allowed. The field in the control statement is also limited to 10 characters.

An example follows:

```
//DCD.CTLCDVER DD *  
ALTER  
CASE GO TO  
EXIT PROGM  
GO TO  
PERFORM  
STOP RUN  
/*
```

Most verbs should be coded with one verb with no spaces. Those which require spaces like 'CASE GO TO' are shown above.

Control Statements for Using the MBRFETCH PROC

The PDS Member Fetcher PROC MBRFETCH is available for pulling one or several members from a partitioned data set (PDS) and creating one sequential data set. This data set may be used as input to the DCDCOBOL PROC. (This PROC is also embedded as one step in the DCDJCL PROC which is covered in the section on JCL PROC Analysis Reports Facility.)

The DD name for these control statements is CTLCDMBR.

Control statements are used to indicate which members are to be selected. The format of these control statements is as follows:

Columns 1 through 4	-The constant (INDD)
Column 5	- An equal sign (=)
Columns 6 through <i>a</i>	- The name of a made-up DDNAME
Column <i>a</i> +1	- An equal sign (=) with no spaces before or after it.
Columns <i>b</i> through <i>c</i>	- One of the following words: 1. ALL 2. COBOL 3. COBRECS 4. MEMBER 5. PREFIX

The remaining two fields are required if PREFIX was used as the last operand and are optional if COBOL or COBRECS was used as the last operand. They are not permitted if either ALL or MEMBER is used as an operand.

Column <i>c</i> +1	-An equal sign (=) with no spaces before or after it.
Columns <i>d</i> through <i>e</i>	-A 1 to 7 characters prefix which will be used to limit selection

An illustration of the possible combinations is listed below:

1. INDD=ddname=ALL
2. INDD=ddname=COBOL
3. INDD=ddname=COBRECS
4. INDD=ddname=MEMBER
5. INDD=ddname=PREFIX=prefix
6. INDD=ddname=COBOL=prefix
7. INDD=ddname=COBRECS=prefix

An example showing the use of these control statements is listed below:

```
//STEP1 EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCDMBR DD *
INDD=USERDD=COBOL
//MBR.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

The above example will pull off all COBOL programs from the library specified by the DD name USERDD and pass them out to a sequential data set.

If members are wanted from more than one library, then multiple DD names may be used. An example is listed below:

```
//STEP2 EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCDMBR DD *
INDD=USERDD=COBOL
INDD=OTHERDD=COBOL
//MBR.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
//MBR.OTHERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

The use of the following words for selection is described here:

ALL	COBRECS	MEMBER
COBOL	COBRECS=PREFIX	PREFIX
COBOL=PREFIX		

ALL ALL indicates that all members will be selected from the partitioned data set. This option should not be used here. Instead, the options COBOL or COBRECS should be used. An exception to this rule occurs when using the READPDS option, in which case, ALL may be used and COBOL and COBRECS may not be used.

COBOL COBOL indicates that all COBOL programs are to be pulled off of the partitioned data set. If non-COBOL programs are found, then they are discarded.

COBRECS COBRECS indicates that all COPY members which are 01 records or record groups beginning with a level number (02-48) are to be pulled off of the PDS. Those members that do not conform to this selection are bypassed for selection. COBRECS may be used when the LOR option is used.

MEMBER MEMBER indicates that the user wants to make further selection with member name control statements. These control statements immediately follow the control statement which contains the word MEMBER. An example is listed below:

```
//STEP3 EXEC MBRFETCH
//MBR.CTLCDMBR DD *
INDD=USERDD=MEMBER
MEMBER1
MEMBER2
MEMB03
MEMBER04
//MBR.USERDD DD DSN=USER.PDS,DISP=SHR
```

PREFIX PREFIX indicates that selection is to be done strictly on the basis of a 1 to 7 character prefix that follows the (=) sign after the word PREFIX. It is recommended that this option not be used here and that **COBOL=prefix** or **COBRECS=prefix** be used in its place. Prefix is described more on the next page.

Three ways to use PREFIX follow:

COBOL=prefix pulls off all COBOL programs from the PDS specified that begins with the same letters that are specified in the prefix. The prefix may not exceed seven characters in length.

COBRECS=prefix pulls off all COPY members that have as their first COBOL record, a level number from (01-48) that have a matching prefix. This is used when using PARM option LOR for LAYOUTS.

PREFIX=prefix pulls off all members that have a matching prefix.

An example follows:

```
//STEP4 EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCDMBR DD *
INDD=USERDD=COBOL=M4
//MBR.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

In the above example, all COBOL programs whose member name begins with the prefix M4 will be selected for passing on to the next step.

See the heading, “JCL Examples”, for an example of combining this PROC with the DCDCOBOL PROC.

Further notes on the member fetcher:

1. Compressing the PDS before using the data set will eliminate the possibility of pulling in older unwanted members from the PDS.
2. Multiple control statements may be used for any of the above formats.
3. To prevent obtaining duplicate members, do not mix formats in the same run.

Control Statements for CALL Hierarchy Report

Read this page completely before setting up to use the CALL Hierarchy Report.

This report produces one or several reports showing top down the CALL Hierarchy for those COBOL programs processed. Within each program the order of CALLs is shown in alphabetical order. These reports may be modified by ADD or DELETE controls statements. The ADD and DELETE commands must be entered one per line. Columns 73-80 are not used. Use spaces to separate the operands. Do not use the single apostrophes or double quotes. Up to 500 control statements may be used.

To add a CALL to the Hierarchy Report, use the following format:

ADD program-a CALLS program-b

To delete a CALL from the Hierarchy Report, use the following format:

DELETE program-b

DELETE pr* (where * indicates that only the characters before the asterisk will be used in making the comparison, and those not compared are considered to be equal for matching)

An example of the use of these control statements is shown below:

```
//DCD.CTLCDCAH DD *  
ADD SUB1P003 CALLS SUB3P091  
DELETE SUB6P*  
DELETE SUB9P*  
/*
```

The CALL Hierarchy Report produces counts showing the number of CALLs present. These counts are valid for all STATIC CALLs and for those DYNAMIC CALLs which use the 'CALL literal' format. When the 'CALL identifier' format is used, multiple CALLs are counted as one call. CALL identifier specifies a dataname, which contains a literal. DCD III builds one CALL for each literal contained in an elementary VALUE clause and/or with the MOVES of a literal to this name. In cases where the identifier is built over several MOVES or VALUE clauses of a group field, then an ADD control statement must be used for each CALL that is missing.

This report works with all the CALLs (and ENTRYs) found within the programs submitted to DCD III. The ENTRY verb (when present) plays a part in determining the CALL Hierarchy only when the program that CALLs a secondary program with the ENTRY verb was not submitted for processing to DCD III. For example, if PROG-A calls an Assembler program PROG-B, which in turn calls PROG-C where PROG-C contains an ENTRY 'PROG-B', then the CALL 'PROG-B' within PROG-A and the ENTRY 'PROG-B' within PROG-C provides all the information to complete the CALL from PROG-A to PROG-C even though Assembler program PROG-B could not be processed by DCD III. In cases where the ENTRY verb is not used (i.e. - the entry is done via PROCEDURE DIVISION USING ... within the sub-program) and the calling program is missing or cannot be processed by DCD III because it is non-COBOL, then ADD control statements must be used to complete the Hierarchy Report. (i.e. ADD PROG-B CALLS PROG-C)

Control Statements for Tracing & Analysis

Two types of control statements are available for Tracing & Analysis. The first type is **SELECTION** control statements and these are mandatory to state which fields are to be selected for the report. The second type is **BYPASSING** and is optional and may be used to break a trace invoked by tracing option YAP.

This report is applicable to any maintenance work done in COBOL for the purpose of tracing how any one data field is used in a COBOL program. There is a limit of one program at a time for this report. Extra programs will be ignored.

SELECT Control Statements

The DD name of the control file for SELECT statements is SELECT. There is a limit of 250 SELECTs, which may be used in any one run. See **TRACE** and '**Tracing & Analysis Supporting Options**' in the PARM section. Also see '**Tracing & Analysis Report**' under '**Overview of All Reports including Tracing & Analysis**'.

There are two formats for SELECTs. Use the first format if you know the name or names to be selected. Use the second format to select by different criteria. If both formats are used together, then all Format#2 SELECTs *must come before* any Format#1 SELECTs. The following rules apply to both SELECT formats. The word SELECT must start in columns 1 to 12. Columns 73-80 are not used. Leave one or more spaces between each entry.

Format#1

user-supplied-name
OR
SELECT IF NAME = user-supplied-name

In case of duplicate names where only 1 name is wanted then use the following format, and fill in the appropriate 01 record name where shown. Using a **GROUP name** will **NOT** substitute for the 01 record name. If further qualification is required, then invoke **Format#2** by adding an AND in the SELECT using START-POSITION (or other) keyword to complete the selection with or without the **OF 01-record-name** shown below.

SELECT IF NAME = user-supplied-name OF 01-record-name

Format#2

SELECT IF keyword operator selection-field
[OR selection-field] ...
[AND keyword operator selection-field
[OR selection-field] ...] ...

The following rules apply to this SELECT format. The brackets [] and ellipses ... in the above format are not to be coded however are there to show the allowance of **multiple ORs** and **multiple ANDs with multiple ORs within the ANDs**. The SELECT statement may be continued on to several lines. **Lower case** entries within the SELECT must be replaced with a **user entry** as follows:

keyword	replace with one of the keywords shown on the KEYWORD list. Within ANY one SELECT, the same keyword may be used once unless operators < and > are used. Use AND as a separator to start a new keyword.
operator	Use = unless the KEYWORD chosen allows the operators < and >.
selection-field	Do not use quotes or apostrophes. Enter the selection field as one entry with no embedded spaces. An asterisk (*) may be used as a wild string if the KEYWORD chosen allows them. The size of the selection-field must not exceed the maximum size allowed for that KEYWORD. Multiple selection-fields may be used by separating them with the word OR .

Consider using PARM option **YTEST** to test all **Format#2** SELECTs before producing a report.

If the user wants to select a field, which has certain characteristics, the following 18 KEYWORDS are provided for finding field which have the selection criteria provided:

#	keyword	operators allowed	maximum length	for selecting on	Comments
1.	PREFIX	=	20	Data-Name	For selecting by begin of Name
2.	SUFFIX	=	20	Data-Name	For selecting by end of Name
3.	BEGIN-NODE	=	20	Data-Name	For selecting by begin of Node*
4.	END-NODE	=	20	Data-Name	For selection by end of Node*
5.	NODE-1	=	20	Data-Name	Wild Strings Allowed, selects from any Node*
6.	NODE-2	=	20	Data-Name	Wild Strings Allowed, selects from any Node*
7.	PICTURE	=	20	PICTURE	Use ORs to select different combinations
8.	LEVEL-NUMBER	=	02	Level Numbers	Use ORs to select different level numbers
9.	WHOLE-DIGITS	=	02	Coded Numeric Size	Selects by # of digits on left of decimal place
10.	DECIMAL-DIGITS	=	02	Coded Numeric Size	Selects by # of digits on right of decimal place
11.	START-POSITION	=, >, <	07	Position in record	May use two START-POSITION keywords separated by AND, 1 using > and 1 using <
12.	END-POSITION	=, >, <	07	Position in record	May use two keywords; one >, and one <
13.	FIELD-SIZE	=, >, <	07	Actual Size of Field	May use two keywords, one >, and one <
14.	GROUP-ONLY	=	01	GROUP fields	Use Y to select only GROUP fields
15.	ELEMENTARY-ONLY	=	01	ELEMENTARY fields	Use Y to select only ELEMENTARY fields
16.	SECTION	=	01	SECTION	Use first letter of SECTION as follows: (F =FILE, W =WORKING-STORAGE, L =LINKAGE, R =REPORT, C =COMMUNICATION)
17.	USAGE	=	02	USAGE	Use two letters for USAGE as follows: (D =DISPLAY, D1 =DISPLAY-1, C =COMP, C1 =COMP-1, C2 =COMP-2, C3 =COMP-3, C4 =COMP-4, IX =INDEX)
18.	CLASS	=	01	CLASS	Use one letter for CLASS as follows: (N =Numeric X =Alphabetic or Alphanumeric, E =Numeric Edited)
19.	NAME	=	30	NAME	(OF 01-record-name) optional after data-name

Table of KEYWORDS for use with free-format SELECT

The most commonly used keywords are those which SELECT on some part of the data-name. Consider using more than one of the first six KEYWORDS in one SELECT separated by ANDs to develop a very powerful selection capability for finding data-names. Four of these six keywords start selection at the **beginning** of a node or data-name and the other two SUFFIX and END-NODE start selection at the **end** of a node or data-name). For NODE-1 and NODE-2, an * may be used after one or more letters in the selection field as a wild string, providing another letter or letters are used following the *. When used, the * indicates that any number from 0 to 18 characters (non-dashes) may be present before picking up selection with the next character. Consider using NODE-1 and NODE-2 together with Wild Strings in one or both PREFIXs. Consider completing selection with one or more non-data-name keywords being ANDed to those used for selecting by data-name.

When selecting by field size, there is a difference, which may be used to an advantage when doing selection. WHOLE-DIGITS and DECIMAL-DIGITS work with NUMERIC fields and work with the size coded in the PICTURE clause, while FIELD-SIZE works with the actual number of bytes that the field occupies within a record. (e.g - 05 TEST-FLD S9(9) COMP-3. takes up 5 bytes in FIELD-SIZE, but has 9 WHOLE-DIGITS in the field)

See Example 10 under 'JCL Examples' in this section, for an example of JCL to use this feature.

* **Node** is defined as a cluster of characters in name separated by dashes.

BYPASS Control Statements

After the first run of Selecting Names, it is possible to have TRACED names (names traced with the use of PARM option YAP) which may not actually be in a valid functioning program path. For example, a date in an 01 record HEADING-1 may accurately trace to a related 01 record PRINT-RECORD, but can inaccurately trace to other 01 records HEADING-2 and DETAIL-RECORD. As a simple means of stopping the trace into and through HEADING-2 and DETAIL-RECORD, bypass control statements should be provided using Format#1 (01-record-name) or in other cases using Format#2 below.

The DD name of the control file for BYPASS statements is BYPASS. Only columns 1-30 may be in BYPASS control statements. **RANGE control statements only use columns 1-17.**

There are three formats for BYPASS control statements. They follow:

Format#1

Columns 1-30

01-record-name

Specifies an 01 record **WHICH IS NOT** implicitly or explicitly **REDEFINED**. When used, all names belonging to this record AND it's redefined (implicit or explicit) 01 records will be BYPASSED.

Format#2

Columns 1-30

data-name

Specifies a COBOL elementary, group, 88 name, or other data division name, which is to be BYPASSED.

Format#3

Columns 1-17

RANGE=nnnnn-nnnnn

Specifies a range of sequence number. Each nnnnn **MUST BE** 5 digits. The first nnnnn specifies a BEGIN program sequence number. The second nnnnn specified an END program sequence number. These sequence numbers are determined AFTER COPY members are expanded. To be safe, use sequence numbers from the 'Alternate Compile Listing' for the program submitted.

Warning - Do not use **RANGE=** when multiple programs are used. If RANGE is used and the program is modified, then the RANGE will change and need to be modified.

This page intentionally left blank

JCL Examples

The examples within this heading use one of three DCD III PROCs. These three PROCs are:

1. **PANCOBOL**
2. **LIBCOBOL**
3. **DCDCOBOL**

Within each of the examples, capitalized letters should be left coded as shown, and non-capitalized letters require the user to enter in the names or entries required to complete the JCL.

Some of the examples using the DCDCOBOL PROC also use the **MBRFETCH** PROC. For WRITPDS and READPDS options, see the heading for “**WRITPDS and READPDS Options**”.

Example 1

Print the System Paragraph Cross Reference and System Data Name Cross Reference Reports from three programs residing on a Panvalet library.

```
//STEP1 EXEC PANCOBOL,SRCLIB='panvalet.library',
//      PARA=,DATA=
//PAN.SYSIN DD *
++WRITE WORK,prog1
++WRITE WORK,prog2
++WRITE WORK,prog3
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
```

Note: The COPYLIB DD is added assuming that there may be COPY members on some of the Panvalet stored programs.

Example 2

Print the CALL Analysis, COPY Analysis and Verb Analysis Reports from two programs residing on a Librarian library.

```
//STEP2 EXEC LIBCOBOL,SRCLIB='librarian.lib',
//      CALL=,COPY=,VERB=
//LIB.SYSIN DD *
-OPT EXEC
-SEL prog1,pswd
-EMOD
-SEL prog2,pswd
-EMOD
-END
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
```

Note: The COPYLIB DD is added assuming there may be COPY members on some of the Librarian stored programs.

Example 3

Print the System Cross Reference for Literals, 01 Record Report and Layout Report from four programs residing on a partitioned data set.

```
//STEP3 EXEC DCDCOBOL,OTHER='LIT',RECORDS=,LAYOUT=
//DCD.COBO LIN DD DSN=user.cobol.library (prog1),DISP=SHR
// DD DSN=user.cobol.library (prog2),DISP=SHR
// DD DSN=user.cobol.library (prog3),DISP=SHR
// DD DSN=user.cobol.library (prog4),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
```

Example 4

Print the 01 Record and Layout Reports from all COBOL programs residing on a partitioned data set whose member names begin with the prefix PAYR or PAYM.

```
//STEP4A EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
INDD=USERDD=COBOL=PAYR
INDD=USERDD=COBOL=PAYM
//MBR.USERDD DD DSN=user.cobol.library,DISP=SHR
/*
//STEP4B EXEC DCDCOBOL,RECORDS=,LAYOUT=
//DCD.COBO LIN DD DSN=&&PASSFILE,DISP=(OLD,PASS)
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
```

Note: The PDS Member Fetcher PROC is discussed under the headings, “**Choice of DCD III PROCs**” and “**Control Statements for Using the MBRFETCH PROC**”.

Example 5

Print the System Record Analysis Report on two sets of records.

```
//STEP5 EXEC DCDCOBOL,SRA=
//DCD.COBO LIN DD DSN=user.cobol.library (prog1),DISP=SHR
// DD DSN=user.cobol.library (prog2),DISP=SHR
// DD DSN=user.cobol.library (prog3),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.CTLCD SRA DD *
*title for report 1
-ALL- mr-record-1
-ALL- ms-record-1
prog3 ms-record-4
*title for the second report
prog2 184
/*
```

Notes:

- In the first report, 2 records are selected from ALL programs along with a record from prog3.
- In the second report, all records in prog2 which have an overall length of 184 will be used.

Example 6

Produce a Data Dictionary Interface File on all records 240 characters in length.

```
//STEP6A EXEC DCDCOBOL,OTHER=DICT
//DCD.DCDDICT DD DSN=user.file,DISP=(,KEEP),
//          VOL=SER=userpk,UNIT=SYSDA,SPACE=(CYL,(2,20)),
//          DCB=(RECFM=FB,LRECL=120,BLKSIZE=3120)
//DCD.COBO LIN DD DSN=user.cobol.library (prog1),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.CTLCD SRA DD *
*set number 1
prog1 240
/*
```

Example 7

Print Layout Reports for selected records from a given program.

```
//STEP7 EXEC DCDCOBOL,LAYOUT=
//DCD.COBO LIN DD DSN=user.cobol.library (prog1),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.CTLCD LAY DD *
ms-record-1
ms-record-2
/*
```

Example 8

Print Layout Reports for all records which have a record size from 220 to 240 characters in length.

```
//STEP8 EXEC DCDCOBOL,LAYOUT=
//DCD.COBO LIN DD DSN=user.cobol.library (prog1),DISP=SHR
//          DD DSN=user.cobol.library (prog2),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.CTLCD LAY DD *
SELECT>219
SELECT<241
/*
```

Example 9

Bring just 01 records into DCD III (not a COBOL program) and print Layout Reports.

```
//STEP9A EXEC MBRFETCH
//MBR.OUTSET DD DCB=(LRECL=80,BLKSIZE=3120)
//MBR.CTLCD MBR DD *
INDD=USERDD=COBRECS
//MBR.USERDD DD DSN=user.copy.library,DISP=SHR
/*
//STEP9B EXEC DCDCOBOL,LAYOUT=,OTHER=LOR
//DCD.COBO LIN DD DSN=&&PASSFILE,DISP=(OLD,PASS)
```

Notes:

- See “**Control Statements for Using the MBRFETCH PROC**”
- All COPY members that do not begin an 01 record or (02-48) level group are bypassed
- With option LOR option indicating records (not COBOL programs), no other reports are allowed

For Examples 10 and 11, see **A Tracing & Analysis Report@** under **A Overview of All Reports including Tracing & Analysis@** in this section.

Example 10

Produce a report using the Tracing & Analysis feature.

```
//STEP10 EXEC DCDCOBOL,TRACE=,OTHER=NOYIN
//DCD.COBO LIN DD DSN=user.cobol.library(prog1),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.SELECT DD *
      SELECT IF NODE-1 = P*R
              AND NODE-2 = Y*R OR YE
              AND FIELD-SIZE = 2
//DCD.BYPASS DD *
HEADING-1-RECORD
DETAIL-RECORD
```

Notes:

- X The SELECT looks for a node with PAYROLL, PR or other combinations and then looks for another node with YEAR, YR, YE or other combinations, and then looks for a field which is two bytes long.
- X The optional BYPASS DD is used to stop the TRACE for 01 records HEADING-1-RECORD and DETAIL-RECORD
- X The option to also produce PROCEDURE DIVISION Narrative is turned off

Example 11

Produce a report using the Tracing & Analysis feature.

```
//STEP11 EXEC DCDCOBOL,TRACE=
//DCD.COBO LIN DD DSN=user.cobol.library(prog1),DISP=SHR
//DCD.COPYLIB DD DSN=user.copy.library,DISP=SHR
//DCD.SELECT DD *
      PAYROLL-MONTH
```

Notes:

- X The SELECT simply looks for the name shown (PAYROLL-MONTH)

JCL to Execute DCD III Without Using PROCs

The JCL necessary to execute the DCDSYSTEM program for use with the features in this section is listed here and is informational. It is recommended however, that PROCs be used for the actual execution of DCD III. The rest of this section is designed for use of the PROCs shown in this section.

```
//DCD   EXEC   PGM=DCDSYSTEM, REGION=8192K,
//      PARM=(options)
//*
//STEPLIB   DD   DSN=user.loadlib,DISP=SHR
//*
//DCDWK01  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK02  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK03  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK04  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK05  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK06  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK07  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK08  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK09  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK10  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK11  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK12  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//*
//DCDWKS1  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS2  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS3  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS4  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS5  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS6  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS7  DD   UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//*
//DCDWKT1  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT2  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT3  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT4  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//*
//PRINT    DD   SYSOUT=*,DCB=BLKSIZE=133
//REPORTS   DD   SYSOUT=*,DCB=BLKSIZE=133
//*
//SYSOUT    DD   SYSOUT=*,DCB=BLKSIZE=121
//SORTMESS  DD   SYSOUT=*,DCB=BLKSIZE=133
//SORTLIB   DD   DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01  DD   UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02  DD   UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03  DD   UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//*
//CONTROL   DD   DSN=user.pds(DCDCNTRL),DISP=SHR
//*
```

See the next page for additional DD statements.

Extra DD statements may be required in addition to those shown on the previous page. These additional DD statements are listed below:

- BYPASS - optionally used to stop tracing levels by BYPASSing a record or field in the Tracing & Analysis report. See the heading '**Control Statements for Tracing & Analysis**'.
- COBOLIN - required unless PARM option READPDS is used without using WRITPDS at the same time. Points to a COBOL program, several COBOL programs or one or more COBOL records when producing only Layout Reports using the LOR option.
- COPYLIB - required if COPY members are present in the COBOL program
- CTLCDCAH- optionally used to modify the CALL Hierarchy Report.
- CTLCDDLT - used to provide data names for use in the Tracing Literals Report.
- CTLCDLAY- optionally used if the LAYOUTS PARM option is used and further selection is wanted through control statements. See the heading, '**Control Statements for Layouts**'.
- CTLCDMBR- see 'MBRFETCH Extra DD Statements ...' below.
- CTLCDSRA - required when either the SRA or DICT PARM option is used. See the heading, '**Control Statements for System Record Analysis**'.
- CTLCDVER- optionally used to select by verb in the VERB Analysis Report.
- DCDPDS - required if the WRITPDS option is used. See the heading, '**WRITPDS and READPDS Options**'.
- DCDREPDS - required if the READPDS option is used. See the heading, '**WRITPDS and READPDS Options**'.
- SELECT - required to select fields for the Tracing & Analysis report. See the heading '**Control Statements for Tracing & Analysis**'.

The REGION is set to 8192K. Consult the Installation Instructions for Guidelines on REGION size.

The DCB=BUFNO is set to 5 for MVS.

See Installation Instructions for information on the CONTROL DD.

JCL for the PDS member fetcher PROC is shown below:

```
//MBR      EXEC  PGM=MBRFETCH,REGION=512K
//SYSPRINT DD   SYSOUT=*
//WORKFILE DD   UNIT=SYSDA,SPACE=(TRK,(4,2))
//OUTSET    DD   DSN=&&PASSFILE,DISP=(,PASS),UNIT=SYSDA
//          SPACE=(CYL,(5,10)),DCB=(LRECL=80,BLKSIZE=3120)
```

MBRFETCH Extra DD statements are listed here:

- CTLCDMBR - required and used to input control to this PROC. See the heading '**Control Statements for Using the MBRFETCH PROC**'.
- userdd - one or more DDnames that are user specified are required if the above CTLCDMBR DD is used.

PROCs Provided for Use in This Section

Four PROCs are provided. They are listed here and are shown on the next pages. If Panvalet or Librarian are not used in your shop, then PROCs PANCOBOL or LIBCOBOL may be ignored.

1. **PANCOBOL** - Running DCD III with input from Panvalet
2. **LIBCOBOL** - Running DCD III with input from Librarian
3. **DCDCOBOL** - Running DCD III without Panvalet or Librarian
4. **MBRFETCH** - Used with PROC DCDCOBOL for selecting COBOL programs or COPY records by means of control statements from partitioned data sets.

```

//PANCOBOL PROC CALL=NO,          * CALL ANALYSIS      REPORT
//          COPY=NO,              * COPY ANALYSIS      REPORT
//          DATA=NO,             * DATA ANALYSIS     REPORT
//          LAYOUT=NO,           * LAYOUTS
//          PARA=NO,              * PARAGRAPH          REPORT
//          RECORDS=NO,          * 01 RECORDS        REPORT
//          SRA=NO,              * SYSTEM RECORD ANALYSIS
//          TRACE=NO,            * TRACING & ANALYSIS
//          RESOLVE=,            * RESOLVE COPY MEMBERS
/*      USE XXX= ABOVE TO TURN ON A REPORT OR OPTION
//          OTHER=' ',            * USED TO ENTER OTHER PARM OPTIONS
//          LINECNT=60,          * NUMBER OF LINES PER PAGE
//          SRCLIB='PANVALET.SOURCE',
//          BUF=5,               * USE BUF=5
//          PRINT='*',           * SEND PRINT OUTPUT TO MSGCLASS
//          REG=8192K,           * REGION SIZE
//          SORTREG=600000,      * SORT REGION
//          WORK=SYSDA          * UNIT=SYSDA, DISK OR OTHER
/*      PROC TO RUN THE SYSTEM REPORTS WITHIN DCD III
/*      MARBLE COMPUTER, INC. 1-800-252-1400 PROC=PANCOBOL REL 3.6
/*                                  LM080108
//PAN      EXEC      PGM=PAN#1
/*      INSERT STEPLIB HERE IF NECESSARY
//SYSPRINT      DD      SYSOUT=&PRINT
//PANDD1        DD      DSN=&SRCLIB,DISP=SHR
//PANDD2        DD      DSN=&&TEMP,DCB=BLKSIZE=6160,UNIT=&WORK,
//              DISP=(,PASS),SPACE=(CYL,(2,20))
/*
//DCD      EXEC      PGM=DCDSYSTM,REGION=&REG,
//              PARM=(&CALL.CAL,&COPY.COP,&DATA.DAT,
//              &LAYOUT.LAY,&PARA.PAR,&RECORDS.REC,
//              &SRA.SRA,&TRACE.TRA,&RESOLVE=RES,
//              &OTHER,'SOR=&SORTREG','LNC=&LINECNT')
/*      INSERT STEPLIB HERE IF NECESSARY
//CONTROL      DD      DSN=USER.PDS (DCDCNTRL),DISP=SHR
//COBOLIN      DD      DSN=&&TEMP,DISP=(OLD,DELETE)
/*      INSERT COPYLIB HERE IF NECESSARY
//DCDWK01      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK02      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK03      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK04      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK05      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK06      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK07      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK08      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK09      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK10      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK11      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK12      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKS1      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS2      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS3      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS4      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS5      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS6      DD      UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKT1      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT2      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT3      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT4      DD      UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//PRINT        DD      SYSOUT=*,DCB=BLKSIZE=133
//REPORTS      DD      SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT        DD      DUMMY,DCB=BLKSIZE=121
//SORTMESS      DD      DUMMY,DCB=BLKSIZE=133
//SORTLIB      DD      DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01      DD      UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02      DD      UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03      DD      UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)

```

Exhibit 1

PANCOBOL PROC

```

//LIBCOBOL PROC CALL=NO,          * CALL ANALYSIS      REPORT
//          COPY=NO,              * COPY ANALYSIS      REPORT
//          DATA=NO,             * DATA ANALYSIS    REPORT
//          LAYOUT=NO,            * LAYOUTS
//          PARA=NO,              * PARAGRAPH          REPORT
//          RECORDS=NO,           * 01 RECORDS        REPORT
//          SRA=NO,               * SYSTEM RECORD ANALYSIS
//          TRACE=NO,             * TRACING & ANALYSIS
//          RESOLVE=,             * RESOLVE COPY MEMBERS
/* USE XXX= ABOVE TO TURN ON A REPORT OR OPTION
//          OTHER=';',            * USED TO ENTER OTHER PARM OPTIONS
//          LINECNT=60,           * NUMBER OF LINES PER PAGE
//          SRCLIB='LIBRARAN.SOURCE',
//          LIBPGM='LIBRARAN',
//          BUF=5,                * USE BUF=5
//          PRINT='*',            * SEND PRINT OUTPUT TO MSGCLASS
//          REG=8192K,            * REGION SIZE
//          SORTREG=600000,       * SORT REGION
//          WORK=SYSDA            * UNIT=SYSDA, DISK OR OTHER
/* PROC TO RUN THE SYSTEM REPORTS WITHIN DCD III
/* MARBLE COMPUTER, INC. 1-800-252-1400 PROC=LIBCOBOL REL 3.6
/*                                     LM080108
//LIB EXEC PGM=&LIBPGM,PARM==NJTA,NRJS= CHECK LIBRARIAN PARMS
/* INSERT STEPLIB HERE IF NECESSARY
//SYSPRINT DD SYSOUT=&PRINT
//LIST DD SYSOUT=&PRINT
//MASTER DD DSN=&SRCLIB,DISP=SHR
//OSJOB DD DSN=&&TEMP,DCB=BLKSIZE=6160,UNIT=&WORK,
//          DISP=(,PASS),SPACE=(CYL,(2,20))
//DCD EXEC PGM=DCDSYSTEM,REGION=&REG,
//          PARM=(&CALL.CAL,&COPY.COP,&DATA.DAT,
//          &LAYOUT.LAY,&PARA.PAR,&RECORDS.REC,
//          &SRA.SRA,&TRACE.TRA,&RESOLVE=RES,
//          &OTHER,'SOR=&SORTREG','LNC=&LINECNT')
/* INSERT STEPLIB HERE IF NECESSARY
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
//COBOLIN DD DSN=&&TEMP,DISP=(OLD,DELETE)
/* INSERT COPYLIB HERE IF NECESSARY
//DCDWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK04 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK05 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK06 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK07 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK08 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK09 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK10 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK11 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK12 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKS1 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS2 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS3 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS4 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS5 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS6 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS7 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKT1 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT2 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT3 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT4 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//PRINT DD SYSOUT=*,DCB=BLKSIZE=133
//REPORTS DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT DD DUMMY=*,DCB=BLKSIZE=121
//SORTMESS DD DUMMY=*,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)

```

Exhibit 2

LIBCOBOL PROC

```

//DCDCOBOL PROC CALL=NO,      * CALL ANALYSIS REPORT
//      COPY=NO,              * CALL ANALYSIS REPORT
//      DATA=NO,             * DATA ANALYSIS REPORT
//      LAYOUT=NO,           * LAYOUTS
//      PARA=NO,              * PARAGRAPH REPORT
//      RECORDS=NO,          * 01 RECORDS REPORT
//      SRA=NO,               * SYSTEM RECORD ANALYSIS
//      TRACE=NO,            * TRACING & ANALYSIS
//*
//      RESOLVE=,             * RESOLVE COPY MEMBERS
//* USE XXX= ABOVE TO TURN ON A REPORT OR OPTION
//      OTHER=',',           * USED TO ENTER OTHER PARM OPTIONS
//      LINECNT=60,          * NUMBER OF LINES PER PAGE
//      BUF=5,               * USE BUF=5
//      PRINT='*',           * SEND PRINT OUTPUT TO MSGCLASS
//      REG=8192K,           * REGION SIZE
//      SORTREG=600000,      * SORT REGION
//      WORK=SYSDA           * UNIT=SYSDA, DISK OR OTHER
//* PROC TO RUN THE SYSTEM REPORTS WITHIN DCDIII
//* MARBLE COMPUTER, INC. 1-800-252-1400 PROC=DCDCOBOL REL 3.6
//*                                     LM080108
//DCD EXEC PGM=DCDSYSTM,REGION=&REG,
// PARM= ( &CALL.CAL, &COPY.COP, &DATA.DAT,
//      &LAYOUT.LAY,&PARA.PAR,&RECORDS.REC,
//      &SRA.SRA, &TRACE.TRA, &RESOLVE=RES,
//      &OTHER,
//      'TAC=&TRLIMIT', 'SOR=&SORTREG', 'LNC=&LINECNT')
//* INSERT STEPLIB HERE IF NECESSARY
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
//* INSERT COPYLIB HERE IF NECESSARY
//DCDWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK04 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK05 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK06 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK07 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK08 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK09 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK10 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK11 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWK12 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKS1 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS2 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS3 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS4 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS5 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS6 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKS7 DD UNIT=SYSDA,SPACE=(CYL,(2,20)),DCB=BUFNO=5
//DCDWKT1 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT2 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT3 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//DCDWKT4 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DCB=BUFNO=5
//*
//PRINT DD SYSOUT=*,DCB=BLKSIZE=133
//REPORTS DD SYSOUT=*,DCB=BLKSIZE=133
//*
//SYSOUT DD DUMMY=*,DCB=BLKSIZE=121
//SORTMESS DD DUMMY=*,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)

```

Exhibit 3

DCDCOBOL PROC

```
//MBRFETCH PROC PRINT='*',WORK=SYSDA
//MBR      EXEC PGM=MBRFETCH,REGION=512K
//SYSPRINT DD SYSOUT=&PRINT
//WORKFILE DD UNIT=&WORK,SPACE=(TRK, (4,2))
//OUTSET   DD DSN=&&PASSFILE,DISP=(,PASS),UNIT=&WORK,
//          SPACE=(CYL,(5,10))
```

Exhibit 4

MBRFETCH PROC

Notes on MBRFETCH PROC

There are two uses for the MBRFETCH PROC within this section.

The first is for inputting COBOL programs. See 'Choice of DCDIII PROCs' use. The user is prompted to add the following DCB to the OUTSET DD as shown below:

DCB=(LRECL=80,BLKSIZE=3120)

The second is for using DCDIII's internal Data Dictionary. See "WRITPDS and READPDS" options in this section. When used here, the record size if specified must be 3120 as shown on the JCL example in the write-up on "WRITPDS and READPDS" options.

Efficient Use of Work Space

When large systems are put through the DCD III system, questions may arise over the use of work file space. This heading makes an attempt to answer these questions.

DCD III work files DCDWK01 through DCDWK07 are used for one program at a time and do not hold cumulative files from several programs at any one time. For this reason, the SPACE allocation for these files may be kept small (both primary and secondary allocation). The only exception to this rule occurs when a user attempts to run Layout Reports on several programs at once, in which case, DCDWK01 will hold all the Layout Reports produced prior to printing them.

DCD III work files DCDWK08 through DCDWK12 are used mostly by the Tracing & Analysis Feature and may typically be kept small.

DCD III work files DCDWKS1 through DCDWKS7 hold cumulative information in a condensed format for all programs run through DCD III at any one time. The files, besides being in a condensed format where three or more occurrences of spaces or zeros are reduced to two characters, are further reduced by only holding the fields that are needed for the particular reports wanted. Further, most of the work files DCDWKS1 through DCDWKS7 are not used when just one or two reports are selected.

Because of the file structure used for files DCDWKS1 through DCDWKS7, the SPACE in the PROCs for these files have very small primary increments and very large secondary increments.

DCDWKT1 through DCDWKT4 are used for Direct Tracing of Literals invoked by PARM option DIT.

The following chart shows which options use which work files:

PARM field	WS1	WS2	WS3	WS4	WS5	WS6	WS7
CAHIER	----	----	----	----	----	YES	----
CALL	----	----	----	----	----	YES	----
COPY	----	----	YES	----	----	----	----
DATA	----	----	----	----	YES	----	YES
FGCONSTS	----	----	YES	----	----	----	----
LAYOUTS	----	----	----	----	----	----	YES
LITERALS	----	----	YES	----	----	----	----
PARAGRAPH	----	YES	----	----	----	----	----
RECORDS	YES	----	----	----	----	----	----
SPREGS	----	----	YES	----	----	----	----
SRANAL	----	----	----	YES	----	----	----
VERB	----	YES	----	----	----	YES	----

The DDNAME DCDPDS which is used when the WRITPDS option is used, will contain condensed information from all of the above seven condensed files depending on which options are specified during the WRITPDS run.

This page intentionally left blank

Overview of All Reports including Tracing & Analysis

The COBOL reports that are made available through this section are listed below:

1.	CALL Analysis Report	B-46
2.	CALL Hierarchy Report	B-47
3.	COPY Analysis Report	B-48
4.	System Data Name Cross-Reference	B-50
5.	System Cross-Reference for Figurative Constants	B-52
6.	Table of Contents for Layouts	B-53
7.	Layout Report	B-54
8.	System Cross-Reference for Literals	B-56
9.	System Paragraph Cross-Reference	B-57
10.	01 Record Report	B-58
11.	System Cross-Reference for Special Registers	B-61
12.	System Record Analysis Report	B-62
13.	Tracing Literal Paths Reports	B-64
14.	Verb Analysis Report	B-67
15.	Tracing & Analysis Report	B-68

The PARM options that are required to run these reports are listed under the heading, “**PARM OPTIONS**”. For most reports, the PARM options require just one option per report (e.g. use option **CALL** for the CALL Analysis Report, **COPY** for COPY Analysis Report, **DATA** for System Data Name Cross-Reference). The following options are applicable to all reports:

C68	LNCNT=	RESOLVE=	VR3
EIB	QUOTE	SORTREG=	VS2
ERRORS	READPDS	STOP	WRITPDS

The Layout Report is affected by many options and optional control statements.

The System Record Analysis Report requires the use of control statements. Also, the RA1, RAB and RAD options are applicable to the System Record Analysis Report.

The System Data Name Cross-Reference and the System Paragraph Cross-Reference have the UNREF option available for including or omitting unreferenced names. The Data Name Cross-Reference report may be limited to control statement use by using the DAS parm option.

The CALL Hierarchy Report allows control statements for Adding and Deleting CALLs.

The Direct Literal Tracing Report requires control statements for specifying data names used for tracing the possible values of literals and the paths used to get a literal to a particular field.

The Verb Analysis Report allows the use of control statements to limit VERBs selected.

The Tracing & Analysis Report (**TRACE Option**) is available for finding data fields by their name (or by a very powerful selection capability), then bringing in all REDEFINES, GROUP fields, and overlapping fields for the name selected, doing Tracing through infinite levels to find all related fields and finally showing all Procedure Division references for all names selected or brought in. Two separate types of control statements are available for **SELECTing** and **BYPASSing**.

DCD III – Tracing & Analysis and Other Reports

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-NUMBER	0146	CALL 'EMPVERIFY' USING	WS-SYSTEM-DATE PT-EMPLOYEE-NBR WS-VERIFY-CODE	
	B010-COMPUTE-FEDERAL-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)		

A B C D E

CALL Analysis Report - Program-Id / CALL-Name Sequence

CALL ANALYSIS	CALL-NAME / PROGRAM-ID	SEQUENCES	01/01/2001	PAGE	1
PROGRAM-ID	PARAGRAPH-NAME	SEQ-#	CALL / ENTRY STATEMENT	PARAMETERS	
PAYROLL	A010-VERIFY-EMPLOYEE-NUMBER	0146	CALL 'EMPVERIFY' USING	WS-SYSTEM-DATE PT-EMPLOYEE-NBR WS-VERIFY-CODE	
PAYROLL	B010-COMPUTE-FEDERAL-TAX	0226	CALL 'FEDTAX' USING	WS-SYSTEM-DATE PN-RECORD	
PAYMONTH	A070-COMPUTE-DEDUCTIONS	0169	CALL 'PAYDEDUC' USING	PN-RECORD PN-WEEKLY-DEDUCTION-1 PN-WEEKLY-DEDUCTION-2	
PAYROLL	A070-COMPUTE-DEDUCTIONS	0184	CALL 'PAYDEDUC' USING	PN-RECORD PN-WEEKLY-DEDUCTION-1 PN-WEEKLY-DEDUCTION-2	
PAYMONTH		0126	PROCEDURE DIVISION ENTRY (PAYMONTH)		
PAYROLL		0124	PROCEDURE DIVISION ENTRY (PAYROLL)		

A B C D E

CALL Analysis Report - CALL-Name / Program-Id Sequence

Exhibit 5 – (Both Sequences)

- A - COBOL Program-Id
- B - Paragraph name where the CALL or ENTRY resides
- C - Sequence number of the CALL, ENTRY or Procedure Division statement
- D - CALL or ENTRY statement
- E - Parameters used

01/01/2001

CALL HIERARCHY REPORT

PAGE 1

COUNT	HIERARCHY OF CALLS
	DCDMAIN
	+ DCDCOBOL
	+ + DCD12CMP
	+ + DCD14IND
12	+ + ST
	+ DCDCOPY
	+ + DCDMCOPY
	+ + + READCOPY
	+ + + READREPL
	+ + + + READCOPY
	+ + DCDMREPL
12	+ + ST
	+ DCDHSKPG
	+ + BANNER
	+ + DCD07PRM
19	+ + + ER
	+ + DCD08OPT
	+ + DCDTR
24	+ + + ER
	+ + ER
	+ + GETNM
	+ + PW
7	+ + + ER
	+ DCDPRHP1
	+ DCDPXREF
	+ + DCDDDCCR
	+ + DCDOTHER
	+ + DCDPDCCR
	+ + DCDSETRF
18	+ + ST
	+ DCDSCAN1
	+ + DCDCICS
	+ + SCANTOK
6	+ + + PASSTOK
	+ + + + LASTIME
125	+ + + + NEXTTOK
A	B C

Exhibit 6

CALL Hierarchy Report

- A - Count of times program is called from above program (Blank for 1 time)
 B - Plus (+) closest to program-id is followed uphill to find calling module
 C - COBOL Program-id

COPY ANALYSIS	MEMBER	SEQUENCE	01/01/2001	PAGE	1
MEMBER	PROGRAM	SEQ-#	LEVEL	IDENTIFIER	SECTION
BANKCOPY	DCDLOGIC	0061	01	BC-BANNER-FIELDS	WORKING-STORAGE SECTION
CORRECT	DCDLOGIC	0110	01	COB-REC	WORKING-STORAGE SECTION
CPYREC	DCDLOGIC	0079	01	COB-REC	WORKING-STORAGE SECTION
H02IF	DCDLOGIC	0134		030-COPY-HERE	PROCEDURE DIVISION
INFOTYP7	DCDLOGIC	0012	FD	FILE-1	FILE SECTION
TESTCOPY	PAYMONTH	0024	01	TEST-COPY-REC	WORKING-STORAGE SECTION
	PAYMONTH	0036	01	TEST-COPY-REC-2	WORKING-STORAGE SECTION
TGO2	DCDLOGIC	0008		SOURCE-COMPUTER	ENVIRONMENT DIVISION
A	B	C	D	E	F

Exhibit 7

COPY Analysis Report - Member Sequence

- A - COPY member name
- B - COBOL Program-id
- C - Sequence number where the COPY is used in the program
- D - Level number associated with the COPY (if applicable)
- E - DATA name associated with the COPY (if applicable)
- F - DIVISION or SECTION where the COPY member resides

COPY ANALYSIS		PROGRAM SEQUENCE		01/01/2001	PAGE 1
PROGRAM	MEMBER	SEQ-#	LEVEL	IDENTIFIER	SECTION
DCDLOGIC	BANKCOPY	0061	01	BC-BANNER-FIELDS	WORKING-STORAGE SECTION
	COBREC1	0110	01	COB1-REC	WORKING-STORAGE SECTION
	CPYREC	0079	01	COB-REC	WORKING-STORAGE SECTION
	H021F	0134		030-COPY-HERE	PROCEDURE DIVISION
	INFOTYP7	0012	FD	FILE-1	FILE SECTION
	TGO2	0008		SOURCE-COMPUTER	ENVIRONMENT DIVISION
PAYMONTH	TESTCOPY	0024	01	TEST-COPY-REC	WORKING-STORAGE SECTION
	TESTCOPY	0036	01	TEST-COPY-REC-2	WORKING-STORAGE SECTION
A	B	C	D	E	F

Exhibit 8

COPY Analysis Report - Program Sequence

- A - COBOL
- B - COPY member name
- C - Sequence number where the COPY is used in the program
- D - Level number associated with the COPY when applicable
- E - Data name associated with the COPY when applicable
- F - Division or section where the COPY member resides

DCD III – Tracing & Analysis and Other Reports

SYSTEM DATA-NAME CROSS REFERENCE								
PROGRAM	SECTION	SEQ-#	LV	DATA-NAME	NARRATIVE	01/01/2001	PAGE	05
(CONTINUED)	WS-ALL DEDUCTIONS				PN-WEEKLY-FED-TAX @97 (189), PN-WEEKLY-FICA @89 (189), PN-WEEKLY-STATE-TAX @89 (189) COMPUTE PN-WEEKLY-PAY-AMOUNT @102 = # (203) IF # > PN-WEEKLY-GROSS @96 (195,199) SUBTRACT PN-WEEKLY-DEDUCTION-1 @100 FROM # (200), PN-WEEKLY-DEDUCTION-2 @101 FROM # (196)			
PAYROLL	WORKING-STORAGE	0118	88	WS-EMPLOYEE-NBR-INVALID	IF # (150)	VALUE (MULTIPLE VALUES)		
PAYROLL	WORKING-STORAGE	0116	88	WS-EMPLOYEE-NBR-VALID		VALUE (MULTIPLE VALUES)		
PAYMONTH	WORKING-STORAGE	0116	88	WS-END-OF-FILE	PERFORM-UNTIL # (138)	VALUE 'Y'		
PAYROLL	WORKING-STORAGE	0114	88	WS-END-OF-FILE	PERFORM-UNTIL # (137)	VALUE 'Y'		
PAYMONTH	WORKING-STORAGE	0115	05	WS-END-OF-FILE-SW	PIC X MOVE 'Y' TO # (135,202)	VALUE SPACE		
PAYROLL	WORKING-STORAGE	0113	05	WS-END-OF-FILE-SW	PIC X MOVE 'Y' TO # (134,220)	VALUE SPACE		
PAYMONTH	WORKING-STORAGE	0110	05	WS-FICA-AMOUNT	PIC S9(4) COMP-3 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)	VALUE +0		
PAYROLL	WORKING-STORAGE	0108	05	WS-FICA-AMOUNT	PIC S9(4) COMP-3 COMPUTE # = .0715 (173), 3131.70 (179), PN-WEEKLY-GROSS @96 (173), WS-FICA-YTD-TOTAL @109 (179) COMPUTE WS-FICA-YTD-TOTAL @109 = # (175) MOVE # TO PN-WEEKLY-FICA @89 (181) MOVE ZEROS TO # (171)	VALUE +0		
PAYROLL	WORKING-STORAGE	0109	05	WS-FICA-YTD-TOTAL	PIC S9(4) COMP-3 COMPUTE # = PN-YTD-FICA @82 (175), WS-FICA-AMOUNT @108 (175) COMPUTE WS-FICA-AMOUNT @108 = # (179) IF # > 3131.70 (178)	VALUE +0		
PAYMONTH	WORKING-STORAGE	0110	05	WS-FICA-AMOUNT	PIC S9(4) COMP-3 COMPUTE # = .0715 (158), 3131.70 (164), PN-WEEKLY-GROSS @89 (158), WS-FICA-YTD-TOTAL @111 (164) COMPUTE WS-FICA-YTD-TOTAL @111 = # (160) MOVE # TO PN-WEEKLY--FICA @101 (166) MOVE ZEROS TO # (156)	VALUE +0		
PAYMONTH	WORKING-STORAGE	0111	05	WS-FICA-YTD-TOTAL	PIC S9(4) COMP-3 COMPUTE # = PN-YTD-FICA @84 (160), WS-FICA-AMOUNT @110 (160) COMPUTE WS-FICA-AMOUNT @110 = # (164) IF # > 3131.70 (163)	VALUE +0		

A B C D E F G H I

Exhibit 9

System Data Name Cross Reference

- A - COBOL program-id
- B - Section
- C - Sequence number of the data name
- D - Level number of the data name
- E - Data name
- F - All related Procedure Division narrative
- G - PICTURE clause
- H - USAGE clause
- I - VALUE clause

01/01/2001 11:30

SYSTEM CROSS REFERENCE FOR FIGURATIVE CONSTANTS

PAGE 01

FIGURATIVE CONSTANTS	PROGRAM	REFERENCES
SPACE (S)	PAYMONTH	115
	PAYROLL	113
ZERO (S)	PAYMONTH	117 123 156 182 186
	PAYROLL	115 121 130 171 197

A**B****C**

Exhibit 10
System Cross- Reference for Figurative Constants

- A - Figurative constant
- B - COBOL program-id
- C - Sequence numbers where the figurative constant is used

01/01/2001		TABLE OF CONTENTS FOR LAYOUT REPORT	PAGE 1
PAYMONTH	FILE	PAYROLL-TIME-FILE.....	6
PAYROLL	FILE	PAYROLL-TIME-FILE.....	1
PAYMONTH	FILE	PAYROLL-UPDATED-FILE.....	8
PAYROLL	FILE	PAYROLL-UPDATED-FILE.....	3
PAYMONTH	FILE	PN-RECORD.....	8
PAYROLL	FILE	PN-RECORD.....	3
PAYMONTH	FILE	PT-RECORD.....	6
PAYROLL	FILE	PT-RECORD.....	1
PAYMONTH	WORK-ST	WS-AMOUNTS.....	10
PAYROLL	WORK-ST	WS-AMOUNTS.....	5

A**B****C****D****Exhibit 11****Table of Contents for Layouts**

- A - COBOL program-id
- B - Section where record name is located
- C - Record name
- D - Page number where record layout starts

DCD III – Tracing & Analysis and Other Reports

FILE SECTION		LAYOUT REPORT		PAGE		1
FD (FILE NAME)		PAYROLL-UPDATED-FILE				
LABEL RECORDS ARE		STANDARD				
RECORD CONTAINS		170 CHARACTERS				
BLOCK CONTAINS		0 RECORDS				
DATA RECORD IS		PN-RECORD				
LEVEL	DATA NAME	LENGTH	TYPE	FROM	TO	
72 01	PN-RECORD	170	GROUP	1	170	CALL 'PAYDEDUC' USING # (169) MOVE PT-RECORD @28 TO # (148) WRITE # (197)
73 05	PN-EMPLOYEE-NBR	5	N	1	5	
74 05	PN-EMPLOYEE-LAST-NAME	20	X	6	25	
75 05	PN-EMPLOYEE-FIRST-NAME	15	X	26	40	
76 05	PN-ENDING-DATE	6	GROUP	41	46	
77 05	PN-ENDING-YEAR	2	N	41	42	
78 05	PN-ENDING-MONTH	2	N	43	44	
79 05	PN-ENDING-DAY	2	N	45	46	
80 05	PN-CUMULATIVE-YTD-AMOUNTS	44	GROUP	47	90	
81 10	PN-YTD-GROSS PIC S9(6)V99	8	SNE	47	54	ADD PN-WEEKLY-GROSS @98 TO # (192)
82 10	PN-YTD-FED-TAX PIC S9(6)V99	8	SNE	55	62	
83 10	PN-YTD-STATE-TAX PIC S9(6)V99	8	SNE	63	70	
84 10	PN-YTD-FICA PIC S9(4)V99	6	SNE	71	76	ADD PN-WEEKLY-FICA @101 TO # (193) COMPUTE WS-FICA-YTD-TOTAL @111 = # (160) IF # > 4380.00 (155)
85 10	PN-YTD-DEDUCTION-1 PIC S9(5)V99	7	SNE	77	83	ADD PN-WEEKLY-DEDUCTION-1 @102 TO # (194)
86 10	PN-YTD-DEDUCTION-2 PIC S9(5)V99	7	SNE	84	90	
87 10	PN-CURRENT-EMPL-DATA	31	GROUP	91	121	
PAYMONTH				01/01/2001		

A B C D E F G H

Exhibit 12 Layout Report

- A - Sequence number of the data name
- B - Level number of the data name
- C - Data name
- D - Length in bytes of this field for computational or packed comp-3 fields, two lengths will be shown in this field. The first one will be in parentheses, and represents what is coded in the PICTURE clause.
- E - TYPE code indicating how the field is used (see chart below)
- F - From position of this field
- G - To position of this field
- H - Corresponding COBOL narrative (requires use of LNR option to produce this narrative)

Layout **TYPE** Chart

<u>TYPE</u>	<u>Meaning</u>
C	COMP or COMPUTATIONAL
C1	COMP-1
C2	COMP-2
C3	COMP-3
C4	COMP-4
N	NUMERIC (DISPLAY USAGE)
NE	NUMERIC EDITED
PR	USAGE IS POINTER
UI	USAGE IS INDEX
X	ALPHANUMERIC or ALPHABETIC
GROUP	GROUP FIELD

Note: The first seven fields (through NE above) will have an S preceding them
if the PICTURE clause is signed.

01/01/2001 11:30

SYSTEM CROSS REFERENCE FOR LITERALS

PAGE 01

LITERALS	PROGRAM	REFERENCES					
+0	PAYMONTH	110	111	112			
	PAYROLL	108	109	110			
'Y'	PAYMONTH	047	091	116	135	202	
	PAYROLL	047	089	114	134	220	
.0715	PAYMONTH	158					
	PAYROLL	173					
00	PAYMONTH	118					
	PAYROLL	116					
04	PAYMONTH	120					
	PAYROLL	118					
2.75	PAYROLL	201					
3131.70	PAYMONTH	163	164				
	PAYROLL	178	179				
43800.00	PAYMONTH	155					
	PAYROLL	170					

A **B** **C**

Exhibit 13**System Cross Reference for Literals**

- A - Literal
- B - COBOL program-id
- C - Sequence numbers where literal is used

SYSTEM PARAGRAPH CROSS REFERENCE							01/01/2001	PAGE	01
STMT	PROGRAM	SEQ-#	PARAGRAPH	SECTION	SEQ-#	TYPE TRANSFER	PARAGRAPH OF TRANSFER		
	PAYMONTH	145	A-PROCESS-PAYROLL-RECORDS	SECTION	137	PERFORM	040-PROCESS-ALL-RECORDS		
	PAYROLL	144	A-PROCESS-PAYROLL-RECORDS	SECTION	136 138	PERFORM PERFORM	040-PROCESS-ALL-RECORDS 040-PROCESS-ALL-RECORDS		
	PAYROLL	168	A060-COMPUTE-FICA	A-PROCESS-PAYROLL-RECORDS	164	GO TO	A050-COMPUTE-STATE-TAX		
NUMBER	PAYROLL	217	A800-READ-NEXT-RECORD	A-PROCESS-PAYROLL-RECORDS	151	GO TO	A010-VERIFY-EMPLOYEE-		
	PAYROLL	224	B-COMPUTE-FEDERAL-TAX	SECTION	159	PERFORM	A040-COMPUTE-FEDERAL-		
TAX	PAYROLL	231	C-COMPUTE-STATE-TAX00	SECTION	166	PERFORM	A050-COMPUTE-STATE-TAX		
	A	B	C	D	E	F	G		

Exhibit 14

System Paragraph Cross Reference

- A - COBOL program-id
- B - Sequence number of the paragraph
- C - Paragraph or section name
- D - Section name that paragraph name belongs to
- E - Sequence number of COBOL verb referencing this paragraph or section
- F - COBOL verb referencing this paragraph or section
- G - Paragraph name that contains the COBOL verb

01 RECORD REPORT LENGTH SEQUENCE 01/01/2001 PAGE 1

LENGTH	PROGRAM	SEQ-#	SECTION	RECORD-NAME
170	PAYMONTH	00028	FILE	PT-RECORD
170	PAYMONTH	00072	FILE	PN-RECORD
170	PAYROLL	00028	FILE	PT-RECORD
170	PAYROLL	00070	FILE	PN-RECORD
09	PAYMONTH	00109	WORKING-STORAGE	WS-AMOUNTS
09	PAYROLL	00107	WORKING-STORAGE	WS-AMOUNTS
06	PAYMONTH	00123	WORKING-STORAGE	WS-SYSTEM-DATE
06	PAYROLL	00121	WORKING-STORAGE	WS-SYSTEM-DATE
03	PAYMONTH	00114	WORKING-STORAGE	WS-SWITCHES
03	PAYROLL	00112	WORKING-STORAGE	WS-SWITCHES

A B C D E

Exhibit 15

01 Record Report - Length Sequence

- A - Length in bytes of the 01 record
- B - COBOL program-id for this record
- C - Sequence number where the record resides
- D - Data Division section that contains the record
- E - 01 record name

01 RECORD REPORT	ENTRY	SEQUENCE	01/01/2001	PAGE	1
PROGRAM	SEQ-#	SECTION	RECORD-NAME	LENGTH	
PAYMONTH	0028	FILE	PT-RECORD	170	
PAYMONTH	00072	FILE	PN-RECORD	170	
PAYMONTH	00109	WORKING-STORAGE	WS-AMOUNTS	09	
PAYMONTH	00114	WORKING-STORAGE	WS-SWITCHES	03	
PAYMONTH	00123	WORKING-STORAGE	WS-SYSTEM-DATE	06	
PAYROLL	00028	FILE	PT-RECORD	170	
PAYROLL	00070	FILE	PN-RECORD	170	
PAYROLL	00107	WORKING-STORAGE	WS-AMOUNTS	09	
PAYROLL	00112	WORKING-STORAGE	WS-SWITCHES	03	
PAYROLL	00121	WORKING-STORAGE	WS-SYSTEM-DATE	06	

A B C D E

Exhibit 16

01 Record Report - Entry Sequence

- A - Length in bytes of the 01 record
- B - COBOL program-id for this record
- C - Sequence number where the record resides
- D - Data Division section that contains the record
- E - 01 record name

This page intentionally left blank

01/01/2001 11:30

SYSTEM CROSS REFERENCE FOR THE SPECIAL REGISTERS

PAGE 01

SPECIAL REGISTERS PROGRAM REFERENCES

DATE	PAYMONTH	131	
	PAYROLL	129	
RETURN-CODE	PAYROLL	130	
SORT-RETURN	PAYROLL	156	173

A

B

C

Exhibit 17

System Cross Reference for Special Registers

- A - Special register
- B - COBOL program-id
- C - Sequence number where special register is used

01/01/2001	11:30	SYSTEM RECORD ANALYSIS - 01 RECORD SUMMARY						PAGE	1
FROM-TO	PROGRAM	01 RECORD NAME	SEQ-#	LV	DATA NAME	PICTURE	DIRECT-FLAGS SET / USE / TEST	INDIRECT FLAGS	
1-70	PAYMONTH	PN-RECORD	0072	01	PN-RECORD		S U SI	UI TI	
	PAYMONTH	PT-RECORD	0028	01	PT-RECORD		U		
	PAYROLL	PN-RECORD	0070	01	PN-RECORD		S U SI	UI TI	
A	B	C	D	E	F	G	H	I	

Exhibit 18

System Record Analysis Report - 01 Record Summary

- A - Beginning and ending record positions for the data name(s) listed
- B - COBOL program-id where record resides
- C - 01 record name
- D - Sequence number where the field resides
- E - Level of the data name
- F - Data name
- G - PICTURE of the data name
- H - Direct flags indicating Procedure Division involvement with this field
- I - Indirect flags indicating Procedure Division involvement with other fields that are a part of this field

Explanation of **S** (Set), **U** (Used) and **T** (Tested) flags:

- S - Data was moved out of this field or the field was in some way modified
- U - Data was moved out of this field or the field was used in some way to modify another field
- T - The field was tested or compared to another field

DCD III – Tracing & Analysis and Other Reports

01/01/2001 11:30

SYSTEM RECORD ANALYSIS - DATA NAME SUMMARY

PAGE 3

FROM-TO	PROGRAM	01 RECORD NAME	SEQ-#	LV	DATA NAME	PICTURE	DIRECT FLAGS SET/USE/TEST	INDIRECT FLAGS
84-90	PAYMONTH	PN-RECORD	0086	10	PN-YTD-DEDUCTION-2	PIC S9(5) V99	S	SI UI
	PAYMONTH	PT-RECORD	0042	10	PT-YTD-DEDUCTION-2	PIC S9(5) V99		SI UI
	PAYROLL	PN-RECORD	0084	10	PN-YTD-DEDUCTION-2	PIC S9(5) V99		SI UI
91-121	PAYMONTH	PN-RECORD	0087	05	PN-CURRENT-EMPL-DATA			SI UI
	PAYMONTH	PT-RECORD	0043	05	PT-CURRENT-EMPL-DATA			UI
	PAYROLL	PN-RECORD	0085	05	PN-CURRENT-EMPL-DATA			SI UI TI
91-96	PAYMONTH	PN-RECORD	0088	10	PN-CURRENT-HOUR-RATE	PIC S9(4) V99	S	SI UI
	PAYMONTH	PT-RECORD	0044	10	PT-CURRENT-HOUR-RATE	PIC S9(4) V99		UI
	PAYROLL	PN-RECORD	0086	10	PN-CURRENT-HOUR-RATE	PIC S9(4) V99	S	SI UI
97-106	PAYMONTH	PN-RECORD	0089	10	PN-EMP-TAX-CODES	PIC X (10)		SI UI
	PAYMONTH	PT-RECORD	0045	10	PT-EMP-TAX-CODES	PIC X (10)		UI
	PAYROLL	PN-RECORD	0087	10	PN-EMP-TAX-CODES	PIC X (10)		SI UI
107-107	PAYMONTH	PN-RECORD	0090	10	PN-NO-STATE-TAX-FLAG	PIC X	T	SI UI
	PAYMONTH	PT-RECORD	0046	10	PT-NO-STATE-TAX-FLAG	PIC X		UI
	PAYROLL	PN-RECORD	0088	10	PN-NO-STATE-TAX-FLAG	PIC X		SI UI
108-109	PAYMONTH	PN-RECORD	0092	10	PN-CURR-DEDUCTION-CODE1	PIC 99		SI UI
	PAYMONTH	PT-RECORD	0048	10	PT-CURR-DEDUCTION-CODE1	PIC 99		UI
	PAYROLL	PN-RECORD	0090	10	PN-CURR-DEDUCTION-CODE1	PIC 99		SI UI
110-114	PAYMONTH	PN-RECORD	0093	10	PN-CURR-DEDUCTION-AMT1	PIC S9(3)V99		SI UI
	PAYMONTH	PT-RECORD	0049	10	PT-CURR-DEDUCTION-AMT1	PIC S9(3)V99		UI
	PAYROLL	PN-RECORD	0091	10	PN-CURR-DEDUCTION-AMT1	PIC S9(3)V99		SI UI
115-116	PAYMONTH	PN-RECORD	0094	10	PN-CURR-DEDUCTION-CODE2	PIC 99		SI UI
	PAYMONTH	PT-RECORD	0050	10	PT-CURR-DEDUCTION-CODE2	PIC 99		UI
	PAYROLL	PN-RECORD	0092	10	PN-CURR-DEDUCTION-CODE2	PIC 99		SI UI
117-121	PAYMONTH	PN-RECORD	0095	10	PN-CURR-DEDUCTION-AMT2	PIC S9(3)V99		SI UI
	PAYMONTH	PT-RECORD	0051	10	PT-CURR-DEDUCTION-AMT2	PIC S9(3)V99		UI
	PAYROLL	PN-RECORD	0093	10	PN-CURR-DEDUCTION-AMT2	PIC S9(3)V99		SI UI
122-125	PAYMONTH	PN-RECORD	0096	05	PN-WEEKLY-HOURS	PIC S9(3)V99	S	SI UI
	PAYMONTH	PT-RECORD	0052	05	PT-WEEKLY-HOURS	PIC S9(3)V99		UI
	PAYROLL	PN-RECORD	0094	05	PN-WEEKLY-HOURS	PIC S9(3)V99	S	SI UI
126-165	PAYMONTH	PN-RECORD	0097	05	PN-COMPUTED-HOURS			SI UI TI
	PAYMONTH	PT-RECORD	0053	05	PT-COMPUTED-HOURS			UI
	PAYROLL	PN-RECORD	0095	05	PN-COMPUTED-HOURS			SI UI TI
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-WEEKLY-FED-TAX	PIC S9(4)V99	S	SI UI
	PAYMONTH	PT-RECORD	0055	10	PT-WEEKLY-FED-TAX	PIC S9(4)V99		UI
	PAYROLL	PN-RECORD	0097	10	PN-WEEKLY-FED-TAX	PIC S9(4)V99	S U	SI UI

A B C D E F G H I

Exhibit 19 System Record Analysis Report - Data Name Summary

01/01/2001 POSSIBLE LITERALS FOR EACH FIELD TRACED PAGE 1

SEQ-#	FIELD BEING TRACED	LITERAL
816	HIST-2ND-ACCOUNT	VALUE UNKNOWN 'A142165513' 'BB----1235' 'CUST-E3E' 'HIST-LD-2G' 'TKY' 'TY1236RTY6' ALL '4' CURRENT-DATE SPACES ZEROS

A

B

C

Exhibit 20

Possible Literals for Each Field Report

- A - Sequence Number for the field being traced
- B - Field Name
- C - Possible Ending Literals available for this field

01/01/2001	POSSIBLE LITERALS SHOWING OWNER FIELDS		PAGE	2
TRACED FIELD & RELATED FIELDS	LITERAL	HOW FORMED	HOW FORMED FLAGS	
816 HIST-2ND-ACCOUNT				
A	B			
1134 C3-IDENT-ACCT	VALUE UNKNOWN	UNKNOWN VALUE		UNK
C3-IDENT-ACCT	ALL '4'	MOVE LITERAL	ML	
1116 C3-SUB-ID-1A	VALUE UNKNOWN	UNKNOWN VALUE		UNK
C3-SUB-ID-1A	SPACES	MOVE FIG CONST		
C3-SUB-ID-1A	ZEROS	MOVE FIG CONST		
411 CUST-IDNT-ALPHA-1	'A142165513'	INITIAL VALUE	INI	
512 CUST-MAIN-ACCOUNT-3RD-ID	'CUST-E3E'	INITIAL VALUE	INI	
CUST-MAIN-ACCOUNT-3RD-ID	SPACES	MOVE FIG CONST		
431 CUST-STORAGE-IDENT	'BB----1235'	INITIAL VALUE	INI	
816 HIST-2ND-ACCOUNT	'HIST-LD-2G'	INITIAL VALUE	INI	
1125 S2-CUST-ACCOUNT-TA	'TY1236RTY6'	INITIAL VALUE	INI	
S2-CUST-ACCOUNT-TA	'TKY'	MOVE LITERAL		ML
S2-CUST-ACCOUNT-TA	CURRENT-DATE	MOVE SPEC REG		
C	D	E	F	G

Exhibit 21

Possible Literals Showing Owner Fields Report

- A** - Traced Field COBOL Sequence Number
- B** - Traced Field
- C** - Related Field COBOL Sequence Number
- D** - Related Field
- E** - Possible Literal tied to Related Field
- F** - How the Literal might be Formed
- G** - FLAGS for easy reference

01/01/2001 POSSIBLE LITERALS WITH POSSIBLE PATHS PAGE 3

LITERAL	HOW FORMED	SEQ-#	FIELD-NAMES BY TRACING LEVELS
'HIST-LD-2G'	INITIAL VALUE	816	HIST-2ND-ACCOUNT
'CUST-E3E'	INITIAL VALUE	512	- CUST-MAIN-ACCOUNT-3RD-ID
VALUE UNKNOWN	UNKNOWN VALUE	1116	- C3-SUB-ID-1A
'A142165513'	INITIAL VALUE	411	- CUST-IDNT-ALPHA-1
SPACES	MOVE FIG CONST	512	- CUST-MAIN-ACCOUNT-3RD-ID
'BB----1235'	INITIAL VALUE	431	- CUST-STORAGE-IDENT
VALUE UNKNOWN	UNKNOWN VALUE	1134	- C3-IDENT-ACCT
VALUE UNKNOWN	UNKNOWN VALUE	1116	- C3-SUB-ID-1A
SPACES	MOVE FIG CONST	1116	- C3-SUB-ID-1A
ZEROS	MOVE FIG CONST	1116	- C3-SUB-ID-1A
'A142165513'	INITIAL VALUE	411	- CUST-IDNT-ALPHA-1
'TK1236RTY6'	INITIAL VALUE	1125	- S2-CUST-ACCOUNT-TA
VALUE UNKNOWN	UNKNOWN VALUE	1134	- C3-IDENT-ACCT
ALL '4'	MOVE LITERAL	1134	- C3-IDENT-ACCT
VALUE UNKNOWN	UNKNOWN VALUE	1116	- C3-SUB-ID-1A
SPACES	MOVE FIG CONST	1116	- C3-SUB-ID-1A
ZEROS	MOVE FIG CONST	1116	- C3-SUB-ID-1A
'TY1236RTY6'	INITIAL VALUE	1125	- S2-CUST-ACCOUNT-TA
'TKY'	MOVE LITERAL	1125	- S2-CUST-ACCOUNT-TA
CURRENT-DATE	MOVE SPEC REG	1125	- S2-CUST-ACCOUNT-TA
ALL '4'	MOVE LITERAL	1134	- C3-IDENT-ACCT
SPACES	MOVE FIG CONST	1116	- C3-SUB-ID-1A
ZEROS	MOVE FIG CONST	1116	- C3-SUB-ID-1A
'TKY'	MOVE LITERAL	1125	- S2-CUST-ACCOUNT-TA
CURRENT-DATE	MOVE SPEC REG	1125	- S2-CUST-ACCOUNT-TA

A**B****C****D****Exhibit 22****Possible Literals with Possible Paths Report**

- A - Possible Literal
- B - How the Literal might be formed
- C - Sequence Number of the Related COBOL Field
- D - Tracing Path Levels - (*Follow dash up page to next shorter indentation*)

DCD III – Tracing & Analysis and Other Reports

VERB ANALYSIS REPORT						01/01/2001	PAGE 1
VERB	PROGRAM	SEQ-#	PARAGRAPH	SEQ-1	FIELD-1	SEQ-2	FIELD-2
MOVE	PAYROLL	00201	A080-COMPUTE-WEEKLY-PAY		2.75	00100	PN-WEEKLY-DEDUCT
MOVE	PAYROLL	00220	A800-READ-NEXT-RECORD		'Y'	00113	WS-END-OF-FILE-SW
OPEN	PAYMONTH	00128	010-OPEN-FILES	00020	PAYROLL-TIME-FILE		
OPEN	PAYMONTH	00128	010-OPEN-FILES	00064	PAYROLL-UPDATED-FILE		
OPEN	PAYROLL	00126	010-OPEN-FILES	00020	PAYROLL-TIME-FILE		
OPEN	PAYROLL	00126	010-OPEN-FILES	00064	PAYROLL-UPDATED-FILE		
PERFORM	PAYMONTH	00137	040-PROCESS-ALL-RECORDS	00145	A-PROCESS-PAYROLL-RECS		
.							
SUBTRACT	PAYMONTH	00181	A080-COMPUTE-WEEKLY-PAY	00103	PN-WEEKLY-DEDUCTION-2	00112	WS-ALL-DEDUCTIONS
SUBTRACT	PAYMONTH	00185	A080-COMPUTE-WEEKLY-PAY	00102	PN-WEEKLY-DEDUCTION-1	00112	WS-ALL DEDUCTIONS

A B C D E F G H

Exhibit 23

Verb Analysis Report

- A - COBOL verb
- B - COBOL program-id
- C - Sequence number where the verb resides
- D - Paragraph name that contains the verb
- E - Sequence number in Data Division where first operand resides
- F - Data name of first operand
- G - Sequence number in Data Division where second operand resides
- H - Data name of second operand

COUNT	VERB	PERCENT	VERB ANALYSIS REPORT	(SUMMARY PAGE)
15	MOVE	13.8 %		
10	IF	9.2 %		
04	READ	3.7 %		
02	WRITE	1.8 %		
02	ACCEPT	1.8 %		
30	COMPUTE	27.5 %		
.				
.				
04	PERFORM	3.7 %		
02	GO TO	1.8 %		
109	* TOTAL *	100.0 %		

A B C

Exhibit 24

Verb Analysis Report - Summary Page

- A - Count of COBOL verbs
- B - COBOL verb
- C - Percent of verb usage compared to all verbs within the program(s)

01/01/2001 11:30

TRACING & ANALYSIS SELECT STATEMENTS

PAGE 1

1 SELECT IF NODE-1 = D*T AND NODE-2 = Y*R AND FIELD-SIZE = 1

* * End of SELECTS * *

/ NO ERRORS /

D*T
Finds
DATEY*R
finds
YEAR1
finds
PIC Xresides in pos 2 - 2 ... is key to
automatic selection of all REDEFINES,
GROUP, & Overlapping fields after this field

Page 1

Program

Seq#

Level

Data-Name

TRACING & ANALYSIS - FOLLOW A DATA FIELD

Directly Referenced Narrative /

01/01/2001

Section Sel#

NR30PGM1

0248

15

CW-DATE-2-YEAR-FORMAT

PIC X

WORK-ST

1

resides in pos 2 - 2 in record CW-RECORD

Narrative show all Procedure

If # = SPACES (1128,1278)

Division References in program ----->

If # not = 'R' (1310)

Move # to RN-DATES-N2-IN @367 (1287), RN-DATES-N2-P3 @368 (1196,1223)

\ Points to
1 SELECT

overlap

0244

01

CW-RECORD

WORK-ST

uses field pos 1 - 80

If # not = SPACES (792)

Move HIGH-VALUES to # (985,1380), SPACES to # (832,889,901,967,1481,1503)

overlap

0245

05

CW-MASTER-RECORD-1ST-OCC

WORK-ST

uses field pos 1 - 72

If # not = SPACES (1451,1489)

0249 88 CW-DATE-TYPE-3

(88 LEVEL HAS MULTIPLE VALUES) WORK-ST

If # (1286,1300,1308)

0252 88 CW-DATE-TYPE-2

VALUE 'M'

WORK-ST

If # (1280,1327)

If not # (1290,1304,1567,1581,1605)

/ redefines

0253

15

CW-DATE-2-YMD-2

PIC X

REDEFINES field at 0248 WORK-ST

If # not = SPACES (1076)

Move >E= to # (1172,1288), 'N' to # (1175,1293), 'U' to # (1177,1298)

overlap

0263

05

CW-MASTER-REC-2

REDEFINES field at 0244 WORK-ST

uses field pos 1 - 80

\----- identification of an Overlapping field -----/

\----- identification of a REDEFINES

Exhibit 25

Tracing & Analysis Report

Notes:

- Everything that can happen to CW-DATE-2-YEAR-FORMAT is shown on this page
- One field CW-DATE-2-YEAR-FORMAT is selected, and *every other possible field is listed* that is a **REDEFINES**, a **GROUP field** for this field, or in any way **Overlaps this field**
- **Narrative** - All Procedure Division References for every field identified is listed underneath each field
- See next page for infinite levels of TRACING

Page 2 **TRACE # 00001** **TRACING & ANALYSIS - FOLLOW A DATA FIELD**
 Program Seq# Level Data-Name Directly Referenced Narrative

01/01/2001
 Section

Tracing 1st Pass isolated the following affected fields

traced	0353	01	RN-RECORD		WORK-ST
				used field pos 1-80	
				Move SPACES to # (1409)	
				Move # to MT-FORMAT-RECORD @471 (1923)	[TRACED to Page 3 below]
traced	0367	05	RN-DATES-N2-IN	PIC X	WORK-ST
				uses field pos 2 - 2	
				If # = ZEROS (1623)	
				Move SPACES to # (1508,1627)	
				Move CW-DATE-2-YEAR-FORMAT to # (1287)	[TRACED from Page 1 on left]
traced	0367	05	RN-DATES-N2-P3	PIC X	WORK-ST
				uses field pos 3 - 3	
				If # = ZEROS (1624)	
				Move SPACES to # (1629)	
				Move CW-DATE-2-YEAR-FORMAT to # (1196,1223)	

Page 3 **TRACE # 00002** **TRACING & ANALYSIS - FOLLOW A DATA FIELD**
 Program Seq# Level Data-Name Directly Referenced Narrative

01/01/2001
 Section

Tracing 2nd Pass isolated the following affected fields

traced	0471	01	MT-FORMAT-RECORD		WORK-ST
				used field pos 1-68	
				Move SPACES to # (1386)	
				Move RN-RECORD @353 to # (1923)	[TRACED from Page 2 above]
traced	0476	05	MT-DATES-M3-P1	PIC X	WORK-ST
				uses field pos 2 - 2	
				If # = ZEROS (1923)	
traced	0476	05	MT-DATES-M3-P2	PIC X	WORK-ST
				uses field pos 3 - 3	
				If # = ZEROS (1923,2031)	
				Move SPACES to # (2052)	

Tracing 3rd Pass found no additional fields - No additional tracing passes possible

Exhibit 25 (Continued)

Notes:

- All Possible Uses are **traced** infinite levels and shown on this page
- **First tracing pass finds fields RN-DATES-N2-IN & RN-DATES-N2-P3** (*from previous page*)
- **Also picked up in first tracing pass is overlapping field RN-RECORD**
- **Second tracing pass finds field MT-FORMAT-RECORD** (*from this page*)
- **Also found in 2nd pass are overlapping fields MT-DATE-M3-1 & MT-DATE-M3-2**
- **Third tracing pass finds NO MATCHES and tracing is stopped here**

This page intentionally left blank

Data Dictionary Interface File

With the use of either PARM option DICT or SRA, a file is produced which may be used to interface with commercial Data Dictionary systems.

The file produced contains basically the same information, which is printed on the System Record Analysis Report except that it is in a file format rather than a printed report. The record size is 120 characters. The block size must be a multiple of 120. 3120 is recommended. The DDNAME is DCDDICT. The file is opened EXTEND for each program which allows for several programs to be run through during one run. See the PARM OPTIONS (DDF and CDD) for using the OPEN EXTEND across several DCD III runs.

See the COBOL COPY record SYS1RECD and the associated description of these fields under this heading for a description of the fields within these records. Also, see the System Record Analysis Report under the heading, “Overview of DCD III Other COBOL Reports”.

To create a Data Dictionary Interface File, control statements must be used. These control statements are the same control statements used in producing a System Record Analysis Report. In fact, the System Record Analysis Report will be printed unless the REPORTS DD is dummied out. Even though the control statements require program-id, only one program at a time may be used when creating a Data Dictionary Interface File.

See the next heading, “Control Statements for System Record Analysis”, for creating the control statements necessary for creating a Data Dictionary Interface File.

The SYS1RECD COPY record shown here may be used as a guide for creating a COBOL or other program to use this file. This COPY member is provided on the installation tape within the member CASEMBRS.

DCD III – Tracing & Analysis and Other Reports

000100**		COPY	SYS1RECD	
000200	01	SRA-SYSTEM-RECORD-ANAL-REC.		
000300	05	SRA-SET-NUMBER	PIC 999.	
000400	05	SRA-FROM-TO-LOCATIONS.		
000500	10	SRA-FROM-POSITION	PIC 9(6).	
000600	10	SRA-TO-POSITION	PIC 9(6).	
000700	05	SRA-PROGRAM-ID	PIC X(8).	
000800	05	SRA-RECORD-NAME	PIC X(30).	
000900	05	SRA-DATA-NAME	PIC X(30).	
001000	05	SRA-EXP-CMPLR-NBR	PIC 9(5).	
001100	05	SRA-LEVEL-NUMBER	PIC XX.	
001200	05	SRA-TYPE-USAGE	PIC X(2).	
001300	88	SRA-TYPE-COMP		VALUE 'C'.
001400	88	SRA-TYPE-COMP-1		VALUE 'C1'.
001500	88	SRA-TYPE-COMP-2		VALUE 'C2'.
001600	88	SRA-TYPE-COMP-3		VALUE 'C3'.
001700	88	SRA-TYPE-COMP-4		VALUE 'C4'.
001800	88	SRA-TYPE-DEFAULT-DISPLAY		VALUE ' '.
001900	88	SRA-TYPE-DISPLAY		VALUE 'D'.
002000	88	SRA-TYPE-DISPLAY-1		VALUE 'D1'.
002100	88	SRA-TYPE-INDEX		VALUE 'IX'.
002200	88	SRA-TYPE-USAGE-IS-INDEX		VALUE 'UI'.
002300	88	SRA-TYPE-POINTER		VALUE 'PR'.
002400	05	SRA-FLAGS.		
002500	10	SRA-SET-FLAG	PIC X.	
002600	88	SRA-DIRECT-SET		VALUE 'S'.
002700	10	SRA-USED-FLAG	PIC X.	
002800	88	SRA-DIRECT-USED		VALUE 'U'.
002900	10	SRA-TESTED-FLAG	PIC X.	
003000	88	SRA-DIRECT-TESTED		VALUE 'T'.
003100	10	SRA-INDIR-SET-FLAG	PIC X.	
003200	88	SRA-INDIRECT-SET		VALUE 'S'.
003300	10	SRA-INDIR-USED-FLAG	PIC X.	
003400	88	SRA-INDIRECT-USED		VALUE 'U'.
003500	10	SRA-INDIR-TESTED-FLAG	PIC X.	
003600	88	SRA-INDIRECT-TESTED		VALUE 'T'.
003700	05	SRA-PICTURE-CLAUSE	PIC X(21).	
003800	05	SRA-FILLER	PIC X.	

The following is a description of the fields within the SYS1RECD COPY member:

1. SRA-SET-NUMBER Sequential number starting at 1 showing which set of control cards produced this set of records.
2. SRA-FROM -TO- LOCATIONS From and to record positions for this data name.
3. SRA-PROGRAM-ID Taken from the program-id in the Identification Division.
4. SRA-RECORD-NAME The name of the 01 record which contains this data name.
5. SRA-DATA-NAME Self-explanatory
6. SRA-EXP-CMPLR-NBR Compiler number or sequence number in the program of this data name.
7. SRA-LEVEL-NUMBER 01-49 level number associated with this data name.
8. SRA-TYPE-USAGE Usage of this data name.
9. **SRA-FLAGS** Contains flags showing how this data field is used in the program.
 - **Direct flags** indicate this field was involved in the Procedure Division statement which caused this flag.
 - **Indirect flags** indicate this field was not involved in the Procedure Division statement which caused this flag.
 - **SET** indicates that this field is changed.
 - **USED** indicates that this field was used to modify in some way another field.
 - **TESTED** indicates that this field was used in a comparison.
10. SRA-PICTURE-CLAUSE The first 21 characters of the PICTURE clause associated with this data name.
11. SRA-FILLER Not used at this time.

WRITPDS and READPDS Options

In DCDIII, a feature is available to allow the storing of work file information for all reports with the exception of the Tracing & Analysis report. The storing of information is done in condensed format as members on partitioned data sets for use at any time for printing reports from. This feature is controlled by two options and requires extra JCL.

To store information using this feature, do the following:

1. Catalog a partitioned data set with the following attributes:
RECFM=FB
LRECL=3120
BLKSIZE=3120 or a multiple of 3120
DSORG=PO
2. Add a DD statement after the EXEC DCDCOBOL PROC as follows:
//DCD.DCDPDS DD DSN=PDS.WITH.3120.LRECL,DISP=SHR
3. Run one or several DCD III runs using one or several programs each time. With each run, do the following:
 - a) Add the WRITPDS PARM option to the DCDCOBOL PROC as shown in the example below. Also add parm option BEGIN or CONT. BEGIN is used during the first run to clear everything on the user created PDS. The CONTINUE option is used for all subsequent runs and will not clear the user PDS.
 - b) Add every one of the report options that will be wanted as stored information on the PDS. It will be impossible to later run a READPDS for a report option not specified on the WRITPDS run. Conversely, it is important not to specify unwanted options as the workspace on the PDS will be wasted with unused information.

An example follows:

```
//STEP1 EXEC DCDCOBOL,DATA=,LAYOUT=,PARA=,OTHER='WRI,BEGIN'  
//DCD.COBLIN DD DSN=USER.COBL.LIBRARY(PROG1),DISP=SHR  
// DD DSN=USER.COBL.LIBRARY(PROG2),DISP=SHR  
//DCD.COPYLIB DD DSN=USER.COPY.LIB,DISP=SHR  
//DCD.DCDPDS DD DSN=USER.CREATED.PDS,DISP=SHR
```

The above example will store the information necessary to produce the System Data Name Cross Reference, System Paragraph Cross Reference and Layout Reports for programs PROG1 and PROG2 on the partitioned data set for later printing. The reports will not appear in this run. An exception to this is the report showing the control statements used when using the SRA option for the System Record Analysis Report, which will appear when using the WRITPDS option. It will not appear when using the READPDS option.

To **print** reports using this feature, do the following:

1. Put a step invoking the Member Fetcher PROC ahead of the EXEC DCDCOBOL PROC to bring in the members wanted. Point the user PDS in the Member Fetcher PROC to the same PDS used in the DCDPDS ddname in the DCDCOBOL step with the WRITPDS option. For information on using the PROC MBRFETCH, see the heading “Control Statements for Using the MBRFETCH PROC” in this section.

The possible combinations of control statements that may be used with the READPDS option are shown here:

- a) INDD=ddname=ALL
- b) INDD=ddname=MEMBER
- c) INDD=ddname=PREFIX=prefix

Do **NOT** use the COBOL or COBRECS combination here.

2. Add an extra DD with the name of DCDREPDS after the DCDCOBOL PROC as shown in the example that follows:
3. Add the READPDS option by using OTHER=REA on the DCDCOBOL PROC line.
4. Only use PARM options that have also been used consistently in WRITPDS DCD III runs.

An example follows:

```
//STEP1 EXEC  MBRFETCH
//MBR.OUTSET DD  DCB=(LRECL=3120,BLKSIZE=3120)
//MB.CTLCDMBR DD  *
INDD=USERDD=PREFIX=TRN20
INDD=USERDD=PREFIX=TRN30
//MBR.USERDD DD  DSN=PDS.WITH.3120.LRECL,DISP=SHR
//*
//STEP2 EXEC  DCDCOBOL,DATA=,PARA=,OTHER=REA
//DCD.DCDREPDS DD  DSN=&&PASSFILE,DISP=(OLD,PASS),
//              DCB=LRECL=3120
```

In the previous example, both the System Data Name Cross Reference and System Paragraph Cross Reference Reports will be produced using the information created during the WRITPDS run.

Do not use the WRITPDS option along with the READPDS option. The information created by using the WRITPDS option will not be available for use by the READPDS option.

PDSFETCH - Newer Software to replace MBRFETCH

This software is newer software with additional features to use in place of our older MBRFETCH software. The JCL is different and with one additional work file. The format of the control statements is almost the same with additional features. Older control statements within MBRFETCH need to have INDD= replaced with PDS= for those statements to work within this new software.

Additional Features within this software include the following:

- 1) A Member Count is shown for each DDNAME processed
- 2) Enhanced Error Messages to identify any error in processing control statements or for errors found when handling any of the files including the Partitioned Data Sets pointed to by the PDS=DDNAME= control statements.
- 3) A MULTIPLE option to allow multiple PREFIXs for three types of control statements; COBOL=, COBRECS=, and PREFIX=. Use of this option can allow one pass for multiple PREFIXs, as opposed to many passes with the older MBRFETCH software.

Important Note: Partitioned Data Sets with this software as with the older MBRFETCH software may NOT be concatenated. In place of concatenation, additional control statements pointing to additional DDNAMES must be used to point to each DDNAME that would otherwise be concatenated.

JCL for the PDSFETCH software

The following JCL is used for running this software.

```
//STEP EXEC PGM=PDSFETCH,REGION=1024K
//OUTPUT DD DSN=&&PASSFILE,DISP=(,PASS),UNIT=SYSDA,
// SPACE=(CYL,(5,10))
//PRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//WORKFIL1 DD UNIT=SYSDA,SPACE=(TRK,(1,2))
//WORKFIL2 DD UNIT=SYSDA,SPACE=(CYL,(1,2))
//CTLINFO DD *
Control statements inserted here
/*
```

Additional DD statements are required that point to Partitioned Data Sets used to get MEMBERS from. The DDNAMES for these DDs are inserted into the control statements the user adds. See examples provided over the next several pages.

A PROC for invoking this JCL is shown on the next page.

PROC for the PDSFETCH software

The following PROC (PDSFETCH) is available on the file DCDIII.CNTL provided by our install files.

```
//PDSFETCH PROC PRINT=*,WORK=SYSDA
//PDS      EXEC PGM=PDSFETCH,REGION=1024K
//OUTPUT   DD DSN=&&PASSFILE,DISP=(,PASS),UNIT=&WORK,
//          SPACE=(CYL,(5,10))
//PRINT     DD SYSOUT=&PRINT
//SYSOUT    DD SYSOUT=&PRINT
//WORKFIL1 DD UNIT=&WORK,SPACE=(TRK,(1,2))
//WORKFIL2 DD UNIT=&WORK,SPACE=(CYL,(1,2))
```

The PDS Member Fetcher PROC MBRFETCH is available for pulling one or several members from a partitioned data set (PDS) and creating one sequential data set. This data set may be used as input to the DCDCOBOL PROC. (This PROC is also embedded as one step in the DCDJCL PROC which is covered in the section on JCL PROC Analysis Reports Facility.)

Control Statements for Using the MBRFETCH PROC

The DD name for these control statements is CTLINFO. This is different from the DD name used with the MBRFETCH program.

As with the MBRFETCH program, control statements are used to indicate which members are to be selected. The first these control statements is shown below. The second format, sometimes used for adding a PREFIX or MEMBER name is discussed where appropriate in the next several pages and examples.

Columns 1 through 3	-The constant (PDS)
Column 4	- An equal sign (=)
Columns 5 through <i>a</i>	- The name of a made-up DDNAME
Column <i>a</i> +1	- An equal sign (=) with no spaces before or after it.
Columns <i>b</i> through <i>c</i>	- One of the following words: 1. ALL 2. COBOL 3. COBRECS 4. MEMBER 5. PREFIX

The remaining two fields are required if PREFIX was used as the last operand and are optional if COBOL or COBRECS was used as the last operand. They are not permitted if either ALL or MEMBER is used as an operand.

Column <i>c</i> +1	-An equal sign (=) with no spaces before or after it.
Columns <i>d</i> through <i>e</i>	-A 1 to 7 characters prefix which will be used to limit selection or an eight character word (MULTIPLE) to indicate that multiple PREFIX control statements will follow this control statement.

An illustration of the possible combinations is listed below:

1. PDS=ddname=ALL
2. PDS=ddname=COBOL
3. PDS=ddname=COBRECS
4. PDS=ddname=MEMBER
5. PDS=ddname=PREFIX=prefix
6. PDS=ddname=COBOL=prefix
7. PDS=ddname=COBRECS=prefix
8. PDS=ddname=PREFIX=MULTIPLE
9. PDS=ddname=COBOL=MULTIPLE
10. PDS=ddname=COBRECS=MULTIPLE

An example showing the use of these control statements is listed below:

```
//STEP1 EXEC PDSFETCH
//PDS.OUTPUT DD DCB=(LRECL=80,BLKSIZE=3120)
//PDS.CTLINFO DD *
INDD=USERDD=COBOL
//PDS.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

The above example will pull off all COBOL programs from the library specified by the DD name USERDD and pass them out to a sequential data set defined by the OUTPUT DD.

If members are wanted from more than one PDS library, then multiple DD names may be used. An example is listed below:

```
//STEP2 EXEC PDSFETCH
//PDS.OUTPUT DD DCB=(LRECL=80,BLKSIZE=3120)
//PDS.CTLINFO DD *
INDD=USERDD=COBOL
INDD=OTHERDD=COBOL
//PDS.USERDD DD DSN=USER.COBOL.LIBRARY1,DISP=SHR
//PDS.OTHERDD DD DSN=USER.COBOL.LIBRARY2,DISP=SHR
```

The use of the following words for selection is described here:

ALL, COBOL, COBRECS, MEMBER, PREFIX=prefix, COBOL=prefix, COBRECS=prefix, PREFIX=MULTIPLE, COBOL=MULTIPLE, and COBRECS=MULTIPLE

- | | |
|----------------|---|
| ALL | ALL indicates that all members will be selected from the partitioned data set. This option should not be used here. Instead, the options COBOL or COBRECS should be used. An exception to this rule occurs when using the READPDS option, in which case, ALL should be used and COBOL and COBRECS <u>may not</u> be used. |
| COBOL | COBOL indicates that all COBOL programs are to be pulled off of the partitioned data set. If non-COBOL programs are found, then they are discarded. |
| COBRECS | COBRECS indicates that all COPY members which are 01 records or record groups beginning with a level number (02-48) are to be pulled off of the PDS. Those members that do not conform to this selection are bypassed for selection. COBRECS may be used when the LOR option is used. |

- MEMBER** MEMBER indicates that the user wants to make further selection with member name control statements. These control statements immediately follow the control statement which contains the word MEMBER and must begin in column 1 and contain a 1-8 character MEMBER NAME. An example is listed below:
- ```
//STEP3 EXEC PDSFETCH
//PDS.CTLINFO DD *
PDS=USERDD=MEMBER
MEMBER1
MEMBER2
MEMB03
MEMBER04
//PDS.USERDD DD DSN=USER.PDS,DISP=SHR
```
- PREFIX** PREFIX indicates that selection is to be done strictly on the basis of a 1 to 7 character prefix that follows the (=) sign after the word PREFIX, or that selection is done with multiple prefixes if the characters MULTIPLE follow the (=) sign. It is recommended that this option not be used here and that **COBOL=prefix** or **COBRECS=prefix** be used in its place. Prefix is discussed more below and is shown in some of the COBOL= (prefix or MULTIPLE) examples that follow.

Six ways to use selection via a prefix follow:

- COBOL=prefix** pulls off all COBOL programs from the PDS specified that begins with the same letters that are specified in the prefix. The prefix may not exceed seven characters in length.
- COBRECS=prefix** pulls off all COPY members that have as their first COBOL record, a level number from (01-49) that have a matching prefix. This is used when using PARM option LOR for LAYOUTS. Again prefix may not exceed seven characters in length.
- PREFIX=prefix** pulls off all members that have a matching 1-7 character prefix.
- COBOL=MULTIPLE** requires that one or more control statements follow this control statement and that each control statement begin in column 1 and contain a 1-7 character prefix. Used for pulling off COBOL programs.
- COBRECS=MULTIPLE** requires that one or more control statements follow this control statement and that each control statement begin in column 1 and contain a 1-7 character prefix. Used for pulling off COBOL records.
- PREFIX=MULTIPLE** requires that one or more control statements follow this control statement and that each control statement begin in column 1 and contain a 1-7 character prefix.

An example where selection is done with the use of one prefix follows:

```
//STEP4 EXEC PDSFETCH
//PDS.OUTPUT DD DCB=(LRECL=80,BLKSIZE=3120)
//PDS.CTLINFO DD *
INDD=USERDD=COBOL=M4
//PDS.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

In the above example, all COBOL programs whose member name begins with the prefix M4 will be selected for passing on to the next step.

An example where selection is done for multiple prefixes on one PDS follows:

```
//STEP5 EXEC PDSFETCH
//PDS.OUTPUT DD DCB=(LRECL=80,BLKSIZE=3120)
//PDS.CTLINFO DD *
INDD=USERDD=COBOL=MULTIPLE
M4
M5
M6A
M6B
//PDS.USERDD DD DSN=USER.COBOL.LIBRARY,DISP=SHR
```

In the above example, all COBOL programs whose member name begins with the prefixes M4, M5, M6A, and M6B will be selected for passing on to the next step.

Further notes on this PDS member fetch software follow:

1. Compressing the PDS if possible before accessing the data set may be helpful.
2. Multiple control statements may be used for any of the above formats.
3. To prevent obtaining duplicate members, do not mix formats in the same run.
4. When using PDSFETCH with the READPDS option, (see the heading 'WRITPDS and READPDS Options' in this section), replace LRECL=80 with LRECL=3120 in the  
//PDS.OUTPUT DD.
5. A listing of Error Message that may be generated from this utility are listed on the next two pages.
6. Look for a return code of Zero if there are no errors. Look for a return code of 0008 if errors are found.

**Error Messages that may be generated from PDSFETCH utility.**

| <u>Message #</u> | <u>Message</u>                                                                        |
|------------------|---------------------------------------------------------------------------------------|
| 040              | Expecting extra control statements for previous PDS=                                  |
| 050              | No Members were found                                                                 |
| A011             | Expecting extra control statements for just previous PDS=                             |
| A021             | Found blank control statement                                                         |
| A022             | Looking for PDS= in columns 1-5                                                       |
| A023             | Looking for first MEMBER in columns 1-8                                               |
| C011             | DDNAME found has more than 8 CHARS                                                    |
| C012             | User DDNAME not found after PDS=                                                      |
| C013             | DDNAME has invalid Non-Alphanumeric CHAR                                              |
| C014             | Undefined error in previous entry                                                     |
| C021             | Type DDNAME not found after DDNAME=                                                   |
| C022             | Type DDNAME has invalid characters                                                    |
| C023             | Expecting ALL, COBOL, COBRECS, MEMBER or PREFIX                                       |
| C041             | PREFIX not found after XXXX<br>(where XXXX = ( COBOL= ), ( COBRECS= ), or ( PREFIX= ) |
| C042             | Prefix must be 7 characters or less                                                   |
| C043             | Invalid Character found in PREFIX                                                     |
| C050             | More than 20 PDS= records                                                             |
| C060             | Duplicate DDNAME Invalid - BYPASSING                                                  |
| D011             | Unidentified Control Record - BYPASSING                                               |
| D012             | Found more than 100 Member or Prefix records for last PDS=                            |
| D013             | Invalid Character found in previous entry - Previous Entry ignored                    |
| D014             | Previous PREFIX over 7 characters in length and ignored                               |
| D015             | Previous MEMBER over 8 characters in length and ignored                               |
| E011             | ERROR with DDNAME (DDNAME) Error in finding                                           |
| E012             | ERROR - DDNAME (DDNAME) was not found                                                 |
| E013             | ERROR - DDNAME (DDNAME) Error found while validating DDNAME                           |
| E031             | ERROR with internal work file with DDNAME (WORKFIL1)                                  |
| E032             | ERROR - while reading PDS with DDNAME (DDNAME)                                        |

|      |                                                                                  |
|------|----------------------------------------------------------------------------------|
| EA31 | ERROR - Member (Member-Name) not found on DDNAME (DDNAME) for just previous PDS= |
| EA32 | ERROR while opening file specified in DDNAME (DDNAME)                            |
| EA33 | ERROR while accessing file specified in DDNAME (DDNAME)                          |
| EA34 | ERROR - Problem while reading all members from UNCONCATENATED DDNAME (DDNAME)    |
| EB41 | ERROR - Member (Member-Name) not found on DDNAME (DDNAME) for just previous PDS= |
| EB42 | ERROR while opening file specified in DDNAME (DDNAME)                            |
| EB43 | ERROR while accessing file specified by DDNAME (DDNAME)                          |
| EB44 | ERROR - problem while reading all Members from UNCONCATENATED DDNAME (DDNAME)    |
| G011 | ERROR while reading record from DDNAME (DDNAME)                                  |
| G012 | ERROR with LRECL found for PDS for DDNAME (DDNAME) - should be 80 or 3120        |
| G013 | ERROR while reading record in DDNAME (DDNAME)                                    |
| G030 | ERROR - Unexpected Record Length of (nnnnn) - Must be 80 or 3120                 |
| H011 | ERROR while reading from DDNAME (DDNAME)                                         |
| H012 | ERROR with LRECL found for PDS for DDNAME (DDNAME) - Should be 80 or 3120        |
| H013 | ERROR while reading record in DDNAME (DDNAME)                                    |
| H040 | ERROR - Unexpected Record Length of (nnnnn) - Must be 80 or 3120                 |
| I020 | ERROR in logic for M2-NBR-CHARS = 'NNNN'                                         |

Release 3.6

DCD III

Computer Aided Software Engineering

(CASE) Files

Use for CASE Files

These files were made available with the new DCD III system for two reasons. The first reason is in response to our existing users of our older DCD II system. The second reason is in response to the general direction that the industry is going with respect to re-engineering of COBOL programs.

These files are for the most part not intended to be an end result of re-engineering. Instead, these files are designed to facilitate the users in getting to the end result chosen for their particular company. The CASE files provide a very complete and accurate set of files. They cross reference both the DATA and the PROCEDURE divisions of one or more COBOL programs.

At **MARBLE Computer Inc.**, the observation obtained from people who use our software is that no two users follow the exact same course in re-engineering. The usage varies greatly; it is dependent upon the needs of each particular company.



DCD III

**Computer Aided Software Engineering (CASE) Files**

**TABLE OF CONTENTS**

|                                                           |      |
|-----------------------------------------------------------|------|
| Use of This Section .....                                 | C-4  |
| List of Exhibits.....                                     | C-4  |
| Rules and Restrictions When Producing CASE Files .....    | C-5  |
| Sample JCL to Access and Print CASE Files .....           | C-6  |
| Data Division Only Files                                  |      |
| Data Division Information File .....                      | C-7  |
| Data Analysis Information File .....                      | C-11 |
| Procedure Division Only Files                             |      |
| Procedure Division Information File.....                  | C-15 |
| Basic Paragraph Information File.....                     | C-23 |
| Paragraph Range Information File .....                    | C-27 |
| Transfer of Control Narrative File.....                   | C-30 |
| Procedure Division to Data Division Cross Reference Files |      |
| Procedure Division Reference to the Data Division .....   | C-33 |
| Procedure Division Narrative for Each Data Field .....    | C-36 |
| Expanded Procedure Division Narrative .....               | C-38 |
| Data Dictionary Interface File .....                      | C-41 |
| PICTURE and Literal File .....                            | C-45 |

### Use of This Section

This section is provided to make the user aware of the work files that are available. They may be used for in-house projects which deal in the re-engineering of COBOL software.

Within this section, the information provided in these files is shown in two formats:

1. A listing of a COBOL COPY record which was used internally within DCD III to create this file.
2. A description of the fields within that record

The COBOL COPY records that were used internally to build these files are made available with this software package. All eleven COPY members are contained within the COPY member CASEMBRS on the second file of the installation tape. See Installation Guide sub-heading, “**DCD III JCL Procedure Library**”.

### List of Exhibits

|     |                                                                         |      |
|-----|-------------------------------------------------------------------------|------|
| 1.  | COPY Member for the Data Division Information File .....                | C-8  |
| 2.  | COPY Member for the Data Analysis Information File .....                | C-12 |
| 3.  | COPY Member for the Procedure Division Information File.....            | C-16 |
| 4.  | COPY Member for the Basic Paragraph Information File .....              | C-24 |
| 5.  | COPY Member for the Paragraph Range Information File .....              | C-28 |
| 6.  | COPY Member for the Transfer of Control Narrative File .....            | C-31 |
| 7.  | COPY Member for Procedure Division References to the Data Division..... | C-34 |
| 8.  | COPY Member for Procedure Division Narrative for Each Data Field .....  | C-37 |
| 9.  | COPY Member for the Expanded Procedure Division Narrative.....          | C-39 |
| 10. | COPY Member for the Data Dictionary Interface File.....                 | C-42 |
| 11. | COPY Member for the PICTURE and Literal File.....                       | C-46 |

### **Rules and Restrictions When Producing CASE Files**

These work files are available by using DCD III following the instructions given in one of two earlier sections and by following the additional instructions given here. The earlier sections are:

1. Alternate Compile Listing Facility
2. Other COBOL Reports Facility

When creating these files, a DCD III report or reports must be run from one of the two previously mentioned sections using the instructions given there. Additionally, other instructions and certain restrictions are listed in this section.

If the report generated is not wanted and the file generated is from the Other COBOL Reports Facility Section, then the REPORTS DD may be dummied out. The PRINT DD should never be dummied out as errors that occur during running are listed there.

The following are restrictions that must be followed when creating these records:

1. Only one program at a time may be run when producing the CASE files. (An exception to this is the Data Dictionary Interface File).
2. When accessing a particular file and the instructions indicate that a certain report option needs to be turned on, other report options should not be turned on unless indicated. In most cases, turning on another report option will cause the file wanted by the user to be destroyed before the user can access the file.
3. Do not pre-catalog, change the LRECL or modify the DISP for the work file which contains the information desired. The work files in DCD III are used multiple times within one DCD III run with varying record sizes. When the DCD III step is finished, the user may copy the file to another file using the LRECL and BLKSIZE specified individually for that file within this section.
4. When using the file created, the first record in each file must be bypassed if characters 1 through 6 in that record contain the constant MARBLE. These records are control records which contain date and time information and LRECL of the file.

More specific instructions are given with each available work file.

Note: The Data Dictionary Interface File mentioned in the Table of Contents does not specifically follow rules given here. Instead, it is more fully documented in the Other COBOL Reports Facility Section in this manual.

### Sample JCL to Access and Print CASE Files

The following JCL is provided as an example of how to access the CASE files shown in this section.

The sample provided here traps the Data Analysis Information File and prints it out in a following IEBGENER step. To catalog a CASE file or otherwise save the information, provide other JCL in the SYSUT2 DD of the IEBGENER step.

```
//STEP1 EXEC DCDCOBOL,DATA=
//DCD.DCDWK03 DD DSN=&&TEMP03,DISP=(,PASS),DCB=(LRECL=0,RECFM=U)
//DCD.COBLIN DD DSN=USER.COBL.PDS (USERPROG) ,DISP=SHR
//DCD.COPYLIB DD DSN=USER.COPY.LIB,DISP=SHR
//*
//STEP2 EXEC PGM=IEBGENER,REGION=256K
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=&&TEMP03,DISP=(OLD,DELETE) ,
// DCB=(RECFM=FB,LRECL=130,BLKSIZE=3120)
//SYSUT2 DD SYSOUT=*, DCB=(RECFM=FB,LRECL=130,BLKSIZE=3120)
```

Note - As of release 3.2, a full DCB is required on both SYSUT1 and SYSUT2 when accessing CASE files AND a DCB as shown above on the //DCD.DCDWKnn file must be present [ DCB=(LRECL=0,RECFM=U) ].

### **Data Division Information File**

This file contains a description of the Data Division attributes for each field in the Data Division.

To produce the file:

1. Use one of the PROCs in the Other COBOL Reports Facility Section of this manual, (e.g. PROC DCDCOBOL).
2. Use the PROC Symbolic DATA= on the EXEC DCDCOBOL (or other PROC name) line to turn on the DATA PARM option.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK01.
2. The LRECL of the file is 104.

The following pages show the internally used Data Division COPY member that was used to create the file and a description on the related fields within the record.

```

** COPY INFOTYP1 (IN1)
01 CIE-1-DATA-DIVISION-INFO.
 05 CIE-1-SOURCE-LINE-NUMBER PIC 9(5) .
 05 CIE-1-EXPANDED-LINE-NUMBER PIC 9(5) .
 05 CIE-1-TYPE-FLAG PIC X.
 88 CIE-1-TYPE-NORMAL VALUE SPACE.
 88 CIE-1-TYPE-FILE-NAME VALUE 'F' .
 88 CIE-1-TYPE-REC-CONTAINS VALUE 'R' .
 88 CIE-1-TYPE-BLK-CONTAINS VALUE 'B' .
 88 CIE-1-TYPE-LABEL-RECORDS VALUE 'L' .
 88 CIE-1-TYPE-RECORDING-MODE VALUE 'M' .
 88 CIE-1-TYPE-VALID VALUES ARE ' '
 'F'
 'R'
 'L'
 'M'
 'B' .

 05 CIE-1-LEVEL-NUMBER PIC 99.
 05 CIE-1-DATA-NAME PIC X(30) .
 05 CIE-1-START-ADDRESS PIC 9(9) .
 05 CIE-1-END-ADDRESS PIC 9(9) .
 05 CIE-1-OCCURS-FACTOR PIC 9(9) .
 05 CIE-1-REDEFINES-SRC-LINE PIC 9(9) .
 05 CIE-1-REDEFINES-EXP-LINE PIC 9(5) .
 05 CIE-1-ITEM-TYPE PIC X.
 88 CIE-1-ITEM-GROUP VALUE 'G' .
 05 CIE-1-SECTION-IND PIC X.
 88 CIE-1-SECTION-FILE VALUE 'F' .
 88 CIE-1-SECTION-WS VALUE 'W' .
 88 CIE-1-SECTION-LINKAGE VALUE 'L' .
 88 CIE-1-SECTION-REPORT VALUE 'R' .
 88 CIE-SECTION-COMM VALUE 'C' .
 05 CIE-1-TYPE-USAGE PIC XX.
 88 CIE-1-TYPE-COMP VALUE 'C' .
 88 CIE-1-TYPE-COMP-1 VALUE 'C1' .
 88 CIE-1-TYPE-COMP-2 VALUE 'C2' .
 88 CIE-1-TYPE-COMP-3 VALUE 'C3' .
 88 CIE-1-TYPE-COMP-4 VALUE 'C4' .
 88 CIE-1-TYPE-DEFAULT-DISPLAY VALUE ' ' .
 88 CIE-1-TYPE-DISPLAY VALUE 'D' .
 88 CIE-1-TYPE-DISPLAY-1 VALUE 'D1' .
 88 CIE-1-TYPE-INDEX VALUE 'IX' .
 88 CIE-1-TYPE-USAGE-IS-INDEX VALUE 'UI' .
 88 CIE-1-TYPE-POINTER VALUE 'PR' .
 05 CIE-1-CLASS PIC X.
 88 CIE-1-CLASS-ALPHA VALUE 'X' .
 88 CIE-1-CLASS-EDITED VALUE 'E' .
 88 CIE-1-CLASS-NUMERIC VALUE 'N' .
 05 CIE-1-INTEGGER-DIGITS PIC 99.
 05 CIE-1-FRACTION-DIGITS PIC 99.
 05 CIE-1-ALT-OCCURS-NBR-GROUP-6.
 10 CIE-1-USE-COMP-FIELD-SW PIC X (01).
 88 CIE-1-USE-COMP-FIELD VALUE 'Y' .
 10 CIE-1-OCCURS-NBR-TIMES PIC 9(05).

** NEXT 4 LINES REDEFINE *****ABOVE 5 BYTES. *****
 10 CIE-1-OCCURS-NBR-GROUP-5 REDEFINES
 CIE-1-OCCURS-NBR-TIMES.
 15 CIE-1-COMP-UNUSED-BYTE PIC X(10).
 15 CIE-1-OCCURS-NBR-TIMES-COMP COMP PIC 9(09).

```

## Exhibit 1

### COPY Member for the Data Division Information File

**Description of Individual Fields**

|     |                              |                                                                                                                                                    |
|-----|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | CIE-1-SOURCE-LINE-NUMBER     | Do not use.                                                                                                                                        |
| 2.  | CIE-1-EXPANDED-LINE-NUMBER   | The expanded sequence number (after COPY members have been resolved and brought into the program) where this data field is defined in the program. |
| 3.  | CIE-1-TYPE-FLAG              | An indicator which has a value of SPACE for most records and a non-space value for records containing FD information.                              |
| 4.  | CIE-LEVEL-NUMBER             | Level number associated with the data field in this record.                                                                                        |
| 5.  | CIE-DATA-NAME                | Data name.                                                                                                                                         |
| 6.  | CIE-1-START-ADDRESS          | Beginning position relative to 1 for this record where this field starts within the 01 record.                                                     |
| 7.  | CIE-1-END-ADDRESS            | Ending position of this field within the record.                                                                                                   |
| 8.  | CIE-1-OCCURS-FACTOR          | How many times this field OCCURS (dictated by 1 or more OCCURS).                                                                                   |
| 9.  | CIE-1-REDEFINES-SRC-LINE     | Do not use.                                                                                                                                        |
| 10. | CIE-1-REDEFINES-EXP-LINE     | For fields containing the REDEFINES clause, this field contains the expanded sequence number of the field being redefined.                         |
| 11. | CIE-1-ITEM-TYPE              | Contains a 'G' for group fields.                                                                                                                   |
| 12. | CIE-1-SECTION-IND            | Contains one letter indicating the section where this record or data name is in the Data Division.                                                 |
| 13. | CIE-1-TYPE-USAGE             | Two characters which indicate USAGE.                                                                                                               |
| 14. | CIE-1-CLASS                  | Contains one character which helps to define numeric or numeric edited fields.                                                                     |
| 15. | CIE-1-INTEGGER-DIGITS        | For numeric fields, the number of whole digits to the left of the decimal.                                                                         |
| 16. | CIE-1-FRACTION-DIGITS        | For numeric fields, the number of decimal digits to the right of the decimal.                                                                      |
| 17. | CIE-1-ALT-OCCURS-NBR-GROUP-6 | Do not use.                                                                                                                                        |
| 18. | CIE-1-USE-COMP-FIELD-SW      | Contains a 'Y' if CIE-1-OCCURS-NBR-TIMES-COMP should be used in place of CIE-1-OCCURS-NBR-TIMES.                                                   |
| 19. | CIE-1-OCCURS-NBR-TIMES       | The number of OCCURS as indicated within the OCCURS clause.                                                                                        |

*This Page intentionally left blank*



### **Data Analysis Information File**

This file is similar to the Data Division Information File and has the PICTURE and VALUE clauses as additional fields.

To produce the file:

1. Use one of the PROCs in the Other COBOL Reports Facility section of this manual (e.g. PROC DCDCOBOL).
2. Use the PROC Symbolic DATA= on the EXEC DCDCOBOL (or other PROC name) line to turn on the DATA PARM option.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK03.
2. The LRECL of the file is 130.

The following pages show the internally used Data Division COPY member that was used to create the file and a description of the related fields within the record.

```

** COPY NRRECORD.
01 NR-1-DATA-DIVISION-INFO.
05 NR-DATA-NAME PIC X(30).
05 NR-PROGRAM-NAME PIC X(08).
05 NR-CMPLR-NBR PIC 9(05).
05 NR-SEQ-NBR PIC 9(04).
88 NR-DATA-DIV-REC VALUE 0000.
05 NR-NARRATIVE.
10 NR-PICTURE-CLAUSE.
15 NR-PICTURE-POS1 PIC X(01).
15 FILLER PIC X(20).
10 NR-VALUE-LITERAL.
15 NR-OCCURS-NBR PIC 9(08).
15 NR-OCCURS-FACTOR PIC 9(08).
15 NR-OCCURS-ALT-END PIC 9(08).
15 NR-VALUE-LAST-6 PIC X(06).
10 NR-LEVEL-NBR PIC X(02).
10 NR-LEVEL-NBR-N REDEFINES NR-LEVEL-NBR PIC 9(02).
10 NR-FROM-NBR PIC 9(06).
10 NR-TO-NBR PIC 9(06).
10 NR-GROUP-FLAG PIC X(01).
88 NR-GROUP VALUE 'G'.
10 NR-SECTION-IND PIC X(01).
88 NR-SECTION-FILE VALUE 'F'.
88 NR-SECTION-W-S VALUE 'W'.
88 NR-SECTION-LINKAGE VALUE 'L'.
88 NR-SECTION-REPORT VALUE 'R'.
88 NR-SECTION-COMM VALUE 'C'.
10 NR-USAGE-IND PIC X(02).
88 NR-USAGE-COMP VALUE 'C'.
88 NR-USAGE-COMP-3 VALUE 'C3'.
88 NR-USAGE-COMP-4 VALUE 'C4'.
88 NR-USAGE-DEFAULT-DISPLAY VALUE ' '.
88 NR-USAGE-DISPLAY VALUE 'D'.
88 NR-USAGE-DISPLAY-1 VALUE 'D1'.
88 NR-USAGE-INDEX VALUE 'IX'.
10 NR-REDEFINES-CMPLR PIC 9(05).
10 NR-TYPE-FLAG PIC X(01).
88 NR-TYPE-NORMAL VALUE ' '.
88 NR-TYPE-FILE-NAME VALUE 'F'.
88 NR-TYPE-RECORD-CONTAINS VALUE 'R'.
88 NR-TYPE-BLOCK-CONTAINS VALUE 'B'.
88 NR-TYPE-LABEL-RECORDS VALUE 'L'.
88 NR-TYPE-RECORDS-MODE VALUE 'M'.
88 NR-TYPE-SPECIAL-NAMES VALUE 'S'.
88 NR-TYPE-88-THROUGH VALUE 'T'.
10 NR-CLASS PIC X(01).
88 NR-CLASS-ALPHA VALUE 'X'.
88 NR-CLASS-EDITED VALUE 'E'.
88 NR-TYPE-NUMERIC VALUE 'N'.
10 NR-WHOLE-DIGITS PIC 9(02).
10 NR-FRACTION-DIGITS PIC 9(02).
10 NR-OCCURS-NBR-PRESENT-SW PIC X(01).
88 NR-OCCURS-NBR-PRESENT VALUE 'Y'.
10 NR-SELECT-BY-CTL-CDS-SW PIC X(01).
88 NR-SELECT-BY-CTL-CDS VALUE 'Y'.
10 NR-VALUE-ALL-SW PIC X(01).
88 NR-VALUE-ALL VALUE 'Y'.

```

## Exhibit 2

### COPY Member for the Data Analysis Information File

### **Description of Individual Fields**

The fields in this record are used to print the first line for each Data Division field in the System Data Name Cross -Reference. See the report for the example of these fields.

Most of the fields here are described under the heading, “Data Division Information File” NR-FROM-NBR and NR-TO-NBR correspond to CIE-START-ADDRESS and CIE-1-END-ADDRESS respectively. Refer to that heading for a description of most fields.

The following fields are new to this record:

The field NR-PICTURE-CLAUSE contains the first 21 characters [e.g. X(15), 9(02), ZZ,ZZ9] coded within the PICTURE clause.

The field NR-VALUE-LITERAL contains the first 30 characters of literals coded within the VALUE clause. This field also contains numbers indicating the number of OCCURS for fields with no VALUE that are within an OCCURS clause.

The field NR-SELECT-BY-CTL-CDS indicates the use of the DAS option which limits the DATA Analysis Reports to just those records selected by the SRA control cards.

The field NR-VALUE-ALL indicates the presence of the word ALL prior to the VALUE literal.

*This page intentionally left blank*

### **Procedure Division Information File**

This file has one or more records for each Procedure Division statement and contains information relevant to those statements.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility section of this manual (e.g. PROC DCDACL).
2. Use SOURCE=NO as a PROC Symbolic on the EXEC DCDACL (or other PROC name) line to turn off the SOURCE PARM option.
3. All other report options must be on. This is normally the default.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK05.
2. The LRECL of the file is 120.

The following pages show the internally used Data Division COPY member that was used to create the file and a description of the related fields within the record.

```

** COPY INFOTYPE6 (IN6)
01 CIE-6-PROCEDURE-DIV-INFO.
 05 CIE-6-TYPE-RECORD PIC 9.
 88 CIE-6-TYPE-NORMAL VALUE 0.
 88 CIE-6-TYPE-QUAL-1 VALUE 1.
 88 CIE-6-TYPE-QUAL-2 VALUE 2.
 88 CIE-6-TYPE-ASSOC-NAME-QUALS VALUE 7.
 88 CIE-6-TYPE-SUBSCRIPT-QUALS VALUE 8.
 88 CIE-6-TYPE-SUBSCRIPT-INDEX VALUE 9.

 05 CIE-6-QUAL-NUMBER PIC 99.
 88 CIE-6-VALID-VALUES VALUES ARE 00
 THRU 50.

 05 CIE-6-PROC-SOURCE-LINE-NUMBER PIC 9(5).
 05 CIE-6-PROC-EXP-LINE-NUMBER PIC 9(5).

 05 CIE-6-DATA-1-NAME PIC X(30).
 05 CIE-6-DATA-2-NAME PIC X(30).

 05 CIE-6-NON-CICS-FORMAT-1.
 10 CIE-6-REFERS-DATA-NAME-1-OR-2 PIC X.
 10 CIE-6-PROC-LTRL-LINE-NUMBER PIC 9(5).
 10 CIE-6-PROC-LTRL-COLUMN-NUMBER PIC 9(5).
 05 CIE-6-CICS-FORMAT-1 REDEFINES CIE-6-NON-CICS-FORMAT-1.
 10 CIE-6-IND-CICS-OR-DL1 PIC X.
 88 CIE-6-IND-FOR-CICS VALUE 'C'.
 88 CIE-6-IND-FOR-DL1 VALUE 'D'.
 88 CIE-6-IND-FOR-SQL VALUE 'S'.
 10 CIE-6-CICS-VERB PIC X(10).

 05 CIE-6-VERB-CODE PIC 999.
 05 CIE-6-TYPE-ACTIVITY PIC X.
 88 CIE-6-MODIFIED VALUE 'M'.
 88 CIE-6-USED VALUE 'U'.
 88 CIE-6-COMPARED VALUE 'C'.
 88 CIE-6-SUBSCRIPT VALUE 'S'.
 88 CIE-6-INDEX VALUE 'I'.

 05 CIE-6-COMPARE-OPERATOR PIC X.
 88 CIE-6-OPER-EQUAL TO VALUE '='.
 88 CIE-6-OPER-GREATER-THAN VALUE '>'.
 88 CIE-6-OPER-LESS-THAN VALUE '<'.
 88 CIE-6-OPER-IS-NUMERIC VALUE 'N'.
 88 CIE-6-OPER-IS-ZERO VALUE 'Z'.
 88 CIE-6-OPER-IS-POSITIVE VALUE 'P'.
 88 CIE-6-OPER-IS-NEGATIVE VALUE 'G'.
 88 CIE-6-OPER-IS-KANJI VALUE 'K'.
 88 CIE-6-OPER-IS-ALPHABETIC VALUE 'A'.
 88 CIE-OPER-88-OR-OTHER VALUE SPACE.

 05 CIE-6-ALL-INDICATOR PIC X.
 88 CIE-6-ALL-INDICATED VALUE 'A'.

 05 CIE-6-CORR-INDICATOR PIC X.
 88 CIE-6-CORR-INDICATED VALUE 'C'.

 05 CIE-6-DATA-1-ACTUAL-EXP-NBR PIC 9(5).
 05 CIE-6-DATA-1-ACTUAL-COL-NBR PIC 99.
 05 CIE-6-DATA-2-ACTUAL-EXP-NBR PIC 9(5).
 05 CIE-6-DATA-2-ACTUAL-COL-NBR PIC 99.

```

**Exhibit 3**  
**COPY Member for the Procedure Division Information File (continued on next page)**

|    |                                |           |                    |
|----|--------------------------------|-----------|--------------------|
| 05 | CIE-6-NON-CICS-FORMAT-2.       |           |                    |
| 10 | CIE-6-NOT-INDICATOR            | PIC X.    |                    |
| 88 | CIE-6-NOT-BEFORE-A             |           | VALUE 'A'.         |
| 88 | CIE-6-NOT-BEFORE-B             |           | VALUE 'B'.         |
| 10 | CIE-6-EXHIBIT-FORM             | PIC X.    |                    |
| 88 | CIE-6-EXHIBIT-NAMED            |           | VALUE 'N'.         |
| 88 | CIE-6-EXHIBIT-CHANGED          |           | VALUE 'C'.         |
| 88 | CIE-6-EXHIBIT-CHANGED-NAME     |           | VALUE 'B'.         |
| 10 | CIE-6-EXAMINE-FORM             | PIC X.    |                    |
| 88 | CIE-6-EXAMINE-TALLYING         |           | VALUE 'T'.         |
| 88 | CIE-6-EXAMINE-REPLACING        |           | VALUE 'R'.         |
| 88 | CIE-6-EXAMINE-BOTH             |           | VALUE 'B'.         |
| 10 | CIE-6-GIVING-REMAINDER-SW      | PIC X.    |                    |
| 88 | CIE-6-GIVING                   |           | VALUE 'G'.         |
| 88 | CIE-6-REMAINDER                |           | VALUE 'R'.         |
| 10 | CIE-6-INTO-BY-SW               | PIC X.    |                    |
| 88 | CIE-6-INTO                     |           | VALUE 'I'.         |
| 88 | CIE-6-BY                       |           | VALUE 'B'.         |
| 88 | CIE-6-WITH-POINTER             |           | VALUE 'P'.         |
| 10 | CIE-6-OPEN-TYPE                | PIC X.    |                    |
| 88 | CIE-6-OPEN-INPUT               |           | VALUE 'I'.         |
| 88 | CIE-6-OPEN-I-O                 |           | VALUE 'B'.         |
| 88 | CIE-6-OPEN-OUTPUT              |           | VALUE 'O'.         |
| 88 | CIE-6-OPEN-EXTEND              |           | VALUE 'E'.         |
| 10 | CIE-6-SET-TYPE                 | PIC X.    |                    |
| 88 | CIE-6-SET-TO                   |           | VALUE 'T'.         |
| 88 | CIE-6-SET-UP                   |           | VALUE 'U'.         |
| 88 | CIE-6-SET-DOWN                 |           | VALUE 'D'.         |
| 10 | CIE-6-INSPECT-FORM             | PIC X.    |                    |
| 88 | CIE-6-INSPECT-TALLYING         |           | VALUE 'T'.         |
| 88 | CIE-6-INSPECT-REPLACING        |           | VALUE 'R'.         |
| 88 | CIE-6-INSPECT-BOTH             |           | VALUE 'B'.         |
| 88 | CIE-6-INSPECT-CONVERTING       |           | VALUE 'C'.         |
| 10 | CIE-6-INSPECT-INITIAL-SW       | PIC X.    |                    |
| 88 | CIE-6-INSPECT-BEFORE           |           | VALUE 'B'.         |
| 88 | CIE-6-WRITE-BEFORE             |           | VALUE 'B'.         |
| 88 | CIE-6-INSPECT-AFTER            |           | VALUE 'A'.         |
| 88 | CIE-6-WRITE-AFTER              |           | VALUE 'A'.         |
| 10 | CIE-6-SORT-MERGE-SW            | PIC X.    |                    |
| 88 | CIE-6-SORT-MERGE-VALID-KEYS    |           | VALUE 'A'.         |
|    |                                |           | 'D'.               |
| 88 | CIE-6-SORT-MERGE-ASCENDING     |           | VALUE 'A'.         |
| 88 | CIE-6-SORT-MERGE-DESCENDING    |           | VALUE 'D'.         |
| 88 | CIE-6-SORT-MERGE-USING         |           | VALUE 'U'.         |
| 88 | CIE-6-SORT-MERGE-GIVING        |           | VALUE 'G'.         |
| 10 | CIE-6-SEND-WITH-SW             | PIC X.    |                    |
| 88 | CIE-6-SEND-WITH-ESI            |           | VALUE 'S'.         |
| 88 | CIE-6-SEND-WITH-EMI            |           | VALUE 'M'.         |
| 88 | CIE-6-SEND-WITH-EGI            |           | VALUE 'G'.         |
| 88 | CIE-6-SEND-WITH-IDENT          |           | VALUE 'I'.         |
| 10 | CIE-6-INITIALIZE-SW            | REDEFINES | CIE-6-SEND-WITH-SW |
|    |                                | PIC X(1). |                    |
| 88 | CIE-6-INIT-NO-OPERAND-2        |           | VALUE '0'.         |
| 88 | CIE-6-INIT-REPL-ALPHABETIC     |           | VALUE '1'.         |
| 88 | CIE-6-INIT-REPL-ALPHANUMERIC   |           | VALUE '2'.         |
| 88 | CIE-6-INIT-REPL-NUMERIC        |           | VALUE '3'.         |
| 88 | CIE-6-INIT-REPL-ALPHANU-EDITED |           | VALUE '4'.         |
| 88 | CIE-6-INIT-REPL-NUMERIC-EDITED |           | VALUE '5'.         |
| 88 | CIE-6-INIT-REPL-EGCS           |           | VALUE '6'.         |

**Exhibit 3****COPY Member for the Procedure Division Information File  
(Continued)**

|    |                          |                           |                          |
|----|--------------------------|---------------------------|--------------------------|
| 10 | CIE-6-UNSTRING-SW        |                           | PIC X.                   |
|    | 88                       | CIE-6-UNSTR-DELIMITED-BY  | VALUE 'D'.               |
|    | 88                       | CIE-6-UNSTR-INTO          | VALUE 'I'.               |
|    | 88                       | CIE-6-UNSTR-DELIMITER-IN  | VALUE 'L'.               |
|    | 88                       | CIE-6-UNSTR-COUNT-IN      | VALUE 'C'.               |
|    | 88                       | CIE-6-UNSTR-WITH-POINTER  | VALUE 'P'.               |
|    | 88                       | CIE-6-UNSTR.-TALLYING     | VALUE 'T'.               |
| 10 | CIE-6-CALL-ENTRY-UTIL-SW |                           | PIC X(1).                |
|    | 88                       | CIE-6-9-P-D-USING-ENTRY   | VALUE 'P'.               |
| 10 | CIE-6-VARYING-FROM-BY-SW |                           | PIC X(1).                |
|    | 88                       | CIE-6-P-D-VARYING         | VALUE 'V'.               |
|    | 88                       | CIE-6-P-D-VARYING-FROM    | VALUE 'F'.               |
|    | 88                       | CIE-6-P-D-VARYING-FROM-BY | VALUE 'B'.               |
| 10 | FILLER                   |                           | PIC X(1).                |
| 05 | CIE-CICS-FORMAT-2        | REDEFINES                 | CIE-6-NON-CICS-FORMAT-2. |
| 10 | CIE-6-CICS-OPERAND       |                           | PIC X(15).               |

### Exhibit 3

#### **COPY Member for the Procedure Division Information File** (Continued from previous page)



**Table of Verb Codes**

|     |   |                  |     |   |                          |
|-----|---|------------------|-----|---|--------------------------|
| 001 | = | MOVE             | 034 | = | UNSTRING                 |
| 002 | = | IF               | 035 | = | PROGRAM-ENTRY            |
| 003 | = | READ             | 036 | = | CALL                     |
| 004 | = | WRITE            | 037 | = | RESERVED-GO-TO-DEPENDING |
| 005 | = | ACCEPT           | 038 | = | VARYING                  |
| 006 | = | ADD              | 039 | = | AFTER                    |
| 007 | = | SUBTRACT         | 040 | = | INITIALIZE               |
| 008 | = | MULTIPLY         | 041 | = | TIMES                    |
| 009 | = | DIVIDE           | 042 | = | ON                       |
| 010 | = | COMPUTE          | 043 | = | EVALUATE                 |
| 011 | = | CLOSE            | 044 | = | EVALUATE-WHEN            |
| 012 | = | DELETE           | 046 | = | EXEC                     |
| 013 | = | DISPLAY          | 047 | = | COPY                     |
| 014 | = | EXAMINE          | 050 | = | PARAGRAPH-SECTION-NAME   |
| 015 | = | EXHIBIT          | 051 | = | ALTER                    |
| 016 | = | REWRITE          | 052 | = | CANCEL                   |
| 017 | = | RELEASE          | 053 | = | EXIT                     |
| 018 | = | RETURN           | 054 | = | GO                       |
| 019 | = | WHEN             | 055 | = | GOBACK                   |
| 020 | = | UNTIL            | 056 | = | PERFORM                  |
| 021 | = | HANDLE-SUBSCRIPT | 057 | = | STOP                     |
| 022 | = | OPEN             | 060 | = | INITIATE                 |
| 023 | = | SET              | 061 | = | GENERATE                 |
| 024 | = | INSPECT          | 062 | = | TERMINATE                |
| 025 | = | MERGE            | 063 | = | CONTROL-IN-RWRD          |
| 026 | = | RECEIVE          | 064 | = | RESET-IN-RW0149          |
| 027 | = | SEARCH           | 065 | = | SOURCE-IN-RW0149         |
| 028 | = | SEND             | 066 | = | SUM-IN-RW0149            |
| 029 | = | SORT             | 067 | = | SUM-UPON-IN-RW0149       |
| 030 | = | START            | 068 | = | TYPE-CH-IN-RW0149        |
| 031 | = | SEEK             | 069 | = | TYPE-CF-IN-RW0149        |
| 032 | = | STRING           |     |   |                          |
| 033 | = | TRANSFORM        |     |   |                          |

NOTE: Numbers are not completely contiguous. See CIE-6-VERB-CODE for a description of the field.

**Description of Individual Fields**

- |     |                          |                                                                                                                                                                                                                                                                                                                    |
|-----|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | CIE-6-TYPE-RECORD        | An indicator showing the type of data name this record covers (regular qualification of a data name, qualification for a subscript or a subscript on an index). Ignore type 7 records.                                                                                                                             |
| 2.  | CIE-6-QUAL-NUMBER        | For qualification records, this number shows the level of qualification (first, second, etc.).                                                                                                                                                                                                                     |
| 3.  | CIE-6-SOURCE-LINE-NUMBER | Do not use.                                                                                                                                                                                                                                                                                                        |
| 4.  | CIE-6-EXP-LINE-NUMBER    | The expanded sequence number (after COPY members has been resolved and brought into the program) where this data field is in the program.                                                                                                                                                                          |
| 5.  | CIE-6-DATA-1-NAME        | The syntax of every Procedure Division statement is broken into one or more groupings where one or two data names are directly involved. This field represents the first data name or literal. (For example: MOVE A TO B.)<br>Note: Literals are truncated to 28 characters.                                       |
| 6.  | CIE-6-DATA-2-NAME        | This field will be blank or will contain the second data name.                                                                                                                                                                                                                                                     |
| 7.  | CIE-6-NON-CICS-FORMAT-1  | Do not use.                                                                                                                                                                                                                                                                                                        |
| 8.  | CIE-6-CICS-FORMAT-1      | This field has information from CICS, DL1 or SQL statements.                                                                                                                                                                                                                                                       |
| 9.  | CIE-6-VERB-CODE          | A three-digit code indicating the COBOL verb used in the Procedure Division that caused this record to be generated. See the Table of Verb Codes.                                                                                                                                                                  |
| 10. | CIE-6-TYPE-ACTIVITY      | This code (while using different letters) is similar to the Set/Used/Tested indicators used in the System Record Analysis Report. “M” indicates the field was modified, “U” indicates the field is used to modify another field, “C” indicates a comparison, “S” indicates a subscript and “I” indicates an INDEX. |
| 11. | CIE-6-COMPARE-OPERATOR   | This code is filled in for many comparisons to indicate the type of comparison.                                                                                                                                                                                                                                    |
| 12. | CIE-6-ALL-INDICATOR      | Do not use.                                                                                                                                                                                                                                                                                                        |
| 13. | CIE-6-CORR-INDICATOR     | Used to indicate the presence of CORRESPONDING.                                                                                                                                                                                                                                                                    |

- |            |                                                                       |                                                                                                                |
|------------|-----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 14.<br>and | CIE-6-DATA-1-ACTUAL-EXP-NBR<br>through<br>CIE-6-DATA-2-ACTUAL-COL-NBR | These represent the actual line number<br>column number where data names 1 and 2<br>reside in the source code. |
| 15.        | CIE-6-NON-CICS-FORMAT-2                                               | Individual flags which indicate how the<br>verb was coded (i. e. what format of the<br>verb was used).         |
| 16.        | CIE-6-CICS-FORMAT-2                                                   | Contains the first CICS operand.                                                                               |

*This page intentionally left blank*

### **Basic Paragraph Information File**

This file has basic information on the location of paragraphs and sections and the referencing of those paragraphs through Procedure Division statements.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility section of this manual (e.g. PROC DCDACL).
2. Use the following PROC Symbolics on the EXEC DCDACL (or other PROC name) line to turn off all report PARM options as follows:

```
SOURCE=NO,
CALL=NO,
COPY=NO,
FIGCON=NO,
LITERAL=NO,
SPREGS=NO,
CDATADV=NO,
CPROC DV=NO,
OTHER='NOVER,NOIRE'
```

To access the file:

1. The file is available on the file with the DDNAME of DCDWK06.
2. The LRECL of the file is 104.

The following pages show the internally used Data Division COPY member that was used to create the file and a description of the related fields within the record.

```

** COPY PARATYP1 (PF1)
01 PAR-1-BASIC-PARAGRAPH-INFO.

 05 PAR-1-SOURCE-LINE-NUMBER PIC 9(5).
 05 PAR-1-EXPANDED-LINE-NUMBER PIC 9(5).
 05 PAR-1-TOKEN-COLUMN-NUMBER PIC 9(2).

 05 PAR-1-PARA-OR-SECT PIC X(30).

 05 PAR-1-RELEVANT-SECTION PIC X(30).

 05 PAR-1-VERB-CODE PIC 999.
 88 PAR-1-PARAGRAPH-NAME VALUE 001.
 88 PAR-1-SECTION-NAME VALUE 002.
 88 PAR-1-PRIOR-TOKEN-OF-PARA-SECT VALUE 005.
 88 PAR-1-LAST-TOKEN-IN-PROGRAM VALUE 006.
 88 PAR-1-DECLARATIVES VALUE 008.
 88 PAR-1-END-DECLARATIVES VALUE 009.
 88 PAR-1-GO-TO VALUE 011.
 88 PAR-1-GO-TO-DEPENDING-ON VALUE 012.
 88 PAR-1-PERFORM VALUE 021.
 88 PAR-1-PERFORM-THRU VALUE 022.
 88 PAR-1-INPUT-PROCEDURE VALUE 025.
 88 PAR-1-INPUT-PROCEDURE-THRU VALUE 026.
 88 PAR-1-OUTPUT-PROCEDURE VALUE 027.
 88 PAR-1-OUTPUT-PROCEDURE-THRU VALUE 028.
 88 PAR-1-ALTER-PARA VALUE 031.
 88 PAR-1-ALTER-PARA-TO-PROCEED-TO VALUE 032.
 88 PAR-1-GOBACK VALUE 041.
 88 PAR-1-STOP-RUN VALUE 042.
 88 PAR-1-EXIT-PROGRAM VALUE 043.
 88 PAR-1-EXIT VALUE 048.
 88 PAR-1-CALL VALUE 051.
 88 PAR-1-PROGRAM-ENTRY VALUE 052.
 88 PAR-1-ENTRY VALUE 053.
 88 PAR-1-CANCEL VALUE 054.
 88 PAR-1-INITIATE VALUE 060.
 88 PAR-1-GENERATE VALUE 061.
 88 PAR-1-TERMINATE VALUE 062.
 88 PAR-1-EXEC-CICS VALUE 071.
 88 PAR-1-EXEC-DLI VALUE 072.
 88 PAR-1-EXEC-SQL VALUE 073.

 05 PAR-1-CONDITIONAL-SW PIC X.
 88 PAR-1-CONDITIONAL VALUE 'Y'.

 05 PAR-1-MULTIPLE-OPERANDS-SW PIC X.
 88 PAR-1-MULTIPLE-OPERANDS VALUE 'Y'.

 05 PAR-1-NAME-EXP-LINE-NUMBER PIC 9(5).
 05 PAR-1-NAME-COLUMN-NUMBER PIC 9(2).
 05 PAR-1-FILLER PIC X(20).

```

#### Exhibit 4

#### COPY Member for the Basic Paragraph Information File

### Description of Individual Fields

- |     |                            |                                                                                                                                                                                                                                                                       |
|-----|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | PAR-1-SOURCE-LINE-NUMBER   | Do not use.                                                                                                                                                                                                                                                           |
| 2.  | PAR-1-EXPANDED-LINE-NUMBER | An expanded sequence number (after COPY members have been resolved and brought into the program) of a statement number in the Procedure Division representing a paragraph or section, or a statement with a reference to a paragraph or section or external transfer. |
| 3.  | PAR-1-TOKEN-COLUMN-NUMBER  | The column number of the beginning of the paragraph or verb which references the paragraph.                                                                                                                                                                           |
| 4.  | PAR-1-PARA-OR-SECT         | The name of a paragraph, a section or a comment generated to correspond to the code used in PAR-1-VERB-CODE.                                                                                                                                                          |
| 5.  | PAR-1-RELEVANT-SECTION     | The name of the parent section (if one exists) for paragraph records.                                                                                                                                                                                                 |
| 6.  | PAR-1-VERB-CODE            | A three-digit code explaining what verb or condition generated this record.                                                                                                                                                                                           |
| 7.  | PAR-1-CONDITIONAL-SW       | Used to indicate that the verb causing this record is part of a conditional statement such as READ/ AT END OR IF.                                                                                                                                                     |
| 8.  | PAR-1-MULTIPLE-OPERANDS-SW | Used to indicate that one of multiple operands is listed and others are not in the paragraph field. An example of this is CALL USING.                                                                                                                                 |
| 9.  | PAR-1-NAME-EXP-LINE-NUMBER | An expanded sequence number of the actual paragraph name for verb codes where a paragraph name is used as an operand. Otherwise, this field will be zero.                                                                                                             |
| 10. | PAR-1-NAME-COLUMN-NUMBER   | A column number which corresponds to the above sequence number.                                                                                                                                                                                                       |

*This page intentionally left blank*



### **Paragraph Range Information File**

This file has most of the information in the Basic Paragraph Information File and the beginning and ending sequence numbers showing the range of those paragraphs or sections.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility Section of this manual (e.g. PROC DCDACL).
2. Use the PROC Symbolic SOURCE= on the EXEC DCDACL (or other PROC name) line to turn on the SOURCE PARM option. This option is normally already on. Other options may also be left on.

To access the File:

1. The file is available on the file with the DDNAME of DCDWK03.
2. The LRECL of the file is 104.

The following pages show the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

```

**
01 PARA-2-RANGES-FOR-PARAS. COPY PARATYP2 (PF2)

05 PAR-2-SOURCE-LINE-NUMBER PIC 9(5).
05 PAR-2-EXPANDED-LINE-NUMBER PIC 9(2).
05 PAR-2-TOKEN-COLUMN-NUMBER PIC 9(2).

05 PAR-2-PARA-OR-SECT PIC X(30).

05 PAR-2-RELEVANT-SECTION PIC X(30).

05 PAR-2-VERB-CODE PIC 999.

 88 PAR-2-PARAGRAPH-NAME VALUE 001.
 88 PAR-2-SECTION-NAME VALUE 002.
 88 PAR-2-PRIOR-TOKEN-OF-PARA-SECT VALUE 005.
 88 PAR-2-LAST-TOKEN-IN-PROGRAM VALUE 006.
 88 PAR-2-DECLARATIVES VALUE 008.
 88 PAR-2-END-DECLARATIVES VALUE 009.
 88 PAR-2-GO-TO VALUE 011.
 88 PAR-2-GO-TO-DEPENDS-ON VALUE 012.
 88 PAR-2-PERFORM VALUE 021.
 88 PAR-2-PERFORM-THRU VALUE 022.
 88 PAR-2-INPUT-PROCEDURE VALUE 025.
 88 PAR-2-INPUT-PROCEDURE-THRU VALUE 026.
 88 PAR-2-OUTPUT-PROCEDURE VALUE 027.
 88 PAR-2-OUTPUT-PROCEDURE-THRU VALUE 028.
 88 PAR-2-ALTER-PARA VALUE 031.
 88 PAR-2-ALTER-TO-PROCEED-TO VALUE 032.
 88 PAR-2-GOBACK VALUE 041.
 88 PAR-2-STOP-RUN VALUE 042.
 88 PAR-2-EXIT-PROGRAM VALUE 043.
 88 PAR-2-EXIT VALUE 048.
 88 PAR-2-CALL VALUE 051.
 88 PAR-2-PROGRAM-ENTRY VALUE 052.
 88 PAR-2-ENTRY VALUE 053.
 88 PAR-2-CANCEL VALUE 054.
 88 PAR-2-INITIATE VALUE 060.
 88 PAR-2-GENERATE VALUE 061.
 88 PAR-2-TERMINATE VALUE 062.
 88 PAR-2-EXEC-CICS VALUE 071.
 88 PAR-2-EXEC-DLI VALUE 072.
 88 PAR-2-EXEC-SQL VALUE 073.

05 PAR-CONDITIONAL-SW PIC X.
 88 PAR-2-CONDITIONAL VALUE 'Y'.

05 PAR-2-MULTIPLE-OPERANDS-SW PIC X.
 88 PAR-MULTIPLE-OPERANDS VALUE 'Y'.

05 PAR-2-NAME-EXP-LINE-NUMBER PIC 9(5).
05 PAR-2-NAME-COLUMN-NUMBER PIC 9(2).

05 PAR-2-SOURCE-BEGIN-NBR PIC 9(5).
05 PAR-2-EXPANDED-BEGIN-NBR PIC 9(5).

05 PAR-2-SOURCE-END-NBR PIC 9(5).
05 PAR-2-EXPANDED-END-NBR PIC 9(5).

```

## Exhibit 5

### COPY Member for the Paragraph Range Information File

## Description of Individual Fields

|     |                            |                                                                                                                                                                                                                                                                      |
|-----|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | PAR-2-SOURCE-LINE-NUMBER   | Do not use.                                                                                                                                                                                                                                                          |
| 2.  | PAR-2-EXPANDED-LINE-NUMBER | An expanded sequence number (after COPY members have been resolved and brought into the program) of a statement number in the Procedure Division representing a paragraph or section or a statement with a reference to a paragraph or section or external transfer. |
| 3.  | PAR-2-TOKEN-COLUMN-NUMBER  | The column number of the beginning of the paragraph or verb which references the paragraph.                                                                                                                                                                          |
| 4.  | PAR-2-PARA-OR-SECT         | The corresponding paragraph or section name.                                                                                                                                                                                                                         |
| 5.  | PAR-2-RELEVANT-SECTION     | When PAR-2-PARA-OR-SECT contains a paragraph name, this field will contain the section name that the paragraph name resides in.                                                                                                                                      |
| 6.  | PAR-2-VERB-CODE            | A three-digit code explaining what verb or condition generated this record.                                                                                                                                                                                          |
| 7.  | PAR-2-CONDITIONAL-SW       | Used to indicate that the verb causing this record is part of a conditional statement such as READ / AT END or IF.                                                                                                                                                   |
| 8.  | PAR-2-MULTIPLE-OPERANDS-SW | Used to indicate that one of multiple operands is listed and others are not in the paragraph field. An example of this is CALL USING.                                                                                                                                |
| 9.  | PAR-2-NAME-EXP-LINE-NUMBER | An expanded sequence of the actual paragraph name for verb codes where a paragraph name is used as an operand. Otherwise this field will be zero.                                                                                                                    |
| 10. | PAR-2-NAME-COLUMN-NUMBER   | A column number which corresponds to the above sequence number.                                                                                                                                                                                                      |
| 11. | PAR-2-SOURCE-BEGIN-NBR     | Do not use.                                                                                                                                                                                                                                                          |
| 12. | PAR-2-EXPANDED-BEGIN-NBR   | For verbs which make reference to a paragraph or section this field shows the beginning expanded sequence number of the paragraph or section being referenced.                                                                                                       |
| 13. | PAR-2-SOURCE-END-NBR       | Do not use.                                                                                                                                                                                                                                                          |
| 14. | PAR-2-EXPANDED-END-NBR     | For verbs that require both begin and end sequence numbers, (e.g. PERFORM), this field shows the ending expanded sequence number.                                                                                                                                    |

### **Transfer of Control Narrative File**

This file has a narrative showing transfer of control information for paragraphs, sections and transfer of control statements such as CALL, PERFORM and GO TO.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility section of this manual (e.g. PROC DCDACL).
2. Use the PROC Symbolic SOURCE= on the EXEC DCDACL (or other PROC name) line to turn on the SOURCE PARM option. This is normally the default. Other options may be left on.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK04.
2. The LRECL of the file is 40.

The following page shows the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

\*\* COPY PARATYPE3. (PF3)

01 PAR-3-NARRATIVE-FOR-SRC-LIST

05 PAR-3-EXPANDED-LINE-NUMBER PIC 9(5).  
05 PAR-3-NARRATIVE PIC X(34).  
05 PAR-3-FILLER PIC X(1).

### **Exhibit 6**

#### **COPY Member for the Transfer of Control Narrative File**

##### **Description of Individual Fields**

- |                               |                                                                                                                                       |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 1. PAR-3-EXPANDED-LINE-NUMBER | An expanded sequence number of the Procedure Division statement for which the narrative is for.                                       |
| 2. PAR-3-NARRATIVE            | Transfer of control narrative belonging to the COBOL verb in the Procedure Division statement referenced by the expanded line number. |

*This page intentionally left blank*

### **Procedure Division References to the Data Division**

This file is a narrative file showing the references for each Procedure Division line back to the Data Division.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility Section of this manual (e.g. PROC DCDACL).
2. Use the PROC Symbolic SOURCE= on the EXEC DCDACL (or other PROCname) line to turn on the SOURCE PARM option. This is normally the default. Other options may be left on.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK02.
2. The LRECL of the file is 52.

The following pages show the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

```

01 RR-WORK-RECORD.
05 RR-P-D-CPMLR-NBR PIC 9(5).
05 RR-NARR-LENGTH PIC 99.

05 RR-SQ-NBR-WORK-AREA.
10 RR-DATA-DIV-SQ-NBRS-6 OCCURS 7 TIMES.
15 RR-DATA-DIV-CMPLR-6 PIC ZZZZ99.
10 FILLER PIC XX.

05 RR-WORK-AREA-2 REDEFINES RR-SQ-NBR-WORK-AREA.
10 RR-DATA-DIV-SQ-NBRS-5 OCCURS 8 TIMES
15 RR-DATA-DIV-CMPLR-5 PIC ZZZ99.
10 FILLER PIC X(4).
05 RR-WORK-AREA-3 REDEFINES RR-SQ-NBR-WORK-AREA.
10 RR-DATA-DIV-SQ-NBRS-4 OCCURS 10 TIMES.
15 RR-DATA-DIV-CMPLR-4 PIC ZZ99.
10 FILLER PIC X(4).

05 RR-WORK-AREA-4 REDEFINES RR-SQ-NBR-WORK-AREA.
10 RR-DATA-DIV-SQ-NBRS-3 OCCURS 14 TIMES.
15 RR-DATA-DIV-CMPLR-3 PIC Z99.
10 FILLER PIC X(2).

05 RR-WORK-AREA-5 REDEFINES RR-SQ-NBR-WORK-AREA.
10 RR-POS OCCURS 44 TIMES.
 PIC X.
05 RR-WORK-AREA-6 REDEFINES RR-SQ-NBR-WORK-AREA.
10 FILLER PIC X(29).
10 RR-CA3-FINAL-5-POS PIC ZZZZ99.
10 RR-CA3-LAST-CHAR PIC X(01).
10 FILLER PIC X(01).
10 RR-FINAL-5-POS PIC ZZZZ99.
10 RR-LAST-CHAR PIC X(01).

05 RR-WORK-AREA-7 REDEFINES RR-SQ-NBR-WORK-AREA.
10 FILLER PIC X(30).
10 RR-CA3-FINAL-4-POS PIC ZZZ99.
10 FILLER PIC X(03).
10 RR-FINAL-4-POS PIC ZZZ99.
10 FILLER PIC X(01).

05 RR-WORK-AREA-8 REDEFINES RR-SQ-NBR-WORK-AREA.
10 FILLER PIC X(31).
10 RR-CA3-FINAL-3-POS PIC ZZ99.
10 FILLER PIC X(04).
10 RR-FINAL-3-POS PIC ZZ99.
10 FILLER PIC X(01).

05 RR-LAST-FILLER PIC X.

```

## Exhibit 7

### COPY Member for Procedure Division References to the Data Division



### **Description of Individual Fields**

1. RR-P-D-CMPLR-NBR  
An expanded sequence number (after COPY members have been resolved and brought into the program) of a statement in the Procedure Division which has references to the Data Division.
2. RR-NARR-LENGTH  
The length in characters of the narrative that follows in the next field.
3. RR-SQ-NBR-WORK-AREA  
An area where numbers representing references back to the Data Division are placed. The numbers will have only as many digits reserved for the number as the largest sequence number is present in the Data Division with a minimum of 2 digits.  
One Space is left between all numbers. In cases where smaller numbers are present, more than one space will be used. The best example of this field is by looking at this file after running a user's COBOL program through DCD III.

### **Procedure Division Narrative for Each Data Field**

This file has a narrative for each Data Division field showing all Procedure Division references to this field. The length of the narrative is 43 characters.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility section of this manual (e.g. PROC DCDACL).
2. Use the PROC Symbolic SOURCE= on the EXEC DCDACL (or other PROC name) line to turn on the SOURCE PARM option. This option is normally already on. Other options may be left on.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK05.
2. The LRECL of file is 60.

The following page shows the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

|    |              |            |
|----|--------------|------------|
| 01 | NR-RECORD    |            |
| 05 | NR-CMPLR-NBR | PIC 9(5).  |
| 05 | NR-NARRATIVE | PIC X(55). |

### **Exhibit 8**

#### **COPY Member for Procedure Division Narrative for Each Data Field**

##### **Description of Individual Fields**

- |    |              |                                                                                            |
|----|--------------|--------------------------------------------------------------------------------------------|
| 1. | NR-CMPLR-NBR | An expanded sequence number of the Data Division statement for which the narrative is for. |
| 2. | NR-NARRATIVE | COBOL narrative belonging to the data name indicated by the sequence number above.         |

### **Expanded Procedure Division Narrative**

This file has a narrative for each Data Division field showing all Procedure Division references to this field. The length of the narrative is 83 characters.

To produce the file:

1. Use one of the PROCs in the Other COBOL Reports Facility section of this manual (e.g. PROC DCDCOBOL).
2. Use the PROC Symbolic DATA= on the EXEC DCDCOBOL (or other PROC name) line to turn on the DATA PARM option.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK05.
2. The LRECL of the file is 130.

The following page shows the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

|    |                 |            |
|----|-----------------|------------|
| 01 | NR-RECORD.      |            |
| 05 | NR-DATA-NAME    | PIC X(30)  |
| 05 | NR-PROGRAM-NAME | PIC X(08). |
| 05 | NR-CMPLR-NBR    | PIC 9(05). |
| 05 | NR-SEQ-NBR      | PIC 9(04). |
| 05 | NR-NARRATIVE    | PIC X(83). |

### **Exhibit 9**

#### **COPY Member for the Expanded Procedure Division Narrative**

##### **Description of Individual Fields**

- |    |                 |                                                                           |
|----|-----------------|---------------------------------------------------------------------------|
| 1. | NR-DATA-NAME    | Data name that the COBOL narrative is for.                                |
| 2. | NR-PROGRAM-NAME | COBOL program-id.                                                         |
| 3. | NR-CMPLR-NBR    | Expanded sequence number.                                                 |
| 4. | NR-SEQ-NBR      | A number from 1 to 9999, re-starting at 1 for every data name.            |
| 5. | NR-NARRATIVE    | The COBOL narrative that appears on the System Data Name Cross-Reference. |

*This page intentionally left blank*

### **Data Dictionary Interface File**

This file produces the same information produced on the System Record Analysis Report. Specifically, direct and indirect flags are available which show data movement and comparison for each field within selected records.

To produce the file:

1. See the heading, “**Data Dictionary Interface File**”, in the Other COBOL Reports Facility section. Control statements are required.

To access the file:

1. The file is available on the file with the DDNAME of DCDDICT.
2. The LRECL of the file is 120.

The following pages show the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

```

** COPY SYS1RECD

01 SRA-SYSTEM-RECORD-ANAL-REC.
 05 SRA-SET-NUMBER PIC999.
 05 SRA-FROM-TO-LOCATIONS.
 10 SRA-FROM-POSITION PIC 9(6).
 10 SRA-TO-POSITION PIC 9(6).
 05 SRA-PROGRAM-ID PIC X(8).
 05 SRA-RECORD-NAME PIC X(30).
 05 SRA-DATA-NAME PIC X(30).
 05 SRA-EXP-CMPLR-NBR PIC 9(5).
 05 SRA-LEVEL-NUMBER PIC XX.
 05 SRA-TYPE-USAGE PIC X(2).
 88 SRA-TYPE-COMP VALUE 'C '.
 88 SRA-TYPE-COMP-1 VALUE 'C1'.
 88 SRA-TYPE-COMP-2 VALUE 'C2'.
 88 SRA-TYPE-COMP-3 VALUE 'C3'.
 88 SRA-TYPE-COMP-4 VALUE 'C4'.
 88 SRA-TYPE-DEFAULT-DISPLAY VALUE ' '.
 88 SRA-TYPE-DISPLAY VALUE 'D '.
 88 SRA-TYPE-DISPLAY-1 VALUE 'D1'.
 88 SRA-TYPE-INDEX VALUE 'IX'.
 88 SRA-TYPE-USAGE-IS-INDEX VALUE 'UT'.
 88 SRA-TYPE-POINTER VALUE 'PR'.
 05 SRA-FLAGS.
 10 SRA-SET-FLAG PIC X.
 88 SRA-DIRECT-SET VALUE 'S'.
 10 SRA-USED-FLAG PIC X.
 88 SRA-DIRECT-USED VALUE 'U'.
 10 SRA-TESTED-FLAG PIC X.
 88 SRA-DIRECT-TESTED VALUE 'T'.
 10 SRA-INDIR-SET-FLAG PIC X.
 88 SRA-INDIRECT-SET VALUE 'S'.
 10 SRA-INDIR-USED-FLAG PIC X.
 88 SRA-INDIRECT-USED VALUE 'U'.
 10 SRA-INDIR-TESTED-FLAG PIC X.
 88 SRA-INDIRECT-TESTED VALUE 'T'.
 05 SRA-PICTURE-CLAUSE PIC X(21).
 05 SRA-FILLER PIC X.

```

## Exhibit 10

### COPY Member for the Data Dictionary Interface File



### Description of Individual Fields

- |                          |                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------|
| 1. SRA-SET-NUMBER        | Sequential number starting at 1 showing which set of control cards produced this set of records. |
| 2. SRA-FROM-TO-LOCATIONS | From and to record positions for this data name.                                                 |
| 3. SRA-DATA-ID           | Taken from the program-id in the Identification Division.                                        |
| 4. SRA-RECORD-NAME       | The name of the 01 record which contains this data name.                                         |
| 5. SRA-DATA-NAME         | Self-explanatory.                                                                                |
| 6. SRA-EXP-CMPLR-NBR     | Compiler number or sequence number in the program of this data name.                             |
| 7. SRA-LEVEL-NUMBER      | 01-49 level number associated with this data name.                                               |
| 8. SRA-TYPE-USAGE        | USAGE of this data name.                                                                         |
| 9. SRA-FLAGS             | Contains flags showing how this data field is used in the program.                               |

Direct flags indicate this field was involved in the Procedure Division statement, which caused this flag.

Indirect flags indicate this field was not involved in the Procedure Division statement, which caused this flag.

**SET** indicates that this is changed.

**USED** indicates that this field was used to modify, in some way, another field.

**TESTED** indicates that this field was used in a comparison.

- |                        |                                                                               |
|------------------------|-------------------------------------------------------------------------------|
| 10. SRA-PICTURE-CLAUSE | The first 21 characters of the PICTURE clause associated with this data name. |
| 11. SRA-FILLER         | Not used at this time.                                                        |

*This page intentionally left blank*

## **PICTURE and Literal File**

This file contains information on PICTURES, VALUE literals and literals used in the Procedure Division.

To produce the file:

1. Use one of the PROCs in the Alternate Compile Listing Facility Section of this manual (e.g. PROC DCDACL).
2. Use SOURCE=NO as a PROC Symbolic on the EXEC DCDACL (or other PROC name) line to turn off the SOU PARM option.
3. All other report options must be on. This is normally the default.

To access the file:

1. The file is available on the file with the DDNAME of DCDWK02.
2. The LRECL of the file is 104.

The following pages show the internally used Data Division COPY member that was used to create the file, and a description of the related fields within the record.

| ** | COPY                         | INFOTYP3   | (IN3)      |
|----|------------------------------|------------|------------|
| 01 | CIE-3-PICTURE-LITERAL-INFO.  |            |            |
| 05 | CIE-3-SOURCE-LINE-NUMBER     | PIC 9(5).  |            |
| 05 | CIE-3-EXPANDED-LINE-NUMBER   | PIC 9(5).  |            |
| 05 | CIE-3-FIELD-COLUMN-NUMBER    | PIC 9(2).  |            |
| 05 | CIE-3-TYPE-CHANGE-IND        | PIC X.     |            |
| 88 | CIE-3-CHANGE-PICTURE         |            | VALUE 'P'. |
| 88 | CIE-3-CHANGE-LITERAL         |            | VALUE 'L'. |
| 88 | CIE-3-CHANGE-RECORD          |            | VALUE 'R'. |
| 88 | CIE-3-CHANGE-BLOCK           |            | VALUE 'B'. |
| 88 | CIE-3-LITERAL-FOR-ELEM       |            | VALUE '5'. |
| 88 | CIE-3-LITERAL-FOR-88         |            | VALUE '8'. |
| 88 | CIE-PICTURE-FOR-ELEM         |            | VALUE 'C'. |
| 88 | CIE-3-COPY-STATEMENT         |            | VALUE 'Q'. |
| 88 | CIE-3-COPY-FOLLOW-REC        |            | VALUE 'V'. |
| 88 | CIE-3-SYS-FIG-CONST          |            | VALUE 'F'. |
| 88 | CIE-3-SYS-SP-REG             |            | VALUE 'S'. |
| 05 | CIE-3-OLD-FIELD-LENGTH       | PIC 9(5).  |            |
| 05 | CIE-3-AMT-OF-FIELD-INCREASE  | PIC 9(2).  |            |
| 05 | CIE-EXPANSION-INFORMATION.   |            |            |
| 10 | CIE-3-EXPANSION-CODE         | PIC 9(3).  |            |
| 88 | CIE-3-EXPANSION-CODE-IGNORED |            | VALUE 000. |
| 10 | CIE-3-EXPANSION-BEGIN-POS    | PIC 99.    |            |
| 10 | CIE-3-EXPANSION-END-POS      | PIC 99.    |            |
| 05 | CIE-3-DIVISION-INDICATOR     | PIC X.     |            |
| 88 | CIE-3-ID-DIVISION            |            | VALUE 'I'. |
| 88 | CIE-3-ENVIRONMENT-DIVISION   |            | VALUE 'E'. |
| 88 | CIE-3-DATA-DIVISION          |            | VALUE 'D'. |
| 88 | CIE-3-PROCEDURE-DIVISION     |            | VALUE 'P'. |
| 05 | CIE-3-DATA-NAME              | PIC X(30). |            |
| 05 | CIE-3-COPY-MEMBER-NAME       | PIC X(8).  |            |
| 05 | CIE-3-PROGRAM-ID-NAME        | PIC X(8).  |            |
| 05 | CIE-3-SOU-DATA-NAME-LINE-NBR | PIC 9(5).  |            |
| 05 | CIE-3-EXP-DATA-NAME-LINE-NBR | PIC 9(5).  |            |
| 05 | CIE-3-LEVEL-NUMBER           | PIC 99.    |            |
| 05 | CIE-3-ALL-LITERAL-SW         | PIC X.     |            |
| 88 | CIE-3-ALL-LITERAL            |            | VALUE 'Y'. |
| 05 | CIE-3-COPY-SUPPRESS-SW       | PIC X.     |            |
| 88 | CIE-3-COPY-SUPPRESS          |            | VALUE 'S'. |
| 05 | CIE-3-COPY-REPLACING-SW      | PIC X.     |            |
| 88 | CIE-3-COPY-REPLACING         |            | VALUE 'R'. |
| 05 | CIE-3-VALUE-THRU-SW          | PIC X.     |            |
| 88 | CIE-3-VALUE-THRU             |            | VALUE 'T'. |
| 05 | CIE-3-FILLER-TO-104          | PIC X(14)  |            |

## Exhibit 11

### COPY Member for the PICTURE and Literal File

**Description of Individual Fields**

|     |                              |                                                                                                                                                                                      |
|-----|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | CIE-3-SOURCE-LINE-NUMBER     | Do not use.                                                                                                                                                                          |
| 2.  | CIE-3-EXPANDED-LINE-NUMBER   | The expanded sequence number (after COPY members have been resolved and brought into the program) of the actual line containing the literal or PICTURE clause.                       |
| 3.  | CIE-3-FIELD-COLUMN-NUMBER    | The column number on this line where the PICTURE or literal begins.                                                                                                                  |
| 4.  | CIE-3-TYPE-CHANGE-IND        | A code indicating the type of information stored on this record. The “P” code will appear in this record as a “C”. The “L” code will appear (for the Data Division) as a “5” or “8”. |
| 5.  | CIE-3-OLD-FIELD-LENGTH       | Do not use.                                                                                                                                                                          |
| 6.  | CIE-3-AMT-OF-FIELD-INCREASE  | Do not use.                                                                                                                                                                          |
| 7.  | CIE-3-EXPANSION-INFORMATION  | Do not use.                                                                                                                                                                          |
| 8.  | CIE-3-DIVISION-INDICATOR     | Indicates what division this record comes from.                                                                                                                                      |
| 9.  | CIE-3-DATA-NAME              | The PICTURE clause or actual literal.                                                                                                                                                |
| 10. | CIE-3-COPY-MEMBER-NAME       | If this field is part of a COPY member, then the COPY member name is shown here.                                                                                                     |
| 11. | CIE-3-PROGRAM-ID-NAME        | The program-id from this COBOL program.                                                                                                                                              |
| 12. | CIE-3-SOU-DATA-NAME-LINE-NBR | Do not use.                                                                                                                                                                          |
| 13. | CIE-3-EXP-DATA-NAME-LINE-NBR | The expanded sequence number of the level number that belongs to and represents this PICTURE or literal.                                                                             |
| 14. | CIE-3-LEVEL-NUMBER           | The level number.                                                                                                                                                                    |
| 15. | CIE-3-ALL-LITERAL-SW         | Do not use.                                                                                                                                                                          |
| 16. | CIE-3-COPY-SUPPRESS-SW       | Used to indicate the use of SUPPRESS on a COPY statement.                                                                                                                            |
| 17. | CIE-3-COPY-REPLACING-SW      | Used to indicate the use of REPLACING on a COPY statement.                                                                                                                           |
| 18. | CIE-3-VALUE-THRU-SW          | Indicates the presence of an ALL before the literal.                                                                                                                                 |

*This page intentionally left blank*

Release 3.6

DCD III  
JCL PROC Analysis Reports  
Facility

*This page intentionally left blank*



DCD III

**JCL PROC Analysis Reports Facility**

**TABLE OF CONTENTS**

|                                                        |      |
|--------------------------------------------------------|------|
| Use of This Section .....                              | D-4  |
| Explanation of JCL PROC Analysis Reports .....         | D-4  |
| User Options Available .....                           | D-5  |
| Use of Control Statements for Selection of PROCs ..... | D-7  |
| Use of Control Statements for Excluding DDNAMEs .....  | D-10 |
| JCL Examples .....                                     | D-11 |
| DCDJCL PROC .....                                      | D-12 |
| Sample Reports .....                                   | D-13 |

### **Use of This Section**

This section is provided to make the user aware of reports which are available for JCL analysis.

### **Explanation of JCL PROC Analysis Reports**

The JCL PROC Analysis Reports Facility produces JCL analysis reports on either JCLPROC libraries or JCL JOB libraries. See, “Sample Reports”, under this heading for a sample of these reports.

The DCDJCL PROC explained in this section contains two steps. The first step isolates the wanted JCL members from a partitioned data set. This first step is the same program that is executed by the MBRFETCH PROC described in another section of this manual. The second step executes the program to print the JCL PROC Analysis Reports for either a PROC library or a JOB library.

A PROC library is defined as a library with existing PROCs or other JCLs which contain EXEC PGM= and DD statements.

A JOB library is defined as a library with JOB cards and EXEC proc-member-name statements.

### User Options Available

There are two basic JCL Analysis Reports:

**The first report is the JOBLIB report** showing only two fields – JOB name and PROC name. The report when run will be listed in two sequences – 1) by JOB name and 2) by PROC name. To run, specify RPTYPE=JOBLIB on the EXEC line. Do not specify other report or J-options when running this report. LNCNT and SORTREG symbolics may be used here.

A sample EXEC line for the JOBLIB report is shown below:

```
//STEP1 EXEC DCDJCL,RPTYPE=JOBLIB,LNCNT=60
```

**The second report is the PROC or basic JCL report** which shows a breakdown of user JCL for DDNAME, DSN & DISPOSITION along with related program name, proc name, step name and step number. There are three sequences to this report and several other related options.

The three sequences are:

- |             |          |
|-------------|----------|
| 1. DDNAME   | sequence |
| 2. DSNNAME  | sequence |
| 3. Unsorted | sequence |

To produce any of these sequences, enter one of the following symbolics on the EXEC DCDJCL statement:

1. RPTYPE=DDN
2. RPTYPE=DSN
3. RPTYPE=JNOSORT
4. RPTYPE=JCL (for all three reports)

To produce any two of the reports, specify two reports options in the RPTYPE symbolic within single apostrophes as shown in this example:

```
RPTYPE='DDN,JNOSORT'
```

### Other Options

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| JDSNONLY  | When the DSNNAME is missing from a DD and replaced with the use of DATA, DUMMY, SYSOUT=, DDNAME= or *, these fields will be shown as the related DSN. To exclude these DD statements, use the JDSNONLY option within the RPTYPE= symbolic within single apostrophes along with the report options selected. (Note options may be abbreviated to the first three characters).                                                                                            |
| JAPROC    | Without the use of this option or the JBPROGRAM option, the three different reports will have no primary sort ahead of DDNAME, DSNNAME or DDNAMES in their unsorted order within each EXEC. Using this option will sort the JCL report with PROC name as the primary (first) sort field.                                                                                                                                                                                |
| JBPROGRAM | See the JAPROC option. When this option is used, the report will be sorted on program name ahead of DDNAME, DSNNAME or DDNAMES in their unsorted order within each EXEC. If JAPROC is also used with this option, the PROC name for the JAPROC option will be sorted first. The program name for this JBPROGRAM option will be second, followed by DDNAME, or the unsorted DDNAME order.                                                                                |
| JEXEC     | PROC name and PROGRAM will not be shown normally if there are no DDNAMES that follow a PROC or PROGRAM name. Using this JEXEC option will force out the PROC name and/or PROGRAM name even if no DDNAMES are present. Using this option with the DSN or DDN option will cause PROGRAM/PROC records with blank DDNAME and blank DSNNAME to appear at the front of the report or scattered throughout the report depending on the use of the JAPROC and JBPROGRAM option. |

#### Example using additional options

```
//STEP1 EXEC DCDJCL,RPTYPE='DDNAME,DSNNAME,JDSN,JAP,JEX'
```

#### Other options for either report type

The other options that are available are number of lines per page (LNCNT=) and SORT region size (SORTREG=). The default for these are LNCNT=60 and SORTREG=600000.

### Use of Control Statements for Selection of PROCs

The first step in producing a JCL PROC Analysis Report is the selection of JCL members. This is accomplished through the use of a member fetcher PROC which isolates and selects members off of a partitioned data set.

The DD name for these control statements is CTLCDMBR.

Control statements must be used to indicate which members are to be selected. The format of these control statements is as follows:

|         |                           |   |                                                                 |
|---------|---------------------------|---|-----------------------------------------------------------------|
| Columns | 1 through 4               | - | The constant ( INDD )                                           |
| Column  | 5                         | - | An equal sign ( = )                                             |
| Columns | 6 through <i>a</i>        | - | The name of a made-up DDNAME                                    |
| Column  | <i>a</i> +1               | - | An equal sign ( = ) with no spaces before or after it           |
| Columns | <i>b</i> through <i>c</i> | - | One of the following words:<br>1. ALL<br>2. MEMBER<br>3. PREFIX |

The remaining two fields are required if PREFIX is used as the last operand. Otherwise, they may not be used.

|         |                     |   |                                                                 |
|---------|---------------------|---|-----------------------------------------------------------------|
| Columns | <i>c</i> +1         | - | An equal sign ( = ) with no spaces before it or after it        |
| Columns | <i>d</i> - <i>e</i> | - | A 1 to 7 character prefix which will be used to limit selection |

An illustration of the possible combinations follows:

1. INDD=ddname=ALL
2. INDD=ddname=MEMBER
3. INDD=ddname=PREFIX=prefix

An example showing the use of these control statements follows:

```
//STEP1 EXEC DCDJCL
//MBR.CTLCDMBR DD *
INDD=USERDD=PREFIX=SP
//MBR.USERDD DD DSN=USER.JCL.LIBRARY,DISP=SHR
```

The above example will pull off all members from the library, specified by the DD name USERDD, which begins with the prefix SP and passes them out to a sequential data set.

It is left up to the user to ensure that all members beginning with the prefix used are definitely JCL members and not something else, (e.g., COBOL programs, COPY members, Assembler programs).

If members are wanted from more than one library, then multiple DDNAMEs may be used. An example follows:

```
//STEP2 EXEC DCDJCL
//MBR.CTLCDMBR DD *
INDD=USERDD=ALL
INDD=OTHERDD=ALL
//MBR.USERDD DD DSN=USER.JCL.LIBRARY, DISP=SHR
//MBR.OTHERDD DD DSN=USER.JCL.LIBRARY2, DISP=SHR
```

The use of the following words for selection is described here:

1. **ALL**
2. **MEMBER**
3. **PREFIX**

1. **ALL** indicates that all members will be selected from the partitioned data set.
2. **MEMBER** indicates that the user wants to make further selection with member name control cards. These control statements immediately follow the control statement which contains the word **MEMBER**. An example follows:

```
//STEP3 EXEC DCDJCL
//MBR.CTLCDMBR DD *
INDD=USERDD=MEMBER
MEMBER1
MEMBER2
MEMB03
MEMBER04
//MBR.USERDD DD DSN=USER.PDS,DISP=SHR
```

3. **PREFIX** indicates that selection is to be done strictly on the basis of a 1 to 7 character prefix that follows the ( = ) sign after the word **PREFIX**. An example follows:

```
//STEP4 EXEC DCDJCL
//MBR.CTLCDMBR DD *
INDD=USERDD=PREFIX=TRR
//MBR.USERDD DD DSN=USER.JCL.PROC.LIBRARY,DISP=SHR
```

Further notes on the member fetcher:

1. Compressing the PDS before using the data set will eliminate the possibility of pulling in older unwanted members from the PDS.
2. Multiple control statements may be used for any of the above formats.
3. To prevent obtaining duplicate members, do not mix formats in the same run.

**Use of Control Statements for Excluding DDNAMEs**

When using the DCDJCL PROC to produce a report on a PROC or other JCL which contains DD statements, it is possible to exclude DDNAMEs from the report that do not necessarily add to the value of the report. In some cases, just add paper and distract from the value of the report, (e.g., DDNAMEs like SYSOUT, SYSPRINT and SORTLIB).

To invoke this feature, add a DDNAME with the name of EXCLUDE at the end of the JCL and add the DDNAMEs that are to be excluded as shown in the example below:

```
//STEP1 EXEC DCDJCL,RPTYPE=JCL
//MBR.CTLCDMBR DD *
INDD=JCLDD=ALL
//MBR.JCLDD DD DSN=USER.JCLPROC.LIBRARY,DISP=SHR
//JCLRPT.EXCLUDE DD *
SYSUDUMP
SYSOUT
SYSPRINT
SORTLIB
JOB CAT
STEP CAT
/*
```



## JCL Examples

The following example is provided for producing the JCL PROC Analysis Reports:

### Example 1

```
//STEP1 EXEC DCDJCL,RPTYPE=DDN
//MBR.CTLCDMBR DD *
INDD=USERDD=MEMBER
JCLPROC1
PROC2
JCLPROC3
PROC4
//MBR.USERDD DD DSN=USER.JCL.LIBRARY,DISP=SHR
/*
```

The above example produces the JCL PROC Analysis Report in DDNAME sequence for the four PROCs listed above.

### Example 2

```
//STEP2 EXEC DCDJCL,RPTYPE=JCL
//MBR.CTLCDMBR DD *
INDD=ALPHA=PREFIX=TR
INDD=ALPHA=PREFIX=PR
//MBR.ALPHA DD DSN=USER.JCL2.LIBRARY,DISP=SHR
//JCLRPT.EXCLUDE DD *
SYSOUT
SYSPRINT
/*
```

The above example produces the JCL PROC Analysis Report in both DDNAME and DSNAME sequence for all PROCs that begin with the prefix TR or PR on the library USER.JCL2.LIBRARY. It will omit the DDNAMEs, SYSOUT and SYSPRINT from that report.

**DCDJCL PROC**

The following is a listing of the DCDJCL PROC:

```
//DCDJCL PROC BUF=5, ALLOW 5 BUFFERS FOR MVS
// LINECNT=60, NBR OF LINES PER PAGE
// OTHER=,
// PRINT='*',
// REG=2048K,
// RPTYPE=JCL, (JCL) GIVES THE REPORT IN THREE SEQUENCES
// SORTREG=600000,
// WORK=SYSDA
//*
//* MARBLE COMPUTER, INC. DCDJCL PROC FOR PROC ANALYSIS REPORTS
//* LM080108
//MBR EXEC PGM=MBRFETCH,REGION=512K
//*
//* INSERT STEPLIB IF NECESSARY
//*
//SYSPRINT DD SYSOUT=&PRINT
//WORKFILE DD UNIT=&WORK,SPACE=(TRK,(4,2))
//OUTSET DD DSN=&&PASSFILE,DISP=(,PASS),UNIT=&WORK,
// SPACE=(CYL,(5,10))
//*
//JCLRPT EXEC PGM=DCDJCL,REGION=®,COND=(4,LT,MBR),
// PARM=('LNC=&LINECNT','SOR=&SORTREG','RPTYPE,,&OTHER)
//*
//* INSERT STEPLIB IF NECESSARY
//*
//PRINT DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//PRTMSG DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//RPTFILE DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//SYSOUT DD SYSOUT=&PRINT,DCB=BLKSIZE=121
//SYSUDUMP DD SYSOUT=&PRINT,DCB=BLKSIZE=133
//SORTMESS DD DUMMY,DCB=BLKSIZE=121
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&WORK,SPACE=(TRK,(400),,CONTIG)
//SORTWK02 DD UNIT=&WORK,SPACE=(TRK,(400),,CONTIG)
//SORTWK03 DD UNIT=&WORK,SPACE=(TRK,(400),,CONTIG)
//PROCFILE DD DSN=&&PASSFILE,DISP=(OLD,DELETE),DCB=BUFNO=&BUF
//WKFILE DD DISP=(,DELETE),UNIT=&WORK,SPACE=(3240,(200,120)),
// DCB=(LRECL=108,BLKSIZE=3240,RECFM=FB,BUFNO=&BUF)
//SYSIN DD DDNAME=EXCLUDE
//*
//CONTROL DD DSN=USER.PDS(DCDCNTRL),DISP=SHR
```

### Sample Reports

DATA SET ANALYSIS \*\* 01/01/2001 \*\*\*\*\* PAGE 1  
 DSNNAME\*\*\*\*\*PROGRAM\* PROC\*\* STEP-NBR STEP-NAME DDNAME\*\* DISPOSITION

|                 |          |        |   |          |          |            |
|-----------------|----------|--------|---|----------|----------|------------|
|                 | DCDJCL   | DCDJCL | 2 | DCDRPT   | WKFILE   | NEW DELETE |
| &&COMPLIST      | DCDMAIN  | COMACL | 1 | DCD      | INFILE   | OLD PASS   |
|                 | GIVEBACK | COMACL | 2 | RETCMPLR | INFILE   | OLD DELETE |
| &&PASSFILE      | DCDJCL   | DCDJCL | 2 | JCLRPT   | PROFILE  | OLD DELETE |
|                 | MBRFETCH | DCDJCL | 1 | MBR      | OUTSET   | MOD PASS   |
| &&TEMP          | 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   |
| DDNAME          | 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   |            |
|                 |          |        | 2 | JCLRPT   | RPTFILE  |            |
|                 |          |        | 2 | JCLRPT   | SYSOUT   |            |
|                 |          |        | 2 | JCLRPT   | SYSUDUMP |            |
|                 | DCDMAIN  | COMACL | 1 | DCD      | PRINT    |            |
|                 |          |        | 1 | DCD      | PRTCMPLR |            |
|                 |          |        | 1 | DCD      | SORTMESS |            |
|                 |          |        | 1 | DCD      | SYSOUT   |            |
|                 |          |        | 2 | DCD      | SYSOUT   |            |
|                 |          |        | 1 | DCD      | PRINT    |            |
|                 |          |        | 1 | DCD      | SORTMESS |            |
|                 |          |        | 1 | DCD      | SYSOUT   |            |
|                 |          |        | 1 | DCD      | SYSOUT   |            |
|                 |          |        | 2 | DCD      | PRINT    |            |
|                 | DCDACL   |        | 1 | DCD      | PRINT    |            |
|                 |          |        | 1 | DCD      | SORTMESS |            |
|                 | DCDCOBOL | LIBACL | 1 | DCD      | SYSOUT   |            |
|                 |          |        | 2 | DCD      | PRINT    |            |
|                 | PANACL   |        | 2 | DCD      | SORTMESS |            |
|                 |          |        | 2 | DCD      | SYSOUT   |            |
|                 |          |        | 2 | DCD      | SYSOUT   |            |
|                 |          |        | 2 | DCD      | PRINT    |            |
|                 |          |        | 2 | DCD      | SORTMESS |            |
|                 |          |        | 2 | DCD      | SYSOUT   |            |

### JCL PROC Analysis Report - DSN Sequence

## DCD III – JCL PROC Analysis Reports Facility

DATA SET ANALYSIS \*\*\* \*01/01/2001\*\*\*\*\*PAGE 1  
 DDNAME\* PROGRAM PROC \*\*\*\*\* STEP-NBR STEP-NAME DSNAME\*\*\*\*\* DISPOSITION

|         |          |          |   |          |                    |            |
|---------|----------|----------|---|----------|--------------------|------------|
| COBOLIN | DCDMAIN  | LIBACL   | 2 | DCD      | &&TEMP             | OLD DELETE |
|         |          | PANACL   | 2 | DCD      | &&TEMP             | OLD DELETE |
| CONTROL | DCDJCL   | DCDJCL   | 2 | JCLRPT   | USER.PDS(DCDCNTRL) | SHR        |
|         |          | PANACL   | 2 | JCLRPT   | USER.PDS(DCDCNTRL) | SHR        |
|         | DCDMAIN  | COMACL   | 1 | DCD      | USER.PDS(DCDCNTRL) | SHR        |
|         |          | DCDAACL  | 1 | DCD      | USER.PDS(DCDCNTRL) | SHR        |
|         |          | LIBACL   | 2 | DCD      | USER.PDS(DCDCNTRL) | SHR        |
|         |          | PANACL   | 2 | DCD      | USER.PDS(DCDCNTRL) | SHR        |
|         | DCDSYSTM | DCDCOBOL | 1 | DCD      | USER.PDS(DCDCNTRL) | SHR        |
| INFILE  | DCDMAIN  | COMACL   | 1 | DCD      | &&COMPLIST         | OLD PASS   |
|         | GIVEBACK | COMACL   | 2 | RETCMPLR | &&COMPLIST         | OLD DELETE |
| LIST    | LIBRARAN | LIBACL   | 1 | LIB      | SYSOUT             |            |
| MASTER  | LIBRARAN | LIBACL   | 1 | LIB      | LIBRARAN.SOURCE    | SHR        |
| OSJOB   | LIBRARAN | LIBACL   | 1 | LIB      | &&TEMP             | NEW PASS   |
| OUTSET  | MBRFETCH | DCDJCL   | 1 | MBR      | &&PASSFILE         | MOD PASS   |
| PAN001  | PAN#1    | PANACL   | 1 | PAN      | PANVALET.SOURCE    | SHR        |
| PAN002  | PAN#1    | PANACL   | 1 | PAN      | &&TEMP             | NEW PASS   |
| PRINT   | DCDJCL   | DCDJCL   | 2 | JCLRPT   | SYSOUT             |            |
|         | DCDMAIN  | COMACL   | 1 | DCD      | SYSOUT             |            |
|         |          | DCDAACL  | 1 | DCD      | SYSOUT             |            |
|         |          | LIBACL   | 2 | DCD      | SYSOUT             |            |
|         |          | PANACL   | 2 | DCD      | SYSOUT             |            |
|         | DCDSYSTM | DCDCOBOL | 1 | DCD      | SYSOUT             |            |
| PROFILE | DCDJCL   | DCDJCL   | 2 | JCLRPT   | &&PASSFILE         | OLD DELETE |
| PRTCMLR | DCDMAIN  | COMACL   | 1 | DCD      | SYSOUT             |            |
| PRTFILE | GIVEBACK | COMACL   | 2 | RETCMPLR | SYSOUT             |            |
| PRTMSG  | DCDJCL   | DCDJCL   | 2 | JCLRPT   | SYSOUT             |            |
| REPORTS | DCDSYSTM | DCDCOBOL | 1 | DCD      | SYSOUT             |            |

### JCL PROC Analysis Report - DDNAME Sequence

# DCD III – JCL PROC Analysis Reports Facility

| JCL DATA | SET ANALYSIS     | 01/01/2001 | INPUT     | SEQUENCE           | (NO SORTING) | PAGE 1      |
|----------|------------------|------------|-----------|--------------------|--------------|-------------|
| DDNAME** | PROGRAM* PROC*** | STEP-NBR   | STEP-NAME | DSNAME*****        | *****        | DISPOSITION |
| CONTROL  | DCDMAIN DCDACL   | 1          | DCD       | USER.PDS(DCDCNTRL) |              | SHR         |
| PRINT    |                  | 1          |           | SYSOUT             |              |             |
| SORTLIB  |                  | 1          |           | SYS1.SORTLIB       |              | SHR         |
| SORTMESS |                  | 1          |           | SYSOUT             |              |             |
| SYSOUT   |                  | 1          |           | SYOUT              |              |             |
| SYSPRINT | LIBRARAN LIBACL  | 1          | LIB       | SYSOUT             |              |             |
| LIST     |                  | 1          |           | SYSOUT             |              |             |
| MASTER   |                  | 1          |           | LIBRARAN.SOURCE    |              | SHR         |
| OSJOB    |                  | 1          |           | &&TEMP             |              | NEW PASS    |
| COBOLIN  | DCDMAIN          | 2          | DCD       | && TEMP            |              | OLD DELETE  |
| CONTROL  |                  | 2          |           | USERPDS(DCDCNTRL)  |              | SHR         |
| PRINT    |                  | 2          |           | SYSOUT             |              |             |
| SORTLIB  |                  | 2          |           | SYS1.SORTLIB       |              | SHR         |
| SORTMESS |                  | 2          |           | SYSOUT             |              |             |
| SYSOUT   |                  | 2          |           | SYSOUT             |              |             |
| SYSPRINT | MBRFETCH DCDJCL  | 1          | MBR       | SYSOUT             |              |             |
| OUTSET   |                  | 1          |           | &&PASSFILE         |              | MOD PASS    |
| PRINT    | DCDJCL           | 2          | JCLRPT    | SYSOUT             |              |             |
| PRTMSG   |                  | 2          |           | SYSOUT             |              |             |
| RPTFILE  |                  | 2          |           | SYSOUT             |              |             |
| SYSOUT   |                  | 2          |           | SYSOUT             |              |             |
| SYSUDUMP |                  | 2          |           | SYSOUT             |              |             |
| SYSOUT   |                  | 2          |           | SYSOUT             |              |             |
| SORTMESS |                  | 2          |           | DUMMY              |              |             |
| SORTLIB  |                  | 2          |           | SYS1.SORTLIB       |              | SHR         |
| PROCFE   |                  | 2          |           | &&PASSFILE         |              | OLD DELETE  |
| WKFILE   |                  | 2          |           |                    |              | NEW DELETE  |
| SYSIN    |                  | 2          |           | DDNAME=EXCLUDE     |              |             |
| CONTROL  |                  | 2          |           | USER.PDS(DCDCNTRL) |              | SHR         |

## JCL PROC Analysis Report - Unsorted Sequence

| JCL JOBLIB<br>JOBNAME | ANALYSIS<br>PROCNAME                                                                | 01/01/2001 | IN JOB NAME SEQUENCE | PAGE 1 |
|-----------------------|-------------------------------------------------------------------------------------|------------|----------------------|--------|
| DCD115D               | >-----<<br>COPYPROC<br>DVN2PROC<br>PROC003                                          |            |                      |        |
| DCD116D               | >-----<<br>DTNUPR2                                                                  |            |                      |        |
| DCD116NA              | >-----<<br>DTNPRC60<br>DTNUPR2                                                      |            |                      |        |
| DCD117D               | >-----<<br>DSN1PR1<br>DSN2PR1<br>DSN3PR6<br>DSN4PR6<br>PROC01<br>PROC02<br>TRAC03NA |            |                      |        |
| DCD118L               | >-----<<br>TRUN02A<br>TRUN03<br>YTOKENS                                             |            |                      |        |

### Job Library Report - JOB NAME Sequence

Release 3.6

## **Error Messages were completely revamped in Release 3.6**

### DCDIII

### Error Messages

**Notice** – Error Messages have been **completely changed** in Release 3.6

For users using an older version of DCD III, consult 3.5 Manual.  
To obtain 3.5 Manual, go to our web site: [www.marblecomputer.com](http://www.marblecomputer.com)

*This page intentionally left blank*



### **Error Messages**

Each error message has a 10-character alphanumeric code associated with it in the following format:

Positions 1 - 3 - DCD  
Position 4 - character indicating the sub-system the message comes from  
Position 5 - character indicating a program within specified sub-system  
Position 6 - 8 - 3 characters indicating the location within the program of the message  
Position 9 - hyphen (-)  
Position 10 - one of the following letters: I, W, C, E, or D where:

- I** indicates an informational message that requires no action from the user.
- W** indicates a warning, which the user should look at, then correct if necessary.
- C** indicates an error was found and it should be looked at and if possible, corrected.
- E** indicates an error was found and it should be looked at and corrected.
- D** indicates an error was found that is causing DCD III to stop processing prior to normal end of job. This error must be corrected before DCD III will successfully run.

Error messages when they are present will print on the PRINT DD file starting on the second or third page.

If the PRINT DD is somehow omitted, then the following messages will occur on the SYSOUT file:

```
***** DDNAME (PRINT) MISSING
***** DDNAME (PRINT) MISSING
```

DCD III messages are listed on the following pages in alphanumeric sequence of the 10-character codes associated with each error message, with numbers sorting ahead of letters.

If you should encounter error messages in your DCD III run that are not self explanatory or require further explanation, look the message up and follow any comments that are given for that particular error message.

The Error messages listed below are major errors that are not listed on the following pages, and are unlikely to occur. Should you receive one of these errors, we request that you contact Marble Computer, Inc. for further assistance, should re-running or correcting any JCL errors not immediately correct the problem or remove the error.

|             |            |            |            |            |
|-------------|------------|------------|------------|------------|
| DCDA4022-D  | DCDGJD01-E | DCDJPW03-D | DCDMCB09-D | DCDNHB12-E |
| DCDCCC005-E | DCDGJF02-D | DCDJPA01-D | DCDMCC02-D | DCDNHK01-D |
| DCDCCC02-E  | DCDGJH07-E | DCDKBR10-E | DCDMCF02-D | DCDNL004-D |
| DCDCCG03-E  | DCDGJH17-E | DCDKBR30-E | DCDME030-D | DCDPB050-D |
| DCDEA030-D  | DCDGKA02-D | DCDKBY40-D | DCDMGC02-D | DCDPEY70-D |
| DCDEAC04-E  | DCDGKC01-E | DCDKBY50-D | DCDMGM85-E | DCDPEY80-D |
| DCDEAE01-E  | DCDGKD01-E | DCDKM020-D | DCDMGM86-E | DCDPNU20-D |
| DCDEB030-D  | DCDGP040-D | DCDKW040-D | DCDMGM95-E | DCDPNU21-D |
| DCDEFB91-E  | DCDGQA02-D | DCDM3030-D | DCDMHC02-D | DCDTBM85-E |
| DCDEFH00-E  | DCDGR020-D | DCDM5050-D | DCDMHM85-E | DCDTBM86-E |
| DCDEFH21-E  | DCDGS010-D | DCDM5A02-D | DCDMHM86-E | DCDTBM95-E |
| DCDESDDA-D  | DCDGS030-D | DCDM5B12-E | DCDMHM95-E | DCDTDM85-E |
| DCDESDD-E   | DCDGSE04-E | DCDM7A02-D | DCDMNA02-D | DCDTDM86-E |
| DCDESDDK-E  | DCDGSFA1-E | DCDM7B04-D | DCDMP030-D | DCDTDM95-E |
| DCDESYLB-E  | DCDGSG01-E | DCDM7C02-D | DCDMQA02-D | DCDTJ040-D |
| DCDG8030-D  | DCDGSG09-E | DCDM7F02-D | DCDMQB10-E | DCDTLC02-D |
| DCDGBA02-D  | DCDGSGA1-D | DCDM8A02-D | DCDMRA02-D | DCDTLM85-E |
| DCDGBA10-E  | DCDGV040-D | DCDM8B13-D | DCDMRB10-E | DCDTLM86-E |
| DCDGBB03-D  | DCDJAA02-D | DCDM8C02-D | DCDMX020-D | DCDTLM95-E |
| DCDGBB11-E  | DCDJAA06-D | DCDM8D13-D | DCDMX030-D | DCDUDA01-D |
| DCDGBB20-E  | DCDJCA02-D | DCDM8M01-D | DCDMX040-D | DCDUF010-D |
| DCDGBB50-E  | DCDJCA06-D | DCDM8M02-D | DCDMX050-D | DCDUFA02-D |
| DCDGBC01-D  | DCDJE050-D | DCDM8Q01-E | DCDNB020-D | DCDUFD01-E |
| DCDGE030-D  | DCDJEA01-D | DCDMBC02-D | DCDNB060-D | DCDUHD01-E |
| DCDGF030-D  | DCDJEB10-E | DCDMBM85-E | DCDND040-D | DCDUJA02-E |
| DCDGF060-D  | DCDJGA02-D | DCDMBM86-E | DCDNF020-D | DCDV3055-D |
| DCDGJA02-D  | DCDJIA02-D | DCDMBM95-E | DCDNH050-D | DCDVA021-D |
| DCDGJC01-E  | DCDJPA02-D | DCDMCA02-D | DCDNHA13-E | DCDVAW03-D |

If you need to contact Marble Computer, Inc. for technical support, you may follow one of the below suggested procedures:

1. Send an email to us at [support@marblecomputer.com](mailto:support@marblecomputer.com) with any attachments needed.
2. You may call Marble Computer, Inc. toll-free at 800-252-1400. We recommend having the following information available when calling.
  - Release number and release date of the version of DCD III that you are running
  - The error message number(s)
  - If you are using DCD III as a remote job entry to a data center, specify the location of the data center that you are using.
  - Exactly what conditions existed when the error messages were encountered (what PARM options were in effect, what PROC was being executed, any other related conditions).

|                   |                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCD5E040-C</b> | <b>PARM LENGTH GREATER THAN 100 - ACCEPTED</b>                                                                                                                                                                                                                                                      |
| Explanation:      | The number of characters in the total PARM field is greater than 100 characters. Only the first 100 characters are meaningful to DCD III. The characters over 100 are ignored.                                                                                                                      |
| User Action:      | Make adjustments to the PARM field to reduce the total length to be under 100 characters.                                                                                                                                                                                                           |
| <b>DCD5E052-I</b> | <b>PARM (R3N) OPTION TURNS OFF (PMO) OPTION ELIMINATING PRINTING OF OFFSETS TO THE FAR RIGHT OF SOURCE CODE</b>                                                                                                                                                                                     |
| Explanation:      | Parm option R3N is inconsistent with parm option PMO. Option R3N carries PL and SL information from the right hand side of the compile listing into right side of DCD III listing, while option PMO traps information from the compile OFFSET option and prints into right side of DCD III listing. |
| User Action:      | Determine whether PMO option is needed, and if not needed turn that options off, or turn option R3N off.                                                                                                                                                                                            |
| <b>DCD5E056-W</b> | <b>PARM OPTION ( NIS ) MAY NOT BE USED WITH OPTION ( IREFS ) - OPTION ( IREFS ) IS TURNED OFF</b>                                                                                                                                                                                                   |
| Explanation:      | Parm option NIS (which is used to provide a sequential sequence) for Procedure Division Narrative is not compatible with the IREFS option which provides indirect References.                                                                                                                       |
| User Action:      | The IREFS option is being turned off. To turn it back on, do not use PARM option NIS. Determine whether NIS option is needed, and if not, turn that options off.                                                                                                                                    |
| <b>DCD5E060-D</b> | <b>PARM OPTION (LOR) MAY NOT BE USED WHEN OTHER REPORTS ARE ALSO TURNED ON</b>                                                                                                                                                                                                                      |
| Explanation:      | The option LOR indicates that non-COBOL programs are being used as input for creating the Layouts Reports.                                                                                                                                                                                          |
| User Action:      | Do not use this option when other report options are also turned on.                                                                                                                                                                                                                                |
| <b>DCD5E061-W</b> | <b>PARM OPTION (LHR) TREATED AS (LHX) WHEN OPTION (LNR) FOR NARRATIVE IS IN EFFECT</b>                                                                                                                                                                                                              |
| Explanation:      | Options LHR and LNR are incompatible. See these options in this User's Manual.                                                                                                                                                                                                                      |
| User Action:      | Do not use these options together.                                                                                                                                                                                                                                                                  |
| <b>DCD5E062-D</b> | <b>PARM OPTION (TRACE) NOT ALLOWED WHEN OPTIONS (WRITPDS OR READPDS) ARE USED</b>                                                                                                                                                                                                                   |
| Explanation:      | Option TRACE is incompatible with option WRITPDS or READPDS. See these options in this User's Manual.                                                                                                                                                                                               |
| User Action:      | Do not use these options together. Eliminate TRACE or eliminate READPDS or WRITPDS.                                                                                                                                                                                                                 |
| <b>DCD5E063-D</b> | <b>AT LEAST ONE PRINT REPORT OPTION MUST BE TURNED ON WHEN RUNNING THE OTHER COBOL REPORTS</b>                                                                                                                                                                                                      |
| Explanation:      | No PARM options were specified (DATA, LAYOUT, etc...) for printing of reports or for the specification of the Data Dictionary file. Even when using WRITPDS or READPDS options, report options must be specified.                                                                                   |
| User Action:      | Specify one or more report options.                                                                                                                                                                                                                                                                 |
| <b>DCD5E065-D</b> | <b>(WRITPDS) PARM OPTION ALSO REQUIRES (BEGIN) OR (CONTINUE) PARM OPTION</b>                                                                                                                                                                                                                        |
| Explanation:      | Parm option WRITPDS requires either option BEG or CON. See these options in this User's Manual.                                                                                                                                                                                                     |
| User Action:      | Add parm option BEGIN or parm option CONTINUE.                                                                                                                                                                                                                                                      |
| <b>DCD5E067-D</b> | <b>PARM OPTION (S03), (S15) OR (S50) MAY NOT BE USED WITH (LAYOUT) OR (RECORDS) OPTION</b>                                                                                                                                                                                                          |
| Explanation:      | PARM options S013, S15, S50, S3H and S1T alter the maximum size of OCCURS within a record which contains an OCCUR clause for use with the System Record Analysis Report. Using these options with the RECORD or LAYOUT reports will produce incorrect results.                                      |
| User Action:      | Turn off the RECORD and LAYOUT options or remove the S03, S15, S50, S3H and S1T options being used.                                                                                                                                                                                                 |
| <b>DCD5E800-I</b> | <b>(STOP) PARM OPTION IN EFFECT</b>                                                                                                                                                                                                                                                                 |
| Explanation:      | The PARM option STOP was entered and processing in DCD III is halted.                                                                                                                                                                                                                               |
| User Action:      | None. To process with DCD III, remove the STOP option.                                                                                                                                                                                                                                              |
| <b>DCD5EA11-E</b> | <b>PARM OPTION (parm field) INVALID WHEN FOLLOWING (option) AND IS IGNORED</b>                                                                                                                                                                                                                      |
| Explanation:      | The two PARM fields shown in the message conflict with each other. The first PARM option will be used and the second one will be ignored.                                                                                                                                                           |
| User Action:      | Remove either of the two options shown.                                                                                                                                                                                                                                                             |
| <b>DCD5EA20-C</b> | <b>INTERNAL ERROR FOR PARM (parm field)</b>                                                                                                                                                                                                                                                         |
| Explanation:      | An internal error has occurred during the storing of PARM fields.                                                                                                                                                                                                                                   |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User Action:      | Save printed output and contact MARBLE Computer, Inc.                                                                                                                                                                 |
| <b>DCD5EA50-D</b> | <b>FIELD IN PARM RECORD TOO LARGE (parm field)</b>                                                                                                                                                                    |
| Explanation:      | The PARM field displayed is in excess of 10 characters. All valid PARM fields are 10 characters or under in length.                                                                                                   |
| User Action:      | Determine what PARM field was intended and correct.                                                                                                                                                                   |
| <b>DCD5EA70-D</b> | <b>PARM-FIELD IS INVALID (parm field)</b>                                                                                                                                                                             |
| Explanation:      | The first three characters of the PARM field displayed do not match to a valid PARM field.                                                                                                                            |
| User Action:      | Consult the User's Manual if necessary and then re-enter a valid PARM field.                                                                                                                                          |
| <b>DCD5EA78-C</b> | <b>FOUND MORE THAN 1 PARM OPTION IN THE RANGE IR1 - IR6 –USING LAST FOUND</b>                                                                                                                                         |
| Explanation:      | More than one parm option (IR1, IR2, IR3, IR4, IR5, IR6) was used.                                                                                                                                                    |
| User Action:      | Remove one or more parm options, so that only one is present.                                                                                                                                                         |
| <b>DCD5EA90-W</b> | <b>PARM OPTION (parm field) MISCODED. ACCEPTED AS (parm field)</b>                                                                                                                                                    |
| Explanation:      | The first three characters of the PARM field matches with a valid PARM option, however, the remaining characters do not match.                                                                                        |
| User Action:      | Check to be sure the PARM option used was the one intended. Correct the PARM field for later runs.                                                                                                                    |
| <b>DCD5EC01-E</b> | <b>PARM OPTION (parm field) NOT FOLLOWED BY (=) AND (VALUE)</b>                                                                                                                                                       |
| Explanation:      | The PARM field displayed is a keyword type option which requires both an (=) sign and value after the (=) character. Either the (=) character is missing or the value after the (=) character is miscoded or missing. |
| User Action:      | Ensure that the fully coded PARM field contains the (=) character and a valid value. It may also help to ensure that the entire keyword and value are enclosed in single apostrophes.                                 |
| <b>DCD5EF01-C</b> | <b>PROBLEM WITH CLN= (USING CLN=60)</b>                                                                                                                                                                               |
| Explanation:      | The value found after CLN= was not numeric or was numerically greater than 999.                                                                                                                                       |
| User Action:      | Correct by using a valid number. For this run a default of 60 is used.                                                                                                                                                |
| <b>DCD5EF02-C</b> | <b>PROBLEM WITH LNC= (USING LNC=60)</b>                                                                                                                                                                               |
| Explanation:      | The value found after LNC= was not numeric or was numerically greater than 999.                                                                                                                                       |
| User Action:      | Correct by using a valid number. For this run a default of 60 is used.                                                                                                                                                |
| <b>DCD5EF03-C</b> | <b>PROBLEM WITH SOR= (USING SOR=150000)</b>                                                                                                                                                                           |
| Explanation:      | The value found after SOR= was not numeric or was 10 million or greater.                                                                                                                                              |
| User Action:      | Use a valid number for SORT region. Make sure the overall REGION is 3072 to 4096K- larger than the SORT region.                                                                                                       |
| <b>DCD5EF25-C</b> | <b>PROBLEM WITH TAC= (USING TAC=025)</b>                                                                                                                                                                              |
| Explanation:      | The value found after TAC = was not numeric or was numerically greater than 999.                                                                                                                                      |
| User Action:      | Correct by using a valid number. For this run, default of 025 is used.                                                                                                                                                |
| <b>DCD5G010-D</b> | <b>DDNAME (CONTROL) MISSING</b>                                                                                                                                                                                       |
| Explanation:      | A DD card is missing from the user JCL.                                                                                                                                                                               |
| User Action:      | See the User's Manual and Installation Addendum for instructions on adding a CONTROL DD card.                                                                                                                         |
| <b>DCD5G025-D</b> | <b>//CONTROL DD CONTROL INFO MISSING</b>                                                                                                                                                                              |
| Explanation:      | The CONTROL DD file is empty. One control record is required.                                                                                                                                                         |
| User Action:      | See the Installation Addendum for format and provide the control record required.                                                                                                                                     |
| <b>DCD5G030-D</b> | <b>//CONTROL DD CONTROL INFO . MUST BE IN VALID FORMAT (MMYY). FOUND (xxxx)</b>                                                                                                                                       |
| Explanation:      | The password control information contains invalid information in columns 1-4. MMYY for month and year is expected.                                                                                                    |
| User Action:      | Enter the information in correct format (MMYY) and run again. If necessary consult the Installation Addendum for further information.                                                                                 |
| <b>DCD5G046-D</b> | <b>CONTROL MONTH/YEAR EXPIRED - MM/YYYY = XX/XXXX</b>                                                                                                                                                                 |
| Explanation:      | Six characters of control information matching to MMYYYY are invalid.                                                                                                                                                 |
| User Action:      | See the Installation Addendum or contact MARBLE Computer, Inc.                                                                                                                                                        |
| <b>DCD5G048-D</b> | <b>SIX CHARACTER CODE (6-char) IN //CONTROL DD CONTROL INFO IS INVALID.</b>                                                                                                                                           |
| Explanation:      | Six characters of control information are invalid.                                                                                                                                                                    |
| User Action:      | See the Installation Addendum or contact MARBLE Computer, Inc.                                                                                                                                                        |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCD5H050-E</b> | <b>(TRACEIN) COMMANDS MUST BEGIN WITH AN (AFTER) COMMAND</b>                                                                                                                           |
| Explanation:      | When using the debugging control statements under direction of Marble Computer, each command must begin with the word ( AFTER )                                                        |
| User Action:      | Check the TRACEIN DD statements and make corrections to the TRACEIN commands as necessary.                                                                                             |
| <b>DCD5H070-D</b> | <b>ONE OR MORE ERRORS FOUND IN (TRACEIN) COMMANDS</b>                                                                                                                                  |
| Explanation:      | One or more errors occurred in processing the debugging control statements.                                                                                                            |
| User Action:      | Check the related error messages and make corrections to the TRACEIN commands as necessary.                                                                                            |
| <b>DCD5HA01-E</b> | <b>PROGRAM-NAME MISSING AFTER (AFTER) COMMAND IN (TRACEIN) FILE</b>                                                                                                                    |
| Explanation:      | The program name must be used in the TRACEIN file. It was not there.                                                                                                                   |
| User Action:      | Check the TRACEIN file and make corrections as necessary.                                                                                                                              |
| <b>DCD5HA02-E</b> | <b>EXPECTING WORD (PRINT) WITHIN (TRACEIN) FILE FOUND (field-contents)</b>                                                                                                             |
| Explanation:      | The word PRINT must be used in the TRACEIN file. It could not be found.                                                                                                                |
| User Action:      | Check the TRACEIN file and make corrections as necessary.                                                                                                                              |
| <b>DCD5HA03-E</b> | <b>EXPECTING (DCDWK01) THRU (DCDWK06) WITHIN (TRACEIN) FILE - FOUND (field-contents)</b>                                                                                               |
| Explanation:      | The file names to be TRACEed must be DCDWK01 through DCDWK06.                                                                                                                          |
| User Action:      | Check the file names and correct as necessary.                                                                                                                                         |
| <b>DCD5HA05-E</b> | <b>ONLY 10 CONTROL CARDS ALLOWED WITHIN (TRACEIN) FILE</b>                                                                                                                             |
| Explanation:      | The number of control cards in the TRACEIN file exceeds 10.                                                                                                                            |
| User Action:      | Check the TRACEIN file and delete excess cards so there are a total of 10 cards or less.                                                                                               |
| <b>DCD5HA06-E</b> | <b>EXPECTING NEW (AFTER) COMMAND</b>                                                                                                                                                   |
| Explanation:      | The end of one command in the TRACEIN file was found. The next token (if present) must be the start of a new command.                                                                  |
| User Action:      | If multiple commands are used, make sure each new command starts with the word AFTER.                                                                                                  |
| <b>DCD5HA31-E</b> | <b>DCDWKXX NUMBER MUST BE 01 TO 06 WITHIN (TRACEIN) FILE</b>                                                                                                                           |
| Explanation:      | The DCDWKXX file that has been specified in the TRACEIN file does not exist.                                                                                                           |
| User Action:      | Check the file names and make corrections as necessary.                                                                                                                                |
| <b>DCD5HA35-E</b> | <b>PROGRAM OR FILE NUMBER IN CONTROL CARD IS INVALID</b>                                                                                                                               |
| Explanation:      | The program-id did not match or the record length was not numeric.                                                                                                                     |
| User Action:      | Check the control card and make corrections as necessary.                                                                                                                              |
| <b>DCD5HA90-I</b> | <b>BYPASSING TOKENS WITHIN (TRACEIN) FILE UNTIL NEXT (AFTER) OR END OF FILE</b>                                                                                                        |
| Explanation:      | A previous documented syntax error was found while scanning an AFTER command within the TRACEIN file. Further scanning for errors will not be done until a new AFTER command is found. |
| User Action:      | None.                                                                                                                                                                                  |
| <b>DCD5X001-C</b> | <b>EXCESSIVE ERROR MESSAGES IN HSKPG</b>                                                                                                                                               |
| Explanation:      | The table of error messages was exceeded for the first phase of processing within DCD III.                                                                                             |
| User Action:      | Examine the previous error messages and make corrections as necessary to eliminate the errors.                                                                                         |
| <b>DCD63070-E</b> | <b>THE VALUES USED WITHIN SELECT CARDS ESTABLISHES A RANGE THAT IS NEGATIVE</b>                                                                                                        |
| Explanation:      | Control cards used to select records for Layout Reports by record size are incorrect.                                                                                                  |
| User Action:      | Correct either the SELECT> or the SELECT< card.                                                                                                                                        |
| <b>DCD63080-I</b> | <b>USING CONTROL CARDS FOR SELECTION BY DATA-NAME FOR LAYOUTS</b>                                                                                                                      |
| Explanation:      | This is an informational message reminding the user that control cards are being used for the selection of Layout Reports.                                                             |
| User Action:      | None.                                                                                                                                                                                  |
| <b>DCD63081-I</b> | <b>USING CONTROL CARDS FOR SELECTION BY RECORD-SIZE FOR LAYOUTS</b>                                                                                                                    |
| Explanation:      | This is an informational message reminding the user that control cards are being used for the selection of Layout Reports.                                                             |
| User Action:      | None.                                                                                                                                                                                  |
| <b>DCD63A03-I</b> | <b>DATA NAME IS GREATER THAN 30 CHARACTERS</b>                                                                                                                                         |
| Explanation:      | Programmer-supplied names must be 30 characters or less in length.                                                                                                                     |
| User Action:      | Check the data name to make sure that it less than 30 characters.                                                                                                                      |

|                   |                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCD63A06-E</b> | <b>UNIDENTIFIED FIELD AFTER DATA NAME ( record-name )</b>                                                                                      |
| Explanation:      | Field in control statement after data name indicated is in error.                                                                              |
| User Action:      | Check the control statement, correct, and re-submit.                                                                                           |
| <b>DCD63A07-E</b> | <b>ONLY 750 CONTROL CARDS ALLOWED WITHIN FILE</b>                                                                                              |
| Explanation:      | There are too many control statements used.                                                                                                    |
| User Action:      | Check the file and eliminate excess cards so there are a total of 750 statements or less.                                                      |
| <b>DCD63AA1-E</b> | <b>CONFLICTS WITH A PREVIOUS (SELECT) CONTROL CARD</b>                                                                                         |
| Explanation:      | The control card conflicts with a previous SELECT control card.                                                                                |
| User Action:      | Check the control cards and make necessary changes.                                                                                            |
| <b>DCD63AA2-E</b> | <b>CONFLICTS WITH A PREVIOUS (SELECT) CONTROL CARD</b>                                                                                         |
| Explanation:      | The control card conflicts with a previous SELECT control card.                                                                                |
| User Action:      | Check the control cards and make necessary changes.                                                                                            |
| <b>DCD63AA3-E</b> | <b>SYNTAX ERROR IN SELECT CARD</b>                                                                                                             |
| Explanation:      | The SELECT card has a non-numeric field or a number over 6 characters in length.                                                               |
| User Action:      | The correct format requires positions 1-6 to be numeric, columns 7 to be a dash, and 8-13 to be numeric.                                       |
| <b>DCD66003-D</b> | <b>DDNAME (CTLCD SRA) IS MISSING</b>                                                                                                           |
| Explanation:      | In the JCL, the DD card for CTLCD SRA is missing.                                                                                              |
| User Action:      | Check JCL to ensure the DD card for CTLCD SRA is there and correctly placed. This control card is needed when PARM option DICT or SRA is used. |
| <b>DCD66A03-E</b> | <b>PROGRAM-ID IN EXCESS OF 8 CHARACTERS</b>                                                                                                    |
| Explanation:      | The program name/record name control card statement is invalid.                                                                                |
| User Action:      | A 1 - 8 character program name must begin in column 1.                                                                                         |
| <b>DCD66A04-E</b> | <b>RECORD NAME OR LENGTH IN A CONTROL CARD IS MISSING</b>                                                                                      |
| Explanation:      | The control card is missing either the record name or length.                                                                                  |
| User Action:      | Check the control card for the record name or length to ensure one is present.                                                                 |
| <b>DCD66A05-E</b> | <b>RECORD NAME IN EXCESS OF 30 CHARACTERS</b>                                                                                                  |
| Explanation:      | The program name/record name control card statement is invalid.                                                                                |
| User Action:      | A 1 - 8 character program name must begin in column 1. A 1 - 30 character 01 record name must follow after one or more spaces.                 |
| <b>DCD66A06-E</b> | <b>NUMERIC FIELD IS IN EXCESS OF 6 CHARACTERS</b>                                                                                              |
| Explanation:      | A control card field used for the System Record Analysis Report has a numeric field greater than 6 digits in length.                           |
| User Action:      | Shorten the number to 6 digits.                                                                                                                |
| <b>DCD66A07-E</b> | <b>MAXIMUM NUMBER OF CONTROL CARDS WITHIN A FILE HAS BEEN EXCEEDED</b>                                                                         |
| Explanation:      | The number of control cards in the file has been exceeded.                                                                                     |
| User Action:      | Check the file and delete excess cards.                                                                                                        |
| <b>DCD66AB2-E</b> | <b>A FIELD IS IN EXCESS OF 6 CHARACTERS OR NOT NUMERIC</b>                                                                                     |
| Explanation:      | A numeric field within an SRA control statement range is too big or not numeric.                                                               |
| User Action:      | Correct the control statement and rerun.                                                                                                       |
| <b>DCD66AB3-E</b> | <b>THE RANGE OF NUMBERS IN SRA CTL CD IS INVALID</b>                                                                                           |
| Explanation:      | A range of numbers found in an SRA control statement was found to be invalid.                                                                  |
| User Action:      | Determine what should be entered, fix, and rerun.                                                                                              |
| <b>DCD6AB01-D</b> | <b>POSITIONS 33 THROUGH 72 IN CONTROL CARD MUST BE BLANK</b>                                                                                   |
| Explanation:      | Control Statements for option DIT must have positions 33 up through position 72 blank.                                                         |
| User Action:      | Look at control statements used, consult manual if need be, and correct.                                                                       |
| <b>DCD6AC01-D</b> | <b>LEADING SPACES ARE NOT PERMITTED FOR THESE CONTROL STATEMENTS</b>                                                                           |
| Explanation:      | Control Statements for option DIT must have the command used start in column 1.                                                                |
| User Action:      | Look at control statements used and correct.                                                                                                   |
| <b>DCD6AD10-D</b> | <b>DDNAME (CTLCD DLT) IS MISSING WHEN PARM OPTION ( DIT ) IS SPECIFIED</b>                                                                     |
| Explanation:      | Option DIT was used indicating control statements for Literal Tracing, however the required DDNAME CTLCD DLT is missing.                       |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User Action:      | Remove option DIT, or add DDNAME CTLCDDL with valid control statements.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>DCD6AD20-D</b> | <b>DDNAME (DCDWKT1) IS MISSING WHEN PARM OPTION ( DIT ) IS SPECIFIED</b>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Explanation:      | Option DIT requires an extra work file DCDWKT1 and this file was not found.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| User Action:      | Add an extra work file, similar to work file DCDWK01, 02, etc. with the name DCDWKT1. Also insure that DD names DCDWKT2, DCDWKT3, and DCDWKT4 are added similar to DCDWKT1.                                                                                                                                                                                                                                                                                                                          |
| <b>DCD6C030-D</b> | <b>DDNAME (SELECT) IS MISSING WHEN PARM OPTION (TRACE) IS SPECIFIED</b>                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Explanation:      | Parm Option TRACE indicates SELECTs are used for 'Follow a Data Field' report and DD SELECT is required.                                                                                                                                                                                                                                                                                                                                                                                             |
| User Action:      | Inspect JCL and add DDNAME <b>SELECT</b> with SELECTs to produce report.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>DCD6C031-D</b> | <b>DDNAME (DCDWK08) MISSING</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Explanation:      | A DDNAME (DCDWK08) is missing from JCL.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| User Action:      | Inspect JCL and add DDNAME DCDWK08 to same.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>DCD6C110-D</b> | <b>CORRECT ERRORS FOUND IN SELECT STATEMENTS - SEE REPORT FILE</b>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Explanation:      | See other file and look at syntax errors there for SELECT control statements used.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| User Action:      | Correct the errors found and resubmit the run.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>DCD6CF02-D</b> | <b>A LIMIT OF 250 SELECT STATEMENTS ARE ALLOWED FOR (TRACE) - THIS NUMBER WAS EXCEEDED</b>                                                                                                                                                                                                                                                                                                                                                                                                           |
| Explanation:      | More than 250 SELECTs were submitted. The limit allowed is 250.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| User Action:      | Reduce run to 250 SELECTs or less and resubmit.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>DCD6CJ01-D</b> | <b>SELECTS WITHOUT ( NAME ) KEYWORD MUST COME BEFORE SELECTS WITH ( NAME ) KEYWORD</b>                                                                                                                                                                                                                                                                                                                                                                                                               |
| Explanation:      | Format#2 SELECTs in format <b>SELECT IF NAME = data-name</b> must come after all other SELECTs using format#1 which uses other keywords with AND/OR logic.                                                                                                                                                                                                                                                                                                                                           |
| User Action:      | Put all SELECT IF NAME control statements last and resubmit.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>DCDA2001-D</b> | <b>DDNAME (INFILE) MISSING FOR (COMPILE) MODE - IF INDEPENDENT MODE, DO NOT USE (COM) OPTION</b>                                                                                                                                                                                                                                                                                                                                                                                                     |
| Explanation:      | Parm option COM was found indicating COMPILE mode. For Compile Mode, a DD file INFILE is required to pass the original compile listing into DCD, and this DD name was not found.                                                                                                                                                                                                                                                                                                                     |
| User Action:      | See the discussion in the User's Manual under the heading 'Compile Mode Considerations' under 'Alternate Compile Listing'.                                                                                                                                                                                                                                                                                                                                                                           |
| <b>DCDA2002-D</b> | <b>DDNAME (DCDWK01) MISSING</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Explanation:      | The running of DCD III requires work files DCDWK01 and more. The DDNAME DCDWK01 was not found..                                                                                                                                                                                                                                                                                                                                                                                                      |
| User Action:      | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>DCDA2003-D</b> | <b>DDNAME (PRTCMPLR) MISSING FOR (COMPILE) MODE - IF INDEPENDENT MODE, DO NOT USE (COM) OPTION</b>                                                                                                                                                                                                                                                                                                                                                                                                   |
| Explanation:      | Parm option COM was found indicating COMPILE mode. For Compile Mode, a DD file PRTCMPLR is required to pass back some reports from the original compile listing , and this DD name was not found.                                                                                                                                                                                                                                                                                                    |
| User Action:      | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>DCDA2120-C</b> | <b>DCD Compile LINECNT (CLN=nn) DOES NOT MATCH (LINECNT=nn) FOUND IN (*OPTIONS IN EFFECT) IN COMPILE LISTING</b>                                                                                                                                                                                                                                                                                                                                                                                     |
| Explanation:      | The DCD III CLN=value (user entered) does not match the line count found within the compile listing.                                                                                                                                                                                                                                                                                                                                                                                                 |
| User Action:      | Use the CLN=PARM option along with the appropriate value. The error should be corrected by entering a CLN=field with a value equal to that shown within the LINECNT=field of the compile step.                                                                                                                                                                                                                                                                                                       |
| <b>DCDA2200-D</b> | <b>SYSPRINT FILE INTO DCD IS NOT COBOL OR IS INVALID. ENSURE THAT DISP=(MOD, PASS) IS USED ON SYSPRINT DD IN COMPILE STEP AND (NONUM) IS COMPILER OPTION (CHECK USE OF VS2/NOVS2 OPTION)</b>                                                                                                                                                                                                                                                                                                         |
| Explanation:      | The INFILE DD within the DD step is not successfully reading in the trapped compiler listing on the SYSPRINT DD from the compile step.                                                                                                                                                                                                                                                                                                                                                               |
| User Action:      | 1 .Consult "Examples" in the Alternate Compile Listing Facility section for correctly setting up the COB.SYSPRINT DD in the compile step.<br>a) Ensure that (MOD,PASS) is used as a disposition within the SYSPRINT DD.<br>b) Ensure that both NONUM and SOURCE are specified as compiler options.<br>c ) Ensure that the compile step ran successfully.<br>d) Ensure that NOVS2 is being defaulted to when using COBOL 68 or COBOL 74. Ensure that VS2 is used when VS COBOL II (COBOL 85) is used. |

2. Consult the COMACL PROC within the User's Manual to ensure the SYSPRINT DD is being used correctly.
3. If Independent Mode was intended, do not use the COM option or the COMACL PROC.

|                   |                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDA2P01-D</b> | <b>VS COBOL II LISTING FOUND, VS2 OPTION NOT IN EFFECT, USE PARM OPT ION (VS2) TO CORRECT</b>                                                                                                                                                                          |
| Explanation:      | PARM Option VS2 is required when VS COBOL II or new COBOL releases are used.                                                                                                                                                                                           |
| User Action:      | Use PARM Option VS2 (e.g. OTHER= 'VS')                                                                                                                                                                                                                                 |
| <b>DCDA4008-D</b> | <b>DDNAME (COBOLIN) MISSING FOR (INDEPENDENT) MODE - IF COMPILE MODE, USE (COM) PARM OPTION</b>                                                                                                                                                                        |
| Explanation:      | In the JCL, the COBOLIN DD card is missing from the file being compiled.                                                                                                                                                                                               |
| User Action:      | If in Independent Mode, check JCL to ensure DD card for COBOLIN is there and correctly placed.<br>If in Compile Mode, run DCDIII with the COM option. The COBOLIN DD is used for specifying the COBOL program to be run through DCDIII, unless the COM option is used. |
| <b>DCDA4009-D</b> | <b>DDNAME (DCDWK01) MISSING</b>                                                                                                                                                                                                                                        |
| Explanation:      | In the JCL, the DD card for DCDWK01 is missing.                                                                                                                                                                                                                        |
| User Action:      | Check the JCL to ensure the DD card for DCDWK01 exists and is correctly placed.                                                                                                                                                                                        |
| <b>DCDA4010-I</b> | <b>ERROR IN PROCESSING BASIS PROGRAMS</b>                                                                                                                                                                                                                              |
| Explanation:      | Informational message preceding BASIS error messages.                                                                                                                                                                                                                  |
| User Action:      | Examine output following error message to determine cause of problem.                                                                                                                                                                                                  |
| <b>DCDA4011-I</b> | <b>NORMAL END OF BASIS PROCESSING</b>                                                                                                                                                                                                                                  |
| Explanation:      | Informational message following the processing of cards from the BASIS DD file.                                                                                                                                                                                        |
| User Action:      | None.                                                                                                                                                                                                                                                                  |
| <b>DCDA4040-D</b> | <b>EMPTY FILE FOUND FOR COBOL PROGRAM</b>                                                                                                                                                                                                                              |
| Explanation:      | COBOL file exists but contains no records                                                                                                                                                                                                                              |
| User Action:      | Check to ensure COBOL file contains records.                                                                                                                                                                                                                           |
| <b>DCDA4B00-D</b> | <b>DDNAME (BASIS) MISSING AND IS REQUIRED WHEN PARM OPTION (BASIS) IS USED</b>                                                                                                                                                                                         |
| Explanation:      | In the JCL, the BASIS DD card is missing.                                                                                                                                                                                                                              |
| User Action:      | Check JCL to ensure the BASIS DD card exists and is correctly placed or remove BASIS as a PARM option.                                                                                                                                                                 |
| <b>DCDA4B02-D</b> | <b>NO COBOL INPUT FOUND ON DDNAME (COBOLIN)</b>                                                                                                                                                                                                                        |
| Explanation:      | COBOL file exists (used with BASIS option) but contains no records.                                                                                                                                                                                                    |
| User Action:      | Check to ensure COBOL file contains records.                                                                                                                                                                                                                           |
| <b>DCDA4B03-D</b> | <b>NO BASIS INPUT FOUND ON DDNAME (BASIS)</b>                                                                                                                                                                                                                          |
| Explanation:      | BASIS DD card was found but no records associated with that DD were found.                                                                                                                                                                                             |
| User Action:      | Check to ensure that the BASIS DD cards are present, formatted correctly and follow the BASIS DD.                                                                                                                                                                      |
| <b>DCDA4B04-D</b> | <b>FIRST CARD IN THE BASIS DD IS NOT (BASIS)</b>                                                                                                                                                                                                                       |
| Explanation:      | The first card found where the BASIS card should appear is not a BASIS card.                                                                                                                                                                                           |
| User Action:      | Check to ensure the BASIS card is correctly placed.                                                                                                                                                                                                                    |
| <b>DCDA4C01-D</b> | <b>BASIS (program name) PROGRAM NOT FOUND ON (COBOLIN) INPUT FILE</b>                                                                                                                                                                                                  |
| Explanation:      | The program name given in the BASIS DD card does not match the program found on the COBOLIN file.                                                                                                                                                                      |
| User Action:      | Check the BASIS DD card and ensure the program name is correct and matches the program pointed to by the COBOLIN file.                                                                                                                                                 |
| <b>DCDA4C02-D</b> | <b>CARD FOLLOWING (BASIS) NOT (INSERT) OR (DELETE)</b>                                                                                                                                                                                                                 |
| Explanation:      | The first card following the BASIS DD card was not an INSERT card or a DELETE card. It must be one of these.                                                                                                                                                           |
| User Action:      | Check to ensure that the INSERT / DELETE cards following the BASIS DD card are correctly placed and that an INSERT or DELETE card immediately follows the BASIS card.                                                                                                  |
| <b>DCDA4E01-D</b> | <b>INSERT (sequence #) OUT OF SEQUENCE</b>                                                                                                                                                                                                                             |
| Explanation:      | An INSERT card was found with a sequence number lower than a previous sequence number on an INSERT or DELETE                                                                                                                                                           |
| User Action:      | Check to ensure the INSERT / DELETE cards are in proper sequence and the sequence numbers are in ascending order.                                                                                                                                                      |



|                   |                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDA4E02-D</b> | <b>INSERT (sequence #) NOT FOUND IN PROGRAM</b>                                                                                                                                       |
| Explanation:      | An INSERT card was found with a sequence number that does not exist in the main program.                                                                                              |
| User Action:      | Check INSERT cards and make sure all sequence numbers are valid and match sequence numbers in the main program.                                                                       |
| <b>DCDA4E03-D</b> | <b>INSERT (sequence #) - NO INSERTS FOUND</b>                                                                                                                                         |
| Explanation:      | An INSERT was found with no COBOL code following it. One line of code must follow an INSERT card.                                                                                     |
| User Action:      | Check INSERT cards to ensure they are properly placed and at least 1 line of COBOL code follows each.                                                                                 |
| <b>DCDA4F01-D</b> | <b>DELETE (sequence #) OUT OF SEQUENCE</b>                                                                                                                                            |
| Explanation:      | A DELETE card was found with a sequence number lower than a previous sequence number on an INSERT or DELETE card.                                                                     |
| User Action:      | Check to ensure INSERT / DELETE cards are in proper sequence and are in ascending order.                                                                                              |
| <b>DCDA4G02-D</b> | <b>(DELETE) FIRST NUMBER IN RANGE (sequence #) NOT FOUND</b>                                                                                                                          |
| Explanation:      | The first sequence number in a DELETE range does not match a valid sequence number in the main program.                                                                               |
| User Action:      | Check to ensure all sequence numbers in DELETE range match valid sequence numbers in the program.                                                                                     |
| <b>DCDA4G04-D</b> | <b>(DELETE) RANGE (1st sequence #) - (2nd sequence #) NOT FOUND</b>                                                                                                                   |
| Explanation:      | The sequence numbers found in a DELETE card do not match valid sequence numbers in the main program.                                                                                  |
| User Action:      | Check sequence numbers on DELETE cards and ensure they match valid sequence numbers in program.                                                                                       |
| <b>DCDA4SA0-I</b> | <b>HANDLING BASIS CARD: basis card</b>                                                                                                                                                |
| Explanation:      | An informational message to assure the user that the BASIS card is being processed by DCDIII.                                                                                         |
| User Action:      | None.                                                                                                                                                                                 |
| <b>DCDB0015-D</b> | <b>DDNAME (COBOLIN) MISSING</b>                                                                                                                                                       |
| Explanation:      | In the JCL, the DD card for COBOLIN is missing.                                                                                                                                       |
| User Action:      | Check the JCL to ensure DD card for COBOLIN exists and is correctly placed.                                                                                                           |
| <b>DCDB0016-D</b> | <b>DDNAME (DCDWK01) MISSING</b>                                                                                                                                                       |
| Explanation:      | In the JCL, the DD card for DCDWK01 is missing.                                                                                                                                       |
| User Action:      | Check the JCL to ensure DD card for DCDWK01 exists and is correctly placed.                                                                                                           |
| <b>DCDB0022-D</b> | <b>DCDMULTI (program name) PROGRAM NOT FOUND ON (COBOLIN) INPUT FILE</b>                                                                                                              |
| Explanation:      | A logic error has occurred. Program member does not match the COBOLIN file.                                                                                                           |
| User Action:      | Check the JCL to ensure DD card member exists and is correctly placed.                                                                                                                |
| <b>DCDB0030-D</b> | <b>EMPTY FILE FOUND FOR COBOL PROGRAM</b>                                                                                                                                             |
| Explanation:      | COBOL file exists but contains no records.                                                                                                                                            |
| User Action:      | Check to ensure COBOL file contains records.                                                                                                                                          |
| <b>DCDB0040-E</b> | <b>( TRACE ) OPTION available for first program found – Ignored for EXTRA programs</b>                                                                                                |
| Explanation:      | Tracing and Analysis is only available for one program at a time. If more programs were inputted, then the extra programs will not have Tracing and Analysis done on them.            |
| User Action:      | Limit Tracing and Analysis to one program at a time.                                                                                                                                  |
| <b>DCDB0041-E</b> | <b>( DIT ) OPTION available for first program found – Ignored for EXTRA programs</b>                                                                                                  |
| Explanation:      | Direct Tracing of Literals is only available for one program at a time. If more programs were inputted, then the extra programs will not have Direct Tracing of Literals done on them |
| User Action:      | Limit Direct Tracing of Literals to one program at a time.                                                                                                                            |
| <b>DCDB0C01-W</b> | <b>START OF ASSEMBLER PROGRAM FOUND WITHIN PROGRAM</b>                                                                                                                                |
| Explanation:      | The start of an Assembler program was found when reading in COBOL programs.                                                                                                           |
| User Action:      | Review the input and ensure that only COBOL programs are being inputted.                                                                                                              |
| <b>DCDB0D02-W</b> | <b>COLUMN 7 OF LINE AFTER THE PROCEDURE DIVISION OF COBOL PROGRAM (XXXXXXXX) IS INVALID - SEE NEXT LINE</b>                                                                           |
| Explanation:      | Within the Procedure Division, a line was found which may be invalid.                                                                                                                 |
| User Action:      | Examine the line and take corrective action if necessary.                                                                                                                             |
| <b>DCDB0F01-D</b> | <b>USE OF PARM OPTION (LOR) REQUIRES COBOL RECS - FOUND COBOL PROGRAM INSTEAD</b>                                                                                                     |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | A COBOL program was found as input when the PARM option LOR indicated Data Division records were being inputted.                                                                                     |
| User Action:      | Review what was intended and correct.                                                                                                                                                                |
| <b>DCDB0XA6-C</b> | <b>COBOL PROGRAM (program name) WAS DUPLICATED</b>                                                                                                                                                   |
| Explanation:      | More than one COBOL program was found with this program name.                                                                                                                                        |
| User Action:      | Determine why duplicate programs are present and remove those that are not wanted.                                                                                                                   |
| <b>DCDB0XA7-I</b> | <b>COBOL PROGRAM-ID (program) BEING PROCESSED - (nnnnn)</b>                                                                                                                                          |
| Explanation:      | Informational message produced for every COBOL program.                                                                                                                                              |
| User Action:      | None.                                                                                                                                                                                                |
| <b>DCDB0XA9-W</b> | <b>INVALID WORD IN PROCEDURE DIVISION</b>                                                                                                                                                            |
| Explanation:      | Found a PIC, PICTURE or VALUE clause in the Procedure Division.                                                                                                                                      |
| User Action:      | Check the program to find out why this condition occurred.                                                                                                                                           |
| <b>DCDCA010-D</b> | <b>DDNAME (DCDWK02) MISSING</b>                                                                                                                                                                      |
| Explanation:      | The running of DCD III requires work files DCDWK02 and others. The DDNAME DCDWK02 was not found.                                                                                                     |
| User Action:      | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                                                                                    |
| <b>DCDCA030-D</b> | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                                                 |
| Explanation:      | A work file being used is in a corrupted state.                                                                                                                                                      |
| User Action:      | Try re-running. If necessary, contact Marble Computer.                                                                                                                                               |
| <b>DCDCADD3-E</b> | <b>OVER 500 REPLACING TOKENS WITHIN ONE COPY REPLACING OPERAND - CHECK FOR ERROR IN CODE</b>                                                                                                         |
| Explanation:      | More than 500 tokens were found when replacing the corresponding tokens in the COPY member before finding a delimiter within pseudo-text-1.                                                          |
| User Action:      | Check the COPY member and the COPY REPLACING clauses to determine why excessive tokens are being replaced within one pseudo text. If this is a valid condition, contact MARBLE Computer, Inc.        |
| <b>DCDCADD4-I</b> | <b>(BY) MISSING IN COPY REPLACING</b>                                                                                                                                                                |
| Explanation:      | The COPY REPLACING statement at the compiler number shown is missing the word BY following identifier-1 and before identifier-2.                                                                     |
| User Action:      | Inspect the COPY REPLACING clauses within the COBOL program and ensure the word BY is correctly placed.                                                                                              |
| <b>DCDCADD6-C</b> | <b>OVER 500 REPLACING TOKENS WITHIN COPY REPLACING OPERAND - CHECK FOR ERROR IN CODE</b>                                                                                                             |
| Explanation:      | More than 500 tokens were found when replacing the corresponding tokens in the COPY member before finding a delimiter within pseudo-text-2.                                                          |
| User Action:      | Check the COPY member and the COPY REPLACING clause to determine why excessive tokens are being replaced within one pseudo text. If this is a valid condition, contact MARBLE Computer, Inc.         |
| <b>DCDCADD9-C</b> | <b>OVER 150 REPLACING WITHIN ONE COPY</b>                                                                                                                                                            |
| Explanation:      | DCD III has a limit of 150 REPLACING entries within one COPY statement.                                                                                                                              |
| User Action:      | Inspect the COPY statements to determine the cause of excessive REPLACING clauses and ensure there are less than 150 per COPY statement. If this is a valid condition, contact MARBLE Computer, Inc. |
| <b>DCDCAH03-E</b> | <b>INTERNAL TABLE EXCEEDED</b>                                                                                                                                                                       |
| Explanation:      | An internal table was exceeded when expanding COPY members.                                                                                                                                          |
| User Action:      | Contact Marble Computer.                                                                                                                                                                             |
| <b>DCDCAH06-E</b> | <b>DATA SET NAME FOR COPYLIB NOT SPECIFIED, RUN TERMINATED</b>                                                                                                                                       |
| Explanation:      | A data set name (DSNAME=) was not provided on the //COPYLIB DD card.                                                                                                                                 |
| User Action:      | Ensure that a DSN=clause is provided on the COPYLIB DD statement and check correct placement of the //COPYLIB card.                                                                                  |
| <b>DCDCAX01-W</b> | <b>COPY REPLACING OPTION IGNORED FOR NESTED COPIES (member-name)</b>                                                                                                                                 |
| Explanation:      | COPY REPLACING is not allowed within NESTED COPYs.                                                                                                                                                   |
| User Action:      | Determine why COPY REPLACING is used in NESTED COPYs and take any appropriate action.                                                                                                                |
| <b>DCDCAX02-E</b> | <b>COPY MEMBER NAME (member-name) APPEARS EARLIER WITHIN THIS ENTIRE NESTED COPY</b>                                                                                                                 |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | Within nested COPYs, a COPY may not name a member already brought in through nested copies above it. The member-name shown is already brought in. If brought in again, it will result in generating endless lines of code. |
| User Action:      | Examine why this is happening and correct.                                                                                                                                                                                 |
| <b>DCDCAX05-I</b> | <b>ERROR WHEN RESOLVING COPY MEMBER (member-name) SEE FOLLOWING MESSAGE</b>                                                                                                                                                |
| Explanation:      | DCD III is unable to resolve the COPY member listed above.                                                                                                                                                                 |
| User Action:      | Look for another (more specific) message following this one.                                                                                                                                                               |
| <b>DCDCAX06-E</b> | <b>COPY MEMBER (member-name) NOT FOUND IN COPYLIB SPECIFIED</b>                                                                                                                                                            |
| Explanation:      | When DCD III attempted to read the COPY member in the COBOL program, it was not found in the partitioned data set specified by DSN= in the COPYLIB DD.                                                                     |
| User Action:      | Check the COPY member name to ensure it is valid and that the correct PDS file is specified within the DSN= in the COPYLIB DD.                                                                                             |
| <b>DCDCAX07-D</b> | <b>//COPYLIB DD IS MISSING OR INCORRECT</b>                                                                                                                                                                                |
| Explanation:      | A data set name (DSNAME=) was not provided on the //COPYLIB DD card.                                                                                                                                                       |
| User Action:      | Ensure that a DSN= clause is provided on the COPYLIB DD statement and check to be sure the placement of the //COPYLIB card is correct.                                                                                     |
| <b>DCDCAX08-E</b> | <b>UNSUCCESSFUL READ WHILE READING RECORD FROM COPY MEMBER (member-name)</b>                                                                                                                                               |
| Explanation:      | The COPY member shown was found within the directory of the PDS specified, however, attempts to read it are unsuccessful.                                                                                                  |
| User Action:      | Ensure that a valid partitioned data set was provided in the COPYLIB DD statement and verify that the member specified is there and can be read successfully.                                                              |
| <b>DCDCAX09-E</b> | <b>UNSUCCESSFUL READ WHEN READING DIRECTORY OF DSN SPECIFIED IN //COPYLIB DD</b>                                                                                                                                           |
| Explanation:      | A problem was encountered when reading the directory of the partitioned data set provided in the DSN= of the COPYLIB DD.                                                                                                   |
| User Action:      | Ensure the PDS specified in the DSN= field is a valid PDS and that the directory is able to be read.                                                                                                                       |
| <b>DCDCAX10-E</b> | <b>NO RECORDS FOUND IN MEMBER (copy member name)</b>                                                                                                                                                                       |
| Explanation:      | When DCD III attempted to resolve this COPY member, it found no records in the member.                                                                                                                                     |
| User Action:      | Check the COPY member and make corrections as necessary.                                                                                                                                                                   |
| <b>DCDCCT01-I</b> | <b>WHEN HANDLING REPLACING WITHIN A COPY, A CONTINUATION TO A NEW LINE IS FORCED AT COMPILER # (compiler #)</b>                                                                                                            |
| Explanation:      | An informational message indicating that a new COBOL line was created to adequately handle the REPLACING on this line.                                                                                                     |
| User Action:      | None.                                                                                                                                                                                                                      |
| <b>DCDCCY06-I</b> | <b>ERROR WHEN RESOLVING COPY MEMBER (copy-member-name)</b>                                                                                                                                                                 |
| Explanation:      | Indicates that an error has occurred in resolving a COPY member.                                                                                                                                                           |
| User Action:      | Look for an associated error message more descriptive in nature than this one.                                                                                                                                             |
| <b>DCDCCY07-E</b> | <b>COPY MEMBER (copy-member) NOT FOUND IN COPYLIB SPECIFIED</b>                                                                                                                                                            |
| Explanation:      | The COPY member name specified was not found in the data sets provided within the COPYLIB DD.                                                                                                                              |
| User Action:      | Either correct the spelling of the member name or determine why the member is missing.                                                                                                                                     |
| <b>DCDCCY08-D</b> | <b>//COPYLIB DD IS MISSING OR INCORRECT</b>                                                                                                                                                                                |
| Explanation:      | A DSN was not found in the JCL on the COPYLIB DD statement or the DD is missing.                                                                                                                                           |
| User Action:      | Correct the JCL and rerun.                                                                                                                                                                                                 |
| <b>DCDCCY09-E</b> | <b>UNSUCCESSFUL READ WHILE READING RECORD FROM COPY MEMBER (copy-member)</b>                                                                                                                                               |
| Explanation:      | An error occurred during the read of a record of the COPY member shown within the record.                                                                                                                                  |
| User Action:      | Determine if the COPY member in question has an I/O problem that can be easily fixed. If so, correct, otherwise, contact System Support at your installation.                                                              |
| <b>DCDCCY0A-E</b> | <b>UNSUCCESSFUL READ WHEN READING DIRECTORY OF DSN SPECIFIED IN //COPYLIB DD</b>                                                                                                                                           |
| Explanation:      | An error occurred during the read of a directory of the data set specified within the COPYLIB DD.                                                                                                                          |
| User Action:      | Contact System Support at your installation for help.                                                                                                                                                                      |
| <b>DCDCCY0B-E</b> | <b>NO RECORDS FOUND IN COPY MEMBER (copy member name)</b>                                                                                                                                                                  |
| Explanation:      | When DCD III attempted to resolve this COPY member, it found no records in the member.                                                                                                                                     |
| User Action:      | Check the COPY member and make corrections as necessary.                                                                                                                                                                   |

|                                                                                  |                                                                                                                                                                 |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDCFE01-E</b>                                                                | <b>PROBLEM IN RESOLVING NESTED COPY MEMBER (member name) - CONTINUING...</b>                                                                                    |
| Explanation:                                                                     | When DCD III attempted to resolve this COPY member, it could not find the member specified.                                                                     |
| User Action:                                                                     | Check to see why the COPY member is missing.                                                                                                                    |
| <b>DCDCR020-E</b>                                                                | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                            |
| Explanation:                                                                     | The first record of a work file being read does not contain a MARBLE header record.                                                                             |
| User Action:                                                                     | Look for possible previous messages to this one. Look for an error in the JCL used.                                                                             |
| <b>DCDEB040-D</b>                                                                | <b>ENTIRE COBOL PROGRAM INCLUDING PROCEDURE DIVISION WAS NOT FOUND</b>                                                                                          |
| Explanation:                                                                     | The PROCEDURE DIVISION was not found within this program.                                                                                                       |
| User Action:                                                                     | Determine why PROCEDURE DIVISION was not found and re-submit.                                                                                                   |
| <b>DCDEBM02-D</b>                                                                | <b>EXCEEDED MAXIMUM NUMBER OF COPYS THAT CAN BE HANDLED - CONTACT MARBLE COMPUTER</b>                                                                           |
| Explanation:                                                                     | A large internal table was exceeded. This is an unusual exception.                                                                                              |
| User Action:                                                                     | Contact Marble Computer.                                                                                                                                        |
| <b>DCDEF016-D<br/>&amp; DCDEF017-D<br/>&amp; DCDEF018-D<br/>&amp; DCDEF019-D</b> | <b>DDNAME (DCDWK0x) MISSING</b>                                                                                                                                 |
| Explanation:                                                                     | The running of DCD III requires work files for processing. The DDNAMEs DCDWK03, DCDWK04, DCDWK05, and DCDWK06 respectively were not found.                      |
| User Action:                                                                     | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                                               |
| <b>DCDEF030-C</b>                                                                | <b>PROGRAM-ID NOT FOUND IN INPUT FILE - LOOK FOR NON-COBOL FILE OR MISSING PROGRAM-ID IN MARGIN-A</b>                                                           |
| Explanation:                                                                     | The PROGRAM-ID clause with Program-id within the Identification Division was not found. Program-Id is used within internal processing of DCD III and is needed. |
| User Action:                                                                     | Check the program and see why this occurred.                                                                                                                    |
| <b>DCDEFB01-W</b>                                                                | <b>NO PERIOD PRECEDING LEVEL NUMBER (nn)</b>                                                                                                                    |
| Explanation:                                                                     | A period did not immediately precede this level number.                                                                                                         |
| User Action:                                                                     | Check the previous line for a missing period.                                                                                                                   |
| <b>DCDEFB02-C</b>                                                                | <b>DATA NAME DOES NOT FOLLOW LEVEL NUMBER (nn)</b>                                                                                                              |
| Explanation:                                                                     | A level number was found and an expected data name following level number was not found.                                                                        |
| User Action:                                                                     | Check the line and make corrections as necessary.                                                                                                               |
| <b>DCDEFB03-C</b>                                                                | <b>(01) FOUND IN MARGIN B - IGNORING</b>                                                                                                                        |
| Explanation:                                                                     | The 01 level number found is within margin B, in column 12 or beyond. This 01 is being ignored as being a valid 01 level number.                                |
| Action:                                                                          | Check the line and make corrections if this is really a 01 level number.                                                                                        |
| <b>DCDEFB66-W</b>                                                                | <b>NO PERIOD PRECEDING LEVEL NUMBER (66)</b>                                                                                                                    |
| Explanation:                                                                     | A period did not immediately precede this level number.                                                                                                         |
| Action:                                                                          | Check the previous line for a missing period.                                                                                                                   |
| <b>DCDEFB67-C</b>                                                                | <b>DATA-NAME DOES NOT FOLLOW 66 LEVEL - BYPASSING</b>                                                                                                           |
| Explanation:                                                                     | Level 66 found with no data name following it.                                                                                                                  |
| Action:                                                                          | Check the line and make corrections if this is really a 66 level number.                                                                                        |
| <b>DCDEFB68-C</b>                                                                | <b>RENAMES EXPECTED WITHIN 66 LEVEL AND NOT FOUND - BYPASSING</b>                                                                                               |
| Explanation:                                                                     | The syntax of 66 (RENAME) statement requires a data name to follow RENAMES word.                                                                                |
| Action:                                                                          | Check the syntax of 66 level line.                                                                                                                              |
| <b>DCDEFB69-C</b>                                                                | <b>TOKEN PAST RENAMES NOT ALPHANUMERIC</b>                                                                                                                      |
| Explanation:                                                                     | The 66 level clause at the compiler number shown contains data following the COBOL verb, RENAMES that is not alphanumeric.                                      |
| User Action:                                                                     | Check the line and make corrections as necessary.                                                                                                               |
| <b>DCDEFB77-W</b>                                                                | <b>NO PERIOD PRECEDING LEVEL NUMBER (level #)</b>                                                                                                               |
| Explanation:                                                                     | The 77 level number at the compiler number shown is not preceded by a period.                                                                                   |
| User Action:                                                                     | Check the line preceding the level number for a missing period and correct.                                                                                     |

## DCD III – Error Messages

---

|                   |                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDEFB78-C</b> | <b>DATA NAME DOES NOT FOLLOW LEVEL NUMBER (level #) - BYPASSING</b>                                                                       |
| Explanation:      | The 77 level number at the compiler number shown is not followed by a data name.                                                          |
| User Action:      | Check the line and make corrections as necessary.                                                                                         |
| <b>DCDEFB79-W</b> | <b>(77) FOUND IN MARGIN B – ACCEPTED</b>                                                                                                  |
| Explanation:      | The 77 level number found at the compiler number shown is in margin B. It should be to the left of margin B.                              |
| User Action:      | Move the level number to the left of margin B or check the line and make corrections as necessary.                                        |
| <b>DCDEFB88-W</b> | <b>NO PERIOD PRECEDING 88 LEVEL NUMBER ( level #) - ACCEPTING</b>                                                                         |
| Explanation:      | The 88 level number at the compiler number shown is not preceded by a period.                                                             |
| User Action:      | Check the line preceding the level number for a missing period and correct.                                                               |
| <b>DCDEFB89-W</b> | <b>DATA NAME DOES NOT FOLLOW 88 LEVEL NUMBER (level #) - BYPASSING</b>                                                                    |
| Explanation:      | The 88 level number at the compiler number shown is not followed by a data name.                                                          |
| User Action:      | Check the line and make corrections as necessary.                                                                                         |
| <b>DCDEFB90-W</b> | <b>(88) FOUND IN MARGIN A – ACCEPTED</b>                                                                                                  |
| Explanation:      | The 88 level number found at the compiler number shown is in margin A. It should be to the right of margin A.                             |
| User Action:      | Move the level number to margin B or check the line and make corrections as necessary.                                                    |
| <b>DCDEFB92-E</b> | <b>VALUE CLAUSE MISSING IN 88 LEVEL - BYPASSING</b>                                                                                       |
| Explanation:      | The 88 level number and clause at the compiler number shown must contain the word VALUE.                                                  |
| User Action:      | Check the line and make corrections as necessary.                                                                                         |
| <b>DCDEFB93-E</b> | <b>CANNOT FIND LITERAL AFTER (VALUE) - BYPASSING</b>                                                                                      |
| Explanation:      | The 88 level number and clause at the compiler number shown must contain a literal following the word VALUE.                              |
| User Action:      | Check the line and make corrections as necessary.                                                                                         |
| <b>DCDEFG23-E</b> | <b>PROBLEM IN RESOLVING OCCURS AT COMPILER # (compiler #) **SEE BELOW**</b>                                                               |
|                   | <b>LEVEL NUMBER = (level #)                      PRIOR END LOCATION = (location)</b>                                                      |
|                   | <b>NUMBER OF OCCURS = (nnn)                      CURRENT END LOCATION = (location)</b>                                                    |
| Explanation:      | An error was found in resolving an OCCURS clause. More information will be displayed underneath this message.                             |
| User Action:      | Look for a syntax error. If none can be found, contact MARBLE Computer, Inc. for support.                                                 |
| <b>DCDEFG32-C</b> | <b>GROUP LEVEL NAME BYPASSED FOR PURPOSES OF CALCULATING RECORD POSITIONS DUE TO SYNTAX ERROR</b>                                         |
| Explanation:      | A problem was encountered in resolving the record positions for a group item. More information will be displayed underneath this message. |
| User Action:      | Look for a syntax error. If none can be found, contact MARBLE Computer, Inc. for support.                                                 |
| <b>DCDEH01A-E</b> | <b>(SYNTAX) ERROR - (MOVE)</b>                                                                                                            |
| Explanation:      | The MOVE statement found at the compiler number shown has the reserved word TO missing or misplaced.                                      |
| User Action:      | Look at the line in question and make any necessary corrections.                                                                          |
| <b>DCDEH01B-C</b> | <b>THE END OF A (MOVE) STATEMENT IS INCOMPLETE OR CONTAINS AN UNRECOGNIZED SPECIAL REGISTER</b>                                           |
| Explanation:      | The MOVE statement found at the compiler number is incomplete.                                                                            |
| User Action:      | Look at the line in question and make any necessary corrections.                                                                          |
| <b>DCDEH03A-C</b> | <b>FILENAME FOLLOWING A (READ) VERB IS NOT ALPHANUMERIC – ACCEPTED AS IS</b>                                                              |
| Explanation:      | A valid file name following the COBOL verb READ is expected at the compiler number shown.                                                 |
|                   | The file name found is in question because it is not alphanumeric.                                                                        |
| User Action:      | Check the file name and correct as necessary.                                                                                             |
| <b>DCDEH03B-E</b> | <b>NO FD ENTRIES FOUND FOR THE FILENAME (FD name)</b>                                                                                     |
| Explanation:      | DCD III has no FILE FDs in its internal table to match to the file name found in the READ statement at the compiler number shown.         |
| User Action:      | Check spelling and if necessary, check the Environment and Data Divisions for a valid FD or SD to match the name used.                    |
| <b>DCDEH03C-E</b> | <b>NO MATCHING FILE-NAME FOR (filename)</b>                                                                                               |

|                   |                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | The file name found after the READ statement at the compiler number shown does not match to a FD file name within the Data Division.         |
| User Action:      | Check the file name and correct to match a file name in the Environment and Data Divisions.                                                  |
| <b>DCDEH03D-C</b> | <b>(INTO IDENTIFIER-1) IDENTIFIER-1 NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                    |
| Explanation:      | The record name following the word INTO in a READ statement is not alphanumeric at the compiler number shown.                                |
| User Action:      | Check the record name and correct as necessary.                                                                                              |
| <b>DCDEH04A-C</b> | <b>NAME FOLLOWING (WRITE) IS NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                           |
| Explanation:      | The record name following the verb WRITE is not alphanumeric at the compiler number shown.                                                   |
| User Action:      | Check the record name and correct as necessary.                                                                                              |
| <b>DCDEH04B-C</b> | <b>(FROM IDENTIFIER-1) IDENTIFIER -1 NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                   |
| Explanation:      | The identifier that follows the word FROM in a WRITE statement is not alphanumeric at the compiler number shown.                             |
| User Action:      | Check the identifier and make corrections as necessary.                                                                                      |
| <b>DCDEH05A-C</b> | <b>IDENTIFIER FOLLOWING (ACCEPT) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                       |
| Explanation:      | The identifier that follows the COBOL verb ACCEPT at the compiler number shown is not alphanumeric.                                          |
| User Action:      | Check the identifier and make corrections as necessary.                                                                                      |
| <b>DCDEH06A-E</b> | <b>THE (ADD) STATEMENT REQUIRES A (TO) OR (GIVING) AND NEITHER WAS FOUND – BYPASSING</b>                                                     |
| Explanation:      | The ADD statement at the compile number shown is missing the word, TO or GIVING.                                                             |
| User Action:      | Check the ADD statement and correct as necessary.                                                                                            |
| <b>DCDEH06B-E</b> | <b>THE END OF THE (ADD) STATEMENT IS INCOMPLETE OR CONTAINS AN UNRECOGNIZED SPECIAL REGISTER - BYPASSING</b>                                 |
| Explanation:      | The ADD statement at the compiler number shown is incomplete.                                                                                |
| User Action:      | Check the ADD statement and correct as necessary.                                                                                            |
| <b>DCDEH07A-E</b> | <b>THE (SUBTRACT) STATEMENT HERE REQUIRES THE WORD (FROM) AND IT WAS NOT FOUND – BYPASSING</b>                                               |
| Explanation:      | The SUBTRACT statement at the compiler number shown requires the word FROM to ensure correct syntax.                                         |
| User Action:      | Check the SUBTRACT statement and correct as necessary.                                                                                       |
| <b>DCDEH07B-E</b> | <b>THE END OF THIS (SUBTRACT) STATEMENT IS INCOMPLETE OR CONTAINS AN UNRECOGNIZED SPECIAL REGISTER - BYPASSING</b>                           |
| Explanation:      | The SUBTRACT statement at the compiler number shown appears to be incomplete.                                                                |
| User Action:      | Check the SUBTRACT statement and correct as necessary.                                                                                       |
| <b>DCDEH08A-E</b> | <b>THE (MULTIPLY) STATEMENT REQUIRES (BY) - BYPASSING</b>                                                                                    |
| Explanation:      | The MULTIPLY statement at the compiler number shown is missing the word BY.                                                                  |
| User Action:      | Check the MULTIPLY statement and correct as necessary.                                                                                       |
| <b>DCDEH08B-E</b> | <b>THE (MULTIPLY) STATEMENT IS INCOMPLETE - BYPASSING</b>                                                                                    |
| Explanation:      | The MULTIPLY statement at the compiler number shown is incomplete.                                                                           |
| User Action:      | Check the MULTIPLY statement and correct as necessary.                                                                                       |
| <b>DCDEH09A-E</b> | <b>THE (DIVIDE) STATEMENT REQUIRES (BY) OR (INTO) - BYPASSING</b>                                                                            |
| Explanation:      | The DIVIDE statement at the compiler number shown requires the word BY or INTO to ensure correct syntax.                                     |
| User Action:      | Check the DIVIDE statement and make corrections as necessary.                                                                                |
| <b>DCDEH09B-E</b> | <b>THE (DIVIDE) STATEMENT IS INCOMPLETE - BYPASSING</b>                                                                                      |
| Explanation:      | The DIVIDE statement at the compiler number shown is incomplete.                                                                             |
| User Action:      | Check the DIVIDE statement and make corrections as necessary.                                                                                |
| <b>DCDEH10A-E</b> | <b>THE (COMPUTE) STATEMENT REQUIRES (=) AND THIS WAS NOT FOUND – BYPASSING</b>                                                               |
| Explanation:      | The COMPUTE statement at the compiler number shown is incomplete and requires the symbol (=) or the word EQUAL or EQUALS for correct syntax. |
| User Action:      | Check the COMPUTE statement and correct.                                                                                                     |

|                                        |                                                                                                                                            |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDEH10B-E</b>                      | <b>THE (COMPUTE) STATEMENT IS INCOMPLETE - BYPASSING</b>                                                                                   |
| Explanation:                           | The COMPUTE statement at the compiler number shown is incomplete.                                                                          |
| User Action:                           | Check the COMPUTE statement and correct.                                                                                                   |
| <b>DCDEH11A-C</b>                      | <b>FILE-NAME AFTER CLOSE (file-name) NOT ALPHANUMERIC - ACCEPTED</b>                                                                       |
| Explanation:                           | The file name following the COBOL verb CLOSE at the compiler number shown is not alphanumeric.                                             |
| User Action:                           | Check the CLOSE statement and make corrections as necessary.                                                                               |
| <b>DCDEH11B-C</b>                      | <b>FILE -NAME AFTER CLOSE (file-name) NOT ALPHANUMERIC - ACCEPTED</b>                                                                      |
| Explanation:                           | The file name following the COBOL verb CLOSE at the compiler number shown is not alphanumeric.                                             |
| User Action:                           | Check the CLOSE statement and make corrections as necessary.                                                                               |
| <b>DCDEH12A-C</b>                      | <b>FILE-NAME AFTER DELETE (file-name) NOT ALPHANUMERIC - ACCEPTED</b>                                                                      |
| Explanation:                           | The file name following the COBOL verb DELETE at the compiler number shown is not alphanumeric.                                            |
| User Action:                           | Check the DELETE statement and make corrections as necessary.                                                                              |
| <b>DCDEH14A-C</b>                      | <b>IDENTIFIER AFTER EXAMINE (field-contents) NOT ALPHANUMERIC - ACCEPTED</b>                                                               |
| Explanation:                           | The field name found after the COBOL verb EXAMINE at the compiler number shown is not alphanumeric.                                        |
| User Action:                           | Check the EXAMINE statement and correct as necessary.                                                                                      |
| <b>DCDEH14B-C</b>                      | <b>(EXAMINE) FOUND (field-contents) FOR (TALLYING) OR (REPLACING) - BYPASSED</b>                                                           |
| Explanation:                           | The EXAMINE statement at the compiler number shown is missing the word TALLYING, REPLACING or is incorrect.                                |
| User Action:                           | Check the EXAMINE statement and correct as necessary.                                                                                      |
| <b>DCDEH14C-C</b>                      | <b>EXAMINE WITH JUST REPLACING REQUIRES (BY); FOUND (field-contents) BYPASSING</b>                                                         |
| Explanation:                           | The EXAMINE statement at the compiler number shown is missing the word BY.                                                                 |
| User Action:                           | Check the EXAMINE statement and correct as necessary.                                                                                      |
| <b>DCDEH15A-C</b>                      | <b>(EXHIBIT) REQUIRES (NAMED) OR (CHANGED) - ACCEPTED AS IS</b>                                                                            |
| Explanation:                           | The EXHIBIT statement at the compiler number shown requires the word NAMED, CHANGED or both to ensure correct syntax.                      |
| User Action:                           | Check the EXHIBIT statement and correct as necessary.                                                                                      |
| <b>DCDEH22A-C</b>                      | <b>THE SYNTAX FOR AN OPEN STATEMENT IS INCOMPLETE</b>                                                                                      |
| Explanation:                           | The syntax found for the OPEN statement at this compiler line number is invalid.                                                           |
| User Action:                           | Look at the OPEN statement and determine the cause of the error.                                                                           |
| <b>DCDEH22B-C</b>                      | <b>EXPECTING INPUT, OUTPUT, I-O OR EXTEND - FOUND (field-contents) -TREATED AS (OUTPUT)</b>                                                |
| Explanation:                           | The field shown was found after the verb OPEN.                                                                                             |
| User Action:                           | If this is correct, ignore or contact MARBLE Computer, Inc. If not, correct as necessary.                                                  |
| <b>DCDEH22C-C<br/>&amp; DCDEH22D-C</b> | <b>THE FILE-NAME (file-name) IS NOT ALPHANUMERIC – ACCEPTED AS IS</b>                                                                      |
| Explanation:                           | The file name found after the COBOL verb OPEN at the compiler number shown is not alphanumeric.                                            |
| User Action:                           | Check the OPEN statement and correct as necessary.                                                                                         |
| <b>DCDEH23A-E</b>                      | <b>THE FORMAT OF THIS (SET) STATEMENT IS UNRECOGNIZED - BYPASSING</b>                                                                      |
| Explanation:                           | Two forms of the SET statement are recognized: (SET index-name TO ?) and (SET index-name UP [or DOWN] BY ?).                               |
| User Action:                           | Correct this SET statement as necessary.                                                                                                   |
| <b>DCDEH24A-C</b>                      | <b>IDENTIFIER AFTER (INSPECT) - (field-contents) NOT ALPHANUMERIC - ACCEPTED</b>                                                           |
| Explanation:                           | The identifier found after the COBOL verb INSPECT at the compiler number shown is not alphanumeric.                                        |
| User Action:                           | Check the INSPECT statement and correct as necessary.                                                                                      |
| <b>DCDEH24B-C</b>                      | <b>EXPECTING (TALLYING) OR (REPLACING) - FOUND (field-contents) – BYPASSING</b>                                                            |
| Explanation:                           | At the compiler number shown, the item found after the first identifier in the INSPECT statement is not REPLACING, TALLYING or CONVERTING. |
| User Action:                           | Check the INSPECT statement and correct as necessary.                                                                                      |
| <b>DCDEH24C-C</b>                      | <b>IDENTIFIER AFTER INSPECT TALLYING (field-contents) NOT ALPHANUMERIC - ACCEPTED</b>                                                      |

|                   |                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | In the INSPECT statement at the compiler number shown, the identifier following the word TALLYING was not alphanumeric.                          |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24D-C</b> | <b>(FOR) MISSING IN INSPECT TALLYING, FOUND (field-contents) - RESULTS UNPREDICTABLE</b>                                                         |
| Explanation:      | The INSPECT statement at the compiler number shown is missing the word FOR following the second identifier.                                      |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24E-C</b> | <b>EXPECTING (CHARACTERS), (ALL) OR (LEADING) FOUND (field-contents)</b>                                                                         |
| Explanation:      | The INSPECT statement at the compiler number shown is missing the word ALL, CHARACTERS or LEADING following the word FOR.                        |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24F-C</b> | <b>EXPECTING ALL, LEADING, FIRST, OR CHARACTERS; FOUND (field-contents)</b>                                                                      |
| Explanation:      | The INSPECT . . . REPLACING statement at the compiler number shown is missing a COBOL word.                                                      |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24G-C</b> | <b>INSPECT WITH (CONVERTING) REQUIRES (TO); FOUND (field-contents)</b>                                                                           |
| Explanation:      | The INSPECT . . . CONVERTING statement found at the compiler number shown requires the word TO following the second identifier.                  |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24H-C</b> | <b>EXPECTING (BY) AFTER (CHARACTERS) - FOUND (field-contents)</b>                                                                                |
| Explanation:      | The INSPECT . . . REPLACING CHARACTERS statement at the compiler number shown requires the word BY after the word CHARACTERS for correct syntax. |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH24I-C</b> | <b>INSPECT WITH (REPLACING) REQUIRES (BY); FOUND (field-contents) - CONTINUING</b>                                                               |
| Explanation:      | The INSPECT . . . REPLACING statement at the compiler number shown requires the word BY after the identifier that follows REPLACING.             |
| User Action:      | Check the INSPECT statement and correct as necessary.                                                                                            |
| <b>DCDEH25A-C</b> | <b>EXPECTING ALPHANUMERIC FILE-NAME; FOUND (file-name) - ACCEPTED AS IS</b>                                                                      |
| Explanation:      | The file name following the COBOL verb MERGE at the compiler number shown is not alphanumeric.                                                   |
| User Action:      | Check the MERGE statement and make corrections as necessary.                                                                                     |
| <b>DCDEH25B-E</b> | <b>NO SD ENTRIES FOUND FOR (SD data-name)</b>                                                                                                    |
| Explanation:      | The file name used at the compiler number shown does not match a file name in the Data Division.                                                 |
| User Action:      | Check the file name and make corrections as necessary.                                                                                           |
| <b>DCDEH25C-E</b> | <b>NO MATCHING FILE-NAME FOUND FOR (file-name)</b>                                                                                               |
| Explanation:      | The file name used at the compiler number shown does not match a file name in the Data Division.                                                 |
| User Action:      | Check the file name and make corrections as necessary.                                                                                           |
| <b>DCDEH25D-C</b> | <b>EXPECTING (ASCENDING) OR (DESCENDING) FOUND (field-contents)</b>                                                                              |
| Explanation:      | The MERGE . . . ON statement at the compiler number shown must be followed by the word ASCENDING or DESCENDING.                                  |
| User Action:      | Check the MERGE statement and make the corrections as necessary.                                                                                 |
| <b>DCDEH25E-E</b> | <b>EXPECTING (USING) IN (MERGE) STMT-FOUND (field-contents) - BYPASSING</b>                                                                      |
| Explanation:      | The MERGE statement at the compiler number shown must be followed by the word, GIVING.                                                           |
| User Action:      | Check the MERGE statement and make corrections as necessary.                                                                                     |
| <b>DCDEH25F-E</b> | <b>EXPECTING (OUTPUT) OR (GIVING) - FOUND (field-contents)</b>                                                                                   |
| Explanation:      | The MERGE statement at the compiler number shown must be followed by the words, OUTPUT or GIVING, for this statement to have correct syntax.     |
| User Action:      | Check the MERGE statement and make corrections as necessary.                                                                                     |
| <b>DCDEH25G-C</b> | <b>(PROCEDURE) IS MISSING AFTER (OUTPUT) - CONTINUING</b>                                                                                        |
| Explanation:      | The MERGE . . . OUTPUT statement at the compiler number shown is missing the word, PROCEDURE after OUTPUT.                                       |
| User Action:      | Check the MERGE statement and make corrections as necessary.                                                                                     |
| <b>DCDEH25H-C</b> | <b>MERGE KEY IS NOT ALPHANUMERIC (key) - ACCEPTED</b>                                                                                            |



## DCD III – Error Messages

---

|                   |                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | The MERGE statement at the compiler number shown contains a key which is not alphanumeric.                                                        |
| User Action:      | Check the MERGE statement and make corrections as necessary.                                                                                      |
| <b>DCDEH26A-C</b> | <b>FILE NAME FOLLOWING (RECEIVE) IS NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                         |
| Explanation:      | The RECEIVE statement at the compiler number shown contains a file name which is not alphanumeric.                                                |
| User Action:      | Check the RECEIVE statement and make corrections as necessary.                                                                                    |
| <b>DCDEH26B-E</b> | <b>NO FD ENTRIES FOR (file-name)</b>                                                                                                              |
| Explanation:      | The File name found at the compiler number shown does not match a file name in the FD section of the Data Division.                               |
| User Action:      | Check the RECEIVE statement at the compiler number shown and make corrections as necessary.                                                       |
| <b>DCDEH26C-E</b> | <b>NO MATCHING FILE-NAME FOR (file-name)</b>                                                                                                      |
| Explanation:      | The file name found at the compiler number shown does not match a file name in the Data Division.                                                 |
| User Action:      | Check the RECEIVE statement at the compiler number shown and make corrections as necessary.                                                       |
| <b>DCDEH26D-C</b> | <b>(INTO IDENTIFIER-1) IDENTIFIER-1 NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                         |
| Explanation:      | The identifier found at the compiler number shown is not alphanumeric.                                                                            |
| User Action:      | Check the RECEIVE . . . INTO statement and correct as necessary.                                                                                  |
| <b>DCDEH27A-C</b> | <b>(SEARCH) OPERAND IS NOT ALPHANUMERIC – ACCEPTED AS IS</b>                                                                                      |
| Explanation:      | The operand or identifier following the COBOL verb SEARCH at the compiler number shown is not alphanumeric.                                       |
| User Action:      | Check the SEARCH statement and correct as necessary.                                                                                              |
| <b>DCDEH28A-C</b> | <b>RECORD NAME FOLLOWING (SEND) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                             |
| Explanation:      | The SEND statement at the compiler number shown contains a record name which is not alphanumeric.                                                 |
| User Action:      | Check the SEND statement and correct as necessary.                                                                                                |
| <b>DCDEH28B-C</b> | <b>(FROM IDENTIFIER-1) IDENTIFIER-1 NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                         |
| Explanation:      | The SEND . . . FROM statement at the compiler number shown contains an identifier following the word FROM that is not alphanumeric.               |
| User Action:      | Check the SEND statement and correct as necessary.                                                                                                |
| <b>DCDEF29A-C</b> | <b>FILE NAME (file-name) IS NOT ALPHANUMERIC AFTER (SORT) - ACCEPTED AS IS</b>                                                                    |
| Explanation:      | The SORT statement at the compiler number shown contains a non-alphanumeric file name.                                                            |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEF29B-E</b> | <b>NO MATCHING SD ENTRIES FOUND FOR (SD name)</b>                                                                                                 |
| Explanation:      | The SORT statement at the compiler number shown contains a file name for the sort that does not match a file name for an SD in the Data Division. |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29C-E</b> | <b>NO MATCHING FILE-NAME FOUND FOR (file-name)</b>                                                                                                |
| Explanation:      | The SORT statement at the compiler number shown contains a file name that does not match a file name in the Data Division.                        |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29D-C</b> | <b>EXPECTING (ASCENDING) OR (DESCENDING) - FOUND (field-contents)</b>                                                                             |
| Explanation:      | The SORT statement at the compiler number shown is missing the word ASCENDING or DESCENDING following the name of the sort file and the word ON.  |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29E-E</b> | <b>EXPECTING (USING) or (INPUT) - FOUND (field-contents)</b>                                                                                      |
| Explanation:      | The SORT statement at the compiler number shown is missing the word INPUT or USING.                                                               |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29F-E</b> | <b>(OUTPUT) OR (GIVING) NOT FOUND WITHIN (SORT)</b>                                                                                               |
| Explanation:      | The SORT statement at the compiler number shown is missing the word GIVING or OUTPUT.                                                             |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29G-C</b> | <b>PROCEDURE) IS MISSING AFTER (OUTPUT) - CONTINUING</b>                                                                                          |
| Explanation:      | The SORT statement at the compiler number shown is missing the word PROCEDURE after OUTPUT.                                                       |
| User Action:      | Check the SORT statement and correct as necessary.                                                                                                |
| <b>DCDEH29H-C</b> | <b>SORT KEY IS NOT ALPHANUMERIC (sort-key) - ACCEPTED</b>                                                                                         |
| Explanation:      | The SORT statement at the compiler number shown contains a sort key that is not alphanumeric.                                                     |

|                                        |                                                                                                                                                                   |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User Action:                           | Check the SORT statement and correct as necessary.                                                                                                                |
| <b>DCDEH30A-C</b>                      | <b>EXPECTING ALPHANUMERIC FILE NAME AFTER (START) - FOUND (file-name)</b>                                                                                         |
| Explanation:                           | The START statement at the compiler number shown contains a file name that is not alphanumeric.                                                                   |
| User Action:                           | Check the START statement and make corrections as necessary.                                                                                                      |
| <b>DCDEH30B-E<br/>&amp; DCDEH30C-E</b> | <b>NO MATCHING FD ENTRY FOUND FOR (file-name)</b>                                                                                                                 |
| Explanation:                           | The START statement at the compiler number shown contains a file name that does not match a file name in the FD section of the Data Division.                     |
| User Action:                           | Check the START statement and make corrections as necessary.                                                                                                      |
| <b>DCDEH30D-C</b>                      | <b>(OR) WITHIN (START) REQUIRES (=), FOUND (field-contents)</b>                                                                                                   |
| Explanation:                           | The START statement at the compiler number shown contains the word OR which must be followed by the phrase EQUAL TO for correct syntax.                           |
| User Action:                           | Check the START statement and make corrections as necessary.                                                                                                      |
| <b>DCDEH31A-C</b>                      | <b>FILE-NAME FOLLOWING (SEEK) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                                               |
| Explanation:                           | The SEEK statement at the compiler number shown contains a file name that is not alphanumeric.                                                                    |
| User Action:                           | Check the SEEK statement and make corrections as necessary.                                                                                                       |
| <b>DCDEH31B-E<br/>&amp; DCDEH31C-E</b> | <b>NO MATCHING FD ENTRIES FOUND FOR (file-name)</b>                                                                                                               |
| Explanation:                           | The SEEK statement at the compiler number shown contains a file name that does not match a file name in the FD section of the Data Division.                      |
| User Action:                           | Check the SEEK statement and make corrections as necessary.                                                                                                       |
| <b>DCDEH32A-C</b>                      | <b>EXPECTING (INTO) FOUND (field-contents)</b>                                                                                                                    |
| Explanation:                           | The STRING statement at the compiler number shown is missing the word INTO after the items that are being strung together.                                        |
| User Action:                           | Check the STRING statement and make corrections as necessary.                                                                                                     |
| <b>DCDEH32B-C</b>                      | <b>DATA-NAME AFTER (INTO) NOT ALPHANUMERIC - ACCEPTED</b>                                                                                                         |
| Explanation:                           | The STRING statement at the compiler number shown contains a non-alphanumeric data name following the word INTO.                                                  |
| User Action:                           | Check the STRING statement and make corrections as necessary.                                                                                                     |
| <b>DCDEH32C-C</b>                      | <b>VALID DATA -NAME SHOULD FOLLOW (WITH POINTER) IN (STRING)</b>                                                                                                  |
| Explanation:                           | The STRING statement at the compiler number shown contains a non-alphanumeric data name following the WITH POINTER clause or the data name is missing or invalid. |
| User Action:                           | Check the STRING statement and make corrections as necessary.                                                                                                     |
| <b>DCDEH33A-C</b>                      | <b>IDENTIFIER AFTER TRANSFORM (field-contents) NOT ALPHANUMERIC - ACCEPTED</b>                                                                                    |
| Explanation:                           | The TRANSFORM statement at the compiler number shown contains an identifier that is not alphanumeric.                                                             |
| User Action:                           | Check the TRANSFORM statement and make corrections as necessary.                                                                                                  |
| <b>DCDEH33B-E</b>                      | <b>(FROM) EXPECTED - FOUND (field-contents) BYPASSING</b>                                                                                                         |
| Explanation:                           | Expecting the word FROM at this point within the TRANSFORM statement.                                                                                             |
| User Action:                           | Check the TRANSFORM statement and make corrections as necessary.                                                                                                  |
| <b>DCDEH33C-E</b>                      | <b>EXPECTING (TO) - FOUND (field-contents)</b>                                                                                                                    |
| Explanation:                           | Expecting the word TO at this point within the TRANSFORM statement.                                                                                               |
| User Action:                           | Check the TRANSFORM statement and make corrections as necessary.                                                                                                  |
| <b>DCDEH34A-C</b>                      | <b>IDENTIFIER (field-contents) IS NOT ALPHANUMERIC</b>                                                                                                            |
| Explanation:                           | The UNSTRING statement at the compiler number shown contains an identifier which is not alphanumeric.                                                             |
| User Action:                           | Check the UNSTRING statement and make corrections as necessary.                                                                                                   |
| <b>DCDEH34B-E</b>                      | <b>(INTO) FOR (UNSTRING) NOT FOUND</b>                                                                                                                            |
| Explanation:                           | The UNSTRING statement at the compiler number shown is missing the word INTO or has the word misplaced.                                                           |
| User Action:                           | Check the UNSTRING statement and make corrections as necessary.                                                                                                   |
| <b>DCDEH35A-C</b>                      | <b>NAME AFTER (USING) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                                                                       |

**& DCDEH35B-C**

Explanation: The ENTRY statement at the compiler number shown contains a data name following the word USING which is not alphanumeric.  
User Action: Check the ENTRY statement and make corrections as necessary.

**DCDEH40A-C THE DATA-NAME FOLLOWING INITIALIZE (identifier) IS NOT ALPHANUMERIC-ACCEPTED AS IS**

Explanation: The INITIALIZE statement at the compiler number shown contains an identifier that is not alphanumeric.  
User Action: Check the INITIALIZE statement and make corrections as necessary.

**DCDEH40B-E THE OPERAND AFTER REPLACING (field-contents) IS UNKNOWN - BYPASSING INITIALIZE**

Explanation: The INITIALIZE statement at the compiler number shown contains an operand following the word REPLACING that is misspelled or unrecognized.  
User Action: Check the INITIALIZE statement and make corrections as necessary.

**DCDEH40C-E EXPECTING (BY) AFTER REPLACING, FOUND (field-contents) - BYPASSING INITIALIZE**

Explanation: The INITIALIZE statement at the compile number shown contains an unrecognized word following the word REPLACING.  
User Action: Check the INITIALIZE statement and make corrections as necessary.

**DCDEH46A-C EXPECTING ALPHANUMERIC FIELD IN MARGIN B AFTER (EXEC field-contents) -FOUND (field-contents) ACCEPTED**

Explanation: The EXEC statement at the compiler number shown contains a data name that is not alphanumeric or is not in the correct margin.  
User Action: Check the EXEC statement and make corrections as necessary.

**DCDEH46B-C EXPECTING ALPHANUMERIC FIELD PRIOR TO LEFT PARENTHESIS - FOUND (Field-contents)**

Explanation: The EXEC statement at the compiler number shown contains a field to the left of the left parenthesis that is not alphanumeric.  
User Action: Check the EXEC statement and make corrections as necessary.

**DCDEH46C-E EXCESSIVE PARAMETERS BYPASSED IN (EXEC) STMT**

Explanation: An excessive number of parameters were used within one EXEC statement.  
User Action: Determine why an excessive number was used and correct. If already correct, contact MARBLE Computer, Inc.

**DCDEH46D-E EXPECTING QUALIFIER NAME AFTER (OF) OR (IN) (EXEC) STMT -BYPASSING QUALIFICATION**

Explanation: The EXEC statement at the compile number shown is missing a qualifier name after OF or IN.  
User Action: Check the EXEC statement and make corrections as necessary.

**DCDEH46E-C EXCESSIVE QUALIFICATION FOR (EXEC) STATEMENT FOUND – PROCESSING CONTINUING**

Explanation: An excessive number of qualifiers were used within one EXEC statement.  
User Action: Determine why an excessive number was used and correct. If already correct, contact MARBLE Computer, Inc.

**DCDEH51A-E ALTER STATEMENT MISSING WORD (TO)**

Explanation: The ALTER statement at the compiler number shown is missing the word TO after the first procedure name.  
User Action: Check the ALTER statement and make corrections as necessary.

**DCDEH51B-E ALTER STATEMENT MISSING PROCEDURE-NAME-2**

Explanation: The ALTER statement at the compiler number shown is missing procedure-name-2 following the word TO.  
User Action: Check the ALTER statement and make corrections as necessary.

**DCDEH57A-C (STOP) REQUIRES (RUN) OR (LITERAL)**

Explanation: The STOP statement at the compiler number shown is missing the word RUN or a literal following it.  
User Action: Check the STOP statement and make corrections as necessary.

**DCDEH60A-C DATA-NAME FOLLOWING (INITIATE) NOT ALPHANUMERIC - ACCEPTED AS IS**

Explanation: The REPORT WRITER verb INITIATE requires an alphanumeric operand following it. This was not found.  
User Action: Check the statement and make necessary corrections.

|                                                             |                                                                                                                             |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>DCDEH61A-C</b>                                           | <b>DATA-NAME FOLLOWING (GENERATE) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                     |
| Explanation:                                                | The REPORT WRITER verb GENERATE requires an alphanumeric operand following it. This was not found.                          |
| User Action:                                                | Check the statement and make necessary corrections.                                                                         |
| <b>DCDEH62A-C</b>                                           | <b>DATA-NAME FOLLOWING (TERMINATED) NOT ALPHANUMERIC - ACCEPTED AS IS</b>                                                   |
| Explanation:                                                | The REPORT WRITER verb TERMINATE requires an alphanumeric operand following it. This was not found.                         |
| User Action:                                                | Check the statement and make necessary corrections.                                                                         |
| <b>DCDEJ10A-I</b>                                           | <b>CONDITIONAL STATEMENT IS INCOMPLETE - IF 88 LEVEL INTENDED, CHECK SPELLING</b>                                           |
| Explanation:                                                | A conditional statement is incomplete. If an 88 level was intended, a matching 88 level in the Data Division was not found. |
| User Action:                                                | Check the conditional statement and make necessary corrections.                                                             |
| <b>DCDEJ100-I</b>                                           | <b>PARSING CANNOT FIND THE BEGIN OF SECOND PART OF NON-88 CONDITIONAL</b>                                                   |
| Explanation:                                                | A conditional statement is incomplete. If an 88 level was intended, a matching 88 level in the Data Division was not found. |
| User Action:                                                | Check the conditional statement and make necessary corrections.                                                             |
| <b>DCDEJ300-C /<br/>&amp; DCDEJ400-C</b>                    | <b>(IF) EXCEEDED 125 OPERANDS FOR 1 CONDITION</b>                                                                           |
| Explanation:                                                | An IF (or IF type) statement contains more than 125 operands within just 1 condition.                                       |
| User Action:                                                | If this is a valid condition, contact MARBLE Computer, Inc.                                                                 |
| <b>DCDEJAPA-E<br/>&amp; DCDEJAPB-E<br/>&amp; DCDEJS1A-C</b> | <b>AN INTERNAL TABLE HAS EXCEEDED 125 DATA-NAMES FOR 1 CONDITION</b>                                                        |
| Explanation:                                                | An IF (or IF type) statement contains more than 125 operands within just 1 condition.                                       |
| User Action:                                                | If this is a valid condition, contact MARBLE Computer, Inc.                                                                 |
| <b>DCDENASA-C</b>                                           | <b>EXCEEDED 75 QUALIFIERS FOR OPERANDS WITHIN JUST 1 CONDITION</b>                                                          |
| Explanation:                                                | More than 75 qualifiers were used within just 1 condition.                                                                  |
| User Action:                                                | If this is a valid condition, contact MARBLE Computer, Inc.                                                                 |
| <b>DCDERW0A-W</b>                                           | <b>NO PERIOD PRECEDING LEVEL NUMBER (level #)</b>                                                                           |
| Explanation:                                                | The level number at the compiler number shown does not have a period preceding it on a previous line.                       |
| User Action:                                                | Check the lines above the level number for a missing period and make corrections as necessary.                              |
| <b>DCDERW0B-C</b>                                           | <b>EXPECTING (ZERO) AFTER (BLANK)</b>                                                                                       |
| Explanation:                                                | The word BLANK at the compiler number shown is missing the word ZERO after it.                                              |
| User Action:                                                | Check the line and make corrections as necessary.                                                                           |
| <b>DCDERW0C-C<br/>&amp; DCDERW0D-C</b>                      | <b>EXPECTING NUMERIC INTEGER AFTER (COLUMN)</b>                                                                             |
| Explanation:                                                | The word COLUMN at the compiler number shown is missing a numeric integer following it.                                     |
| User Action:                                                | Check the line and make corrections as necessary.                                                                           |
| <b>DCDERW0E-C<br/>&amp; DCDERW0F-C</b>                      | <b>EXPECTING NUMERIC INTEGER AFTER (LINE)</b>                                                                               |
| Explanation:                                                | The word LINE at the compiler number shown is missing a numeric integer following it.                                       |
| User Action:                                                | Check the line and make corrections as necessary.                                                                           |
| <b>DCDERW0G-C</b>                                           | <b>EXPECTING (GROUP) AFTER (NEXT)</b>                                                                                       |
| Explanation:                                                | The word NEXT at the compiler number shown is missing the word GROUP following it.                                          |
| User Action:                                                | Check the line and make corrections as necessary.                                                                           |
| <b>DCDERW0H-C<br/>&amp; DCDERW0I-C</b>                      | <b>EXPECTING NUMERIC INTEGER AFTER (NEXT GROUP)</b>                                                                         |
| Explanation:                                                | The phrase NEXT GROUP at the compiler number shown is missing a numeric integer following it.                               |
| User Action:                                                | Check the line and make corrections as necessary.                                                                           |
| <b>DCDERW0J-C</b>                                           | <b>CANNOT FIND VALID PICTURE CLAUSE</b>                                                                                     |

## DCD III – Error Messages

---

|                                        |                                                                                                             |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Explanation:                           | Expecting valid PICTURE clause to follow the word PICTURE and instead found data within the margin a field. |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0K-E</b>                      | <b>EXPECTING (TALLY) OR IDENTIFIER AFTER (SUM)</b>                                                          |
| Explanation:                           | Expecting a valid data name within the SUM clause within REPORT WRITER.                                     |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0L-C</b>                      | <b>EXPECTING IDENTIFIER AFTER (UPON)</b>                                                                    |
| Explanation:                           | Expecting a valid data name within the SUM UPON clause.                                                     |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0M-E<br/>&amp; DCDERW0N-E</b> | <b>EXPECTING (HEADING) OR (FOOTING) AFTER (field-contents)</b>                                              |
| Explanation:                           | Expecting the word HEADING or FOOTING within REPORT WRITER at this line.                                    |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0P-E</b>                      | <b>EXPECTING VALID TYPE AFTER (TYPE IS) - FOUND (field-contents)</b>                                        |
| Explanation:                           | An invalid entry was found after the TYPE IS clause.                                                        |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0Q-C</b>                      | <b>EXPECTING (FINAL) OR IDENTIFIER AFTER (TYPE IS) CLAUSE</b>                                               |
| Explanation:                           | Expecting reserved word FINAL or a valid data name to follow the TYPE IS clause in REPORT WRITER.           |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDERW0R-C</b>                      | <b>EXPECTING VALID OPERAND FOR USAGE CLAUSE - FOUND (field-contents)</b>                                    |
| Explanation:                           | The USAGE clause is in error or contains a type usage unknown to DCD III.                                   |
| User Action:                           | Correct USAGE clause or contact MARBLE Computer, Inc. if the USAGE clause is valid.                         |
| <b>DCDERW0S-E</b>                      | <b>EXPECTING LITERAL AFTER (VALUE) - FOUND (field-contents)</b>                                             |
| Explanation:                           | The entry following VALUE does not complete the VALUE clause within the REPORT SECTION.                     |
| User Action:                           | Determine why the entry is not valid and correct.                                                           |
| <b>DCDERWRA-C</b>                      | <b>MNEMONIC-NAME AFTER CODE MISSING</b>                                                                     |
| Explanation:                           | The operand following WITH CODE is missing.                                                                 |
| User Action:                           | Look at listing and correct.                                                                                |
| <b>DCDERWRB-C</b>                      | <b>EXPECTING INTEGER FOR PAGE CLAUSE</b>                                                                    |
| Explanation:                           | One of the PAGE clauses is missing a number following it.                                                   |
| User Action:                           | Consult the line indicated and correct the format of the PAGE clause.                                       |
| <b>DCDERWRC-C</b>                      | <b>EXPECTING INTEGER AFTER (field-contents)</b>                                                             |
| Explanation:                           | One of the PAGE clauses is missing a number following it.                                                   |
| User Action:                           | Consult the line indicated and correct the format of the PAGE clause.                                       |
| <b>DCDESDDB-E</b>                      | <b>NUMERIC INTEGER DOES NOT FOLLOW OCCURS - BYPASSING</b>                                                   |
| Explanation:                           | The OCCURS clause at the compiler number shown is not followed by a numeric integer.                        |
| User Action:                           | Check the OCCURS clause for a missing integer and make corrections as necessary.                            |
| <b>DCDESDDC-E</b>                      | <b>CANNOT FIND VALID ENTRY TO COMPLETE PICTURE CLAUSE - BYPASSING</b>                                       |
| Explanation:                           | The PICTURE clause is either continued in an invalid column or is incomplete or incorrect.                  |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDESDDD-E</b>                      | <b>ALPHANUMERIC DATA-NAME NOT FOUND AFTER REDEFINES - BYPASSING</b>                                         |
| Explanation:                           | The data name found after the REDEFINES clause in the Data Division is not alphanumeric.                    |
| User Action:                           | Check the line and make corrections as necessary.                                                           |
| <b>DCDESDDF-E</b>                      | <b>ALPHANUMERIC DATA-NAME NOT FOUND AFTER RENAMES - BYPASSING</b>                                           |
| Explanation:                           | The data name that follows the COBOL verb, RENAMES is not alphanumeric.                                     |
| User Action:                           | Check the RENAMES clause and make corrections as necessary.                                                 |
| <b>DCDESDDG-W</b>                      | <b>ENTRY FOUND WITHIN USAGE CLAUSE IS EITHER INVALID OR UNRECOGNIZED BYPASSING</b>                          |
| Explanation:                           | The word following the USAGE IS clause is either incorrect or unrecognized.                                 |
| User Action:                           | Check the USAGE IS clause and make corrections as necessary.                                                |

|                   |                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDESDDH-E</b> | <b>CANNOT FIND LITERAL AFTER (VALUE) - BYPASSING</b>                                                                                                 |
| Explanation:      | The level number and clause at the compiler number shown must contain a literal following the word VALUE.                                            |
| User Action:      | Check the line indicated and make any necessary corrections.                                                                                         |
| <b>DCDESDDI-E</b> | <b>INDEX-NAME IS MISSING OR IS NOT ALPHANUMERIC AFTER (INDEXED BY) CLAUSE- BYPASSING</b>                                                             |
| Explanation:      | The INDEXED BY clause at the compiler number shown contains an index name that is not alphanumeric or is missing.                                    |
| User Action:      | Check the line and make corrections as necessary.                                                                                                    |
| <b>DCDESDDJ-C</b> | <b>ENTRY FOR PICTURE CLAUSE HAS NON-NUMERIC VALUE WITHIN PARENTHESES - (picture)</b>                                                                 |
| Explanation:      | The PICTURE clause at the compiler number shown must contain a numeric value within its parentheses.                                                 |
| User Action:      | Check the PICTURE clause and make corrections as necessary.                                                                                          |
| <b>DCDESDDL-C</b> | <b>EXPECTING ALPHANUMERIC NAME FOR (REPORT IS) CLAUSE</b>                                                                                            |
| Explanation:      | The REPORT IS clause at the compiler number shown contains a word following it that is not alphanumeric.                                             |
| User Action:      | Check the line and make corrections as necessary.                                                                                                    |
| <b>DCDESYLE-E</b> | <b>EXPECTING QUALIFIER TO FOLLOW (OF) OR (IN) - FOUND (field-contents)</b>                                                                           |
| Explanation:      | Expecting qualifier to follow OF or IN.                                                                                                              |
| User Action:      | Examine listing and insert qualifier at correct location.                                                                                            |
| <b>DCDGBA11-E</b> | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                                  |
| Explanation:      | An internal table was not large enough to handle the program being documented.                                                                       |
| User Action:      | Rerun the job with a larger region.                                                                                                                  |
| <b>DCDGBB10-E</b> | <b>AN UNMATCHED RECORD IS BEING BYPASSED WHEN MATCHING FILES - LOOK FOR SYNTAX ERROR</b>                                                             |
| Explanation:      | Match merging of internal files is either out of sequence or an unmatched record is present. This may be due to a syntax error in the input program. |
| User Action:      | Try compiling the program to see if it compiles cleanly. If unsolved, contact MARBLE Computer, Inc.                                                  |
| <b>DCDGDZ10-D</b> | <b>DDNAME (DCDWKT2) IS MISSING WHEN PARM OPTION ( DIT ) IS SPECIFIED</b>                                                                             |
| Explanation:      | A necessary DD is missing for an internal work file.                                                                                                 |
| User Action:      | Insure that DDnames DCDWKT1 through DCDWKT4 are present within the JCL similar to DDnames DCDWK01, DCDWK02, etc.                                     |
| <b>DCDGDZ20-D</b> | <b>OVER nnnn CTLDLTCC CONTROL CARDS WERE USED – TABLE EXCEEDED</b>                                                                                   |
| Explanation:      | Too many control statements were specified.                                                                                                          |
| User Action:      | Limit the number of control statements to the number shown in the message printed.                                                                   |
| <b>DCDGDZ30-D</b> | <b>OVER 150 CONTROL CARDS WERE ADDED TO THE DIT CONTROL STATEMENT FILE – TABLE EXCEEDED</b>                                                          |
| Explanation:      | Too many control statements were specified.                                                                                                          |
| User Action:      | Use less control statements.                                                                                                                         |
| <b>DCDGDZ40-D</b> | <b>OVER 999 DATA NAMES WERE FOUND FOR DOING TRACING ON – TABLE EXCEEDED</b>                                                                          |
| Explanation:      | When tracing literals one data name with a literal directly to 1000 or more data names, an unusual condition.                                        |
| User Action:      | This condition if found and valid may be resolved by contacting MARBLE Computer, Inc.                                                                |
| <b>DCDGF002-D</b> | <b>DDNAME (DCDWK08) MISSING</b>                                                                                                                      |
| Explanation:      | The running of DCD III requires work files DCDWK08. The DDNAME DCDWK08 was not found.                                                                |
| User Action:      | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                                    |
| <b>DCDGJB04-C</b> | <b>EXCESSIVE QUALIFICATION AND/OR SUBSCRIPTING FOUND</b>                                                                                             |
| Explanation:      | Excessive qualification and/or subscripting was found while handling a CORRESPONDING statement at this line number.                                  |
| User Action:      | Determine why excessive qualification and/or subscripting is used and attempt to correct. If necessary, contact MARBLE Computer, Inc.                |
| <b>DCDGJN02-W</b> | <b>EXCESSIVE QUALIFICATION AND/OR SUBSCRIPTING FOUND</b>                                                                                             |

## DCD III – Error Messages

---

|                                    |                                                                                                                                         |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:                       | Excessive qualification and/or subscripting was found while handling a CORRESPONDING statement at this line number.                     |
| User Action:                       | Determine why excessive qualification and/or subscripting was used and attempt to correct. If necessary, contact MARBLE Computer, Inc.  |
| <b>DCDGKB04-C</b>                  | <b>EXCESSIVE QUALIFICATION AND/OR SUBSCRIPTING FOUND</b>                                                                                |
| Explanation:                       | Excessive qualification and/or subscripting was found while handling a CORRESPONDING statement at this line number.                     |
| User Action:                       | Determine why excessive qualification and/or subscripting was used and attempt to correct. If necessary, contact MARBLE Computer, Inc.  |
| <b>DCDGKF02-D</b>                  | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                    |
| Explanation:                       | The first record of a work file being read does not contain a MARBLE header record.                                                     |
| User Action:                       | Look for possible previous message to this one. Look for an error in the JCL used.                                                      |
| <b>DCDGKH07-C</b>                  | <b>INSUFFICIENT SPACE TO HANDLE INITIALIZE DATA DIVISION NAMES (H070) - BYPASSING EXCESS NAMES</b>                                      |
| Explanation:                       | A table for Data Division NAMES was not large enough to handle the program being documented.                                            |
| User Action:                       | Contact MARBLE Computer, Inc.                                                                                                           |
| <b>DCDGKN02-C</b>                  | <b>EXCESSIVE QUALIFICATION AND/OR SUBSCRIPTING FOUND</b>                                                                                |
| Explanation:                       | Excessive qualification and/or subscripting was found while handling a CORRESPONDING statement at this line number.                     |
| User Action:                       | Determine why excessive qualification and/or subscripting was used and attempt to correct. If necessary, contact MARBLE Computer, Inc.  |
| <b>DCDGSA05-E</b>                  | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                     |
| Explanation:                       | An internal table was not large enough to handle the program run through DCD III.                                                       |
| User Action:                       | Rerun the job with a larger region.                                                                                                     |
| <b>DCDGSAC0-E</b>                  | <b>MORE THAN 7000 SECTIONS - EXCESS IGNORED IN PAR2 PROGRAM</b>                                                                         |
| Explanation:                       | More than 7000 SECTIONS were found within one COBOL program.                                                                            |
| User Action:                       | If this is a valid condition, contact Marble Computer.                                                                                  |
| <b>DCDGSB10-E</b>                  | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                     |
| Explanation:                       | An internal table was not large enough to handle the program run through DCD III.                                                       |
| User Action:                       | Rerun the job with a larger region.                                                                                                     |
| <b>DCDGSE02-E</b>                  | <b>PROGRAM ERROR ON MATCH - CHECK FOR PRIOR TABLE EXCEEDED MESSAGE</b>                                                                  |
| Explanation:                       | An internal error has occurred in matching paragraph names. This is probably caused by a previously printed out table exceeded message. |
| User Action:                       | If no other error messages are shown, contact MARBLE Computer, Inc.                                                                     |
| <b>DCDGSFP0-W</b>                  | <b>THE PARAGRAPH (paragraph-name) USED HERE IS NOT FOUND WITHIN THE PROGRAM</b>                                                         |
| Explanation:                       | A matching paragraph name to the one used here is not found within the program.                                                         |
| User Action:                       | Check for a missing name or a spelling error.                                                                                           |
| <b>DCDGSFS0-W</b>                  | <b>THE SECTION (section-name) USED HERE IS NOT FOUND WITHIN THE PROGRAM</b>                                                             |
| Explanation:                       | A matching section name to the one used is not found within the program.                                                                |
| User Action:                       | Check for a missing name or a spelling error.                                                                                           |
| <b>DCDGS03-W / DCDGSG11-W</b>      | <b>A MATCH WAS NOT FOUND FOR PARAGRAPH NAME (paragraph-name)</b>                                                                        |
| Explanation:                       | A matching paragraph name to the one used here is not found within the program.                                                         |
| User Action:                       | Check for a missing name or a spelling error.                                                                                           |
| <b>DCDGWA03-E</b>                  | <b>EXCESSIVE DYNAMIC CALLS FOUND AT THIS LINE NUMBER – REMAINING BYPASSED</b>                                                           |
| Explanation:                       | Over 3000 dynamic CALLs were found in one program. Those over 3000 will be bypassed.                                                    |
| User Action:                       | Determine why such an excessive number of dynamic CALLs were used in one program and attempt to reduce this number.                     |
| <b>DCDGWB02-E &amp; DCDGWD02-E</b> | <b>EXCESSIVE MATCHING LITERALS TO DYNAMIC CALLS FOUND THIS LINE NUMBER</b>                                                              |
| Explanation:                       | The number of VALUE and/or MOVE literals matching to names used in dynamic CALLs exceeds 2000.                                          |
| User Action:                       | Determine why excessive MOVES of literal to dynamic CALLs was used in one program and attempt to reduce this number.                    |

|                                                                                  |                                                                                                                                                             |
|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDHSB11-E</b>                                                                | <b>MORE THAN (NNNNN) DATA NAMES USED IN THE PROCEDURE DIVISION</b>                                                                                          |
| Explanation:                                                                     | An internal table was exceeded within DCD and this table needed to be made larger.                                                                          |
| User Action:                                                                     | Contact Marble Computer, Inc.                                                                                                                               |
| <b>DCDIBCB1-I</b>                                                                | <b>TRACING TABLE EXCEEDED – SOME TRACING WILL BE LOST</b>                                                                                                   |
| Explanation:                                                                     | An internal table was exceeded within DCD and this table needed to be made larger.                                                                          |
| User Action:                                                                     | Contact Marble Computer, Inc.                                                                                                                               |
| <b>DCDIBMA0-E</b>                                                                | <b>INTERNAL 88 LEVEL TABLE EXCEEDED – SOME TRACING MAY BE LOST</b>                                                                                          |
| Explanation:                                                                     | An internal table was exceeded within DCD and this table needed to be made larger.                                                                          |
| User Action:                                                                     | Contact Marble Computer, Inc.                                                                                                                               |
| <b>DCDIF030-I</b>                                                                | <b>NO RECORDS WERE PASSED TO THE SORT FOR THE PROGRAM xxxxxxxx</b>                                                                                          |
| Explanation:                                                                     | No records were selected for the program shown for Direct Tracing of Literals                                                                               |
| User Action:                                                                     | If this is a small program or a small Direct Literal report was anticipated, this may be normal.                                                            |
| <b>DCDJA040-I</b>                                                                | <b>NO DATA NAMES INTO SORT</b>                                                                                                                              |
| Explanation:                                                                     | Probably a very small program was processed resulting in no PROCEDURE DIVISION verbs (Transfer of Control) or (Data Manipulation) were found.               |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJC040-I</b>                                                                | <b>NO DATA-NAMES INTO SORT</b>                                                                                                                              |
| Explanation:                                                                     | This program contains no figurative constants, special registers, CALLs, COPY members or literals referenced from the Procedure Division.                   |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJCTA0-C</b>                                                                | <b>DYNAMIC CALL TABLE EXCEEDED FOR CALL REPORT</b>                                                                                                          |
| Explanation:                                                                     | An internal Marble table is exceeded.                                                                                                                       |
| User Action:                                                                     | Contact Marble Computer, Inc.                                                                                                                               |
| <b>DCDJE020-I</b>                                                                | <b>NO DATA-NAMES INTO SORT</b>                                                                                                                              |
| Explanation:                                                                     | No data names were passed out to the SORT. If the NOUNREF option was used, then unreferenced names were found to pass out to the SORT.                      |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJE021-W</b>                                                                | <b>NO DATA-NAMES WERE AVAILABLE FROM COBOL PROGRAM TO INCLUDE ON INTERNAL TABLE</b>                                                                         |
| Explanation:                                                                     | No data names were included on an internal table. If the NOUNREF option was used, then no referenced names were found.                                      |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJED02-E<br/>&amp; DCDJED03-E<br/>&amp; DCDJED04-E<br/>&amp; DCDJED05-E</b> | <b>OVER 49 QUALIFIERS WERE FOUND FOR ONE USE OF A DATA NAME</b>                                                                                             |
| Explanation:                                                                     | A data name with more than 49 qualifiers was found by DCD III.                                                                                              |
| User Action:                                                                     | If this is a valid condition, contact MARBLE Computer, Inc.                                                                                                 |
| <b>DCDJG040-I</b>                                                                | <b>NO DATA-NAME INTO SORT</b>                                                                                                                               |
| Explanation:                                                                     | There were no references in the Procedure Division to data names in the Data Division.                                                                      |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJI040-I</b>                                                                | <b>NO PARAGRAPH OR SECTION NAMES INTO SORT - IF (NOUNREF) USED, NO PARAGRAPHS WERE REFERENCED</b>                                                           |
| Explanation:                                                                     | There were no references in the Procedure Division to paragraph or section names. If PARM option NOUNREF was used, then no paragraph names were referenced. |
| User Action:                                                                     | None.                                                                                                                                                       |
| <b>DCDJPA34-E</b>                                                                | <b>SR-SECT-RANGE-TABLE EXCEEDED IN NOPARAS PROGRAM</b>                                                                                                      |
| Explanation:                                                                     | An internal table was exceeded and needs to be fixed by Marble Computer, Inc.                                                                               |
| User Action:                                                                     | Contact Marble Computer.                                                                                                                                    |
| <b>DCDJPX01-I</b>                                                                | <b>NO PARAGRAPH OR SECTION NAMES INTO SORT</b>                                                                                                              |
| Explanation:                                                                     | No paragraphs or sections were found.                                                                                                                       |
| User Action:                                                                     | If a small program with none present, ignore message.                                                                                                       |



## DCD III – Error Messages

---

|                                                                                  |                                                                                                                                                                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDKBS02-C</b>                                                                | <b>INDEXES EXCEEDED 125 FOR ONE OCCURS</b>                                                                                                                                  |
| Explanation:                                                                     | DCD III has a limit of 124 indexes present for 1 OCCURS. Documentation is lost.                                                                                             |
| User Action:                                                                     | Check the OCCURS to determine the cause of excessive use and ensure there are less than 125 indexes per OCCURS. If this is a valid condition, contact MARBLE Computer, Inc. |
| <b>DCDKFA02-D</b>                                                                | <b>FILE EMPTY – NO MARBLE HEADER</b>                                                                                                                                        |
| Explanation:                                                                     | The first record of a work file being read does not contain a MARBLE header record.                                                                                         |
| User Action:                                                                     | Look for possible previous messages to this one. Look for an error in the JCL used.                                                                                         |
| <b>DCDKFB07-D</b>                                                                | <b>SELECTED MORE THAN 1000 ENTRIES – REDUCE SELECT</b>                                                                                                                      |
| Explanation:                                                                     | Over 1000 entries were SELECTED from the SELECT control statements entered.                                                                                                 |
| User Action:                                                                     | Reduce the number of SELECTs used, and resubmit the run.                                                                                                                    |
| <b>DCDKFBU1-E</b>                                                                | <b>TRACE TABLE (DIR-) EXCEEDED</b>                                                                                                                                          |
| Explanation:                                                                     | An internal table, which holds detail from cumulative SELECTs, was exceeded.                                                                                                |
| User Action:                                                                     | Consider using less SELECTs or less ANDs and ORs in SELECTs.                                                                                                                |
| <b>DCDKFM02-D</b>                                                                | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                        |
| Explanation:                                                                     | The first record of a work file being read does not contain a MARBLE header record.                                                                                         |
| User Action:                                                                     | Look for possible previous messages to this one. Look for an error in the JCL used.                                                                                         |
| <b>DCDKFP06-E</b>                                                                | <b>OVER 9999 MATCHES - PROBABLE ERROR</b>                                                                                                                                   |
| Explanation:                                                                     | With Trace Option, on one SELECT more than 9999 matches were found.                                                                                                         |
| User Action:                                                                     | Look at SELECT and determine why this happened and fix.                                                                                                                     |
| <b>DCDKFY1A-D</b>                                                                | <b>DDNAME (DCDWK09) MISSING</b>                                                                                                                                             |
| Explanation:                                                                     | The DDNAME DCDWK09 is required by DCD III for program DCDSYSTEM.                                                                                                            |
| User Action:                                                                     | Provide missing DDNAME. See JCL examples in User's Manual.                                                                                                                  |
| <b>DCDKH040-D</b>                                                                | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                        |
| Explanation:                                                                     | An internal work file was found to be empty.                                                                                                                                |
| User Action:                                                                     | Try rerunning.                                                                                                                                                              |
| <b>DCDKH050-D</b>                                                                | <b>FILE EMPTY - NO DATA RECORDS</b>                                                                                                                                         |
| Explanation:                                                                     | An internal work file was found to be empty.                                                                                                                                |
| User Action:                                                                     | Try rerunning.                                                                                                                                                              |
| <b>DCDM5020-I</b>                                                                | <b>NO DATA-NAMES INTO SORT</b>                                                                                                                                              |
| Explanation:                                                                     | No data names were found in the Data Division to be included on an internal table for building COBOL narrative from.                                                        |
| User Action:                                                                     | None.                                                                                                                                                                       |
| <b>DCDM5021-W</b>                                                                | <b>NO DATA-NAMES ON TABLE</b>                                                                                                                                               |
| Explanation:                                                                     | No data names were found in the Data Division to be included on an internal table for building COBOL narrative from.                                                        |
| User Action:                                                                     | None.                                                                                                                                                                       |
| <b>DCDM5D02-C<br/>&amp; DCDM5D03-C<br/>&amp; DCDM5D04-C<br/>&amp; DCDM5D05-C</b> | <b>OVER 49 QUALIFIERS WERE FOUND FOR ONE DATA NAME</b>                                                                                                                      |
| Explanation:                                                                     | A data name with more than 49 qualifiers was found by DCD III.                                                                                                              |
| User Action:                                                                     | If this is a valid condition, contact MARBLE Computer, Inc.                                                                                                                 |
| <b>DCDM5FA0-C</b>                                                                | <b>QUALIFICATION USED IN LINE #nnnnn IS NOT WITHIN 01 RECORD</b>                                                                                                            |
| Explanation:                                                                     | A MOVE or other verb is using qualification and one data name is not within the 01 record.                                                                                  |
| User Action:                                                                     | Replace the data name not within the 01 record to a name within the 01 record.                                                                                              |
| <b>DCDM5J01-W</b>                                                                | <b>THE NAME (reference-name) DOES NOT MATCH TO A DATA DIVISION NAME</b>                                                                                                     |
| Explanation:                                                                     | A Procedure Division reference was made to a Data Division name that does not exist.                                                                                        |
| User Action:                                                                     | Check for a spelling error. If the name shown is a special register, notify MARBLE Computer, Inc. Then, ignore the message.                                                 |
| <b>DCDM8U20-I</b>                                                                | <b>NO DATA NAMES INTO SORT</b>                                                                                                                                              |
| Explanation:                                                                     | A program processing Indirect References had no data names to sort.                                                                                                         |
| User Action:                                                                     | Similar to error message above, if program is small, ignore this error message.                                                                                             |

|                                                                                  |                                                                                                                                          |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDM9A02-I</b>                                                                | <b>NO INDIRECT REFERENCES IN PROGRAM WERE FOUND</b>                                                                                      |
| Explanation:                                                                     | No indirect references were found within the entire COBOL.                                                                               |
| User Action:                                                                     | If program is small with very few references, ignore this error message.                                                                 |
| <b>DCDM9U20-I</b>                                                                | <b>NO DATA NAMES INTO SORT</b>                                                                                                           |
| Explanation:                                                                     | A program processing Indirect References had no data names to sort.                                                                      |
| User Action:                                                                     | Similar to error message above, if program is small, ignore this error message.                                                          |
| <b>DCDMKB02-I</b>                                                                | <b>TABLE TOO SMALL - Contact MARBLE Computer</b>                                                                                         |
| Explanation:                                                                     | An internal table was exceeded for an exceptionally large program.                                                                       |
| User Action:                                                                     | Contact Marble Computer.                                                                                                                 |
| <b>DCDMN002-D</b>                                                                | <b>DDNAME (DCDWK07) MISSING</b>                                                                                                          |
| Explanation:                                                                     | The running of DCD III requires work files DCDWK07 and others. The DDNAME DCDWK07 was not found.                                         |
| User Action:                                                                     | See JCL examples in the 'Alternate Compile Listing' section in the User's Manual and correct JCL.                                        |
| <b>DCDMUC02-W</b>                                                                | <b>FILE EMPTY - NO DATA OR PARAGRAPH RECORDS</b>                                                                                         |
| Explanation:                                                                     | An internal work file found no Data or Paragraph records for this COBOL program. This may be a valid condition for a very small program. |
| User Action:                                                                     | Check program and if very small ignore this message.                                                                                     |
| <b>DCDMXJ01-E</b>                                                                | <b>INDIRECT TABLE FOR REPORT EXCEEDED</b>                                                                                                |
| Explanation:                                                                     | An internal table was exceeded.                                                                                                          |
| User Action:                                                                     | Contact Marble Computer.                                                                                                                 |
| <b>DCDNHD02-C<br/>&amp; DCDNHD03-C<br/>&amp; DCDNHD04-C<br/>&amp; DCDNHD05-C</b> | <b>OVER 49 QUALIFIERS WERE FOUND FOR ONE DATA NAME</b>                                                                                   |
| Explanation:                                                                     | A data name with 49 qualifiers was found by DCD III.                                                                                     |
| User Action:                                                                     | If this is a valid condition, contact MARBLE Computer, Inc.                                                                              |
| <b>DCDNHJ01-W</b>                                                                | <b>THE NAME (reference-name) DOES NOT MATCH TO A DATA DIVISION NAME</b>                                                                  |
| Explanation:                                                                     | A Procedure Division reference was made to a Data Division name that does not exist.                                                     |
| User Action:                                                                     | Check for a spelling error. If the name shown is a special register, notify MARBLE Computer, Inc. Then, ignore the message.              |
| <b>DCDP1R10-E</b>                                                                | <b>EXCESSIVE SELECTIONS FOUND IN TRACING ALL POSSIBLE NAMES -TURN OFF YAP OPTION AND EXAMINE FIELDS SELECTED</b>                         |
| Explanation:                                                                     | Excessive SELECTIONs caused one or more internal tables to be exceeded.                                                                  |
| User Action:                                                                     | Turn off TRACING option to get run to continue to end, then examine to see how SELECTs can be limited.                                   |
| <b>DCDP1SBA-E</b>                                                                | <b>RFP-TABLE NBR ENTRIES EXCEEDED – SELECT LESS ENTRIES OR CONTACT MARBLE COMPUTER</b>                                                   |
| Explanation:                                                                     | An internal SELECT table for Tracing & Analysis was exceeded.                                                                            |
| User Action:                                                                     | Use less SELECTs in an attempt to have program run.                                                                                      |
| <b>DCDP1Y70-I</b>                                                                | <b>NO DIRECT SELECTED NAMES WERE FOUND - TRACING FOR PROGRAM (progrname) NOT DONE</b>                                                    |
| Explanation:                                                                     | An option was on indicating that TRACING should be done, however no names were selected.                                                 |
| User Action:                                                                     | Specify one or more names for SELECTION with SELECT statements.                                                                          |
| <b>DCDPBB03-E</b>                                                                | <b>BYPASS TABLE EXCEEDED –USE 1000 OR LESS BYPASS COMMANDS</b>                                                                           |
| Explanation:                                                                     | Over 1000 BYPASS table entries were found.                                                                                               |
| User Action:                                                                     | Correct the number of BYPASS control statements to limit it to 1000 entries.                                                             |
| <b>DCDPBG01-E</b>                                                                | <b>BYPASS TABLE EXCEEDED - USE 1000 OR LESS BYPASS COMMANDS</b>                                                                          |
| Explanation:                                                                     | The number of BYPASS control statements must be limited to 1000.                                                                         |
| User Action:                                                                     | Reduce the number of BYPASS control statements.                                                                                          |
| <b>DCDPBG02-D</b>                                                                | <b>RANGE= REQUIRES nnnnn-nnnnn AND FOUND RANGE=XXXXXXXXXXXX</b>                                                                          |
| Explanation:                                                                     | The RANGE control statement used is invalid.                                                                                             |
| User Action:                                                                     | Insure the RANGE=NNNNN-NNNNN format is used and resubmit.                                                                                |

## DCD III – Error Messages

---

|                                        |                                                                                                                                                                                                                                       |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDPBG03-E</b>                      | <b>BYPASS FILE – INVALID DATA NAME FOUND (xxxxxxx)</b>                                                                                                                                                                                |
| Explanation:                           | The first character of the name is a space.                                                                                                                                                                                           |
| User Action:                           | Correct Bypass name and resubmit.                                                                                                                                                                                                     |
| <b>DCDPBG04-E</b>                      | <b>BYPASS NAME TABLE EXCEEDED -USE 1000 OR LESS BYPASS COMMANDS</b>                                                                                                                                                                   |
| Explanation:                           | The number of BYPASS control statements must be limited to 1000.                                                                                                                                                                      |
| User Action:                           | Reduce the number of BYPASS control statements.                                                                                                                                                                                       |
| <b>DCDPEY30-D</b>                      | <b>MARBLE HEADER MISSING - CONTACT MARBLE COMPUTER</b>                                                                                                                                                                                |
| Explanation:                           | A header is missing on an internal work file.                                                                                                                                                                                         |
| User Action:                           | If this message occurs more than once and rerunning does not eliminate, contact Marble Computer.                                                                                                                                      |
| <b>DCDPEY40-D</b>                      | <b>NO PROGRAM HEADER FOUND FOR PROGRAM ( progame )</b>                                                                                                                                                                                |
| Explanation:                           | A header is missing on internal work file 09.                                                                                                                                                                                         |
| User Action:                           | If this message occurs more than once and rerunning does not eliminate, contact Marble Computer.                                                                                                                                      |
| <b>DCDPEY60-E<br/>&amp; DCDPEY90-E</b> | <b>TOTAL TRACE ENTRIES EXCEED nnnnn MAX ENTRIES FOR TABLE FOR<br/>PROGRAM progame - SELECT LESS ENTRIES</b>                                                                                                                           |
| Explanation:                           | Too many entries were SELECTed by the SELECT control statements and a table was exceeded.                                                                                                                                             |
| User Action:                           | Redo the SELECTs used and resubmit to select less entries.                                                                                                                                                                            |
| <b>DCDPF040-D</b>                      | <b>MARBLE HEADER MISSING - CONTACT MARBLE COMPUTER</b>                                                                                                                                                                                |
| Explanation:                           | A header is missing on an internal work file.                                                                                                                                                                                         |
| User Action:                           | If this message occurs more than once and rerunning does not eliminate, contact Marble Computer.                                                                                                                                      |
| <b>DCDPF055-D</b>                      | <b>NO PROGRAM HEADER FOUND FOR PROGRAM ( progame )</b>                                                                                                                                                                                |
| Explanation:                           | A header is missing on internal work file.                                                                                                                                                                                            |
| User Action:                           | If this message occurs more than once and rerunning does not eliminate, contact Marble Computer.                                                                                                                                      |
| <b>DCDPFC02-E</b>                      | <b>TOTAL TRACE ENTRIES EXCEEDS nnnnn MAX ENTRIES FOR TABLE FOR PROGRAM<br/>progame</b>                                                                                                                                                |
| Explanation:                           | All entries either directly selected for TRA entries or added through TRACING exceeds the maximum number available in table space.                                                                                                    |
| User Action:                           | Select less entries so the amount of table space is reduced.                                                                                                                                                                          |
| <b>DCDPFC03-D</b>                      | <b>DIRECT ENTRY MISSING FROM INTERNAL TABLE - CONTACT MARBLE COMPUTER</b>                                                                                                                                                             |
| Explanation:                           | This may be an internal logic error.                                                                                                                                                                                                  |
| User Action:                           | Try rerunning, then if this message still appears, contact Marble Computer.                                                                                                                                                           |
| <b>DCDPNP05-W</b>                      | <b>UNABLE TO GUAGE CORRECT USE OF FLOATING POINT COMP-1 OR COMP-2 - USER<br/>MUST CHECK CAREFULLY</b>                                                                                                                                 |
| Explanation:                           | During Tracing and Analysis that follows data fields from one field to another via From-To positions with an 01 record, a field is being traced through Floating point fields where From-To positions may not be accurately followed. |
| User Action:                           | User should look at results carefully and not rely entirely on results given.                                                                                                                                                         |
| <b>DCDPNP06-W</b>                      | <b>FIELD WITH (xxxxxx ) USAGE AT CMPLR NBR nnnnn HAS DIFFERENT LENGTH FROM<br/>FIELD WITH SAME USAGE AT CMPLR NBR nnnnn</b>                                                                                                           |
| Explanation:                           | During Tracing and Analysis that tracing using From-To positions are linking two fields with different lengths together. Our processing will follow only From-To positions from original field through new field.                     |
| User Action:                           | This is a warning for user clarification. No further action required except to notice difference, unless user wants to make program changes to rectify length difference.                                                             |
| <b>DCDPNP08-W</b>                      | <b>BINARY FIELD AT CMPLR NBR nnnnn IS NOT A HALFWORD, FULLWORD, OR<br/>DOUBLEWORD</b>                                                                                                                                                 |
| Explanation:                           | A binary (BINARY or COMPUTATIONAL) field is found where the real calculated positions for the field in question is outside the normal length of 2, 4, or 8 bytes for a halfword, fullword, or doubleword.                             |
| User Action:                           | User should look at usage and make any change deemed necessary or ignore this message.                                                                                                                                                |
| <b>DCDPNP09-W</b>                      | <b>BINARY FIELD AT CMPLR NBR nnnnn IS TOO SMALL FOR FIELD AT CMPLR nnnnn</b>                                                                                                                                                          |
| Explanation:                           | Within Tracing and Analysis, the binary field being traced into, is too small for the record positions of the other field.                                                                                                            |
| User Action:                           | This is a warning only, however the user may want to analysis why this is happening.                                                                                                                                                  |
| <b>DCDPNP0A-W</b>                      | <b>BINARY FIELD AT CMPLR NBR nnnnn IS TOO LARGE FOR FIELD AT CMPLR nnnnn</b>                                                                                                                                                          |

## DCD III – Error Messages

---

|                   |                                                                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | Within Tracing and Analysis, the binary field being traced into, is too large for the record positions of the other field.                                                                                                    |
| User Action:      | This is a warning only, however the user may want to analysis why this is happening.                                                                                                                                          |
| <b>DCDPNP0B-W</b> | <b>FIELD AT CMPLR NBR IS NOT EXACT SIZE AS FIELD AT CMPLR nnnnn</b>                                                                                                                                                           |
| Explanation:      | During Tracing and Analysis that follows data fields from one field to another via From-To positions with an 01 record, a field is being traced through Binary fields where From-To positions may not be accurately followed. |
| User Action:      | This is a warning only, however the user may want to analysis why this is happening.                                                                                                                                          |
| <b>DCDPNY40-D</b> | <b>DDNAME (CODEWK03) MISSING</b>                                                                                                                                                                                              |
| Explanation:      | A DD needed for 'Follow a Data Name' report (CODEWK03) is missing.                                                                                                                                                            |
| User Action:      | Inspect the JCL and add this DD.                                                                                                                                                                                              |
| <b>DCDPNY80-E</b> | <b>NSD – TABLE EXCEEDED - REDO SELECT STMTS TO SELECT LESS DATA NAMES</b>                                                                                                                                                     |
| Explanation:      | An internal table was exceeded.                                                                                                                                                                                               |
| User Action:      | Reduce the number of SELECTs to select less data names so internal table will be reduced in size.                                                                                                                             |
| <b>DCDPQC06-I</b> | <b>TRACING IGNORED AS THE FLAG POSITIONS WERE FOUND OUTSIDE THE MOVE OR COMPARE</b>                                                                                                                                           |
| Explanation:      | A PROCEDURE DIVISION instruction was found with a field that should be traced; however, the field, when being moved or compared, was truncated due to the size of the other field, and tracing is being ignored.              |
| User Action:      | No action is necessary.                                                                                                                                                                                                       |
| <b>DCDPQU30-D</b> | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                                                                          |
| Explanation:      | An internal work file was found to be empty.                                                                                                                                                                                  |
| User Action:      | Try rerunning. If necessary, contact Marble Computer.                                                                                                                                                                         |
| <b>DCDPQUA0-E</b> | <b>INTERNAL TABLE ( RFC-) EXCEEDED - CONTACT MARBLE COMPUTER</b>                                                                                                                                                              |
| Explanation:      | An internal table was exceeded.                                                                                                                                                                                               |
| User Action:      | Reduce the number of SELECTs to select less data names so internal table will be reduced in size.                                                                                                                             |
| <b>DCDPTP06-E</b> | <b>INTERNAL DATA NAME TABLE EXCEEDED - CONTACT MARBLE COMPUTER</b>                                                                                                                                                            |
| Explanation:      | An internal data name table was exceeded.                                                                                                                                                                                     |
| User Action:      | Look at the SELECTs statement used and reduce them to reduce table space.                                                                                                                                                     |
| <b>DCDPTP07-E</b> | <b>OVER 9999 MATCHES - PROBABLE ERROR</b>                                                                                                                                                                                     |
| Explanation:      | Tracing included over 9999 matches.                                                                                                                                                                                           |
| User Action:      | Look at the SELECTs statements used and change or reduce the number used to correct this error.                                                                                                                               |
| <b>DCDPTP08-D</b> | <b>DATA-NAME TABLE EXCEEDED - RUN WITH SMALLER NBR OF CONTROL STATEMENTS OR TURN OFF OPTION ( YAP )</b>                                                                                                                       |
| Explanation:      | An internal tracing table was exceeded. This is probably due to an excessive number of SELECTs being used.                                                                                                                    |
| User Action:      | Reduce the number of SELECTs or turn off the Tracing Option.                                                                                                                                                                  |
| <b>DCDPTY10-D</b> | <b>DDNAME ( DCDWK09 ) MISSING</b>                                                                                                                                                                                             |
| Explanation:      | A DD statement is missing.                                                                                                                                                                                                    |
| User Action:      | Inspect the JCL and add the DD statement.                                                                                                                                                                                     |
| <b>DCDPTY50-D</b> | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                                                                          |
| Explanation:      | An internal work file was found to be empty.                                                                                                                                                                                  |
| User Action:      | Try rerunning.                                                                                                                                                                                                                |
| <b>DCDPTY60-D</b> | <b>EXCEEDED INTERNAL TABLE - USE LESS CONTROL STATEMENTS</b>                                                                                                                                                                  |
| Explanation:      | An internal table was exceeded.                                                                                                                                                                                               |
| User Action:      | Look at the SELECTs statement used and reduce them to reduce table space.                                                                                                                                                     |
| <b>DCDT0080-C</b> | <b>NARRATIVE ONLY ALLOWED FOR ONE PROGRAM AT A TIME</b>                                                                                                                                                                       |
| Explanation:      | Narrative will only allow one program at a time for Layout Reports.                                                                                                                                                           |
| User Action:      | DCD III will print narrative alongside the first program.                                                                                                                                                                     |
| <b>DCDTB015-I</b> | <b>COBOL SORT CALLED FOR PROGRAM (XXXXXXXX) DUE TO INSUFFICIENT GETMAIN SPACE</b>                                                                                                                                             |

## DCD III – Error Messages

---

|                                        |                                                                                                                                                                                                                                                                                             |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:                           | The size of the COBOL program was larger than the table space necessary to do an internal shell sort, so the COBOL sort was called.                                                                                                                                                         |
| User Action:                           | None.                                                                                                                                                                                                                                                                                       |
| <b>DCDTBC02-D</b>                      | <b>FILE EMPTY - NO MARBLE HEADER</b>                                                                                                                                                                                                                                                        |
| Explanation:                           | The first record of a work file being read does not contain a MARBLE header record.                                                                                                                                                                                                         |
| User Action:                           | Look for possible previous message to this one. Look for an error in the JCL used.                                                                                                                                                                                                          |
| <b>DCDTDC18-E</b>                      | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                                                                                                                                                                         |
| Explanation:                           | An internal table was not large enough to handle the program run through DCD III.                                                                                                                                                                                                           |
| User Action:                           | Rerun the job with a larger region.                                                                                                                                                                                                                                                         |
| <b>DCDUBB01 -D</b>                     | <b>THE NBR OF RECORDS SELECTED BY SRA CTL CARDS EXCEEDS 775</b>                                                                                                                                                                                                                             |
| Explanation:                           | Too many control cards were used for DD CTLCDRA.                                                                                                                                                                                                                                            |
| User Action:                           | Use less control statements.                                                                                                                                                                                                                                                                |
| <b>DCDUD020 -I</b>                     | <b>NO DATA-NAME INTO SORT</b>                                                                                                                                                                                                                                                               |
| Explanation:                           | There were no references in the Procedure Division to data names in the Data Division.                                                                                                                                                                                                      |
| User Action:                           | None.                                                                                                                                                                                                                                                                                       |
| <b>DCDUDA02 -W</b>                     | <b>EARLY EOF ON INFO-FILE-2</b>                                                                                                                                                                                                                                                             |
| Explanation:                           | An early end of file was found on an internal work file.                                                                                                                                                                                                                                    |
| User Action:                           | Check results to see if everything is correct. If in doubt, contact MARBLE Computer, Inc.                                                                                                                                                                                                   |
| <b>DCDUDA11-E<br/>&amp; DCDUDA13-E</b> | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                                                                                                                                                                         |
| Explanation:                           | An internal table was not large enough to handle the program run through DCD III.                                                                                                                                                                                                           |
| User Action:                           | Rerun the job with a larger region.                                                                                                                                                                                                                                                         |
| <b>DCDUFB05-E</b>                      | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                                                                                                                                                                         |
| Explanation:                           | An internal table was not large enough to handle the program being documented.                                                                                                                                                                                                              |
| User Action:                           | Rerun the job with a larger region.                                                                                                                                                                                                                                                         |
| <b>DCDUHB05-E</b>                      | <b>TABLE LIMITS EXCEEDED REGION</b>                                                                                                                                                                                                                                                         |
| Explanation:                           | An internal table was not large enough to handle the program being documented.                                                                                                                                                                                                              |
| User Action:                           | Rerun the job with a larger region.                                                                                                                                                                                                                                                         |
| <b>DCDV3050-I</b>                      | <b>COBOL MEMBER (program name) NOT FOUND</b>                                                                                                                                                                                                                                                |
| Explanation:                           | COBOL program name was not found. This occurs when using PARM option READPDS.                                                                                                                                                                                                               |
| User Action:                           | Check to ensure program name is correctly spelled and is there.                                                                                                                                                                                                                             |
| <b>DCDV3051-D</b>                      | <b>DDNAME (DCDREPDS) MISSING</b>                                                                                                                                                                                                                                                            |
| Explanation:                           | In the JCL, the DDNAME for DCDREPDS is missing.                                                                                                                                                                                                                                             |
| User Action:                           | Check in the JCL to ensure the DDNAME is there and correctly placed.                                                                                                                                                                                                                        |
| <b>DCDV3052-D</b>                      | <b>COBOL MEMBER (program name) NOT FOUND</b>                                                                                                                                                                                                                                                |
| Explanation:                           | COBOL program name was not found. This occurs when using PARM option READPDS.                                                                                                                                                                                                               |
| User Action:                           | Check to ensure program name is correctly spelled and is there.                                                                                                                                                                                                                             |
| <b>DCDV3053-D</b>                      | <b>COPY MEMBER NOT FOUND IN COPYLIB SPECIFIED</b>                                                                                                                                                                                                                                           |
| Explanation:                           | When DCD III attempted to read the COPY member in the COBOL program, it was not found in the partitioned data set specified by DSN= in the COPYLIB DD.                                                                                                                                      |
| User Action:                           | Check the COPY member name to ensure it is valid and that the correct PDS file specified within the DSN= in the COPYLIB DD.                                                                                                                                                                 |
| <b>DCDV3054-D</b>                      | <b>NO RECORDS FOUND IN MEMBER (copy member name)</b>                                                                                                                                                                                                                                        |
| Explanation:                           | When DCD III attempted to resolve this COPY member, it found no records in the member.                                                                                                                                                                                                      |
| User Action:                           | Check the COPY member and make corrections as necessary.                                                                                                                                                                                                                                    |
| <b>DCDV3A02-I</b>                      | <b>WORK FILES FOR PROGRAM (XXXXXXXX) DID NOT HAVE OPTION (XXX) IN LAST (WRITPDS) RUN</b>                                                                                                                                                                                                    |
| Explanation:                           | When the READPDS option is used, all files on the data set referenced with the DDNAME of DCDREPDS must have been created with all the same PARM options, which are used in this run using the READPDS option. These work files did not have those options turned on and are being bypassed. |
| User Action:                           | Ignore these messages or re-create the DCDPDS files using the WRITPDS option along with the option shown in the message.                                                                                                                                                                    |

|                                                                                  |                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDV3A03-E</b>                                                                | <b>WORK FILE FOR PROGRAM (XXXXXXXX) IS INCOMPLETE FOR OPTION (XXX) - THIS OPTION IS BEING TURNED OFF</b>                                                                                                                                                |
| Explanation:                                                                     | See explanation for message DCDV3A02-I. Multiple options for reports were used during this READPDS run and not all the options correspond to what was used during the WRITPDS run. This option is being turned off.                                     |
| User Action:                                                                     | Review process, rethink and possibly re-create DCDPDS file.                                                                                                                                                                                             |
| <b>DCDVA020-D</b>                                                                | <b>DDNAME (DCDPDS) MISSING</b>                                                                                                                                                                                                                          |
| Explanation:                                                                     | In the JCL, the card for DCDPDS is missing.                                                                                                                                                                                                             |
| User Action:                                                                     | Check the JCL to ensure the DD card for DCDPDS exists and is correctly placed.                                                                                                                                                                          |
| <b>DCDX1050-I</b>                                                                | <b>NO RECORDS WERE PASSED INTO THE SORT FOR THE JCL PROC ANALYSIS REPORT</b>                                                                                                                                                                            |
| Explanation:                                                                     | Either no DDNAMEs were found or the entire JCL is missing.                                                                                                                                                                                              |
| User Action:                                                                     | Check to see what JCL was used and examine.                                                                                                                                                                                                             |
| <b>DCDX1A04-C</b>                                                                | <b>OVER 500 EXCLUDE DDNAMES WERE USED-THE REMAINING EXCLUDE DDNAMES ARE IGNORED</b>                                                                                                                                                                     |
| Explanation:                                                                     | Too many control cards were used.                                                                                                                                                                                                                       |
| User Action:                                                                     | Reduce the number of control statements.                                                                                                                                                                                                                |
| <b>DCDX1FR0-C</b>                                                                | <b>EXCESSIVE REFERBACKS (OVER 500) WERE USED IN ONE PROC - THOSE OVER LIMIT ARE IGNORED</b>                                                                                                                                                             |
| Explanation:                                                                     | More referbacks were present in one PROC than this software can handle.                                                                                                                                                                                 |
| User Action:                                                                     | Find out why excessive referbacks were used and see if this number can be reduced. If this is a valid condition, contact Marble Computer.                                                                                                               |
| <b>DCDX1FS0-C</b>                                                                | <b>EXCESSIVE SYMBOLICS (OVER 2000) WERE USED IN ONE PROC - THOSE OVER LIMIT ARE IGNORED</b>                                                                                                                                                             |
| Explanation:                                                                     | More symbolics were present in one PROC than this software can handle.                                                                                                                                                                                  |
| User Action:                                                                     | Find out why excessive symbolics were used and see if this number can be reduced.                                                                                                                                                                       |
| <b>DCDX1QA0-W</b>                                                                | <b>TABLE EXCEEDED - ONLY THE FIRST (nnnnn) PROC NAMES WILL BE SHOWN IN THE LIST OF PROC NAMES</b>                                                                                                                                                       |
| Explanation:                                                                     | A table used to hold PROC names for later printing out the names of PROCs used has been exceeded.                                                                                                                                                       |
| User Action:                                                                     | The main report used to report on the JCL is accurate. The sub-report used to show which PROCs were inputted will be truncated.                                                                                                                         |
| <b>DCDX2050-I</b>                                                                | <b>NO RECORDS WERE PASSED INTO THE SORT FOR THE JCLLIB ANALYSIS REPORT</b>                                                                                                                                                                              |
| Explanation:                                                                     | Insufficient JCL was found or the entire JCL is missing.                                                                                                                                                                                                |
| User Action:                                                                     | Check to see what JCL was used.                                                                                                                                                                                                                         |
| <b>DCDYA030-I<br/>&amp; DCDYB030-I<br/>&amp; DCDYC040-I<br/>&amp; DCDYE030-I</b> | <b>NO DATA-NAMES INTO SORT</b>                                                                                                                                                                                                                          |
| Explanation:                                                                     | For VERB, RECORDS, SRA, or COPY report respectively, no data names were found or sorted. This indicates no entries were found to be reported on for the report selected.                                                                                |
| User Action:                                                                     | No required action.                                                                                                                                                                                                                                     |
| <b>DCDYG040-I</b>                                                                | <b>NO DATA NAMES INTO SORT</b>                                                                                                                                                                                                                          |
| Explanation:                                                                     | There were no records passed in to the CALL Report.                                                                                                                                                                                                     |
| User Action:                                                                     | None.                                                                                                                                                                                                                                                   |
| <b>DCDYH050-I</b>                                                                | <b>NO RECORDS WERE PASSED INTO THE SORT FOR THE CALL HIERARCHY REPORT</b>                                                                                                                                                                               |
| Explanation:                                                                     | No records were passed out to the SORT.                                                                                                                                                                                                                 |
| User Action:                                                                     | None.                                                                                                                                                                                                                                                   |
| <b>DCDYH060-D</b>                                                                | <b>UNABLE TO DETERMINE ONE MAIN PROGRAM - SEE PRIOR MESSAGES - UNABLE TO PRODUCE REPORT</b>                                                                                                                                                             |
| Explanation:                                                                     | When determining the order of hierarchy for the CALL Hierarchy Report, a program was called in a fashion that makes it impossible to complete the hierarchy. For example, program A calls program B, which calls program C, which then calls program A. |
| User Action:                                                                     | Determine why this is happening and either correct or do not run this report.                                                                                                                                                                           |
| <b>DCDYHD17-W</b>                                                                | <b>OVER nnnn DIFFERENT CALLS FOUND IN PROGRAM (xxxxxxx) – EXCESS IGNORED</b>                                                                                                                                                                            |
| Explanation:                                                                     | More CALLs were found in one program than can be handled.                                                                                                                                                                                               |
| User Action:                                                                     | Determine why excess CALLs are used. Possibly try using a DELETE control card to reduce this number.                                                                                                                                                    |

|                                    |                                                                                                                                                                 |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DCDYHD30-C</b>                  | <b>THE HIERARCHY EXCEEDED 23 LEVELS DEEP FOR SYSTEM (xxxxxxx) - EXCESS LEVELS IGNORED</b>                                                                       |
| Explanation:                       | The CALL Hierarchy Report is designed to handle CALLs to a level of 23 deep. CALLs going deeper than that are ignored.                                          |
| User Action:                       | Determine why CALLs are going to this depth. If necessary, contact MARBLE Compute, Inc.                                                                         |
| <b>DCDYHD31-E</b>                  | <b>CANNOT ADD CALL TO (xxxxxxx) AS THIS IS WITHIN CURRENT STRING - SEE NOTE BELOW</b>                                                                           |
| Explanation:                       | When determining the trace of CALLs, a CALL to a program was found that itself is found earlier within this chain.                                              |
| User Action:                       | See following messages which show the order of the trace established so far. Then determine why this is happening and either correct or do not run this report. |
| <b>DCDYHD80-I</b>                  | <b>xxxxxxx xxxxxxxx xxxxxxxx</b>                                                                                                                                |
| Explanation:                       | This message appears after message DCDYHD31-E, showing the order of CALLs which caused that message. If necessary, more than one of these messages will appear. |
| User Action:                       | See error message DCDYHD31-E.                                                                                                                                   |
| <b>DCDYHFA0-E</b>                  | <b>EXCEEDED nnnn PROGRAMS FOR CALL HIERARCHY CHART - WILL BE INCOMPLETE</b>                                                                                     |
| Explanation:                       | The number specified is the maximum number of programs that may be submitted for this report.                                                                   |
| User Action:                       | Determine if a smaller number of programs may be used when running this report.                                                                                 |
| <b>DCDYG040-I</b>                  | <b>NO RECORDS WERE PASSED INTO THE SORT FOR THE CALL ANALYSIS REPORT</b>                                                                                        |
| Explanation:                       | No selected records were found to report on for CALL Analysis Report.                                                                                           |
| User Action:                       | Determine if CALLs were not present, then ignore this message.                                                                                                  |
| <b>DCDYIA04-E</b>                  | <b>CONTROL CARDS FOR CTLCAHCC MUST BEGIN WITH (ADD) OR (DELETE)</b>                                                                                             |
| Explanation:                       | Control cards for the CALL Hierarchy Report begin with either ADD or DELETE.                                                                                    |
| User Action:                       | Correct the control statement in error and resubmit.                                                                                                            |
| <b>DCDYIA06-E</b>                  | <b>FIRST PROGRAM NAME MUST BE ALPHANUMERIC FIELD FROM 1 TO 30 CHARACTERS LONG – BYPASSING</b>                                                                   |
| Explanation:                       | The program name used in a control statement exceeded 30 characters in length.                                                                                  |
| User Action:                       | Enter a correct program name and resubmit.                                                                                                                      |
| <b>DCDYIA08-E</b>                  | <b>THE THIRD FIELD ON CTLCAHCC CONTROL CARDS MUST BE THE CONSTANT (CALLS) – BYPASSING</b>                                                                       |
| Explanation:                       | The third field on control cards when used must be the constant (CALLS).                                                                                        |
| User Action:                       | Determine what format was desired for this control card, correct and resubmit.                                                                                  |
| <b>DCDYIA10-E</b>                  | <b>SECOND PROGRAM NAME MUST BE ALPHANUMERIC FIELD FROM 1 TO 30 CHARACTERS LONG – BYPASSING</b>                                                                  |
| Explanation:                       | Any program name used in a control card must not exceed 30 characters in length.                                                                                |
| User Action:                       | Enter a Correct program name within control statement and resubmit.                                                                                             |
| <b>DCDYIA11-E</b>                  | <b>MORE CONTROL CARDS WERE SPECIFIED THAN PERMITTED – REDUCE NUMBER OF CTL CARDS</b>                                                                            |
| Explanation:                       | More than 500 control statements were submitted for the CALL Hierarchy Report.                                                                                  |
| User Action:                       | Use less control statements.                                                                                                                                    |
| <b>DCDYIE01-E</b>                  | <b>AN EXPECTED FIELD WITHIN A CONTROL CARD IS MISSING</b>                                                                                                       |
| Explanation:                       | The format of the control statement requires one more field and this field is missing.                                                                          |
| User Action:                       | Determine the correct format and resubmit with the correct control card.                                                                                        |
| <b>DCDYK040-I</b>                  | <b>NO DATA-NAME INTO SORT</b>                                                                                                                                   |
| Explanation:                       | There were no records passed into the PARAGRAPH Report                                                                                                          |
| User Action:                       | None.                                                                                                                                                           |
| <b>DCDYL030-I</b>                  | <b>NO DATA NAMES INTO SORT</b>                                                                                                                                  |
| Explanation:                       | There were no records passed out to the DATA Report.                                                                                                            |
| User Action:                       | None.                                                                                                                                                           |
| <b>DCDYN100-I &amp; DCDYN110-I</b> | <b>NO RECORDS WERE PASSED INTO SORT FOR THE TRACING AND ANALYSIS REPORT</b>                                                                                     |
| Explanation:                       | No records were passed to an internal sort. This is an informational message.                                                                                   |
| User Action:                       | It may be that no records were selected, which will cause this message. Check SELECT statements used.                                                           |
| <b>DCDYN115-I</b>                  | <b>NO RECORDS WERE SELECTED FOR PRODUCING (TRACING &amp; ANALYSIS) REPORT</b>                                                                                   |

## DCD III – Error Messages

---

|                   |                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Explanation:      | The SELECTs used did not find any matches within this program.                                                                             |
| User Action:      | If processing only one program, insure the correct SELECT is used.                                                                         |
| <b>DCDYNA01-D</b> | <b>MISSING HEADER RECORD FOR DCDWK09 IN PROGRAM DCDTARPT</b>                                                                               |
| Explanation:      | An internal header record is missing from the internal file name shown.                                                                    |
| User Action:      | Try resubmitting run.                                                                                                                      |
| <b>DCDYNB11-E</b> | <b>OVER 250 SELECTIONS FOUND WITHIN RECORD (record-name)</b>                                                                               |
| Explanation:      | Within one record, more than 250 fields were selected.                                                                                     |
| User Action:      | Go back and change the SELECTs used to select less fields. Be more specific in selecting fields.                                           |
| <b>DCDYPA0-D</b>  | <b>FILES ARE NOT IN SEQUENCE BETWEEN DDNAMES CODEWK01 CODEWK02 &amp; CODEWK06</b>                                                          |
| Explanation:      | Internal work files are not in sequence.                                                                                                   |
| User Action:      | Possible logic error. If several programs, try one program. Try resubmitting.                                                              |
| <b>DCDYPB01-C</b> | <b>INVALID FIELD FOUND - BYPASSING</b>                                                                                                     |
| Explanation:      | A field on an internal work file has a field which should be numeric but is not.                                                           |
| User Action:      | If this happens continuously, contact MARBLE Computer.                                                                                     |
| <b>DCDYPB02-D</b> | <b>MATCHING SOURCE LINE NOT FOUND - RUN ENDING</b>                                                                                         |
| Explanation:      | Internal work files for doing code changing require a source code work file and this file is missing.                                      |
| User Action:      | Possible logic error. Try resubmitting.                                                                                                    |
| <b>DCDYPE03-D</b> | <b>INTERNAL TABLE EXCEEDED - CONTACT MARBLE COMPUTER</b>                                                                                   |
| Explanation:      | An internal table was exceeded.                                                                                                            |
| User Action:      | Contact Marble Computer.                                                                                                                   |
| <b>DCDYPU01-D</b> | <b>INITIAL HEADER RECORDS ON DD CODEWK01, CODEWK02, AND/OR CODEWK06 ARE NOT IN SYNC</b>                                                    |
| Explanation:      | When using one or several programs, internal files used during creating CODE CHANGE records were found to be out of order.                 |
| User Action:      | If using several programs, try fewer programs or just one program. If one program, and this happens consistently, contact Marble Computer. |
| <b>DCDYPX02-D</b> | <b>DDNAME (CODECHAG) MISSING</b>                                                                                                           |
| Explanation:      | A DDNAME (CODECHAG) is missing in the JCL and needs to be present when the Code Changer option is being used.                              |
| User Action:      | Inspect the JCL used and see that the DDNAME is supplied.                                                                                  |
| <b>DCDYT030I</b>  | <b>NO DATA-NAMES INTO SORT</b>                                                                                                             |
| Explanation:      | There were no records passed into the PARAGRAPH Report                                                                                     |
| User Action:      | None.                                                                                                                                      |
| <b>DCDYU030-I</b> | <b>NO DATA-NAMES INTO SORT</b>                                                                                                             |
| Explanation:      | There were no records passed into the PARAGRAPH RANGE Report                                                                               |
| User Action:      | None.                                                                                                                                      |
| <b>DCDYWG04-E</b> | <b>RECORD LOCATIONS ARE OFF</b>                                                                                                            |
| Explanation:      | Record locations are off due to incorrect use of an ( * ) within the control statements.                                                   |
| User Action:      | Check the JCL control statements to ensure it is the correct logic.                                                                        |
| <b>DCDYY030-I</b> | <b>NO DATA-NAMES INTO SORT</b>                                                                                                             |
| Explanation:      | There were no records passed in to the LAYOUT Report.                                                                                      |
| User Action:      | None.                                                                                                                                      |
| <b>DCDZZ099-D</b> | <b>See SYSOUT for MISSING DD or other error</b>                                                                                            |
| Explanation:      | An error has occurred in one of Marble's work files DCDWK01 through DCDWK12                                                                |
| User Action:      | To see more information on this error, look at messages near the end of the SYSOUT DD.                                                     |



Release 3.6

# DCD III

## Installation Guide

*This page intentionally left blank*

**DCDIII  
Installation Guide**

**T A B L E   O F   C O N T E N T S**

|                                                                         |      |
|-------------------------------------------------------------------------|------|
| Introduction .....                                                      | F-5  |
| Overview of 5 steps for installing.....                                 | F-5  |
| Step 1 – Newer Electronic Downloading of DCD III Files .....            | F-5  |
| Older Step 1 – Older method of unloading from a physical cartridge..... | F-7  |
| Loading from a cartridge .....                                          | F-8  |
| Obtaining JCL to Unload Eight Cartridge Files .....                     | F-8  |
| JCL (on INSTALL member) to unload Eight Cartridge Files .....           | F-9  |
| JCL to unload just One or Two Cartridge Files .....                     | F-10 |
| Step 2 - Modifying DCD III PROCs.....                                   | F-11 |
| Step 3 - Creating a Control Record.....                                 | F-13 |
| Step 4 - Testing DCD III.....                                           | F-14 |
| Step 5 - Installing and Testing ISPF Panels.....                        | F-15 |
| Modifying CLISTs, Panels and JCL Skeletons .....                        | F-16 |
| Setting up Skeletons for Compile Mode .....                             | F-19 |
| Testing DCD ISPF Panels Prior to Permanent Installation.....            | F-21 |
| Permanent Installation of the ISPF Panels .....                         | F-21 |

**Addendums**

|                                                               |      |
|---------------------------------------------------------------|------|
| 1. DCD III JCL Procedure Library .....                        | F-22 |
| 2. Use of SYSTRACE and TRACEIN DDs for Customer Support ..... | F-24 |
| 3. Installation Addendum (to be inserted) .....               | F-25 |

*This page intentionally left blank*

## Introduction

This installation guide corresponds to release **3.6** of the Data Correlation and Documentation System (DCD III).

It should be noted that many of the modifications incorporated into this release are the direct result of user request. An example of this is the ISPF Panels added with release 1.5 or the new Tracing & Analysis Report feature added with release 2.4 or the addition of **Indirect References** with release 3.1 and the Unused Data Names and Unused Paragraph reports in release 3.4 **and most recently the newer Electronic Installation of DCD III in release 3.6**. Modifications to support additional user requests are currently under development for future releases. It is our policy to provide requested features where such support is technically and economically feasible.

### ---NOTICE---

Installation of this release of DCD III requires DCDCNTRL control file information for your site that is available from a prior release or may be obtained from a green colored “Installation Addendum” page (See **Green\*** colored page - Addendum 3.)

\*For Trial users, Addendum 3 is a **Pink** colored page.

If you have any problems, suggestions or comments, please contact MARBLE Computer, Inc. at 1-800-252-1400

## Overview of 5 steps for installing

- Step 1 - Obtaining all 8 DCD III files electronically. See **Step 1** just below.
- Steps 2 through 4 - Complete Testing of DCD III without SPF panels using first 2 DCD III files.
- Step 5 - Use of the last 6 files to Install ISPF panels for online running of DCD III.\*

\*The use of our SPF panels is recommended for all use of DCD III with the exception of Compile Mode. See discussion of ‘Compile Mode’ in ‘Alternate Compile Facility’.

### Step 1 – Newer Electronic Downloading of DCD III Files

DCD III new releases are now available for downloading electronically from our web site to a user’s PC for then uploading to the mainframe from there. Complete instructions for downloading to a PC and uploading to user’s mainframe from the PC are provided in this download within a [ReadMe.Txt](#) file.

To obtain the latest release, go to Marble web site, [www.marblecomputer.com](http://www.marblecomputer.com) and look for a link to the ‘DCD’s Latest Software Release’ on the Home Page and click on that link.

After installing the DCD III files successfully, and they are uploaded to your mainframe, then continue with Step 2 in this Install Guide.

**Release 3.6 - 08/01/2008**

*This page intentionally left blank*

## Older Step 1 - Older method of unloading from a physical cartridge

If for some unknown reason, the user finds it impossible to electronically download the new release file, please contact Marble Computer and we will make arrangements to create and send an older format physical cartridge. We first however strongly urge trying the electronic download as it has fewer steps.

The eight files for installation of DCD III with ISPF panels may be provided on a 3480 cartridge. The cartridge will contain eight files. The first two files contain the DCD system necessary for installation **without** ISPF panels. The next six files contain files necessary for the installation of the ISPF panels. When re-installing DCD III without changes to the ISPF panels, only the first file or the first two files need to be loaded. When loading all the files, a member INSTALL (an IEBCOPY unloaded Partitioned Data Set) that resides on the third file of this cartridge may be re-loaded first to make installation easier. The JCL for re-loading this member is shown on the next page.

The third file contains the INSTALL member along with 11 CASE COBOL COPY members which are accessed by the ISPF panels. Ideally the 11 COPY members without the INSTALL member should be moved to a new separate COBOL file with the same attributes as the CNTL library. This new file (CNTL or COBOL) then needs to be named within one of the panels and put into one of the JCL skeletons. Instructions for using this new file name are listed later under the heading, “**Modifying CLISTs, Panels and JCL Skeletons**”.

The 4th through 8th files are all related to ISPF. They should be loaded to their own respective partitioned data sets corresponding to five categories of files used with ISPF panels as follows:

|    | <u>DDNAME</u> | <u>Type</u>  |
|----|---------------|--------------|
| 4. | SYSPROC       | CLIST        |
| 5. | ISPLIB        | Panel        |
| 6. | ISPLIB        | Message      |
| 7. | ISPLIB        | JCL Skeleton |
| 8. | ISPLIB        | LOAD module  |

and then one of two methods may be used to access the members contained within the PDSs:

1. The PDSs may be concatenated with other similar PDSs in the TSO LOGON USER PROC.
2. The members may be moved (copied) from the PDSs here to one of the corresponding PDSs already used in the TSO LOGON USER PROC.

File attributes and space requirements for the eight files are as follows:

| Tape File # | Tape Data-Set Name | DCB Information |       |         | Space Requirements |                  |
|-------------|--------------------|-----------------|-------|---------|--------------------|------------------|
|             |                    | RECFM           | LRECL | BLKSIZE | Tracks*            | Directory Blocks |
| 1.          | DCDIII.LOAD        | U               | 0     | 6144    | 120                | 10               |
| 2.          | DCDIII.CNTL        | FB              | 80    | 3120    | 8                  | 10               |
| 3.          | DCDSPF.CNTL        | FB              | 80    | 3120    | 6                  | 10               |
| 4.          | DCDSPF.CLIST       | VB**            | 255   | 6223    | 2                  | 10               |
| 5.          | DCDSPF.PLIB        | FB              | 80    | 6160    | 25                 | 50               |
| 6.          | DCDSPF.MLIB        | FB              | 80    | 6160    | 3                  | 10               |
| 7.          | DCDSPF.SLIB        | FB              | 80    | 6160    | 6                  | 10               |
| 8.          | DCDSPF.LLIB        | U               | 0     | 6144    | 20                 | 10               |

\* SPACE calculation is based on 3390 device types      \*\*This CLIST file is Variable blocked

Note - If a file transfer program or utility is used to unload the cartridge to disk before IEBCOPY is used, then larger block sizes may be needed during the file transfer program (e.g. DCDIII.LOAD might require 6164 rather than 6144).

### Loading from a Cartridge

For users installing from a cartridge, and also not installing or making changes to the ISPF Panels, the JCL under the heading 'JCL to unload just one or two cartridge files' may be used to load just one or two files.

### Obtaining JCL to Load Eight Cartridge Files

When loading all eight cartridge files, the JCL shown below should be used to unload the third file on the tape to a newly created PDS. One member on that PDS, (INSTALL) may then be used to load all eight files.

```
// JOB
//PRESTEP EXEC PGM=IEBCOPY,REGION=512K
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD SPACE=(CYL,(1,1),UNIT=sysda
//SYSUT4 DD SPACE=(CYL,(1,1),UNIT=sysda
//TAPE DD DSN=DCDSPF.CNTL,DISP=(OLD,PASS),
// LABEL=(3,SL,EXPDT=98000) ,UNIT=TAPE,
// VOL=SER=DCDIII
//DISK DD DSN=user.cntl.pds,
// DISP=(NEW,CATLG,DELTE),
// SPACE=(TRK,(10,5,10),,,ROUND),
// UNIT=sysda,VOL=SER=user01
//SYSIN DD *
COPY INDD=TAPE,OUTDD=DISK
```

Modify fields shown in lower case letters in the UNIT=, DSN= and VOL=SER= keywords as necessary for your installation.

The JCL on this obtained member INSTALL is shown on the next page.



**JCL (on *INSTALL* member) to Load Eight Cartridge Files**

```

// JOB
//COPY PROC
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD SPACE=(CYL,(1,1)),UNIT=sysda
//SYSUT4 DD SPACE=(CYL,(1,1)),UNIT=sysda
//TAPE DD DSN=&TFILE,DISP=(OLD,KEEP),
// LABEL=(&TL,SL,EXPDT=98000),UNIT=(tape,,DEFER),
// VOL=(,RETAIN,SER=DCDIII)
//DISK DD DSN=&DFILE,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(&DSPACE),,,ROUND),
// UNIT=sysda,VOL=SER=&DVOLSER
//SYSIN DD DSN=&&TEMPCTL,DISP=(OLD,PASS)
// PEND
//*
//STEPS EXEC PGM=IEBGENER,REGION=128K
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD *
// COPY INDD=TAPE,OUTDD=DISK
//SYSUT2 DD DSN=&&TEMPCTL,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),SPACE=(TRK,(1,1))
//*
//STEP1 EXEC COPY,TFILE='DCDIII.LOAD',TL=1,DSPACE='120,10,10',
// DVOLSER=userxx,DFILE='userzz.DCDIII.LOAD'
//*
//STEP2 EXEC COPY,TFILE='DCDIII.CNTL',TL=2,DSPACE='8,2,10',
// DVOLSER=userxx,DFILE='userzz.DCDIII.CNTL'
//*
//STEP3 EXEC COPY,TFILE='DCDSPF.CNTL',TL=3,DSPACE='6,2,10',
// DVOLSER=userxx,DFILE='userzz.DCDSPF.CNTL'
//*
//STEP4 EXEC COPY,TFILE='DCDSPF.CLIST',TL=4,DSPACE='2,1,10',
// DVOLSER=userxx,DFILE='userzz.DCDSPF.CLIST'
//*
//STEP5 EXEC COPY,TFILE='DCDSPF.PLIB',TL=5,DSPACE='25,4,50',
// DVOLSER=userxx,DFILE='userzz.DCDSPF.PLIB'
//*
//STEP6 EXEC COPY,TFILE='DCDSPF.MLIB',TL=6,DSPACE='3,1,10',
// DVOLSER=userxx,DFILE='userzz.DCDSPF.MLIB'
//*
//STEP7 EXEC COPY,TFILE='DCDSPF.SLIB',TL=7,DSPACE='6,2,10',
// DVOLSER=userxx,DFILE='userxx.DCDSPF.SLIB'
//*
//STEP8 EXEC COPY,TFILE='DCDSPF.LLIB',TL=8,DSPACE='20,3,10',
// DVOLSER=userxx,DFILE='userzz.DCDSPF.LLIB'

```

MODIFY fields as necessary shown in lower case in the UNIT=, DVOLSER= and DFILE=keywords. Replace JOB with a valid JOB card. Submit this JCL to load the eight files.

### JCL to Load Just One or Two Cartridge Files

For users re-installing DCD III without changes to the ISPF Panels, the following JCL may be used to load the first one or two files. If there are no changes to the user JCL, then only the first file needs to be loaded.

```
// JOB
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD SPACE=(CYL,(1,1)),UNIT=sysda
//SYSUT4 DD SPACE=(CYL,(1,1)),UNIT=sysda
//TAPE DD DSN=DCDIII.LOAD,DISP=(OLD,PASS),
// LABEL=(1,SL,EXPDT=98000),UNIT=tape,
// VOL=(,RETAIN,SER=DCDIII)
//DISK DD DSN=new.user.loadlib,
// DISP=(NEW,CATLG,DELETE)
// SPACE=(TRK,(120,10,10),,,ROUND),
// UNIT=sysda,
// VOL=SER=user01
//SYSIN DD *
// COPY INDD=TAPE,OUTDD=DISK
//
//STEP2 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD SPACE=(CYL,(1,1)),UNIT=sysda
//SYSUT4 DD SPACE=(CYL,(1,1)),UNIT=sysda
//TAPE DD DSN=DCDIII.CNTL,DISP=(OLD,PASS),
// LABEL=(2,SL,EXPDT=98000),UNIT=tape,
// VOL=(,RETAIN,SER=DCDIII)
//DISK DD DSN=new.user.cntllib,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(8,2,10),,,ROUND),
// UNIT=sysda,
// VOL=SER=user01
//SYSIN DD *
// COPY INDD=TAPE,OUTDD=DISK
//
```

Modify fields shown in lower case letters in the DSN=, UNIT= and VOL= keywords as necessary for your installation.

The second file is a JCL library, which contains PROCs, which are used for the execution of DCDIII.

Replace JOB with a valid JOB card. Submit this JCL to load one or two files.

## Step 2 - Modifying DCD III PROCs

(Continue here after electronically uploading the new DCD III files)

A list of all members on the DCDIII.CNTL file are listed under the heading, “**DCD III JCL Procedure Library**”, in this section.

The following are PROCs, which require modification:

1. COMACL for executing the Alternate Compile Listing Facility in Compile Mode.
2. DCDACL for executing the Alternate Compile Listing Facility in Independent Mode.
3. LIBACL for executing the Alternate Compile Listing Facility if Librarian is used in your shop.
4. PANACL for executing the Alternate Compile Listing Facility if Panvalet is used in your shop.
5. DCDCOBOL for executing the Other COBOL Reports Facility on one or more COBOL programs.
6. LIBCOBOL for executing the Other COBOL Reports Facility on one or more COBOL Programs if Librarian is used in your shop.
7. PANCOBOL for executing the Other COBOL Reports Facility on one or more COBOL programs if Panvalet is used in your shop.
8. MBRFETCH for selecting members from a partitioned data set to produce reports from the Other COBOL Reports Facility.
9. DCDJCL for executing the JCL PROC Analysis Reports Facility.

These PROCs are listed in this manual in their appropriate section (i.e. - Alternate Compile Listing, Tracing & Analysis and Other COBOL Reports or JCL PROC Analysis).

After selecting the PROCs which will be used in your shop, go into each of the PROCs individually and:

- 1) Insert STEPLIBs where noted to point to the load library where the DCD III, Panvalet or Librarian modules reside.

(Continued on next page...)

- 2) If known COPY member libraries are used in your shop, insert COPYLIB DDs where noted to point to these COPY libraries. These should match to what is used in the SYSLIB step of your installations COBOL compile PROCs.
- 3) If SYS1.SORTLIB is not the standard library for sort routines in your shop, modify as appropriate.
- 4) Modify the CONTROL DD line to the appropriate CONTROL data set (see the Installation Addendum and see – “**Step 3 - Creating a Control Record**”).
- 5) At the beginning of each PROC make appropriate changes to symbolic parameters that are present on the last six to eight lines. The symbolic parameters before this point should generally be left as they are.

The following symbolic parameters in all PROCs affect DCD III resource utilization:

| <u>SYMBOLIC</u> | <u>DEFAULT</u> | <u>MINIMUM</u> | <u>MEANING</u> |
|-----------------|----------------|----------------|----------------|
| REG             | varies         | varies         | REGION SIZE    |
| BUF             | 5              | 2              | DCB BUFNO      |
| SORTREG         | 600000         | 600000         | SORT REGION    |

Decreasing these values increases run time, while decreasing PAGE and/or storage requirements. Do not increase SORTREG above 1,000,000 and do not increase BUF above 9.

For installations concerned about storage resources, the minimum REGION required to generate DCD III reports is 3684K for the Alternate Compile Listing Facility and 5632K for all other DCD III reports. If using this minimum, the BUF and SORTREG parameters must be set to “2” and “600000”, as indicated above. **Using too small a region may cause an OC4.**

When the PROCs are modified and tested, they should be stored on a system PROCEDURE library for general use.

- 6) Consider removing the SORT work files (SORTWK01-03) and SORTLIB from those PROCs which use them. Today, many SORT packages do dynamic allocations of these files. When this is done, they are not needed within our PROCs.

### **Step 3 - Creating a Control Record**

To create a control record, set up one 80-character record as a member named DCDCNTRL, on a permanent JCL or control statement partitioned data set. The LRECL on the PDS must be 80. Fill in only the first thirteen characters on this control record. Leave columns 14 through 80 blank.

See the Installation Addendum to obtain the date and related control characters for this control record.

If this is a new installation or the Installation Addendum is lost, contact MARBLE Computer, Inc., at 1-800-252-1400 for this control information.

### Step 4 - Testing DCD III

At a minimum, adequate testing should include the COMACL, DCDACL, DCDCOBOL and DCDJCL PROCs. If Librarian or Panvalet is used, then the PROCs LIBACL and LIBCOBOL or PANACL and PANCOBOL should also be tested. To test DCD III, do the following:

- 1) Modifying the member DCDCNTRL on the file DCDIII.CNTL just loaded with the one line control statement information provided in the Installation Addendum (this must be done first).
- 2) Pull in the member TDCDACL from the CNTL PDS file unloaded from the cartridge. This contains JCL to test the Independent Mode PROC DCDACL. Modify the JOB statement and modify the DSNs to point to the two PDSs just unloaded from the tape. Submit the job and check the results for COND CODE of 0000.
- 3) Pull in the member TCOMACL from the file used above. Modify the JOB statement and DSNs as above. As necessary, change the COBUC PROC name if this is not a standard PROC for doing a COBOL compile at your installation. Submit the job and check the results for the following COND CODEs:

|    |          |             |              |
|----|----------|-------------|--------------|
| 1. | COB      | 0000        |              |
| 2. | DCD      | <b>0001</b> |              |
| 3. | RETCMPLR | *           | NOT EXECUTED |
- 4) Pull in the member TDCDCOBO from the file used above. This contains JCL to test the other reports produced through DCD III. Modify the JOB statement and modify the DSNs to point to the two PDSs just loaded from the tape. Submit the job and check the results for a COND CODE of 0000.
- 5) Pull in the member TDCDJCL from the CNTL PDS file loaded from the cartridge. Modify the JOB statement and modify the DSNs to point to the two PDSs just loaded from the cartridge. Submit the job and look for a COND CODE of zeros in both steps.
- 6) If Librarian or Panvalet are used, refer to Examples 2 and 3 under the Alternate Compile Listing Facility section and at Examples 1 and 2 under the Other COBOL Reports Facility section in this manual for suggested JCL to test these PROCs. Look for a COND CODE of 0000 from the DCD step.
- 7) See Example 4 under the Tracing & Analysis and Other COBOL Reports Facility section for suggested JCL to test the MBRFETCH PROC.

## Step 5 - Installing and Testing ISPF Panels

### General Notes

#### Naming conventions:

1. All DCD III ISPF panels begin with the letters DZ3.
2. Several user PROFILE names are built and kept after execution of DCD III ISPF panels. Each of these profile names begin with the letters DZ3 followed by the letter A, B, C, D, H, J, or I.
3. The CLIST member DZ3DCD1 assigns an application-id of \$DCD to these names.

#### Variable Blocked CLIST file:

The CLIST file copied from the cartridge in step 4 of the JCL shown on page F-8, is a variable blocked file. If CLISTs are fixed block at your installation, then a new CLIST file should be allocated and catalogued as fixed block and the file loaded from the cartridge should be copied to this new file. The variable blocks CLIST file should be deleted.

#### Modifying members on three of the files:

Before these members can be used, changes need to be made to some CLIST, Panels and JCL members. These changes are covered in the next two headings:

1. Modifying necessary CLISTs, Panels, and JCL Skeletons
2. Setting up Skeletons for Compile Mode

## Modifying CLISTs, Panels and JCL Skeletons

The following members need to be modified during installation.

### CLISTs

- DZ3DCD1 - Modify the five file names within the five LIBDEF statements to the permanent data set names (DSNAMEs) loaded from files 4 through 8 from the installation cartridge.

### Panels

- DZ3HC100 - Replace xxx within this panel with the name of the file (DSNAME) loaded from file 3 from install cartridge. This DSNAME should reflect the final file, which will contain 11 CASE COPY members. (See -Addendum 1- in this section, and see CASEMBRS within **DCD III JCL Procedure Library** there.)
- DZ3Z0801 - Insert within the single apostrophes within the ) INIT section of this panel, one or more production COPY libraries that should always be searched when looking for COPY libraries. Look at the SYSLIB DDNAME within standard compile procs for a list of these DSNAMEs. (These DSNAMEs may be changed and added to by the programmer while using the SPF panels.)
- DZ3Z0806 - Change the literal from (Y) to (N) within the )INIT section of this panel as necessary to block out COMPILE, PANVALET, and LIBRARIAN respectively. If LIB='Y', consider EXT='Y' also, to block out the use of -SEL commands and only allow -EXTRACT commands to be built by DCD III for use within Librarian to bring in programs.

#### - Notes on panel DZ3Z0806 -

1. Most installations set up compile PROCs calling DCD separate and exclusive from the SPF panels here. Additional effort may be used here to allow (not block out) COMPILE mode within DCD SPF panels. **However**, it may be appropriate to initially block out COMPILE mode from SPF panels and add later, if specifically requested by users.
2. Librarian and Panvalet should be allowed or blocked out depending on their use within your shop. If they are used at your installation, they should be allowed here.

- DZ3A0455 - If COMPILE mode is allowed within SPF panels, this panel needs to be modified to allow selection of needed skeletons for COMPILE mode. (This panel, if allowed, should be modified later following the instructions given under the heading, "**Setting up Skeletons for Compile Mode**".)

(Continued...)



### Skeletons

- DZ3CNTRL - Modify this skeleton to point to the DSNNAME which contains the control password for running this software. (See, “**Step 3 - Creating a Control Record**” and ‘Installation Addendum’, for determining the name of this DSNNAME.)
- DZ3STEP - Modify this skeleton to point to the DSNNAME which contains the load modules of the DCD III system. This file was loaded from the first file on the install tape.
- DZ3LSTEP - If Librarian mode is allowed and a STEPLIB is required for LIBRARIAN, modify this skeleton to point to the DSNNAME needed to invoke LIBRARIAN and delete the )CM and blanks ahead of //STEPLIB.
- DZ3PSTEP - If Panvalet mode is allowed and a STEPLIB is required for PANVALET, modify this skeleton to point to the DSNNAME needed to invoke PANVALET and delete the )CM and blanks ahead of //STEPLIB.
- DZ3SORT - If DDNAMEs SORTMESS and SORTLIB are required to be included within the JCL for sort programs at your installations, include these within this skeleton by deleting the )CM and blanks ahead of //SORTMESS and //SORTLIB and verify the accuracy of the DDs provided.
- DCDCASEC - Modify this panel to reflect a correct DSNNAME within the SYSUT1 DDNAME. Replace xxx with the name of the 3rd file loaded from the DCD III install cartridge. (This must be the same file name inserted in panel DZ3HC100 mentioned earlier in this heading.) This DSN may not be changed by the user. Proper testing using this DSN is essential. Testing of this DSN is covered under the heading, “**Testing DCD ISPF Panels Prior to Permanent Installation**”.
- DCDJOB - ADD needed JOBLIB, PROCLIB or other leading JOB JCL where noted in this skeleton. (See standard JCL used at your installation for these lines.) The JCL added here should be the same JCL needed for all JOBS at your installation. Optionally, these JOBLIB, PROCLIB or other JCL may instead be added individually after the )IM DCDJOB lines within the following procs:
1. DCDCASEC
  2. DZ3OCRFB
  3. DZ3ACLF
  4. DCDJC

(Continued..)

- DZ3PANVA - If Panvalet is allowed and supported, verify that the program name DZ3APANR used for executing Panvalet is correct and check if a region parameter is required.
- DZ3LIBRA - If Librarian is allowed and supported, verify that the program name DZ3ALIBR used for executing Librarian is correct. Check if a region parameter is required and check if the PARM field is correct. See modifying panel DZ3Z0806 under **Panels** for forcing users to use EXTRACT only when using Librarian.

Changing UNIT=SYSDA in Skeletons

If UNIT=SYSDA is not valid, make changes for SYSDA globally in the following procs:

1. DZ3PANVA
2. DZ3APANR
3. DZ3LIBRA
4. DZ3ALIBR
5. DZ3MBRFE
6. DZ3RMBRF
7. DZ3OCRF
8. DZ3ACLF
9. DCDJCL

Compile Mode Skeletons DZ3COM01 through DZ3COM09

If compile mode is used, within the ISPF panels, necessary skeletons DZ3COM01 through DZ3COM09 are built and modified using skeletons DZ3COM0A as a guide.

These skeletons, if used, should be built following the instructions given under the heading, **“Setting up Skeletons for Compile Mode”**.

### Setting up Skeletons for Compile Mode

One panel needs to be modified, and one to several skeletons need to be built, if installation is to include building COMPILE JCL streams from DCD III panels.

#### - Recommendation -

**It is recommended that COMPILE mode be blocked out from these SPF panels and have users use COMPILE/DCD procs in batch mode or invoke COMPILE with DCD PROCs from the other panels.** This invokes the COBOL compiler in the manner currently used by programmers for doing their COBOL compiles. Examples of combining Compile and DCD procs are given in the heading, “JCL Examples within the Aternate Compile Listing”, section of this manual. If skeletons are added here or whenever standard COMPILE procs that also use DCD are modified, these skeletons will also have to be changed.

Panel DZ3A0455 should be looked at initially, then modified after all necessary skeletons are created. Instructions are given later.

Skeletons DZ3COM0A are provided as a sample for creating other necessary skeletons and should not be modified. The skeletons created must be named DZ3COM01, then DZ3COM02 in ascending order, not to exceed DZ3COM09.

Whatever compile and DCD combinations of PROCs are used in batch mode must also be set up here. For instance, if both COBOL 74 and the newer VS COBOL II are used, then a minimum of two skeletons must be used, one for each separate PROC executed. Separate skeletons may also be added for COMPILE alone, COMPILE with LINKAGE EDITOR, COMPILE with CICS, SQL, etc... A review of existing COMPILE PROCs used can be used to organize a list of the skeletons required. Also, it may be helpful to review the underlined lines below on exclusion of Librarian, Panvalet and DCD JCL from selected PROCs before organizing a list of these skeletons.

#### Building needed SKELETONs

Build as many skeletons as are needed for your installation. After copying DZ3COM0A to DZ3COM01 (or DZ3COM02 or DZ3COM03 etc...), the following modifications need to be made to the new skeleton: (Look at a copy of DZ3COM0A before continuing.)

1. Replace COBxxxxx on the //COB line with the name of a valid PROC currently used for compiling programs at your installation. The PROC may include steps for doing a pre-compile of CICS, SQL or others and may include steps for doing post compiler steps, such as running the Linkage Editor or software packages i.e., the CA-Optimizer. The PROC must not contain LIBRARIAN steps or Panvalet steps ahead of the COMPILE step. Also, the PROC must not contain JCL to execute DCD III. THE JCL for Librarian or Panvalet, if used, and the JCL for DCD III execution is built separately by the other skeletons, (in conjunction with the panels and associated programs which controls the Panels).

2. If CICS, SQL or other pre-processor steps are needed and are not within the selected COMPILE PROC, then JCL should be added ahead of the //COB line mentioned above to invoke the necessary JCL or PROC. Insure that any JCL or PROCs added do **NOT** invoke Panvalet OR Librarian steps. Also, see number 5 below for modifying the 4 sets of ) SEL through ) ENDSEL lines.
3. Modify the //COBOL.SYSLIN line to insert the name of a partitioned data set that routinely holds object members from COBOL compiles. Leave (&ALINPRG) as it is, unless the object library is a sequential file, in which case it may be removed. If an object file is not created, this line may be removed. Check compiler options carefully before removing this line.
4. Leave the //COB.SYSPRINT line as it is with one exception. The LRECL of 121 and the BLKSIZE of 1210 may need to be changed to 133 and 1330 respectively if VS COBOLII compilers or other newer compilers where SYSPRINT have a record length of 133. Leave DISP=(MOD,PASS) as is. Do NOT convert to New,Pass.
5. Leave the 4 sets of ) SEL & ATYPINP = x through )ENDSEL as they are unless pre-processor steps are involved. These 4 sets of statements provide the COBOL input. If pre-processor steps are involved, change the //COB.SYSIN to //PRE-STEP.INPUTDD to direct the input COBOL to those steps and then insure that correct JCL is provided to bring the output COBOL from the pre-processor steps into the SYSIN dd of the COB step.
6. If a SYSLIB dd is provided within the COMPILE PROC, then remove the //COB.SYSLIB within this skeleton. If a SYSLIB dd is not provided within the COMPILE PROC, then modify the //COB.SYSLIB to include concatenated DDs for all COPY libraries needed by the compile and fill in valid DSNAMES.
7. If CA-Optimizer and/or Linkage Editor steps are wanted, add JCL for these at the end of the skeleton, provided they are not already included in other invoked PROCs. The JCL added may invoke other PROCs provided that DCD is not invoked from these additional PROCs.

### **Modifying Panel DZ3A0455**

Make the following modifications to panel DZ3A0455 after all necessary skeletons have been built:

1. Add, change and delete the current four lines beginning, %1, %2, %3 and %4, so that there is one line for every skeleton built. The numbers 1,2,3,4,5, etc... match the last number of the skeletons built. For instance, 1 matches to skeleton DZ3COM01. Insert appropriate documentation describing the COMPILE PROC next to each numbered line.
2. Modify the VER line next to the bottom line to remove those numbers from 0 through 9 that do not match those lines added above.

### **Testing DCD ISPF Panels Prior to Permanent Installation**

Concatenate the five PDSs into a copy of the TSO LOGON USER PROC. Re-log using the newly created PROC. Go into Dialog Test and invoke the DCD panels using the name DZ3DCD. With all the numerous options, control statements and features that could be tested, a full test of all features is impractical. Therefore, each installer should satisfy himself/herself that the panels are working after submitting at least one batch job in each of the four main options from the main panel DZ3DCD. See the following comments:

In the ACLF section, when testing COMPILE Mode, a return code of 1 from the DCD step is normal. The following RETCMPLR step should not be executed during a successful run.

In the CASE files section, (one of the four main options), option 5- Copy a selected COPY member- must be tested to verify the DSName entered in skeleton DCDCASEC. Testing may be made with any one of the eleven CASE files. An output DSN must be provided for copying.

### **Permanent Installation of the ISPF Panels**

Permanent installation may be done after the DCD members from the five PDSs have been made available within the appropriate DD names within the user's TSO LOGON USER PROC. For permanent installation, find an appropriate selection panel that currently exists with room for at least one entry and add a line similar to the following to that panel:

%n      +Data Correlation & Documentation for COBOL programs.

where n is a letter or number and then add an appropriate entry in the) PROC section of that panel as follows:

n,'CMD(DZ3DCD1)'

Re-test to see that the DZ3DCD panel is brought in correctly.

- Addendum 1 -

**DCD III JCL Procedure Library**

DCD III.CNTL contains the following members:

|                     |                                                                                                                                 |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------|
| CASEMBRS            | Contains 11 (eleven) CASE COPY members, all separated by the keyword CASE. See CASE files for use.                              |
| COBOLPGM & COBOLPG2 | Small COBOL programs which may be used for testing DCD III.                                                                     |
| COMACL              | Sample PROC for executing DCD III in Compile Mode.                                                                              |
| DCDACL              | Sample PROC for executing DCD III in Independent Mode.                                                                          |
| LIBACL              | Sample PROC for executing DCD III in Independent Mode, taking the source program from a Librarian library.                      |
| PANACL              | Sample PROC for executing DCD III in Independent Mode, taking the source program from a Panvalet library.                       |
| DCDCNTRL            | Password control record. This must be modified before doing any testing.                                                        |
| DCDCOBOL            | Sample PROC for executing DCD III to produce Other COBOL Reports.                                                               |
| LIBCOBOL            | Sample PROC for executing DCD III to produce Other COBOL Reports, taking the source programs from a Librarian library.          |
| PANCOBOL            | Sample PROC for executing DCD III to produce Other COBOL Reports, taking the source program from a Panvalet library.            |
| DCDJCL              | Sample PROC for producing JCL PROC Analysis Reports from JCL PROCs.                                                             |
| INVMAST1            | A test COPY member used by test program COBOLPGM.                                                                               |
| MBRFETCH            | Sample PROC for pulling in members from a partitioned data set. These members may be COBOL programs, COPY members or JCL PROCs. |
| SELECT              | A file used in testing Tracing & Analysis with test program COBOLPGM                                                            |
| TCOMACL             | Testing JCL for Compile Mode and the COMACL PROC                                                                                |
| TDCDACL             | Testing JCL for the DCDACL PROC and Independent Mode                                                                            |
| TDCDCOBO            | Testing JCL for the DCDCOBOL PROC for producing Other COBOL Reports                                                             |
| TDCDJCL             | Testing JCL for the DCDJCL PROC for producing JCL PROC Analysis Reports                                                         |

Nine of the members are **PROCs**, which should be tailored for use at the user's installation.

Four of the members (**TCOMACL**, **TDCDACL**, **TDCDACL**, **TDCDCOBO** and **TDCDJCL**) are JCL for testing four of the **PROCs**.

One of the members (**COBOLPGM**) is a sample **COBOL** program, which is accessed by the three members **TCOMACL**, **TDCDACL** and **TDCDCOBO**.

One of the members (**DCDCNTRL**) is an incomplete control record, which must be modified before any testing is done.

For testing the Librarian or Panvalet **PROCs**, consult the examples mentioned under, “**Step 4 - Testing DCDIII**”, and fill in necessary information for Librarian or Panvalet data set name, valid COBOL program name and Librarian password. If **COPY** members are present, include a **//DCD.COPYLIB DD** as shown in Example 2 in either the Alternate Compile Listing Facility section, and the Tracing & Analysis and Other COBOL Reports Facility section.

### **Use of SYSTRACE and TRACEIN DDs for Customer Support**

This page is provided as informational and should not normally require user attention.

A de-bugging language for looking at the intermediate files used within DCD III may be used by adding two additional DD's.

These two DD's are explained below:

```
//DCD.SYSTRACE DD SYSOUT=*,DCB=BLKSIZE=133
```

This is used as a print file for printing out the de-bugging information.

```
//DCD.TRACEIN DD *
AFTER progname PRINT {ALL or nnnn} RECORDS FROM DCDWKnn
/*
```

This DD is used to input 1 to 25 AFTER commands to direct the printing of works files after the execution of internal programs within the DCD III system.

The progname and number (n) within the AFTER command need to be provided by MARBLE Computer, Inc.

No other options need to be turned on to invoke this feature.



Addendum 3 -

DCD III  
INSTALLATION ADDENDUM  
TO BE INSERTED HERE

(For ACTIVE Users – Keep GREEN Installation Addendum)

(For TRIAL Users – Keep Pink Installation Addendum,  
and if purchased, replace with GREEN Installation Addendum)

**(For Active Users)**

**You already have this password on file if you are an "Active" user of DCDIII. If you require assistance in obtaining a password, please call our office at 1-800-252-1400. An account number will be required.**

DCD III will not run unless the following is done:

Modify the DSN in the //CONTROL DD line in all of the DCD III PROCs with the exception of MBRFETCH. Make this DD point to a member of a PDS containing a date record shown below.

An example showing control statement input for illustration only of this feature along with the format of date and password are shown below:

```
//CONTROL DD *
012001-XXXXXX
```

**(this control information is obtained from Marble Computer, Inc.)**

End of Installation Section

**INDEX****- 0 -**

01 Record Report B-5, B-14, B-45  
B-58, B-59  
01 Record Report, Example running  
B-33

**- A -**

ASKIP/NOASKIP PARM Option A-15,  
A-19

**- B -**

BAS PROC Symbolic A-19  
Basic Paragraph Information File  
(CASE) C-24, C-25, C-26  
BASIS, Error Messages E-36, E-37, E-38,  
  
BASIS, Supported by DCD III A-10, A-24  
BASIS/NOBASIS PARM Option A-15,  
A-19  
BUF, Modifying in DCD III PROCs  
(see also BUFNO) F-9  
BUFNO option A-24, B-37, E-30,  
F-9  
BYPASS control statements B-30

**- C -**

CA-Optimizer, Compile Mode  
Considerations A-6  
CA-Optimizer, PARM Options (see  
CA1, CA2, CA3)  
CA-Optimizer, Supported by DCD III  
A-10 A-11 A-30  
CA-Optimizer, Use with NRSPACE  
Option A-17

CA1 PARM Option A-10 A-15, A-16  
A-19  
CA1 Report Example A-13  
CA2 PARM Option A-10, A-15, A-16,  
A-17, A-19  
CA2 Report Example A-13  
CA3 PARM Option A-11, A-15, A-16  
A-19  
CALL Analysis Report B-5, B-12, B-45,  
B-46, B-47  
CALL Analysis Report, Example  
Running B-31  
CALL Statement Reports IV. A-15  
A-30, A-35, A-36  
CALL=PROC Symbolic A-19, B-17, C-23  
CALL=PROC Symbolic, Examples using  
A-22, A-47, A-48, B-31  
CALL/NOCALL PARM Option (see also  
CALL Statements Report,  
CALL Analysis Report)  
A-15, A-19, B-12, B-17  
CASE (section) C-1 through C-47  
CASE Files, Rules and restrictions in  
producing C-5  
CASE Files, Sample JCL to access and  
print C-6  
Catalogued JCL Procedures (see PROCs)  
CDATADV=PROC Symbolic A-19, C-23  
CDATADV=PROC Symbolic, Examples  
using A-21, A-22  
CICS, Fields in CASE files C-16, C-18,  
C-20, C-21, C-24, C-28  
CICS PARM Option A-16, A-19, B-12,  
B-17  
CICS, Supported by DCD III A-10, A-11,  
B-9  
CLIST, No longer supported V.  
CLN=nn PARM Option A-16, A-19  
CLN=nn PARM Option, Error Messages  
E-30, E-35  
COAID/NOCOAID PARM Option (see  
also COBOL/Aid) A-16,  
A-19

- COBOL 68 and earlier (see Version 3  
COBOL, VR3/NOVR3  
PARM Option)
- COBOL 68, Error Message E-36
- COBOL 68, Supported by DCD III  
A-10, A-11, B-9
- COBOL 74, Error Message E-36
- COBOL 74, Supported by DCD III  
A-10, A-11, B-9
- COBOL 85 (see VS COBOL II,  
VS/NOVS2 PARM  
Option)
- COBOL/Aid, Compile Mode  
Considerations A-6
- COBOL/Aid, PARM Options (see CA1,  
CA2, CA3, COAID/  
NOCO Aid)
- COBOL/Aid, Supported by DCD III  
A-10, A-11, A-17, A-30
- Code Changing (Y2K) B-16, B-73
- CODEWK01-06 work files B-36, B-41
- CODECHAG work file B-36, B-41
- COMACL PROC (Compile Mode) A-5  
A-8, A-24, A-25, A-26
- COMACL PROC, Example using A-22
- COMACL PROC, Modifying and testing  
at installation F-8, F-9,  
F-11, F-12, F-13
- COMLCNT=Proc Symbolic A-19
- Compile Mode (see also COMACL PROC  
and COMPILE PARM  
Option) A-5, A-6, A-8,  
A-24
- Compile Mode, Error messages E-35, E-36
- Compile Mode, Used with BASIS A-10
- Compile Mode, Used with CA-Optimizer  
A-10
- Compile Mode, Used with CICS A-11
- Compile Mode, Used with COBOL 68  
and earlier A-11
- Compile Mode, Used with COBOL 85  
A-11
- Compile Mode, Used with DB2 A-12
- Compile Mode, Used with DL1 A-12
- Compile Mode, Used with Librarian  
A-12
- Compile Mode, Used with NOSOURCE  
PARM Option A-17
- Compile Mode, Used with Panvalet A-12
- Compile Mode, Used with SRESOLVE  
PARM Option A-17
- Compile Mode, Used with STOP PARM  
Option A-18
- Compile Mode, Used with VR3 PARM  
Option A-18
- Compile PARM Option (see also Compile  
Mode) A-8, A-16
- Compiler cross-reference (see Cross-  
Reference Compiler)
- Compiler options (see LISTER, NUM/  
NONUM, SOURCE)
- Compiler reports (see PMAP and DMAP)
- Compiler source listing (see Source listing,  
Compiler)
- Computer Aided Software Engineering  
(see CASE)
- COND CODE (see Condition Code)
- COND Parameter (see JCL COND  
Parameter)
- Condition Code A-8, A-18, B-15, F-11
- Control Cards for Data Dictionary  
Interface File B-12, B-78  
B-79, B-80, C-41, C-42,  
C-43
- Control Cards for excluding DDNAMEs  
D-10
- Control Cards for Layouts B-12, B-22,  
B-23, B-37, B-45, E-24
- Control Cards for MBRFETCH PROC  
B-8, B-24, B-25, B-26  
B-27, B-29, B-32, B-35,  
B-37, B-38
- Control Cards for selection of PROCs  
D-7, D-8, D-9
- Control Cards SRA Report B-15, B-18  
B-33, B-34
- 1.

Control Statements Tracing & Analysis  
    B-15, B-16, B-28  
CONTROL DD (see also Control Record)  
    A-24, B-37  
Control Record, Error messages  
    E-30, E-31  
Control Record, Modifying at  
    installation F-9, F-10,  
    F-11, F-12, F-13  
Control Record, Sample JCL for  
    A-23, B-36  
COPY Analysis Report B-5, B-10,  
    B-12, B-45, B-48, B-49  
COPY Analysis Report, Error message  
    E-40  
COPY Analysis Report Example  
    Running B-31  
COPY Statements Report IV, A-12,  
    A-16, A-30, A-35, A-36,  
    A-48  
COPY=PROC Symbolic A-19, B-17,  
    C-23  
COPY=PROC Symbolic, Examples  
    Using A-22, A-48, B-6,  
    B-31  
COPY/NOCOPY PARM Option (see  
    Also COPY Statements  
    Report, COPY Analysis  
    Report) A-16, A-19, B-12  
    B-17  
CPROC DV=PROC Symbolic A-19, C-23  
CPROC DV=PROC Symbolic, Examples  
    Using A-22, A-49  
Cross-reference, compiler IV, A-6, A-30,  
    A-31, A-40, A-42

## **- D -**

Data Analysis Information Files (CASE)  
    C-11, C-12, C-13  
Data Dictionary Interface File B-5, B-12,  
    B-17, B-78, B-79, B-80  
Data Dictionary Interface File (CASE)  
    C-5, C-41, C-42, C-43  
Data Dictionary Interface File, Example  
    Producing B-34  
Data Division Condensed Cross-Reference  
    Report IV, A-6, A-7, A-16  
    A-18, A-30, A-40, A-41  
Data Division Condensed Cross-Reference  
    Report, Examples running  
    A-21, A-22, A-49  
Data Division Information File (CASE)  
    C-7, C-8, C-9  
DATA=PROC Symbolic (see also DATA/  
    NODATA PARM Option)  
    B-17, C-7, C-11, C-38  
DATA=PROC Symbolic, Examples using  
    B-6, B-7, B-8, B-28, B-29,  
    B-31, C-6  
DATA/NODATA PARM Option (see also  
    System Data Name Cross-  
    Reference) B-12, B-17,  
    B-43, B-45, C-7, C-11,  
    C-38  
DB2, Supported by DCD III A-10, A-12,  
    B-9  
DCDACL PROC (for Independent Mode)  
    A-20, A-25, A-27  
DCDACL PROC, Example using A-21,  
    A-22  
DCDACL PROC, Modifying and testing  
    at installation F-8, F-9,  
    F-11, F-12, F-13  
DCDACL PROC, Used with BASIS A-10  
DCDCNTRL (see Control Record)  
DCDCOBOL PROC (for Other COBOL  
    Reports) B-6, B-8, B-24,  
    B-26, B-30  
DCDCOBOL PROC, Example using  
    B-7, B-8, B-32, B-33,  
    B-34, B-35  
DCDCOBOL PROC, Modifying and  
    testing at installation F-8,  
    F-9, F-11, F-12, F-13  
DCDCOBOL PROC, Use with WRITPDS  
    and READPDS Options  
    B-28, B-29  
DCDJCL PROC, (for JCL PROC Analysis  
    Reports) B-24, D-5, D-6,  
    D-7, D-8, D-10, D-11, D-12

DCDJCL PROC, Modifying and testing  
at installation F-8, F-9,  
F-11, F-12, F-13

DCDWK01 through DCDWK07 work  
files A-23, A-26, A-27,  
A-28, A-29, A-50, B-36,  
B-39, B-40, B-41, B-43,  
E-31

DCDWK01 work file A-47, C-7, E-34,  
E-35, E-36

DCDWK02 work file C-33, C-45, E-26

DCDWK03 work file C-6, C-11, C-27,  
E-5

DCDWK04 work file B-12, B-34, B-68,  
C-30, C-41, E-5

DCDWK05 work file C-15, C-36, C-38

DCDWK06 work file C-23, E-5

DCDWK07-12 work files B-36, B-41

DCDWKS1 through DCDWKS7 work  
files B-36, B-39, B-40,  
B-41, B-43

DCD/NODDC PARM Option (see also  
Data Division Condensed  
Cross-Reference Report)  
A-16, A-19

DICT PARM Option (see also Data  
Dictionary Interface File)  
B-12, B-15, B-17, B-37,  
B-68, E-25,

DL1, Fields in CASE files C-16, C-20  
C-24

DL1 PARM Option A-16, A-19, B-12,  
B-17

DL1 Supported by DCD III A-10, A-12,  
B-9, B-10, C-28

DMAP report, compiler A-6, A-10, A-11,  
A-30

## **- E -**

Errors, E-level A-6, A-8

Errors, listed in PRINT DD C-5

Error messages (Section) E-1 through  
E-44

Expanded Procedure Division Narrative  
File (CASE) C-38, C-39

## **- F -**

FGCONSTS/NOFGCONSTS PARM  
Option (see also Figurative  
Constant Report, System  
Cross-Reference for  
Figurative Constants) A-16,  
A-19, B-12, B-17, B-43

FIGCON=PROC Symbolic A-19, C-23

FIGCON=PROC Symbolic, Examples  
using A-22, A-48

Figurative Constants Report (see also  
System Cross-Reference  
for Figurative Constants)  
IV, A-16, A-30, A-37,  
A-38, A-48

Flags, Direct and Indirect in SRA Report,  
IV, B-5, B-15, B-62, B-63  
in Data Dict File, B-69,  
B-70, C-41, C-42, C-43

Following a Data Field Report (Y2K)  
Option B-3, B-4, B-5, B-17,  
B-27, B-35, B-45, B-68

## **- G -**

Group fields, flagged in CASE file C-9

Group fields, flagged in Data Division  
Condensed Cross-  
Reference A-40, A-41

Group fields, flagged in Layout Report  
B-54

Group names, using on control cards for  
Layout Report B-22

## **- H -**

Header control card, for System Record  
Analysis Report B-17  
Hexadecimal record positions in UCLF  
Report A-17, in Layout  
Report B-13  
Highlights, new release II through XV  
Host variables, in SQL stmts A-12, B-9

## **- I -**

IEJECT/NOIEJECT PARM Option  
A-16, A-19  
INDEP PARM Option (see also  
Independent Mode)  
A-16, A-17, A-19  
Independent Mode (see also DCDACL,  
LIBACL, PANACL,  
INDEP PARM Option)  
A-24  
Independent Mode, Error messages  
E-35, E-36,  
Independent Mode, Used with BASIS  
PARM Option A-15  
Independent Mode, Used with  
CA-Optimizer A-10  
Independent Mode, Used with  
SRESOLVE PARM  
Option A-17  
Installation Addendums (to be inserted)  
F-15 through F-16  
Installation of DCD III (section) F-1  
through F-16

## **- J -**

JCL, Changes in new release III, F-6  
JCL COND Parameter A-6, A-8  
JCL for WRITPDS and READPDS  
Options B-28, B-29

JCL, Modifying SYSPRINT for Compile  
Mode A-6, A-7, A-11  
JCL Override in producing Data  
Dictionary Interface File  
B-34  
JCL PROC Analysis Reports (Section)  
D-1 through D-14, F-8  
JCL PROC Analysis Reports, Examples  
running D-11  
JCL PROC Symbolics (see also  
Individual PROC  
symbolics) A-14, A-19,  
A-47, A-48, B-11  
JCL PROC Library for DCD III F-7,  
F-8, F-12  
JCL PROCs (see PROCs)  
JCL to access and print CASE files C-6  
JCL to execute DCD III without using  
PROCs A-23, A-45, B-36  
JCL to unload DCD III from tape at  
Installation F-7

## **- K -**

Keyword/non-keyword PARM fields,  
A-44, A-45, E-30  
Keywords, JCL F-7

## **- L -**

L77 PARM Option B-14, B-17  
Layout Report III, B-5, B-7, B-8, B-9,  
B-12, B-13, B-14, B-17,  
B-22, B-23, B-28, B-37,  
B-43, B-45, B-53, B-54  
B-55  
Layout Report, Error messages E-23,  
E-24, E-29,  
Layout Report, Example running  
B-32, B-34, B-35  
Layout Report, Table of Contents (see  
Also LTC and LTS PARM  
Options) B-14, B-45, B-53

LAYOUT=PROC Symbolic B-17  
LAYOUT=PROC Symbolic, Examples  
    Using B-7, B-8, B-28,  
    B-32, B-34, B-35  
LAYOUTS/NOLAYOUTS PARM  
    Option (see also Layout  
    Reports) B-12, B-13,  
    B-14, B-17, B-37, B-43  
LFD PARM Option B-12, B-17  
LHR PARM Option B-13, B-17  
LHR PARM Option, Error message  
    E-29  
LHX PARM Option B-13, B-17  
LHX PARM Option, Error message  
    E-29  
LIBACL PROC A-5, A-25, A-28  
LIBACL PROC, Example using A-20,  
    A-21  
LIBACL PROC, Modifying and testing  
    at installation F-8, F-9,  
    F-11, F-12, F-13  
LIBOCL PROC B-6, B-7, B-8,  
    B-38, B-40  
LIBCOBOL PROC, Example using  
    B-30, B-31  
LIBCOBOL PROC, Modifying and  
    testing at installation  
    F-8, F-9, F-11, F-12,  
    F-13  
Librarian PROCs (see LIBACL,  
    LIBCOBOL)  
Librarian, Supported by DCD III  
    A-10, A-12, B-9, B-10  
LISTER compiler option A-7  
LITERAL=PROC Symbolic A-19  
    B-17, C-23  
LITERAL=PROC Symbolic Examples  
    Using A-22, B-32  
Literals, in CASE files C-12, C-20,  
    C-45, C-46, C-47  
Literals Report (see also System Cross-  
    Reference for Literals)  
    IV, A-17, A-30, A-37,  
    A-38

LITERALS/NOLITERALS PARM  
    Option (see also Literals  
    Report, System Cross-  
    Reference for Literals)  
    A-17, A-19, B-14, B-17,  
    B-43, F-11, F-12  
LINECNT=PROC Symbolic A-19, B-17  
LINECNT=PROC Symbolic, Error  
    message E-35  
LK1PARM Option B-13, B-17  
LLA PARM Option B-13, B-17  
LL1 PARM Option B-13, B-17  
LNCNT=nn PARM Option A-16, A-17,  
    A-19, B-14, B-17, B-45,  
    D-6  
LNR PARM Option B-9, B-13, B-14,  
    B-17, B-55  
LNR PARM Option, Error message  
    E-29  
LOR PARM Option B-7, B-13, B-14,  
    B-17, B-35, B-37, E-29,  
    E-35  
Lower Case Narrative, NUC option  
    A-17, B-14  
LPP/NOLLP PARM Option B-13, B-17  
LPV/NOLPV PARM Option B-13, B-17  
LS1/NOLS1 PARM Option B-12, B-14,  
    B-17  
LTC/NOLTC PARM Option B-14, B-17  
LTS PARM Option B-14, B-17

## **- M -**

MBRFETCH PROC (Member Fetcher)  
    B-6, B-8, B-24, B-25,  
    B-26, B-29, B-37, B-38,  
    B-42, D-5  
MBRFETCH PROC, Example using  
    B-30, B-32, B-35  
MBRFETCH PROC, Modifying and  
    testing at installation F-8  
Member Fetcher PROC (see MBRFETCH  
    PROC)



MVS operating environment, setting  
DCB-BUFNO parameter  
in A-24, B-37

## - N -

Narrative (Upper/Lower CASE) NUC  
option A-17, B-14  
Non-keyword PARM fields (see keyword)  
NRSPACE/NONSPACE PARM Option  
A-17, A-19  
NUM/NONUM compiler option A-7,  
E-36

## - O -

OTHER=PROC Symbolic A-19, B17,  
B-29  
OTHER=PROC Symbolic Examples  
using A-22, B-7, B-8,  
B-28, B-34, B-35

## - P -

PANACL PROC A-5, A-25, A-29  
PANACL PROC, Example using A-20,  
A-21  
PANACL PROC, Modifying and testing  
at installation F-8, F-9,  
F-11, F-12, F-13  
PANCOBOL PROC B-6, B-7, B-8,  
B-38, B-39  
PANCOBOL PROC, Example using  
B-30, B-31  
PANCOBOLPROC, Modifying and  
testing at installation  
F-8, F-9, F-11, F-12, F-13  
Panvalet PROCs (see PANACL,  
PANCOBOL)  
Panvalet Supported by DCD III A-10,  
A-12, B-9, B-10

PARA=PROC Symbolic B-17  
PARA=PROC Symbolic, Examples  
Using B-6, B-28, B-29,  
B-31  
Paragraph Range Information File (CASE)  
C-27, C-28, C-29  
PARAGRAPH/NOPARAGRAPH PARM  
Option (see also System  
Paragraph Cross-Reference)  
B-14, B-17, B-43  
PARM Options (see also individual PARM  
Options) A-15, A-16, A-17,  
A-18, A-19, B-12, B-13,  
B-14, B-15, B-17,  
PARM Options, Specifying A-14, B-11  
PARM Options, Use of A-44, A-45, A-46  
Partitioned Data Set (see PDS)  
Password, DCD III (see also Control  
Record, Installation  
Addendum) E-31  
Password, Librarian F-13  
PDC/NOPDC PARM Option (see also  
Procedure Division  
Condensed Cross-  
Reference) A-17, A-19  
PDS (see also WRITPDS, READPDS,  
Installation of DCD III)  
PDS, containing COPY members A-5,  
A-24, B-6, B-25, E-27,  
E-28  
PDS, containing COBOL source code B-6,  
B-24, B-26, B-27, D-9  
PDS Member Fetcher PROC (see  
MBRFETCH PROC)  
PICTURE and Literal File (CASE)  
C-45, C-46, C-47  
PMAP report, compiler A-6, A-10, A-11,  
A-30  
PROC Symbolics A-5, A-8, A-9, A-14,  
A-19, A-46, A-47, A-48,  
A-49, B-6, B-11 B-17, F-9  
PROCs (see also individual PROCs, JCL  
PROC Analysis Reports)  
PROCs, Choosing A-5, B-6

PROCs, Examples using A-20, A-21,  
A-22, A-49, B-30, B-31,  
B-32, B-33, B-34, B-35,  
D-11

PROCs, Executing DCD III A-49

PROCs, Modifying and testing at  
Installation F-8, F-9,  
F-10, F-11

PROCs, Sample PROCs for testing  
F-12, F-13

PROCs, Specifying PARM options in  
A-14

Procedure Division Condensed Cross-  
Reference Report IV,  
A-6, A-7, A-17, A-18,  
A-30, A-42, A-43

Procedure Division Condensed Cross-  
Reference Report,  
Examples running A-21,  
A-22, A-49

Procedure Division Information File  
(CASE) C-15, C-16,  
C-17, C-18, C-19, C-20,  
C-21

Procedure Division Narrative for Each  
Data Field File (CASE)  
C-36, C-37

Procedure Division Reference to the Data  
Division File (CASE)  
C-33, C-34, C-35

## - Q -

QUOTE/NOQUOTE PARM Option  
A-17, A-19, B-14, B-17

## - R -

RA1 PARM Option B-15, B-45

RAA PARM Option B-15, B-45

RAB PARM Option B-15, B-45

RAD PARM Option B-15, B-45

RAS/RAT PARM Options B-15

READPDS/WRITPDS B-16, B-82

RECORDS=PROC Symbolic B-17

RECORDS=PROC Symbolic, Example  
using B-32

RECORDS/NORECORDS PARM Option  
(see also 01 Record Report)  
B-14, B-17, B-43

REG, Modifying in DCD III PROCs at  
installation F-9

REGION A-17, A-24, B-37, E-21, E-24,  
E-28, E-29, E-30, F-9

RESOLVE=PROC Symbolic A-19, B-17  
B-45

RESOLVE=PROC Symbolic, Examples  
using A-21, B-7

RESOLVE/NORESOLVE PARM Option  
A-8, A-17, A-19, B-15,  
B-45

RPTYPE=PROC Symbolic D-6

RPTYPE=PROC Symbolic, Example  
using D-10, D-11

## - S -

SORT region (see also SORTREG=PARM  
Option) A-17, A-24, B-15,  
D-6, E-30, F-9

SORTREG, Modifying in DCD III PROCs  
at installation F-9

SORTREG=PROC Symbolic A-19, B-17

SORTREG=nnnnnn PARM Option  
A-17, A-19, B-15, B-17,  
B-45, D-6

SOURCE compiler option E-36,

SOURCE=PROC Symbolic A-8, A-19,  
A-48, C-15, C-23, C-27,  
C-30, C-33, C-36, C-45

SOURCE=PROC Symbolic, Example  
Using A-47

SOURCE/NOSOURCE PARM Option  
(see also Source Listing  
Report) A-17, A-19, A-44,  
E-36,

Source listing, compiler III, A-6, A-7,  
A-8, A-9, A-16, A-17,  
A-24, A-30, E-35, E-36,  
Source Listing Report III, A-6, A-8, A-9,  
A-10, A-11, A-15, A-16,  
A-17, A-30, A-31, A-32,  
A-33, A-34, A-40, A-42,  
A-44, A-45  
Source Listing Report, Example running  
A-22, A-48  
SPACE allocation (see also Work space)  
B-43, E-21, F-6  
Special Registers Report (see also  
System Cross-Reference  
for Special Registers)  
IV, A-17, A-30, A-39  
SPREGS=PROC Symbolic A-19, B-17,  
C-23  
SPREGS=PROC Symbolic, Example  
using A-22  
SPREGS/NOSPREGS PARM Option  
(see also Special Registers  
Report, System Cross-  
Reference for Special  
Registers) A-17, A-19,  
B-15, B-17  
SRA=PROC Symbolic B-17  
SRANAL/NOSRANAL PARM Option  
(see also System Record  
Analysis) B-15, B-17, B-43  
SRESOLVE/NOSRESOLVE PARM  
Option (see also System  
Record Analysis Report)  
A-8, A-17, A-19, A-24  
STOP/NOSTOP PARM Option A-8, A-18,  
A-19, B-15, B-17  
STOP PARM Option Error Message E-29  
Symbolics (see PROC Symbolics)  
System Cross-Reference for Figurative  
Constants (see also  
Figurative Constants  
Report) B-5, B-12, B-45,  
B-52

System Cross-Reference for Literals  
(see also Literals Report)  
B-5, B-12, B-45, B-56  
System Cross-Reference for Literals,  
Example running B-32  
System Cross-Reference for Special  
Registers (see also Special  
Registers Report) B-5,  
B-15, B-45, B-61  
System Data Name Cross-Reference  
B-5, B-9, B-12, B-15,  
B-28, B-29, B-43, B-45,  
B-50, B-51  
System Data Name Cross-Reference,  
Example running B-31  
System Paragraph Cross-Reference  
B-5, B-14, B-15, B-45,  
B-57  
System Paragraph Cross-Reference,  
Example running  
B-28, B-29, B-31  
System Record Analysis Report B-5,  
B-12, B-15, B-17, B-18,  
B-28, B-37, B-45, B-62,  
B-63, B-68  
System Record Analysis Report, Error  
Messages E-25, E-40,  
Example running B-33  
SYSTRACE F-17

## **- T -**

Testing DCD III at installation F-11  
TRA=PROC Symbolic B-17  
TRACE/NOTRACE PARM Option B-15  
TRACE supporting options B-19  
TRACEIN F-17, E-31, E-32, E-33  
Transfer of Control Narrative File  
(CASE) C-30, C-31

## - U -

UNREF=PROC Symbolic A-19  
 UNREF=PROC Symbolic, Examples  
     using A-5, A-21, A-49  
 UNREF/NOUNREF PARM Option  
     A-18, A-19, A-21, A-49,  
     B-15, B-17, B-45  
 UNREF/NOUNREF PARM Option,  
     Error messages  
     E-40, E-41  
 Upper Case Narrative, NUC option  
     A-17, B-14

VS COBOL II, Error Message E-36  
 VS COBOL II, Supported by DCD III  
     A-11, B-9  
 VS COBOL II, Used with CLN=nn  
     Parm Option A-16  
 VS2/NOVS2 PARM Option (see also  
     VS COBOL II, COBOL 85)  
     A-18, A-19, B-15, B-17,  
     B-45  
 VS2 PROC Symbolic A-19, B-17

## - V -

Verb Analysis Report A-18, A-35, B-5,  
     B-15, B-45, B-64, B-65  
 Verb Analysis Report, Error message  
     E-40  
 Verb Analysis Report, Example running  
     B-31  
 VERB=PROC Symbolic B-17  
 VERB=PROC Symbolic, Example  
     using B-31  
 VERB/NOVERB PARM Option  
     (see also Verb Analysis  
     Report) A-18, A-19, B-15,  
     B-17, B-43  
 VERB/NOVERB PARM Option,  
     Example using B-31  
 Version 3 COBOL (see also COBOL 68  
     and earlier, VR3/NOVR3  
     PARM Option) A-11, A-18,  
     B-15  
 VR3/NOVR3 PARM Option (see also  
     Version 3 COBOL 68 and  
     Earlier) A-18, A-19, B-15,  
     B-17, B-45  
 VR3 PROC Symbolic A-19, B-17  
 VS COBOL II (see COBOL 85,  
     VS2/NOVS2 PARM  
     Option)

## - W -

Work files (see also individual work files  
     DCDWK01 - DCDWK07,  
     DCDWKS1 - DCDWKS7)  
     III, A-18, B-43, C-4, C-5,  
     F-17  
 Work files, Storing in condensed format  
     B-28  
 Work space, Efficient use of B-43

## - X Y Z -

Y2K Code Verification B-16, B-72  
 Y2K Option (Following a Data Name  
     Report) B-3, B-4, B-5,  
     B-17, B-27, B-28, B-35,  
     B-45, B-68  
 Zero, Condition Code (see also STOP  
     PARM Option) A-18,  
     B-15

Release 3.6

## DCD III

# SPF (System Productivity Facility) PANELS

*This page intentionally left blank*

With the industry turning more and more to on-line applications, MARBLE Computer, Inc. has taken on the challenge of providing availability to it's customers on an on-line SPF panel interface for our DCD III software.

To make use of these panels, a client must have Version 2 or later of the ISPF Dialog Manager and have ISPF/PDF installed.

### BENEFITS

#### **FULL DCD III FUNCTIONALITY**

In many Data Processing Departments, individual users do not have a DCD III user's manual which causes many features of DCD III to be unused or unnoticed. With the interactive SPF panels the user will be drawn to understand full DCD III functionality. This is due to our IBM/OS DCD III user's manual being placed on either HELP or SELECTION panels to guide users right to the function he/she wants to know about. The manual, while still functional, is no longer needed by each user to understand the capabilities of DCD III.

#### **USER-FRIENDLY INTERFACE**

Selection panels will be designed to make the user aware of every feature of DCD III in a friendly way, while the printed manual is still available as a reference guide.

#### **ALLEVIATION OF JOB CONTROL LANGUAGE MODIFICATIONS**

The user is not required to know OS JCL as all JCL will be built and submitted automatically as a batch job from the ISPF panels. Therefore, there is no need to modify JCL job streams as in the past.

#### **INTERACTIVE ERROR DETECTION**

As errors are detected on-line, error messages will be displayed describing what action needs to be performed on the users behalf. This will speed up the job submission process by both reducing JCL errors and reducing wasted CPU time due to running jobs that do not produce the anticipated results.

#### **SAVES TIME**

On new tasks the users time will be saved by 50% - 75% thus eliminating the need to understand all the option and JCL that are required to achieve desired output reports produced by DCD III. With this time savings, the user can perform project related tasks instead of spending large amounts of time understanding the parameters and JCL that drive DCD III.

*This page intentionally left blank*



### Use of this Section

This section is designed to give a partial overview of how the ISPF panels are used. Full coverage is not attempted here. There are about 240 panels (over 150 are HELP panels and 90 are panels that are called upon as necessary to complete user requests). These panels are assisted with more than 55 individual programs to help insure error free submission of DCD III jobs.

In addition to error free submission of jobs, features that are generally hidden unless one has carefully and thoroughly read the entire DCD III manual, are made available as a user selects one or more features or options.

Selected panels for completing user requests and some associated HELP panels from the UCLF facility only are shown in this section.

#### List of Panel Exhibits

|    |                                                   |     |
|----|---------------------------------------------------|-----|
| 1. | Selected Panels for Using the UCLF .....          | H-6 |
| 2. | Selection Panel for UCLF .....                    | H-6 |
| 3. | Alternate Compile Listing - Support Options ..... | H-7 |
| 4. | Alternate Compile Listing - More Options .....    | H-7 |
| 5. | Specify Type Input.....                           | H-8 |
| 6. | Panvalet Panel .....                              | H-8 |
| 7. | Select from Partitioned Data Set Panel .....      | H-9 |
| 8. | Job Card Panel .....                              | H-9 |

#### List of Help Panel Exhibits

|    |                                                       |      |
|----|-------------------------------------------------------|------|
| 1. | Help Screen for 'Main Selection Panel' .....          | H-10 |
| 2. | Main HELP Panel for UCLF .....                        | H-10 |
| 3. | First HELP Panel for #1 on Main UCLF HELP Panel ..... | H-11 |
| 4. | First HELP Panel for Supporting Options .....         | H-12 |
| 5. | HELP Panel for CASE Work Files .....                  | H-13 |

**Selected Panels for Using the ACLF**

Main Selection Panel

**DCD III (Data Correlation & Documentation)**  
**OPTION = ==>**

—

- 1 ACL - Alternate Compile Listing
- 2 OCRF - Other COBOL Reports Facility
- 3 CASE - Re-engineering work files
- 4 JCL - JCL Reports
- X Exit

DCD III spf release n.n

Selection Panel for ACLF

**Alternate Compile Listing - for 1 COBOL program**  
**OPTION = ==>**

—

- 1 Reports available and overview
- 2 Useful information
- 3 Reselect options & Submit batch job
- 4 Submit batch job
- X Exit

This panel is only displayed if option 3 is selected on the ‘Selection Panel for ACLF’.

### **Alternate Compile Listing - Supporting Options**

#### **Change options if required**

1. CAn - \_ (Enter 1, 2, or 3 for special CA-Optimizer formatting)
2. C68 - \_ (Enter Y if COBOL program is older 1968 COBOL)
3. DNF - \_ (Print an error message if data names are unresolved)
4. EIB - \_ (Insert COPY member DFHEIBLK for CICS program. Enter Y if CICS program is not run through pre-compiler)
5. ERR - \_ (Enter N to not print error messages)
6. FTX - \_ (Enter N to remove or F to force FROM-TO pos in 73-80)
7. LNCNT - \_ (Line count for DCD III reports)
8. NIS - \_ (Sort Narrative in Proc. Div. sequence for each field)
9. **Other** - \_ (Enter Y for less used options not shown on this panel)
10. QUO - \_ (Enter Y if double quote, not single apostrophe is used for non-numeric literals) - Enter D for self-determine
11. R3N - \_ (Use narrative from 85 COBOL rel. 3 or more on report)
12. REGION - \_ (REGION=3072K or larger is recommended)
13. SORTREG - \_ (SORTREG=300000 is recommended)
14. UNR - \_ (Enter N to omit printing unreferenced names in the DATA and PARAGRAPH Analysis reports.)
15. **VS2** - \_ (Enter Y for newer VS COBOL II - 1985 COBOL)
16. 370 - \_ (Enter Y if COBOL 370, MVS & VM, 390, or Enterprise COBOL)

If option 9 is selected the following panel would appear.

### **--Alternate Compile Listing - More Options --**

#### **Change options if required**

1. ASK - \_ (Enter N to ignore SKIP1, SKIP2, SKIP3, lines)
2. CHECK - \_ (Enter N to ignore checking previous seq nbrs)
3. CLN - \_ (Enter Line Count used in COBOL compile step)
4. COAID - \_ (Enter Y if using COBOL/Aid software)
5. F16 - \_ (Enter Y for FROM position of fields in columns 1-6)
6. IDD - \_ (Enter Y to include DATA DIVISION Literals)
7. IEJ - \_ (Enter Y to ignore EJECT lines)
8. JCL - \_ (Enter Y for to change **SPACE** or **BUFNO** in JCL)
9. NRS - \_ (Enter Y to insert a blank line to separate narrative in the DATA DIVISION. Do not use with CA1 OR CA3.)
10. PMO - \_ (Enter N to omit printing HEX OFFSETs in Compile Mode)
11. RES - \_ (Enter N to not resolve COPYs; this may produce errors)
12. SRE - \_ (Enter N to not expand COPY with SUPPRESS members.)
13. STOP - \_ (Enter Y to not process DCD III in compile mode.)
14. VR3 - \_ (Enter Y for older 1974 COBOL (version 3 or earlier) where SKIPs and EJECTs did not generate sequence numbers.)
15. USER - \_ (Enter other options here - Use HELP or Manual)

This panel is always displayed for either option 3 or 4 on the ‘Selection for ACLF’.

Specify Type Input

**Specify mode, type input & COPY libraries**

COMpile mode ==>\_ (Enter Y to run with a COBOL compiler, or  
enter N for Independent of COMpile mode)

Type Input ==>\_ (Enter C, P, L, or S to indicate type input)

|        |                                            |                                     |
|--------|--------------------------------------------|-------------------------------------|
| Select | <b>C -Select from Partitioned Data Set</b> |                                     |
|        | <b>P -Select from Panvalet</b>             | <b>(COBOL program will</b>          |
|        | <b>L -Select from Librarian</b>            | <b>be taken from the type input</b> |
|        | <b>S -Select from Sequential File</b>      | <b>specified)</b>                   |

COPY Library #1==>\_

COPY Library #2==>\_

COPY Library #3==>\_

COPY Library #4==>\_

COPY Library #5==>\_

COPY Library #6==>\_

Notes:     a) Enter fully qualified PDS names of files with COPY members.  
              **b) Do not put quotes around the Data Set name.**  
              c) DSN(s) to use here may be taken from SYSLIB DD IN COMPILE PROC.

The following panel is only displayed if P is entered for ‘Type Input’ above.

Panvalet Panel

----- **Panvalet Data Set Name** -----  
          **AND program name**

Data set name ==>\_

- a) Enter a fully qualified name for a Panvalet Data Set  
which contains user COBOL programs and INCLUDEs.
- b) Do not put quotes around the Data Set Name.**

Specify program ==>\_

- a) Enter PROGRAM name

The following panel is only displayed if **C** is entered for 'Type Input'.

Select from Partitioned Data Set Panel

**-- Partitioned Data Set and Program for --  
input to 'Alternate Compile Listing'**

PDS Name ==>\_

Notes on specifying Partitioned Data Set:

- a) Enter a fully qualified PDS name.
- b) Do not put quotes around the Data Set Name.
- c) The file must be partitioned, catalogued AND contain the program specified.

COBOL Program ==>\_

- a) Enter 1-8 character COBOL program name which resides on the Partitioned Data Set Specified above.

Job Card Panel

**.....Enter/Verify JOB Card .....**

**Enter/Verify JOB card - (Use up to 4 lines)**

FOR LOGON ID: **MACC01**      SUBMIT JOB ==>\_ (Y OR N)

```
//MARBLE02 JOB (MARBLE,R052),MSGLEVEL=1,
// NOTIFY=MARBLE,PRTY=9,TIME=(1,30),MSGCLASS=Q
//*
/*JOBPARM L=999
```

## Selected HELP Panels

### Help Screen for ‘Main Selection Panel’

#### **DCD III (DATA Correlation & Documentation)**

Four Facilities to chose from:

1. ALCF produces a source listing of the program similar to compile listing with meaningful COBOL documentation to the right of the source code.
2. OCRF is a reporting facility on one (or several programs) for various reports **such as** cross referencing of fields, tracing data names through a system of programs showing direct and indirect usage, charting a hierarchy of all the programs, and LAYOUTs of records.
3. CASE for producing work files to use in various aspects of COBOL Re-engineering.
4. JCL produce JCL reports showing DSNAMES and DDNAMEs.

### Main HELP Panel for ACLF

#### **---DCD III “Alternate Compile Listing” ---Overview of Reports --- SELECTION = = =>\_**

The ‘Alternate Compile Listing’ can be run independently of a COBOL compile or it can be run with the compiler to produce one listing combining the compiler and UCLF listing together.

- 1 Source Listing
- 2 CALL Statements
- 3 COPY Statements
- 4 Figurative Constants
- 5 Literals
- 6 Special Registers
- 7 DATA DIVISION Condensed Cross Reference
- 8 PROCEDURE DIVISION Condensed Reference
- 9 CASE Files

First HELP Panel for #1 on Main UCLF HELP Panel**--- DCD III Overview of ACLF Source Listing**

This report is the principal UCLF report. It list the source listing as does the compile listing, and additionally adds narrative generated by DCD.

For the DATA DIVISION, condensed look-alike PROCEDURE DIVISION code is produced along side each data field as shown here:

```
320 05 WS-NEXT-COL PIC S9(4). > ADD 1 TO # (1019)
 IF # > WS-MAX-COL @321 (1005)
 MOVE 02 TO # (1115)
321 05 WS-MAX-COL PIC S9(4). > IF WS-NEXT-COL @320 > # (1005)
322 05 WS-MAX-PD-DIGS PIC S9(3). > IF # =3 (942,965,1022), 4 (945,968,1025,1064)
 PERFORM-UNTIL # = ZERO (951,978)
```

The # in the narrative is substituted for the data name on the left.

An @ number is used to show where another data field is located.

Numbers in parentheses are used to show Procedure Division locations.

Similar Procedure Division statements are grouped together.

Some statements such as the PERFORM are condensed.

**(Continued)**

First HELP Panel for Supporting Options

**DCD III Specifying Supporting Options**

The following options affect the running of DCD III.

1. CA1, CA2, or CA3 - For CA-Optimizer users. Option CA1 and CA3 move PROCEDURE DIVISION displacements into columns 73-80 and print columns 128-133 respectively to save paper when formatting DCD III narrative. Option CA2 does what CA1 does plus selectively saves Some of the DATA DIVISION CA- data in cols 73-80.
2. C68 - Use when older COBOL 68 is used.
3. DFN - May be set NODNF to not print error messages when matching names are missing in the Data Division.
4. EIB - Insert COPY member DFHEIBLK FOR CICS programs to resolve names such as EIBDATE, and EIBTIME.
5. ERR - May be set to NOERROR to omit the printing of all error messages.
6. FTx - Default is to move the FROM-TO position for data fields into columns 73-80 if these positions are blank. Use N to stop this. Use F to force the move.

**(Continued)**



HELP Panel for CASE Work Files

**---- DCD III Trapping Useful Information on Work Files**

DCD III has access to information on each COBOL program it processes.

Much of this information is available to the user in the form of sequential files. These files are named CASE files.

This information may be used for any of the following purposes:

- a) Loading a Data Dictionary
- b) Loading a Repository
- c) Assisting with a field expansion project
- d) Doing any type of Re-engineering project with COLOL programs

The user may rebuild this information to fit their needs. COPY Members (containing 01 records) are available for writing small COBOL programs to access this information. More information is available by looking at the panels under the heading 'DCD III CASE Files'.

**( End of Trapping Useful Information HELP Panel )**

*This page intentionally left blank*

D

C

D

III

D

C

D

III

D

C

D

III

D

C

D

III

D

C

D

III



# Active COBOL Documentation For the Past, Present and Future

The automatic solution  
to COBOL maintenance

---

## DCD III

---

### DATA CORRELATION AND DOCUMENTATION SYSTEM

#### IBM/OS USER'S MANUAL

P.O. Box 920692  
El Paso, TX 79902  
800-252-1400  
FAX: 915-845-7918

