



**Marble**  
*Enterprise Software*

# Control/DCD

## Quick Start Guide

Covers CSE and DDM Modules in Detail

July 2018 Release 2.3

6416 Via De Albur Court  
Suite 100  
El Paso, TX 79912

P. (800) 252-1400  
F. (915) 845-7918

[Support@marblecomputer.com](mailto:Support@marblecomputer.com)  
[www.marblecomputer.com](http://www.marblecomputer.com)

# Table of Contents

I.	Cobol Program Overview using DDM and CSE.....	3
	The Need for Control/DCD .....	3
II.	Understanding what Control/DCD offers for one program.....	4
	By category or name &Operand searching .....	4
	Digital Documentation Manual - DDM.....	4
	Control/DCD Source Editing - CSE.....	5
	Summary Management System .....	5
	Alternate Compile Listing.....	5
	Creating a DDM or building a program with added analysis for CSE.....	5
	List of categories for current &Operands.....	6
III.	Examples of searching on a data-name, key-operand and more .....	7
	Search on data-name .....	7
	Search on performed-routine.....	8
	Search on 3 types of CODE-NOT-USED.....	9
	Search on PERFORM-ANALYSIS and Perform Errors.....	10
	Search on FORWARD-TRACING .....	11
	Search on program OPENS.....	12
	Search on OCCURS / DEPENDING.....	13
	Sample Invalid Data Messages.....	14
	Using the INDEX for finding categories .....	15
	Other Searches .....	16
IV.	Summary Management Reports.....	16
V.	Alternate Compile Listing .....	16
VI.	Using ISPF for One COBOL Program .....	17
	Main ISPF for Control/DCD.....	17
	Almost Duplicate panels for creating DDM or CSE .....	17
	Modify Type Analysis Panel .....	18
	Other Significant Notes.....	20
VII.	Downloading the Digital Documentation Manual to the PC.....	20
	Using COBOL Editing under ISPF on the mainframe.....	21

# Cobol Program Overview using DDM and CSE

---

## The Need for Control/DCD

Managing Source Code for Mainframe Applications has become a difficult task. The shortage of experienced mainframe programmers has left organizations with a major dilemma with respect to maintaining mainframe application code by either an offshore solution or in-house solution with a severe shortage of experienced COBOL programmers.

This creates new problems in the maintenance of the code, the predominance of which is COBOL. According to IBM and The Gartner Group there is roughly 220-250 Billion lines of COBOL code currently in production on Mainframes. Additionally, Micro Focus in a recent survey, says that each year 1.5 Billion new lines of COBOL code are added to mainframe applications.

IBM and other Independent Software Vendors (ISVs) are pushing the concept of moving coding off the mainframe through new Development Environments (IDEs) like Rational, Topaz and other products. These efforts continue to grow slowly. ISPF has been in use on mainframes since the release of IBM's Time Sharing option (TSO). Based on IBM studies the current ratio of mainframe developers using ISPF versus an IDE is 4:1.

Millions of programmers like you, the reader, have used ISPF, and will continue to do so. ISPF is ingrained in the Mainframe Developers DNA, it works and will be supported by IBM if there are mainframes.

Marble Computer designed Control/DCD with you, the ISPF user, in mind. We wanted to take advantage of your many years of experience using ISPF.

With Control/SE Marble Computer has advanced the analysis of a COBOL program way beyond existing software tool solutions that are currently in use in mainframe organizations.

Control/SE gives the COBOL programmer new insights into the code, its logic and data usage. We've designed Control/SE so that unfamiliarity with a COBOL program is no longer an issue. Mean Time to Repair is reduced. Unused code is identified and can be eliminated with certainty.

We also wanted to give you a new and different way of looking at a COBOL program listing. Thus, we developed the Digital Documentation Manual (DDM) for this release.

Since 1959 when COBOL became adopted as a preferred programming language on Computer vendor's mainframes the COBOL listing has been the mainstay of both development and maintenance projects.

The DDM gives the COBOL programmer new insights into the code. We've designed DDM to tell the story of each and every data element in a COBOL program. Additionally, the DDM can be File Transferred in a .pdf format to be stored on a Project based PC library or Server.

Please let us know how you like Control/DCD along with any suggested enhancements to:

[Support@MarbleComputer.com](mailto:Support@MarbleComputer.com).

Thank you,

The Marble Development Team

# Understanding what Control/DCD offers for one program

---

## By category or name **&Operand** searching

This Quick Start Guide provides an understanding of searching by category or name using the **Search &OPERAND** Analysis feature in Control/DCD along with examples of type ANALYSIS produced.

This new analysis feature is available in two separate ways in Control/DCD as described briefly below and in their own heading later.

1. The Digital Documentation Manual, released in Control/DCD release 2.2
2. The COBOL Source Editing feature, released in Control/DCD release 2.1

The **&OPERAND** is available to provide complete ANALYSIS information on key features of a COBOL program, with one SEARCH that includes an **&** *ahead of* key operand, data-name, performed routine, COPY-Member-Name or more. Three types of **&operand** information with description are listed here.

1. A complete hierarchical trace of any one data field showing:
  - a. What that activity is within each performed routine the field is used in along with the Performed Routine that the field resides in
  - b. A trace of the Performs Routines where this field activity occurs
  - c. All Data Attributes associated with this field including COPY member that the field resides in (if it does), 01 record that it belongs to, and From-To position in the 01 record it belongs to
2. PERFORM Errors (of any type) in the COBOL program (Major, Moderate and Minor)
3. Forward Tracing of all used routines in the program

There are a dozen other types of Analysis keywords, that are described in this Quick Start Guide. See the heading, 'List of current &Operands available'.

Examples show best how &Operand searching works. This Quick Start Guide is full of examples.

## Digital Documentation Manual - DDM

This feature uses Control/DCD to create a PDF file using A TXT2PDF utility with JCL generated through ISPF panels on the mainframe in a format where the PDF file is downloaded in binary format to the PC. Once downloaded, it is immediately available for clicking on and using with any type PDF reader.

The output is in 100-character record format. This Digital Documentation Manual is highly recommended.

After a cover page, an INDEX is available, along with the COBOL program, COPY members and a lot of ANALYSIS information available for searching on.

There are no page numbers to deal with. Key searching is done using &Operand as later examples in this Quick Start Guide show.

Users on release 2.1 wanting use of the Digital Documentation Manual are urged to move to release 2.2.

## Control/DCD Source Editing - CSE

Control/SE is designed for editing of the COBOL program on the mainframe in a manner that provides unique **&Operand** searching (while editing the COBOL program) using the following steps:

1. Using ISPF a JOB is submitted providing the name of the COBOL program to be edited.
2. A different PDS named EDITFILE (not COBOL) holds a duplicate of the COBOL program with added &Operand Analysis information at the end of the COBOL program.
3. Editing of the COBOL program is done in ISPF the same way an ISPF programmer has done for many years, with awareness that additional information is available during the analysis part of editing via a 'F &Operand' command. Numerous examples are given in this Quick Start Guide showing the unique analysis information available for CSE (and for the Digital Documentation Manual).
4. When editing is done and AFTER the program is saved back as a member on the EDITFILE PDS file, Option 3, under ISPF an ISOLATE feature is invoked by the COBOL technician to remove just the COBOL program and SAVE it back to the COBOL PDS.

## Summary Management System

Within Control/DCD for one program analysis, a Summary Management System is automatically in place to store traced key information from each program as it is processed by Control/DCD.

See 'Summary Management System' heading later in this guide.

## Alternate Compile Listing

An older method, the 'Alternate Compile Listing' available in multiple older versions of DCD is still available in Control/DCD. See heading later 'Alternate Compile Listing'.

## Creating a DDM or building a program with added analysis for CSE

ISPF is used to create the **&operand** information for all the **features** described in this Overview. See the heading 'Using ISPF for One COBOL Program building' later in this Quick Start Guide.

## List of categories for current &Operands

1. **&Data-Division-Field-Name**
2. **&Performed-Paragraph (or Performed-Section)**
3. **&CALL**
4. **&COPY**
5. **&CODE-NOT-USED**
6. **&ERRORS**
7. **&FORWARD-TRACING**
8. **&HELP**
9. **&INDEX**
10. **&OCCURS**
11. **&OPEN**
12. **&PERFORM-ANALYSIS**
13. **&PERFORMED-ROUTINES**

The next three require more explanation. See 'Notes on #14, #15 and #16' below.

14. **&Cn/**
15. **&n**
16. **&Copy-Member-Name**

### Notes on #14, #15 and #16

- #14. **&Cn/** is useful when COPY members are included. The **n** in **&Cn/** is the sequential number of this COPY member. Use a search on **&INDEX** to find order of COPY members in this program.

A search on **&Cn/** takes the user to the beginning of this COPY member.

COPY members may be included or omitted when running Control/DCD. See the Index for heading 'Using ISPF for One COBOL program'.

- #15. **&n** requires that a specify option be on (Use @-DD-number) on the 'Modify Type Analysis' panel. See the Index for heading 'Using ISPF for One COBOL program'.

When this option is on, a Narrative Line similar to below, shows what this option produces differently within the narrative:

```
001470      Move WS-CURR-NBR @98 to IPT-NBR-OF-ENTRIES (342)
```

A search with &98 may be used to find the narrative for the other data field in a Procedure Division statement involving data-names. The following search will bring up the Narrative for preceding data name WS-CURR-NBR that resides at line 98 in the COBOL program:

**&98**

- #16. **&Copy-Member-Name** is similar in function to the use of **&Cn/** in that it will branch to the beginning of that COPY member, provided COPY members are included.

## Examples of searching on a data-name, key-operand and more

The examples of searching in this section show a box as is found in an PDF Reader file. For using IPSF editing a COBOL program under Control/SE, substitute an ISPF FIND for the box shown. For first example below, use:

**F &IPT-NBR-ENTRIES**

in place of PDF reader box **&IPT-NBR-ENTRIES**

### Search on data-name

shows	Data Division Information,
and shows	Procedure Division activity to this name,
along with	Performed Routine hosting each reference.
Also, for <u>DDM</u>	Forward Tracing of routines where activity exists.

**&IPT-NBR-ENTRIES**

```
(0163) &IPT-NBR-ENTRIES      &163&          ==>   Data Attributes   <==
      In 5-8 of IPT-INTERNAL-HOLD-TBL
      in WORKING-STORAGE
      05  IPT-NBR-ENTRIES
          Pic S9(8)  Value ZERO
          Usage is COMP
      Used in Depending on IPT-NBR-ENTRIES

      01  PROGRAM-ENTRY          ==>   Forward Tracing   <==
      02      C-INITAIL-READ-FILE
      02      N-SORT-INPUT-PROCEDURE
      03          P-LOOK-FOR-MATCH

C-INITIAL-READ-FILE          ==> Activity by Routine <==
      Move +45 to IPT-NBR-ENTRIES (260)
N-SORT-INPUT-PROCEDURE
      Move WS-CURR-HOLD-NBR @125 to IPT-NBR-ENTRIES (342)
      Set IPT-INDEX @166 to IPT-NBR-ENTRIES (344)
P-LOOK-FOR-MATCH
      Move IPT-NBR-ENTRIES to CPY-NBR-ENTRIES @131 (C2/32)
```

### Notes on changing results for Data Name Analysis

See 'Modify Type Analysis' in 'Using ISPF for One COBOL Program' for *FORMAT* use)

- Use **FORMAT 1** or use **OMIT DD-ATTRIBUTES** to remove **DD-ATTRIBUTES**.
- Use **FORMAT 2** or **3**, or use **ADD DD-ATTRIBUTES** to add **DD-ATTRIBUTES**
- Use **FORMAT 1** or **2**, to remove **Performed Routine** and **Routine Tracing**
- **C2/32** in last **P-D activity** line refers to the 32nd line in the 2nd COPY member.

## Search on performed-routine

shows

forward-tracing from that point on

&M-MERGE-THREE-FILES-TO-ONE		
4	276	&M-MERGE-THREE-FILES-TO-ONE
5	560	&FILEIO-ERROR-ROUTINE --> (4 Performs)
6	588	&FILEIO-RTN-2 --> (Perform/VARYING)
7	318	&N-SORT-INPUT-PROCEDURE
8	374	&N220-SEEK-MATCH THRU N299-EXIT --> (Perform/UNTIL)
9	395	&N410-GET-IPT-RANGE-SEQ THRU N499-EXIT --> (Perform/VARYING)
10	411	&N450-RELEASE-SORT-RECORD
11	423	&N500-WRITE-SORT-REC THRU N599-EXIT --> (2 Performs)
<u>(See #5)</u> FILEIO-ERROR-ROUTINE		
(See #10)		N500-WRITE-SORT-REC THRU N599-EXIT --> (2 Performs)
(See #10)		N500-WRITE-SORT-REC THRU N599-EXIT
(See #10)		N500-WRITE-SORT-REC THRU N599-EXIT
(See #8)		N410-GET-IPT-RANGE-SEQ THRU N499-EXIT -->

### Notes on Forward Tracing created

- Forward tracing shows the structure of PERFORMs in the program in **indented** format.
- Multiple PERFORMs **OR** different types of Performs, such as UNTIL, VARYING, or TIMES are noted.
- Sequence numbers (276, 560, 588 etc.) above show where the PERFORMed routine resides.
- **(See #5)** refers to a performed routine that has been performed before AND tracing further to **FILEIO-RTN-2** is not shown after the first occurrence of **FILEIO-ERROR-ROUTINE**. Look at underlined: (See #5) five lines up from the bottom of above example.



## Search on 3 types of CODE-NOT-USED

shows            **Unused 01 Records**  
and              **Unused Data Names (not in a COPY)**  
and              **Dead Procedure Division Code**

### **&CODE-NOT-USED**

#CODE-NOT-USED            &CODE-NOT-USED

          &Unreferenced PARAGRAPHS and SECTIONS for IPTPROG  
Before removing routine, check FALL THRU's not intended

#### Unused SECTIONS

Seq Number	Section/Paragraph Name	Message or Warning
543	R-ENTRY-TO-SUB-ACCT-2	NO Direct Refs to SECTION

Before removing routine, check FALL THRU's not intended

#### Unused PARAGRAPHS

Seq Number	Section/Paragraph Name	Message or Warning
544	R010-BEGIN-SUB-ACCT-2	NO Direct Refs to PARAGRAPH

#### &Unused 01 Records and 77 Entries for Prog: IPTPROG

Seq Number	Lvl	Field Name
146	01	WS-COUNT-SEQS
175	01	S1-RECORD

#### &Unused Data Names for Prog: IPTPROG

Seq Number	Lvl	Field Name
(39)***01**WS-AW-RECORD***** contains unused fields **		
41	10	WS-AW-SORT-PREFIX-ID

### Notes on types of Unused Information

- These three reports show **DEAD** Procedure Division Code, **Unused 01 records** and **Unused (non-COPY Member)** fields.
- In the 01 RECORD unused report, if a redefined record is used, it is reported.

## Search on PERFORM-ANALYSIS and Perform Errors

Shows **Major Perform Errors**  
and **Moderate and Minor Perform Errors**  
and **Normal PERFORM and GO TO Activity**

### **&PERFORM-ANALYSIS**

#### **PERFORM Warnings & Major Errors**

**Count      Type & Seq Nbr(s)**

**01      GO TOs leaving range of PERFORM      MAJOR PERFORM ERROR**  
**477**

**03      Backward GO TO's      MINOR PERFORM WARNING**  
**753              833              912**

#### **PERFORM & GO TO activity**

**Count      Type & Seq Nbr(s)**

**12      PERFORM SECTIONS**  
**242              248              258              304              306              310**  
**332              336              338              479              480              685**

**05      PERFORM Paragraphs**  
**356              400              414              428              452**

**07      GO TO Paragraphs**

### **Notes on types of PERFORM-ANALYSIS**

- The first report in PERFORM-ANALYSIS shows PERFORM Warnings and Major Perform Errors, such as the one **MAJOR PERFORM ERROR** shown that leaves the range of a PERFORM
- Next is a report of normally executed PERFORMs and GO TOs

## Search on FORWARD-TRACING

from Program-Entry

### &FORWARD-TRACING

#FORWARD	TRACING	&FORWARD-TRACING
1	229	PROGRAM-ENTRY
2	265	C-BUILD-NARR-FILE-TO-MERGE
3	487	MERGE-THREE-FILES-TO-ONE
4	554	FILEIO-ERROR-ROUTINE --> (4 Performs)
5	588	FILEIO-RTN-2 --> (Perform/VARYING)
6	612	N-SORT-INPUT-PROCEDURE
7	653	N220-LOOK-FOR-MATCH --> (Perform/UNTIL)
(See #4)		FILEIO-ERROR-ROUTINE --> (2 Performs)
8	892	P-SORT-OUTPUT-ROUTINE
9	1021	G-BUILD-OPEN-NARR-TO-MERGE

### Notes on Forward Tracing

- Forward tracing shows the structure of PERFORMs in the program in **indented** format.
- Multiple PERFORMs **OR** different types of Performs, such as UNTIL, VARYING, or TIMES are noted.
- Sequence numbers (229, 265, 487 etc.) above show where the PERFORMed routine resides.
- **(See #4)** refers to a performed routine that has been performed before AND tracing further is not shown after the first occurrence of **FILEIO-ERROR-ROUTINE**.

## Search on program OPENs

<b>&amp;OPEN</b>
------------------

```
000862    #OPEN REPORT          &OPEN REPORT
000863 (0038)    &MAST-INP-FILE
000864    C-BUILD-NARR-FILE-TO-MERGE
000865          OPEN INPUT MAST-INP-FILE
000867 (0043)    &ACCT-WORK-FILE
000868    C-BUILD-NARR-FILE-TO-MERGE
000869          OPEN OUTPUT ACCT-WORK-FILE
000870    P-SORT-OUTPUT-PROCEDURE
000871          OPEN INPUT ACCT-WORK-FILE
000872    End of FD OPEN & SORT Info
```

### Notes on types of OPEN

- All files that are used in an **OPEN** or a **SORT USING** or **GIVING** are shown with a reference to where they reside.
- In **Format 3**, the most recent Performed Routine name (e.g. **C-HANDLE-MAST-REC**) is shown ahead of the OPEN verb or SORT USING or GIVING clause.

## Search on OCCURS / DEPENDING

### &OCCURS

#OCCURS REPORT            &OCCURS REPORT

Used to verify legal range in OCCURS when DEPENDING on Data-Name is used

```
(0026) &T2-NBR-ENTRIES
      In 453-454 of T1-TABLE
      in WORKING-STORAGE
      05  T2-NBR-ENTRIES
          Pic S9(4)  Value ZERO
          Usage is COMP
      Used in Depending on T2-NBR-ENTRIES (30)       <--- DEPENDING ON field
      Move 50 to T2-NBR-ENTRIES (721)
      Add 1 to T2-NBR-ENTRIES (911)
      Set T2-INDEX to T2-NBR-ENTRIES (633)

(0029) &T2-ENTRY
      In 1-15 of 01 T2-TABLE
      in WORKING-STORAGE
      05  T2-ENTRY  OCCURS  5 TO 30  TIMES
          DEPENDING ON  T2-NBR-ENTRIES               <--- OCCURS/DEPENDING
      MOVE ZEROS to T2-ENTRY (905)

End of  OCCURS/DEPENDING  Info
```

### Notes on OCCURS/DEPENDING

The above OCCURS report shows complete narrative on two fields within the program as follows:

1. Each data field containing an OCCURS clause with a DEPENDING clause
2. Each data field referenced in a DEPENDING clause

The report allows quick verification of whether the table may be exceeded!

Note the highlighted in yellow 50 and 30 in the above report showing that the table is in violation of new COBOL standards.

## Sample Invalid Data Messages

### &ERRORS

SQ-NBR	MSG-NBR	DESCRIPTION
103	CSEM6IE4-I	VALUE '12 6' INVALID for field at 00102
148	CSEM6KC3-I	MOVE '12T44' may reach field at (0481) via MOVE at 01346
237	CSEM6IB2-I	MOVE HIGH-VALUES may be INVALID for field at 00153
341	CSEM6IA2-I	VALUE SPACES has REDEFINED Numeric Overlap Pos at 00045

#### Notes on Invalid Data Messages

- The messages shown above are all messages that portray Invalid Data Messages to assist in converting to COBOL 6.2
- According to IBM, only 25% of shops or installations encounter this problem
- Use ISPF Error Option 'I' in place of Option 'Y' to create these Invalid Data Messages
- Errors are created for the following:
  - DISPLAY NUMERIC fields in the WORKING-STORAGE section with no initial VALUE are shown
  - If a bad non-numeric literal is in a REDEFINES to a numeric DISPLAY entry, this is flagged as such.
  - If a bad non-numeric literal is MOVED to a REDEFINES to a numeric DISPLAY entry, this is flagged as such.
  - If a bad literal exists in a VALUE in the DATA DIVISION and is MOVED to another field that is a REDEFINES to a numeric DISPLAY entry, this also is flagged.
  - If a bad literal is MOVED to a one field, that is later MOVED to another REDEFINES of a numeric DISPLAY field, then this is flagged.

## Using the INDEX for finding categories

&INDEX
--------

#INDEX	&INDEX			
#COBOL PROGRAM	HAS	854 RECORDS		
#HELP	HAS	262 RECORDS		
#CALL REPORT	HAS	28 RECORDS		
#COPY REPORT	HAS	17 RECORDS		
#PERFORM-ANALYSIS	HAS	34 RECORDS		
#COPY=FILEINFO	HAS	18 RECORDS	C1/..	
#COPY=DDACLNBR	HAS	50 RECORDS	C2/..	OMITTED
#COPY=DCDPREC	HAS	139 RECORDS	C3/..	
#OCCURS	HAS	22 RECORDS		
#OPEN REPORT	HAS	16 RECORDS		
#NARRATIVE REPORT	HAS	1,969 RECORDS		
#FORWARD-TRACING	HAS	131 RECORDS		
#CODE-NOT-FOUND	HAS	61 RECORDS	OMITTED	

### Notes on INDEX

- The INDEX report shows a total of all possible records **available** by option. OMITTED is used to indicate when user options were used to OMIT some features.

## Other Searches

### Notes on Other Searches not shown

- Searches on CALLs and COPYs show pertinent information on CALLs and COPYs.
- Searches on ERRORs show syntax errors found when scanning the code, similar to Compile Errors. This option requires that ERRORs be turned on within the Modify Type Analysis Panel.
- Searches on HELP show information available to further explain some features.
- Searches on PERFORM-ROUTINES show a list of all Paragraphs and Sections that are performed.

See the sub-heading 'Notes on #13, #14 and #15' under 'List of categories for current &Operands' shown earlier for next three SEARCHES.

- Searches on ( &Cn/ ) may be used when COPY members are included. See the sub-heading 'Other Significant Notes' later in this guide.
- Searches on ( &n ) allow branching from the narrative on one name to the narrative on another name.
- Searches on ( &Copy-Member-Name ) is similar in function to ( &Cn/ ).

## Summary Management Reports

---

### How to create and access

1. Turn on the Summary Management option in the 'Modify Analysis Selection' panel.
2. After running job, go to the //SUMMARY DD in main JCL to find the DSN used in that DD and then go into BROWSE within ISPF and bring up this program within that file and see the information available there.

## Alternate Compile Listing

---

### How to access and getting more information

1. Go to the //PRINT DD to find the Partitioned Data Set DSN used in that DD. This data set was established during installation of this software. The Alternate Compile listing is stored as a member using the PROGRAM-ID as member name.
2. See the Control/DCD User's Manual for more information on the Alternate Compile Listing.



# Using ISPF for One COBOL Program

## Main ISPF for Control/DCD

```
Control/DCD          The Ultimate COBOL Program and Application Analysis Tool

Option ==>          Enter 1-6, H  or use F3 to exit)

1  Build a Digital Documentation Manual

2  Build COBOL Program with F &operand Analysis  for ISPF editing

3  Save COBOL Program without F &operand Analysis  after ISPF editing

4  Analyze a Cobol Application

5  Abend Analysis for Data Field Errors

6  Verify JCL Accuracy within an Application

H  HELP

Control/DCD Release 2.2
```

Use options 1, 2 and 3 for handling 'One COBOL Program'.

For creating a PDF file for download to the PC, always use Option 1.

For doing Mainframe Editing of the COBOL program under ISPF, use Option 2 and when done with COBOL editing, use Option 3 to save the program.

Using Option 1 or Option 2 brings an almost duplicate panel, with some exceptions. See below.

### Almost Duplicate panels for creating DDM or CSE

The panel below show the key parts of the panel that mostly duplicate each other. Option 2 for CSE uses EDITFILE in place of ANALYSIS in 3 places AND option 4 is different from the option 4 shown below.

```
COBOL Program ==> USERPROG  Press Enter with S to submit; F3 to Exit

COBOL PDS ==> USERID.USERLIB.COBOL

ANALYSIS PDS ==> USERID.USER.ANALYSIS

==> S  Enter 1-6 to make changes before submitting; Leave as S to submit
      1 - Modify Analysis Selection
      2 - Change COPYLIBs for resolving COPY Members
      3 - Modify PARM Options, COBOL Version, or Report Selection
      4 - Modify File Name for ANALYSIS, SUMMARY or Userid for Adobe File
      5 - Edit SPACE= or BUFFER= in JCL
      6 - Edit JOB Statement

      S - Submit JOB
```

First, change the name to the correctly wanted COBOL program name and when no other changes are needed, press the ENTER key to submit the JOB to either build a (PDF) file for downloading to the PC, or for option 2 to create a COBOL program with added documentation on an EDITFILE PDS for editing.

To modify the output or make necessary JCL changes, use options 1-6 on panel shown above.

Most changes needed are accomplished through option 1, 'Modify Analysis Selection'.

Changing the COPYLIBs used to find COPY members is done through Option 2.

More in-depth options controlling Control/DCD are done in Option 3, along with changing the Version of COBOL being processed.

Options 4, 5 and 6 are all JCL related.

This Quick Start Guide covers the 'Modify Analysis Selection' panel (Option 1) in more detail. Look at options (2-6) to determine if other changes are needed and without further assistance from this Guide.

## Modify Type Analysis Panel

**Option 1 invoked from the previous panel, brings up the 'Modify Type Analysis panel' shown below:**

```
Modify Type Analysis provided
Reset Default Values ==> N      (Reset Y/N's & FORMAT value to 3)

1. ==> Y  Add COPY Members          2. ==> Y  CALL Analysis
3. ==> Y  COBOL Source (add)        4. ==> Y  COPY Analysis
5. ==> N  CODE NOT USED (show)      6. ==> Y  DD info (PIC & more)
7. ==> N  ERRORS (Y, N or I)        8. ==> 3  FORMAT (1, 2 or 3)
9. ==> Y  FORWARD TRACING          10. ==> Y  HELP (how to use)
11. ==> N  INDEXED BY (clause)      12. ==> N  INDIRECT REFERENCES
13. ==> N  OMIT Name (use #)        14. ==> Y  OCCURS and OPEN
15. ==> Y  PERFORM Errors           16. ==> N  PERFORMED ROUTINES
17. ==> Y  SQL (show SQL info)
18. ==> Y  Use @-DD-number (of other name in P-D narrative)

(P=ASELE)
```

## Description of 18 numbered options

A brief description of each of these 18 options is given below. Compare these options with earlier examples for a clearer understanding. Also see the sub-heading 'Other Significant Notes' that follow a description of these 18 options.

1. Add COPY Members brings up another panel where selection may be made to include ALL COPY member, SELECT some COPY member, or to individual BYPASS some COPY members. (Note – If this option is already set to Y and COPY selection panel does not come up, turn the option off and then back on to Y to bring up COPY panel).
2. CALL Analysis includes PROGRAM ENTRY and ENTRYs besides CALLs.
3. COBOL Source includes the COBOL Source Code. In Control/SE, the COBOL Source is put first for editing on same.

(continued next page)

4. COPY Analysis shows each COPY along with other useful information.
5. CODE-NOT-USED shows code not used in three different ways shown below:
  - a. 01 Records where there is no field use for entire record
  - b. Data Fields that are not used (non-COPY fields only)
  - c. Possible Procedure Division Dead Code (with comments on how to carefully check same)
6. Data Division Attributes to be included for each data field Narrative. (Includes many attributes including From-To position relative to each 01 record, Usage, Level, Value, COPY member AND much more.)
7. Errors in COBOL Syntax like Compile Errors. This feature may be used to eliminate many compiles that take longer under Cobol 5 or 6). Also, option (I) here creates possible INVALID DATA messages for converting to Cobol 5 or 6.
8. FORMAT 1, 2 or 3. Format 3 is the standard default and usually recommended.
 

**FORMAT 1** (Format 1 offers the most **condensed** Narrative format)

  - FORMAT 1 has the following attributes
    - Data Division Narrative is turned **OFF**
    - Sort Sequence for Procedure Division Narrative is in **Verb** Sequence
    - Unique Forward Tracing of used Performed Routine Names are NOT included
    - Performed Routine Names are **NOT** included

**FORMAT 2**

  - FORMAT 2 has the following attributes
    - Data Division Narrative is turned **ON**
    - Sort Sequence for Procedure Division Narrative is in **Procedure Division** Sequence
    - Unique Forward Tracing of used Performed Routine Names are NOT included
    - Performed Routine Names are **NOT** included
9. Forward Tracing does forward tracing of PERFORMED Routines in the COBOL program. See examples given earlier.
10. HELP also provides information on using this software.
11. INDEXED BY clause adds an INDEXED BY clause to Data Division attributes. See Number 6 above, for 'Data Division Attributes to be included'.
12. INDIRECT REFERENCES show Indirect References when used in any type of REDEFINES or Group (or 01 record) use for elementary fields OR elementary use for Group field use after Direct Narrative use for each data field – Insure that NIS = N in Option 3 (Modify PARM).
13. OMIT Name (recommended) omits the data name and inserts a # character in its place in the PROCEDURE DIVISION activity in data name narrative as shown below:
 

**MOVE # TO OTHER-NAME (324,655), IF # = 'SMITH' (555)**
14. OPEN lists where the OPEN verbs are used in the program. (includes SORT/USING & GIVING). Also, this option when on, shows OCCURS/DEPENDING report for possible OCCURS range violations.
15. PERFORM ERRORS shows MAJOR, MODERATE and MINOR perform errors as well as PERFORM Analysis.
16. PERFORMED ROUTINES shows a count PERFORMED Routines including ENTRYs which by themselves are not PERFORMed.
17. SQL shows usage of data names in SQL statements.
18. Use @-DD-NUMBER insert @seq-nbr after the other (non-searched) name in P-D narrative as shown below:

MOVE # TO OTHER-NAME @216 (1024,1033)

This allows a search on the other field not originally searched for, using &n ( &216 for above )

## Other Significant Notes

Consult the Examples given earlier for a clearer guide on items below, such as 'Data Division Narrative', 'Procedure Division Narrative', and included 'Performed Routine Names'.

Defaults are established and may be reset to same. See 'Reset Default Values ==> N'.

By default, the following reports are turned OFF. All other reports are turned **ON**.

- CODE-NOT-USED
- ERRORS
- PERFORMED-ROUTINES

The NARRATIVE report is defaulted to FORMAT 3

- FORMAT 3 has the following attributes
  - Data Division Narrative is turned **ON**
  - Sort Sequence for Procedure Division Narrative **is in Procedure Division** Sequence
  - **Performed Routine Names are included** ahead of shown Procedure Division Narrative
  - **For Digital Documentation Manual, Tracing of Performed Routines with activity**

Within the NARRATIVE report;

- Listing **INDEXED BY** references are turned **OFF**
- Listing **INDIRECT** references are turned **OFF**. When turning ON, insure **NIS = N**, under Option 3 (Modify PARM Options).
- The presence of **@n** references following other names allowing searches on ( **&n** ) is initially turned **OFF**

To insure COPY's are included, do the following:

- Use Option 1 for 'Modify Type Analysis' to indicate Y for Add COPY Members
  - Turning the Y to an N, and then resetting to Y, brings up another panel for further selecting or bypassing individual members.
- Use Option 2 to re-specify what COPYLIBs are used in the JCL.

## Downloading the Digital Documentation Manual to the PC

---

The filename created in binary format for the PDF to be binary downloaded for Adobe OR other type PDF reader is:

**MANUAL4 .*program* .PDF** ( where ***program*** is the COBOL program-id )

The userid used for this file may be modified under option 4 ( Modify File Name... ).

Use a binary transfer download to the PC and the PDF file is then ready to be accessed by any PDF reader.

## Using COBOL Editing under ISPF on the mainframe

The following panel is used for option 2, for building COBOL program for Cobol Source Editing:

Add Analysis to COBOL Member for ISPF editing											
COBOL Program ==> USERPROG    Press Enter with S to submit; F3 to Exit											
COBOL PDS ==> USERID.USERLIB.COBOL											
EDITFILE PDS ==> USERID.USER.EDITFILE											
==> S    Enter 1-6 to make changes before submitting; Leave as S to submit											
1 - Modify Analysis Selection											
2 - Change COPYLIBs for resolving COPY Members											
3 - Modify PARM Options, COBOL Version, or Report Selection											
4 - Modify File Name for EDITFILE, SUMMARY or BCKUPCOB											
5 - Edit SPACE= or BUFFER= in JCL											
6 - Edit JOB Statement											
S - Submit JOB											
Sample Edit Entry Panel - - - > to use after JOB finishes	<table border="1"><tr><td colspan="2">ISPF Library</td></tr><tr><td>Project . . .</td><td>USERID</td></tr><tr><td>Group . . . .</td><td>UserLib</td></tr><tr><td>Type . . . .</td><td>EDITFILE</td></tr><tr><td>Member . . .</td><td>USERPROG</td></tr></table>	ISPF Library		Project . . .	USERID	Group . . . .	UserLib	Type . . . .	EDITFILE	Member . . .	USERPROG
ISPF Library											
Project . . .	USERID										
Group . . . .	UserLib										
Type . . . .	EDITFILE										
Member . . .	USERPROG										

After option S is used and user presses ENTER key to submit job and the job finishes, the output COBOL program with added &Operand category analysis information is put into a special EDITFILE PDS. Accessing that EDITFILE in a format like the smaller box shown above brings up the COBOL program with added &Operand searching added.

During editing use &HELP for more help. As example, if the user wants to change the source code with an ALL parameter, but not change added documentation, go to &HELP and look at Modifying CHANGE command within the &HELP notes.

When done editing, use option 3, to bring up ISOLATE ISPF shown below, and press ENTER key to re-build the COBOL program after editing is saved back to the EDITFILE PDS. The step below will store the COBOL program back to the main COBOL PDS after removing all added &Operand analysis information.

ISOLATE just COBOL Program from EDITFILE & STORE in MAIN COBOL PDS	
COBOL Program ==> IPTPRG    Press Enter to submit or F3 to Exit	
FROM:	EDITFILE PDS ==> MARBL01.CSE2.EDITFILE
TO:	MAIN COBOL PDS ==> MARBL01.USERLIB.COBOL

*This page intentionally left blank*