Control/DCD Quick Reference Guide

Table of Contents

© 2015 – 2019 Marble Computer, Inc

Introduction to Control/DCD Quick Reference Guide

Purpose and Scope:

The Control/DCD Quick Reference Guide is designed for use by the programmer who wants to get a head start on using Control/DCD.

Additionally, the Guide is a reference point for understanding the functionality of Control/DCD, its components and how each component relates to each other. This Guide covers all of the components which are:

1. The Digital Documentation Manual (the DDM)
2. The Alternate Compile Listing (ACL)
3. The COBOL Source Editor (CSE)
4. Analyze a COBOL Application
5. Abend Analysis for Data Field Errors
6. Verify JCL Accuracy Within an Application

When involved in a COBOL 6.2 the programmer will easily find The Control/DCD specific features in each of the components that will assist in the migration to IBM's COBOL 6 compiler.

Using the Guide:

The Table of Contents is interactive. By clicking on the section heading the Programmers is immediately taken to the desired information on best use of Control/DCD functionality.

User Comments:

Please send any comments or suggestions dealing with the Control/DCD Quick Reference Guide to Support@marblecomputer.com

Full Data Name Analysis

Use & ahead of data-field to get Data Analysis

From DDM (in pdf reader search box)

```
&IPT-NBR-ENTRIES
```

From CSE (from ISPF)

```
F &IPT-NBR-ENTRIES
```

Search Results show:

Data Division Information,
and **Procedure Division activity to this name,**
and **Performed Routine hosting each reference.**
and **Forward Tracing of routines where activity exists.** *(for DDM only)*

```
(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)
```

Four Steps to Understanding DATA NARRATIVE

STEP ONE - Shows data attributes and procedure division activity.

Data Attributes==>
```
(0089) INV-CODE
       In 63-65 of 01 INV-RECORD
       in WORKING-STORAGE
       05  INV-CODE
                  Pic 9(03)
```

Forward-Tracing==>
```
01    VB-UPDATE-INV-FILE --> (Perform/UNTIL)
  02      VE-DO-200-CODES (No Activity)
   03     |   VEA-DO-CODE-250-299 --> (Perform/TIMES)
  02      VF-DO-300-CODES
```

Activity by Routine=>
```
VB-UPDATE-INV-FILE --> (Perform/UNTIL)
       If INV-CODE not numeric (658)
       If INV-CODE < 200 (682)
       If INV-CODE > 300 (682)
       If INV-CODE < 300 (690)
       If INV-CODE > 400 (690)
VEA-DO-CODE-250-299 --> (Perform/TIMES)
       If INV-CODE not = 261 (758), 262 (758)
VF-DO-300-CODES
       Move INV-CODE to WK-INV-CODE (771)
       Move 352 to INV-CODE (774)
```

STEP TWO - Lists related performed routine.

Data Attributes==>
```
(0089) INV-CODE
       In 63-65 of 01 INV-RECORD
       in WORKING-STORAGE
       05  INV-CODE
                  Pic 9(03)
```

Forward-Tracing==>
```
01    VB-UPDATE-INV-FILE --> (Perform/UNTIL)
  02      VE-DO-200-CODES (No Activity)
   03     |   VEA-DO-CODE-250-299 --> (Perform/TIMES)
  02      VF-DO-300-CODES
```

Activity by Routine=>
```
VB-UPDATE-INV-FILE --> (Perform/UNTIL)
       If INV-CODE not numeric (658)
       If INV-CODE < 200 (682)
       If INV-CODE > 300 (682) <-- P-D line number
       If INV-CODE < 300 (690)
       If INV-CODE > 400 (690)
VEA-DO-CODE-250-299 --> (Perform/TIMES)
       If INV-CODE not = 261 (758), 262 (758)
```

```
VF-DO-300-CODES
      Move INV-CODE to WK-INV-CODE (771)
      Move 352 to INV-CODE (774)
```

STEP THREE - Show abbreviated forward tracing for referenced routine.

Data Attributes==>

```
(0089) INV-CODE
      In 63-65 of 01 INV-RECORD
      in WORKING-STORAGE
      05  INV-CODE
                Pic 9(03)
```

Forward-Tracing==>

```
01    VB-UPDATE-INV-FILE --> (Perform/UNTIL)
  02      VE-DO-200-CODES (No Activity)
   03     |   VEA-DO-CODE-250-299 --> (Perform/TIMES)
  02      VF-DO-300-CODES
```

Activity by Routine=>

```
VB-UPDATE-INV-FILE --> (Perform/UNTIL)
      If INV-CODE not numeric (658)
      If INV-CODE < 200 (682)
      If INV-CODE > 300 (682)
      If INV-CODE < 300 (690)
      If INV-CODE > 400 (690)
VEA-DO-CODE-250-299 --> (Perform/TIMES)
      If INV-CODE not = 261 (758), 262 (758)
VF-DO-300-CODES
      Move INV-CODE to WK-INV-CODE (771)
      Move 352 to INV-CODE (774)
```

STEP FOUR - Forward Tracing shows full tracing for abbreviated routines structure.

Control/DCD's **Forward-Tracing** is a **hierarchical listing** of **Performed** Routines

```
1   271     PROGRAM-ENTRY
2   308        A-OPEN-TWO-FILES
3   325        C-BUILD-NARR-FILE-TO-MERGE
4   342        E-CLEAR-OUT-ES-RECORD
5   357        M-MERGE-THREE-FILES-TO-ONE
6   399        .  N-SORT-INPUT-PROCEDURE
7   524        .  P-SORT-OUTPUT-PROCEDURE
8   538        .     P110-CLEAR-QVW-LINE  --> (Perform/UNTIL)
9   352        G-BUILD-OPEN-NARR-TO-MERGE
10  631        V-UPDATE-INV-CODE
11  655           VB-UPDATE-INV-FILE  --> (Perform/UNTIL)
12  706              VC-DO-SPECIAL-EXTENSIONS
13  731              VD-DO-OTHER-ASSOC-ACCTS
14  748              VE-DO-200-CODES
15  755              .  VEA-DO-CODE-250-299  --> (Perform/TIMES)
16  778              .     VW-WRITE-AND-READ-NEXT-REC
17  769              VF-DO-300-CODES
18
```

Click Here to Return to Index

Forward Tracing Chart

From DDM (in pdf reader search box)

```
&FORWARD-TRACING
```

From CSE (from ISPF)

```
F &FORWARD-TRACING
```

Search Results show:

```
#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.**

Click Here to Return to Index

&Operand – Searching made easy
Use of & Character

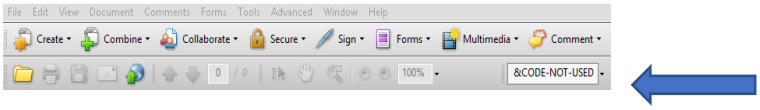
Inserting an & character just ahead of a selectable operand goes right to that Analysis.

See 'Search Operands' for all available operands

Example below shows one operand (CODE-NOT-USED)

From DDM (in pdf reader search box)

&CODE-NOT-USED



From CSE (from ISPF)

F &CODE-NOT-USED

Search Results show:

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

         &Unreferenced PARAGRAPHs and SECTIONs for IPTPROG
   Before removing routine, check FALL THRUs 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 THRUs 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.

Click Here to Return to Index

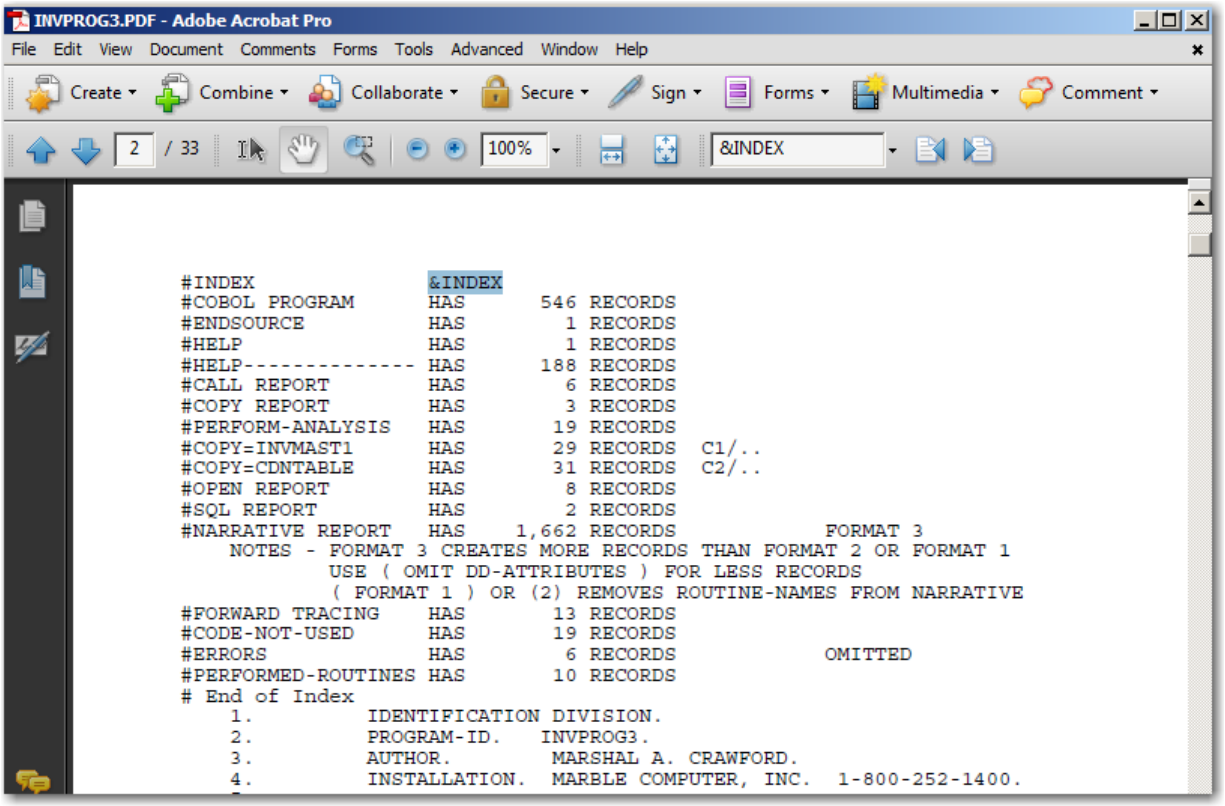
SEARCH Operands

(*Searching with **&** immediately followed by **operand***)

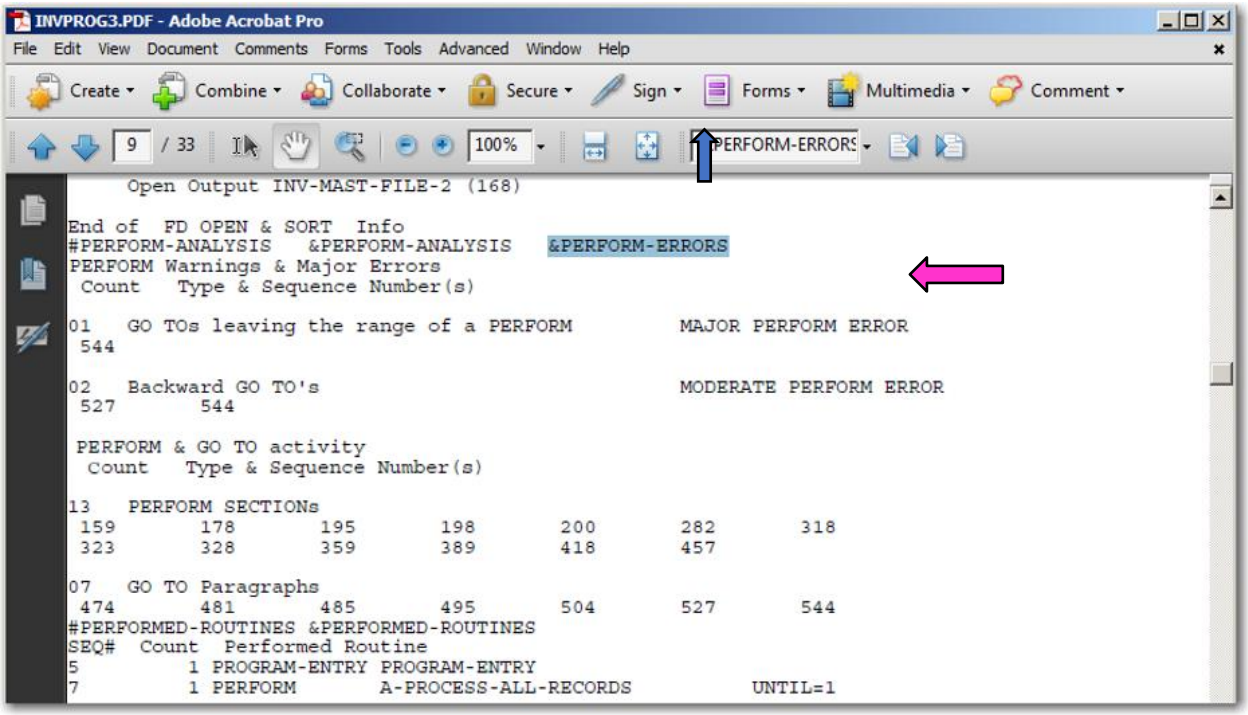
OPERANDS to Choose From

1. any data field
2. any Performed paragraph or section
3. CALL
4. COPY
5. CODE-NOT-USED
6. ERRORS
7. FORWARD-TRACING
8. HELP
9. INDEX
10. OPEN
11. PERFORM-ANALYSIS
12. PERFORMED-ROUTINES
13. SQL
14. Cn/ (see HELP for explanation)
15. n (see HELP for explanation)
16. any COPY member

&INDEX points to DDM Index



&PERFORM-ERRORS shows a major Perform Error

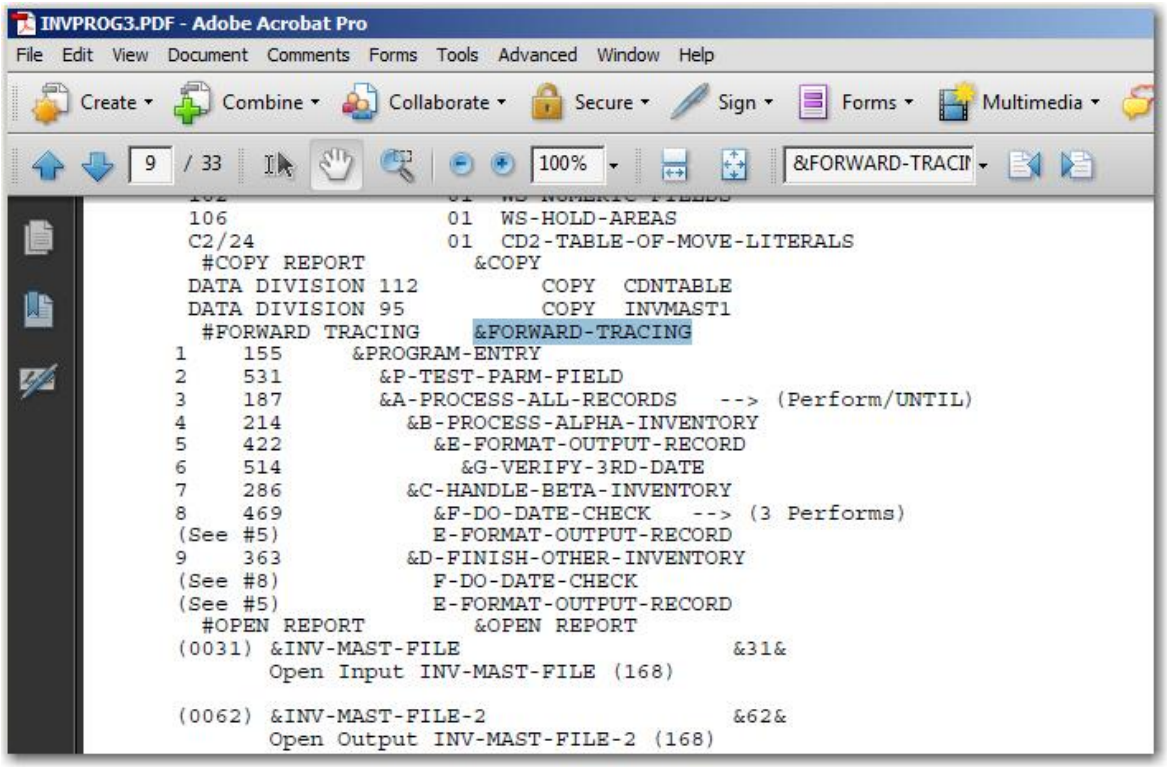


&Forward-Tracing of Performed Routines

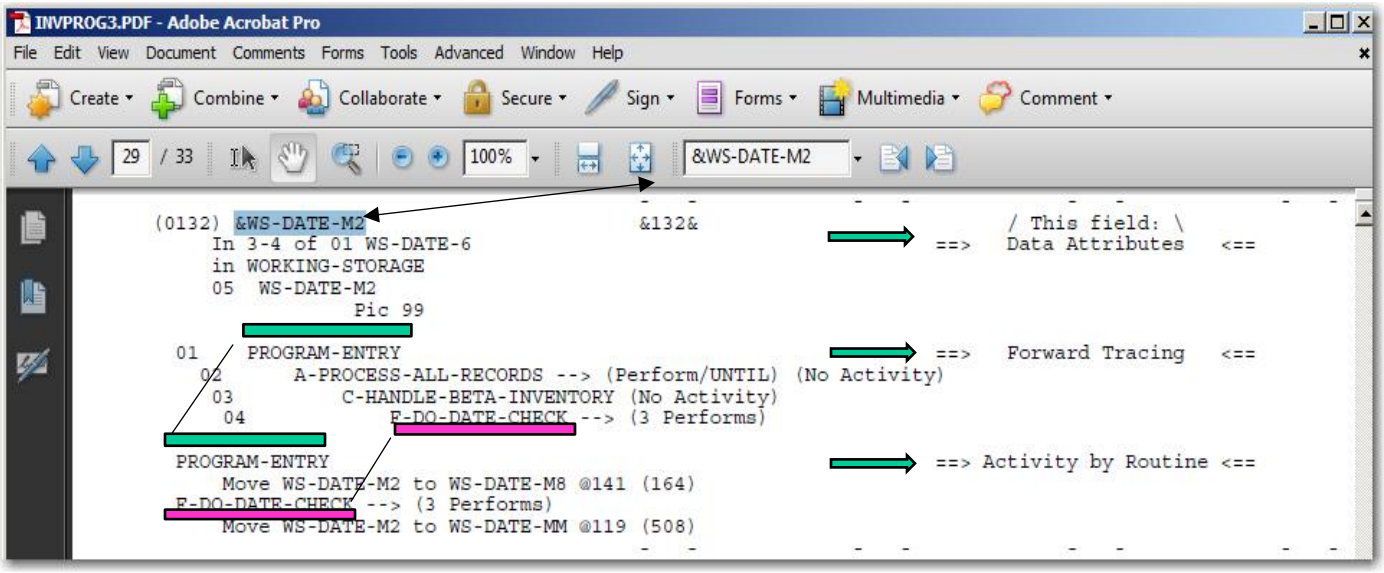
(allows maintaining structured code!)

&user-data-field

Find a selected data field example:



1. Shows Data Attributes
2. Shows Forward Tracing relevant to this field use
3. Shows Procedure Division activity within Performed Routine

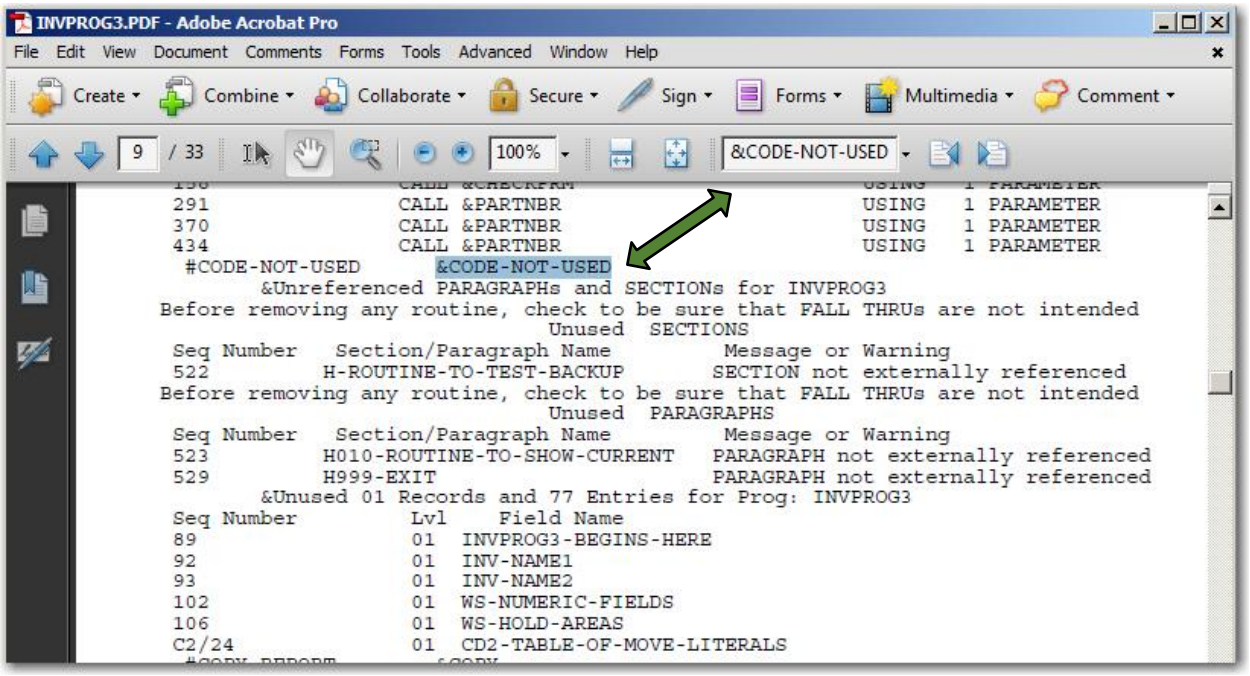


Activity by Routine within Forward Tracing
tells a very complete story of this data usage

&CODE-NOT-USED

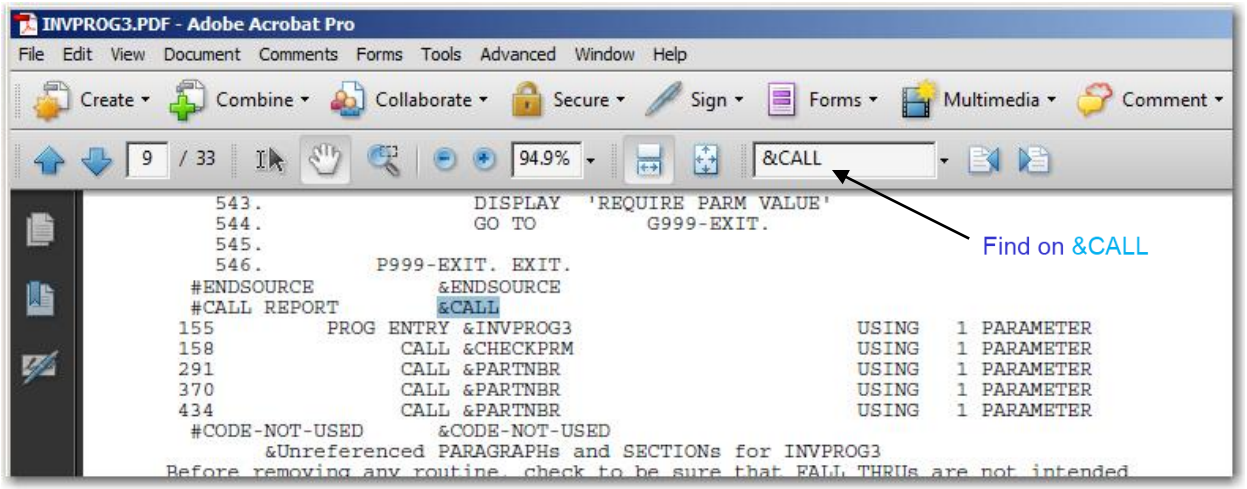
CODE NOT USED example shows:

1. Procedure Division DEAD CODE
2. Unused 01 records
3. Unused (non-COPY-member) data fields



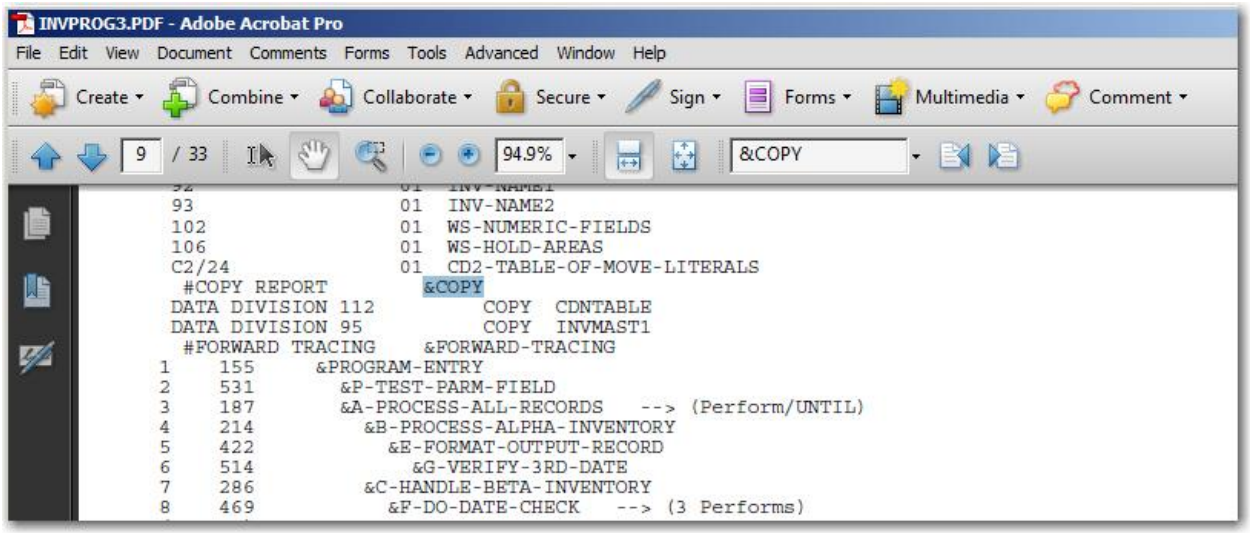
&CALL

Find CALL info example



Program-ENTRY and CALL information is provided at one glance **&COPY**

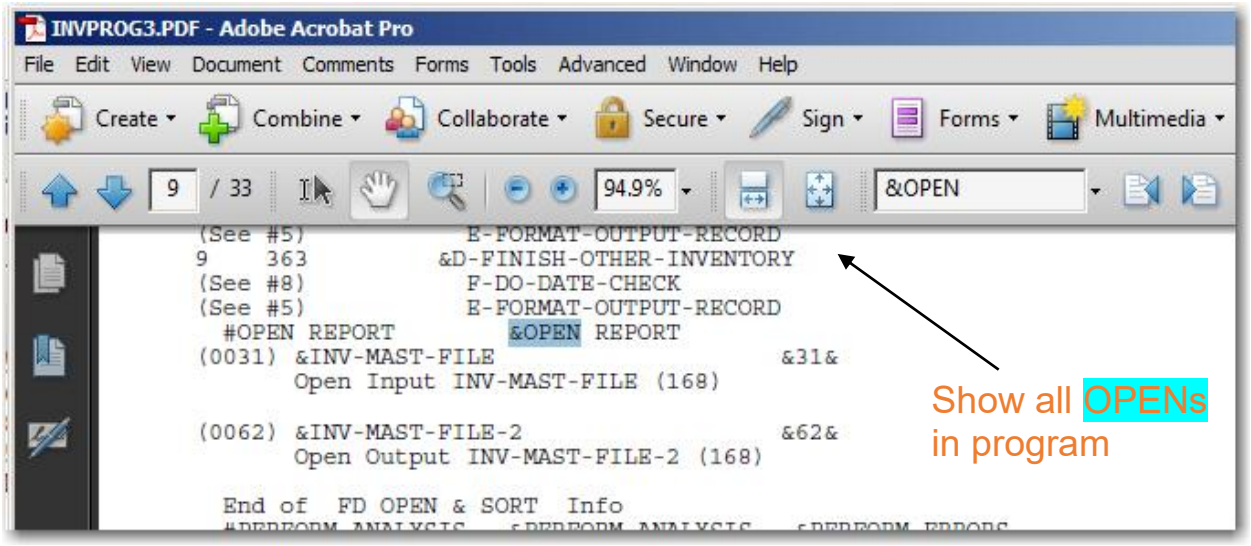
Find COPY information



All COPY members along with supporting information is given.

&OPEN

Get OPEN verb activity



Every OPEN statement in shown along with OPEN INPUT, OUTPUT or other type shown.

Click Here to Return to Index

PERFORM ERRORS – Finding PERFORM logic errors

From DDM (in pdf reader search box)

`&PERFORM-ANALYSIS`

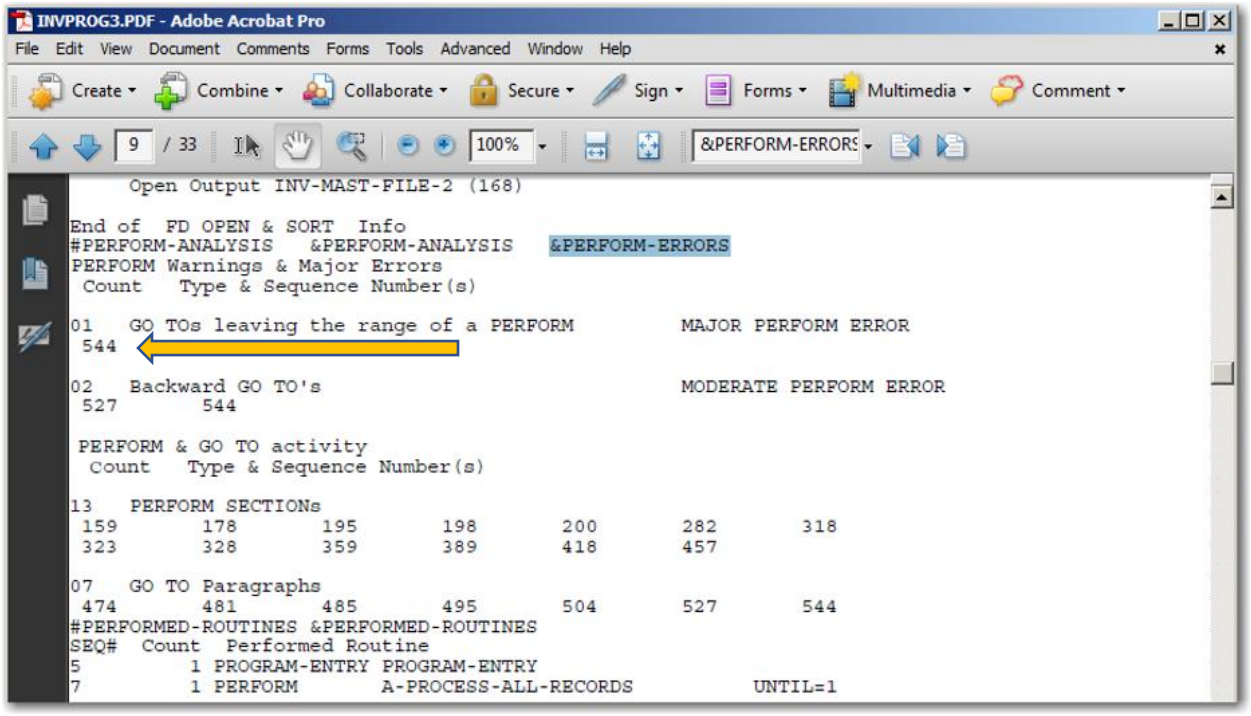
From CSE (from ISPF)

`F &PERFORM-ANALYSIS`

Search Results show:

&PERFORM-ERRORS shows a major Perform Error

In example below, GO TO leaves PERFORM Range



Click Here to Return to Index

Conversion to COBOL 6

Control/DCD has 6 areas that assist in conversion to COBOL 6

COBOL 6.2 Migration Assistance in the DDM

1. INVALID DATA not flagged in earlier COBOL
   Literal '1_3' containing 'F140F3'
   IBM does not flag these fields

2. Excessive Compile Time with newer compilers
   - 5 to 10 times for smaller programs, more for larger!
   - An Efficient Pre-Compiler would save time!
   - Control/DCD Perform analysis can help increase compile speed
   - Forward-Tracing also helps

3. OCCURS / DEPENDING where existing code exceeds table size
   - Examine All Code for Possible Exceeding Table Size

4. Not initially assigning a VALUE in Working-Storage
   - Data Division Code Need to be Looked at Carefully

COBOL 6.2 Migration Assistance from Abend Analysis

1. PACKED and BINARY fields defined too small
   - A Way Needs To Be Found To Identify These Fields

COBOL 6.2 Migration Assistance from Multiple COBOL Program Analysis

1. COBOL PARAMETER inconsistencies
   - Parameter Sizes not the same or parameter missing
   - These Must Be Fixed

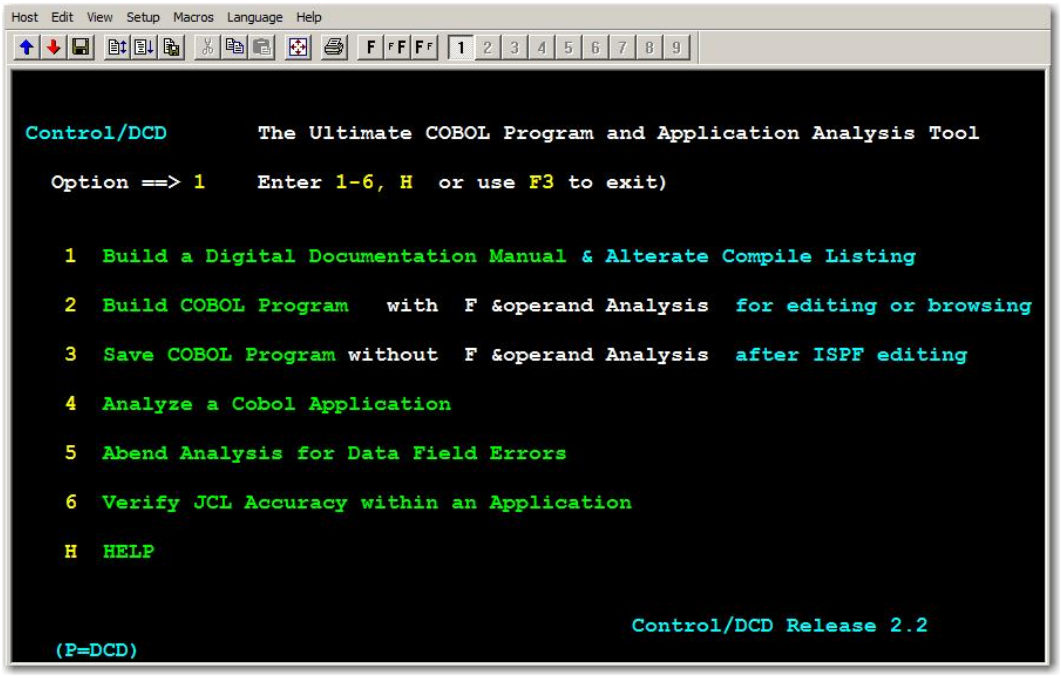
Click Here to Return to Index

Using ISPF panels to build the Analysis

Control/DCD may now be entirely run from ISPF panels

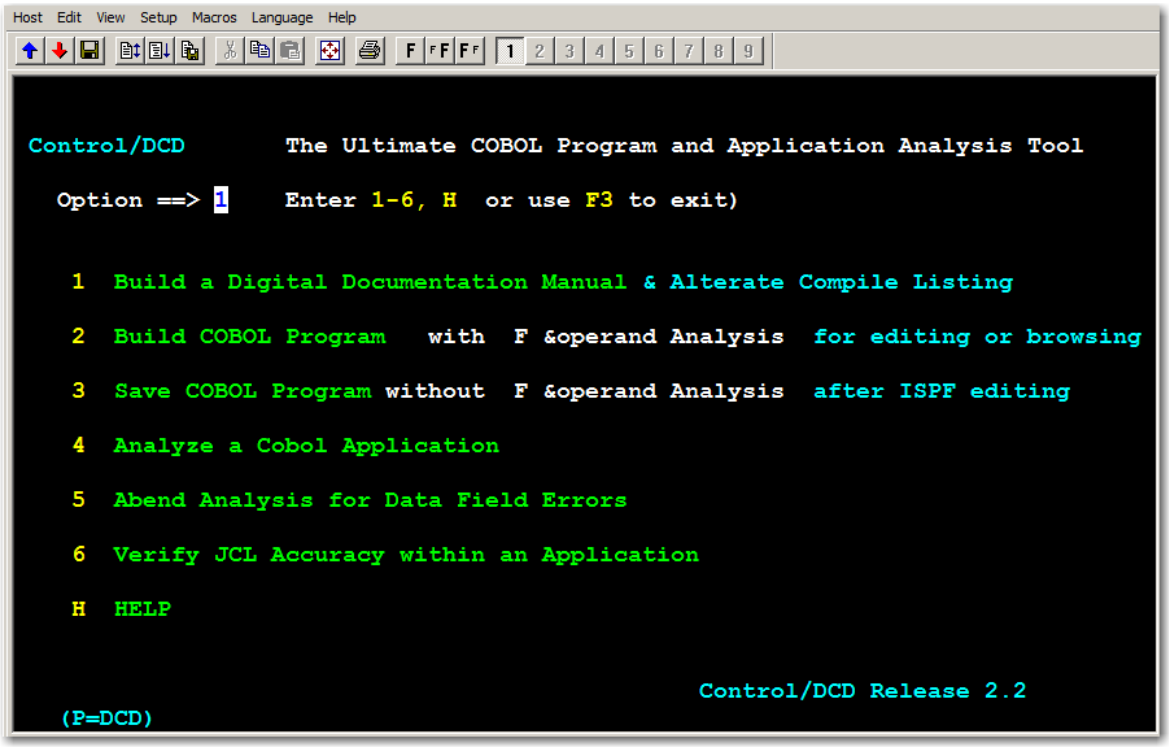
- We recommend using Enhanced ISPF Panels, over JCL when feasible.
- Normally only 2-3 panels need to be accessed to generate your output.

The Control DCD Main Menu



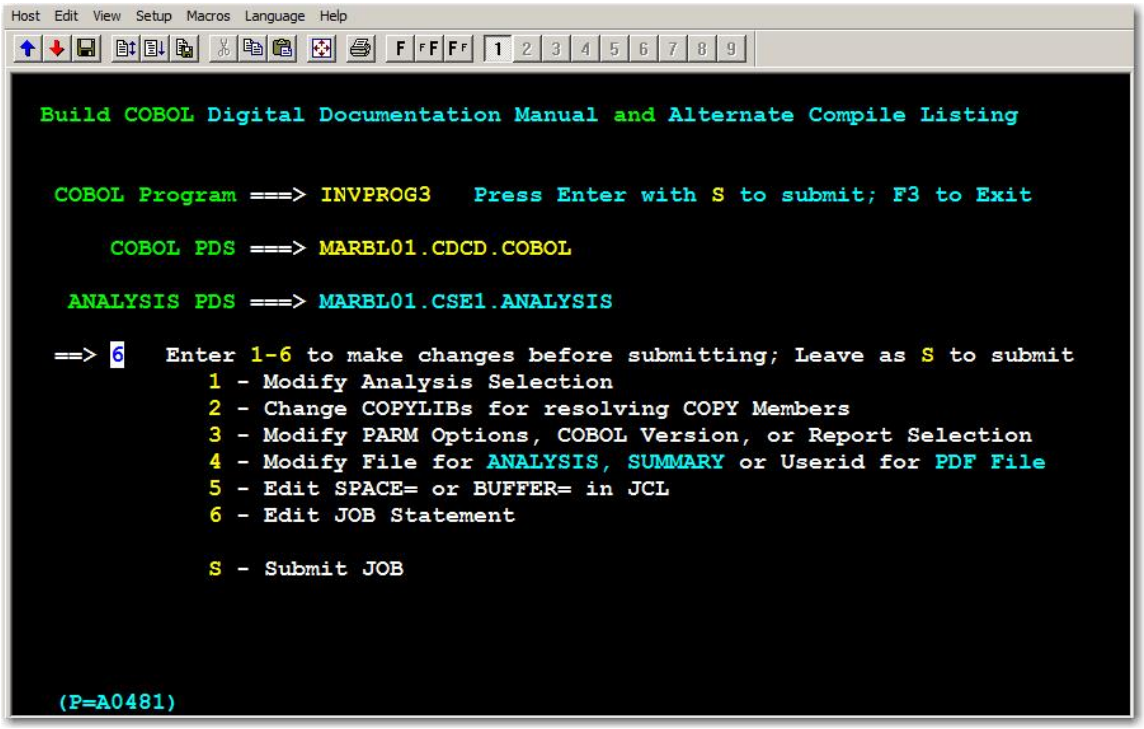
Recap
Main Menu

- Option 1 – Creates the DDM and "ACL"
- Option 2 – Edit and Browse the Source Code. Code includes the Narrative found in the DDM.
- Option 3 – Saves the Source Code without the Narrative.
- Option 4 – Analyzes a COBOL Application
- Option 5 - Abend Analysis
- Option 6 – JCL Accuracy Check
- Option H – Help Screens



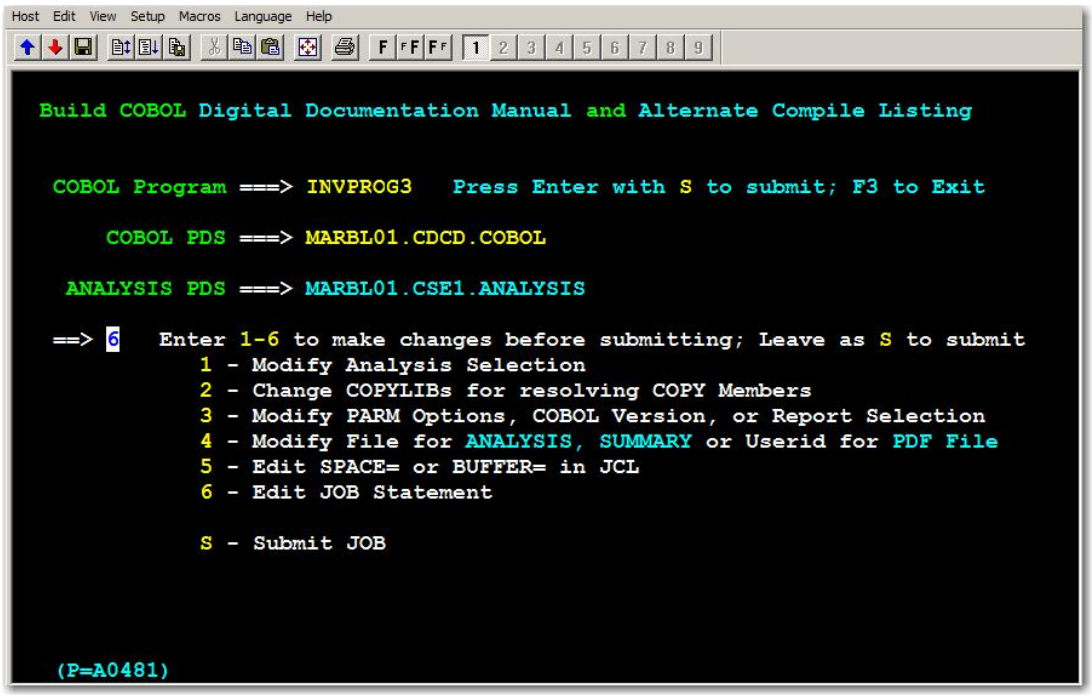
Select 1 and press enter

Use Option 6, to initially set up JOB Statement



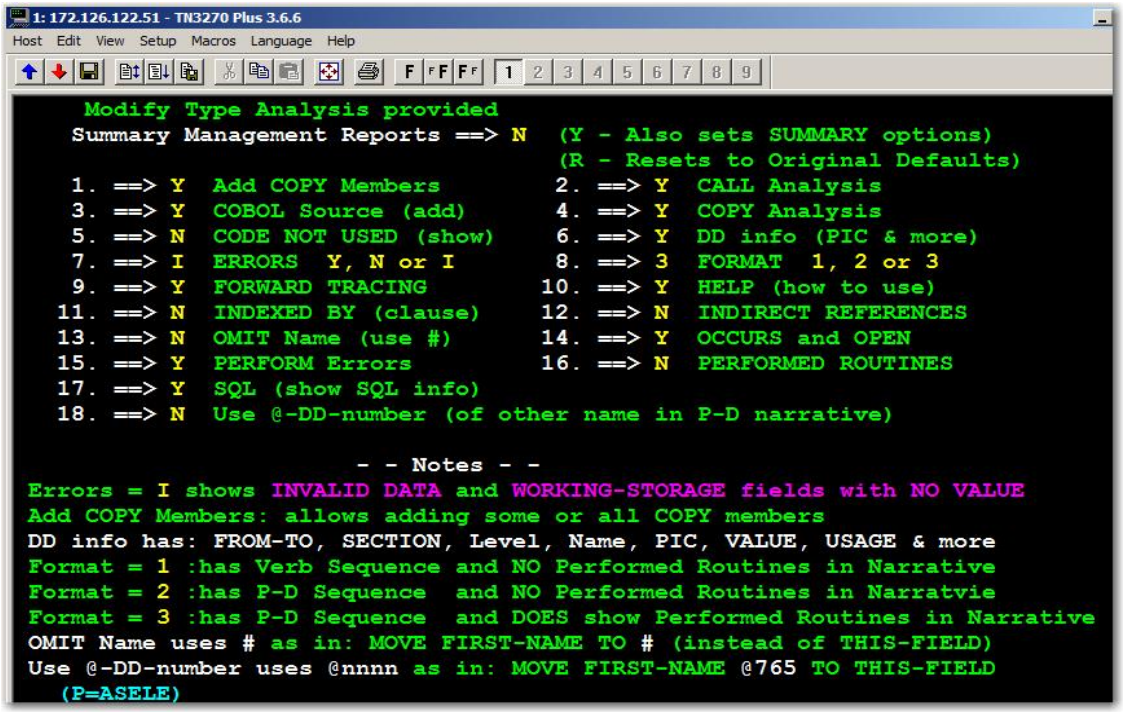
Enter the JOB statement values according to Z/OS requirements and your installation standards. Use all 4 lines.

Use Option 1-6, to Make Changes



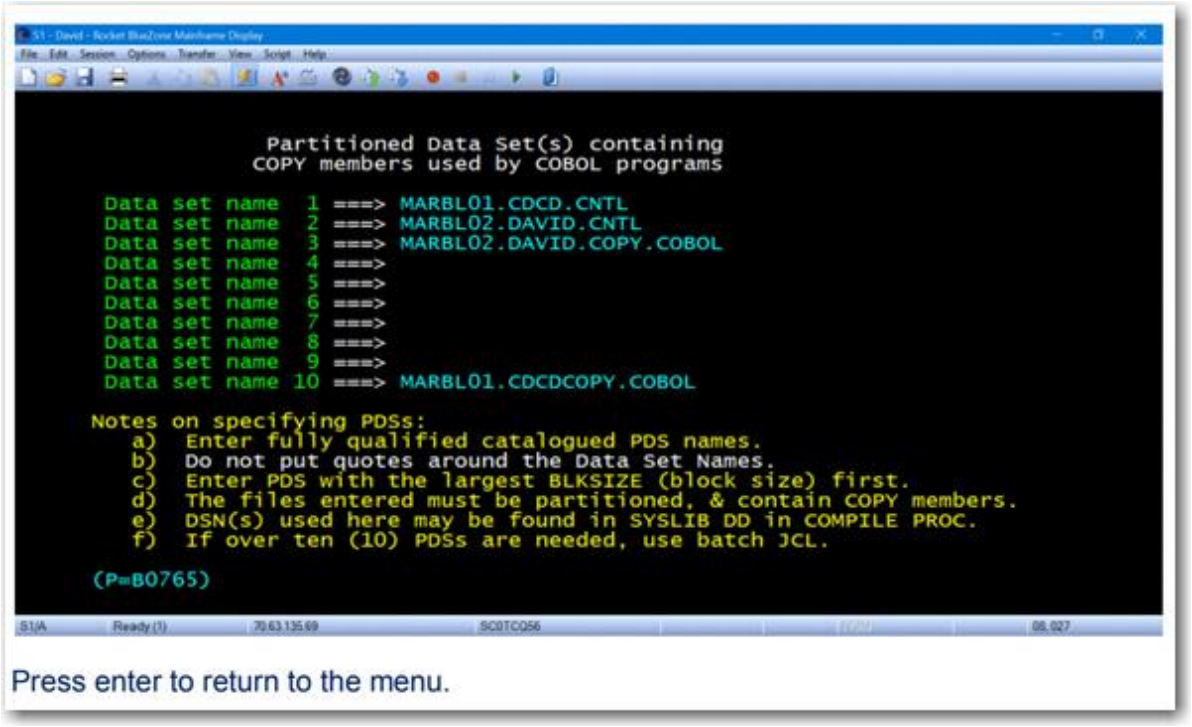
Enter 1 or 2 to Modify Options or Change COPYLIBs

The user may choose what type of analysis information will be included in the ISPF EDITFILE

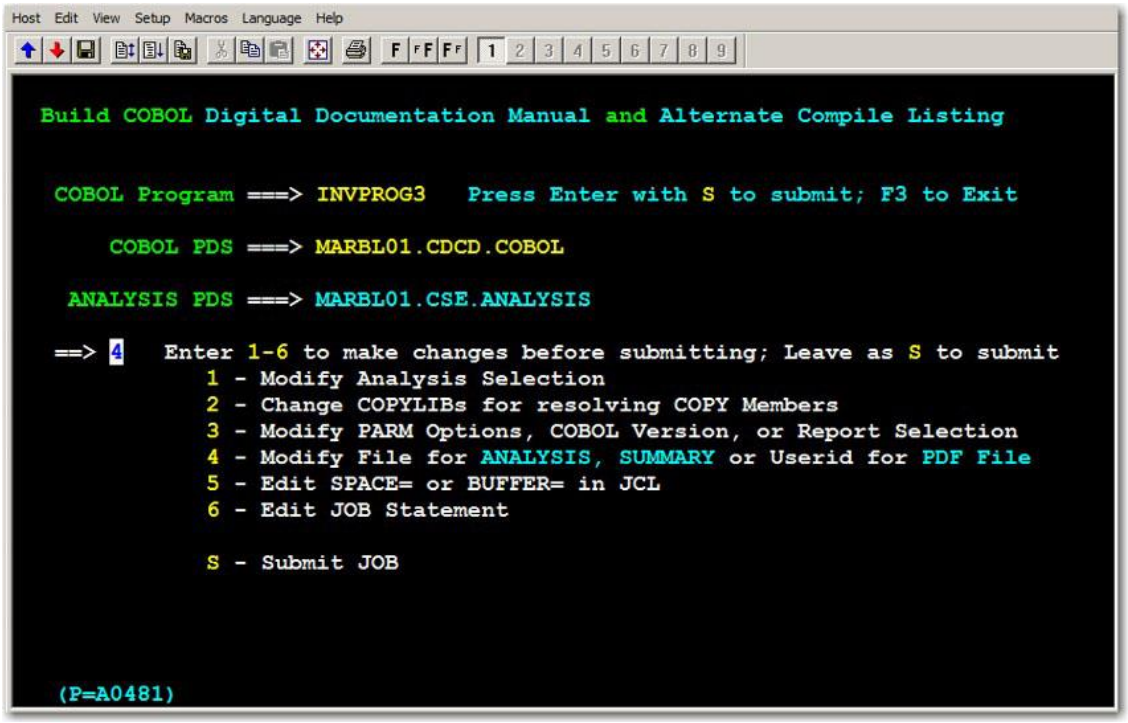


Examples for each type of ANALYSIS follow

Option 2 allows one time entering the correct COPYLIBs

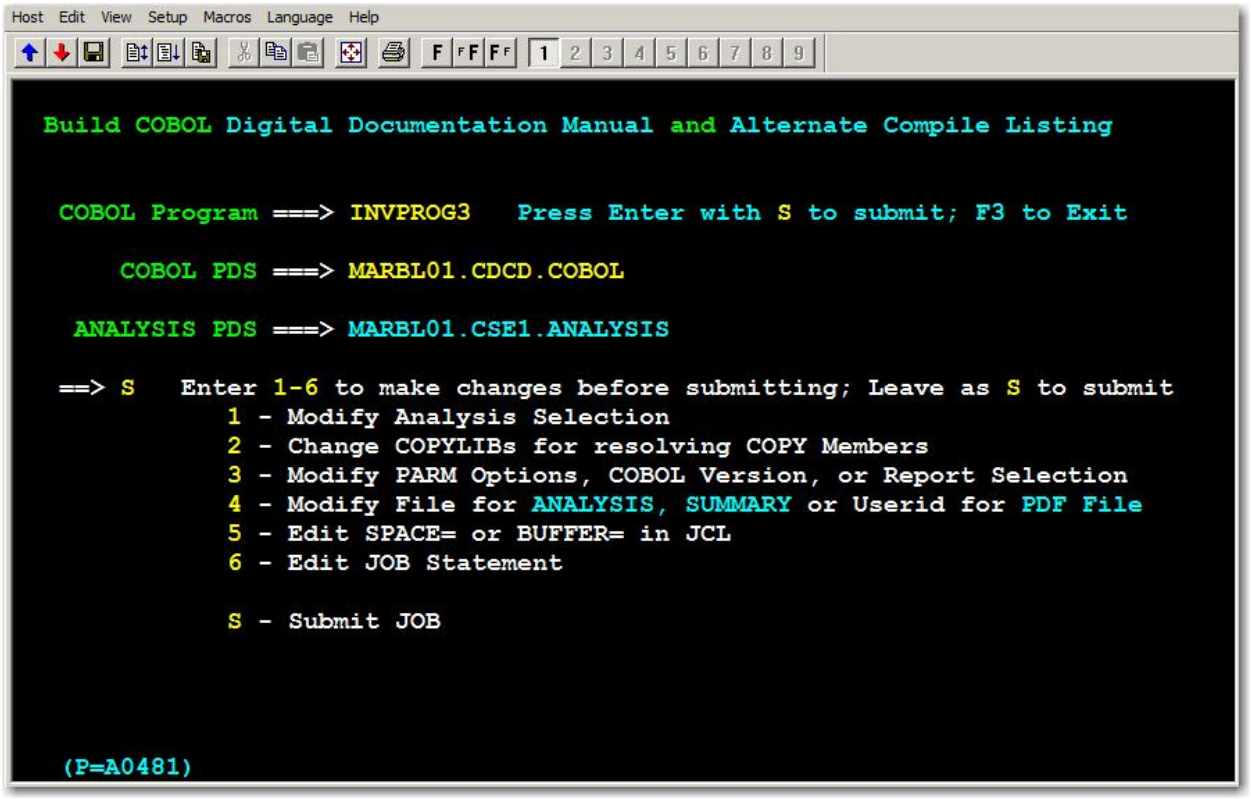


Option 4 is needed initially to provide USERID



Modifying Install file for .ANALYSIS & .SUMMARY is possible here!

1. Enter Program Name
2. Verify or change input COBOL PDS
3. Use ‘S’ option to submit job



Pressing enter with option S submits job

Recap
Run Time Options

1. Choose Analysis Options
2. Specify COPYLIBs (Initially used)
3. Modify PARM options (mostly unused)
4. Modify Analysis, Summary, USERID (Initially used)
5. Edit SPACE or Buffer Size (mostly unused)
6. Enter JOB Statement (Initially used)

S. Submit Command

Checking the Return Code for DDM

Check the return code after JOB finishes for return code of 0000, 0004, 0008, 0012 or 0016 similar to return codes from a COBOL compile.

Also check for two additional Return codes of 0000 used to create file 'userid.MANUAL4.progname.PDF'

Do a binary transfer of file:
'userid.MANUAL4.progname.PDF'
from the mainframe to the PC or other platform.

Click Here to Return to Index

Using the Analysis in a PDF Reader

Overview of Digital Documentation Manual PDF
(Many benefits)

Output is in PDF format available to any PDF reader

(PC, Adobe, MAC or any media)

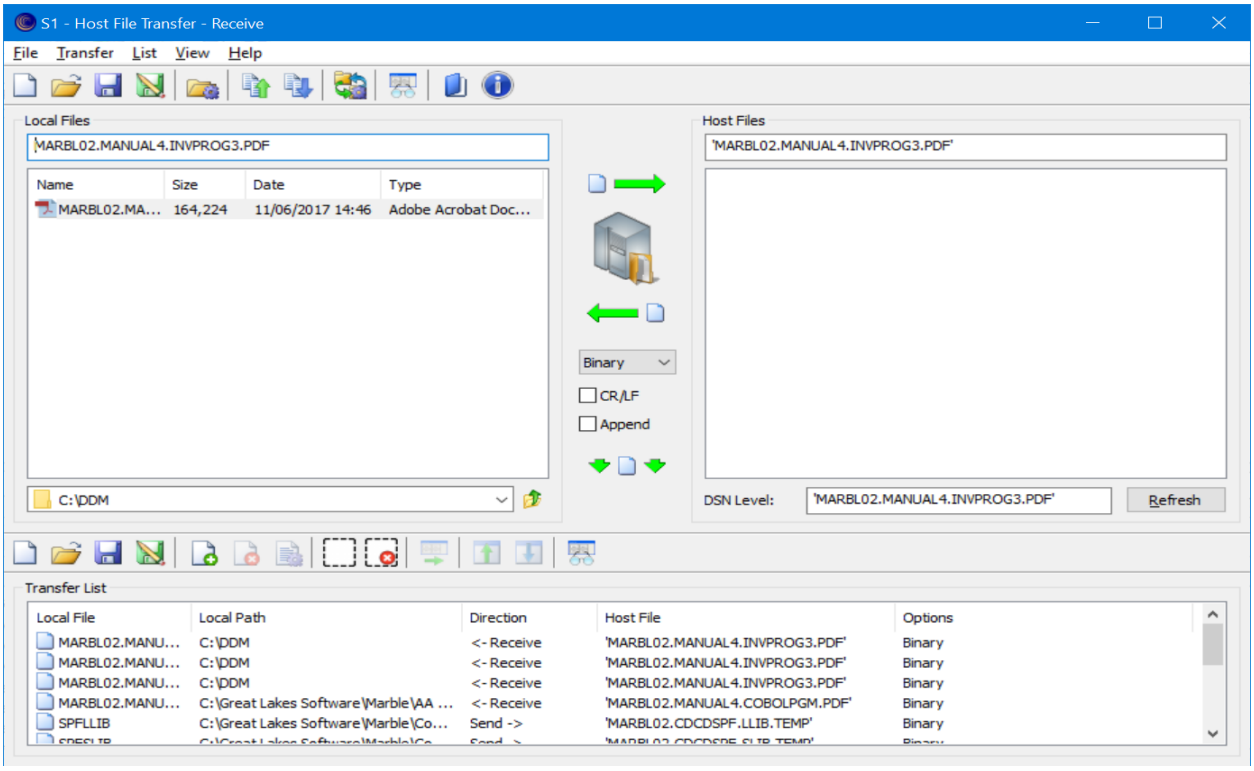
Ideal for Analysis from anywhere
Doing a search with **&operand** is used in the DDM (Digital Documentation Manual) and in the Cobol Source Editor(Option 2)

Only in DDM, **each data name** has its own reduced Forward-Tracing chart

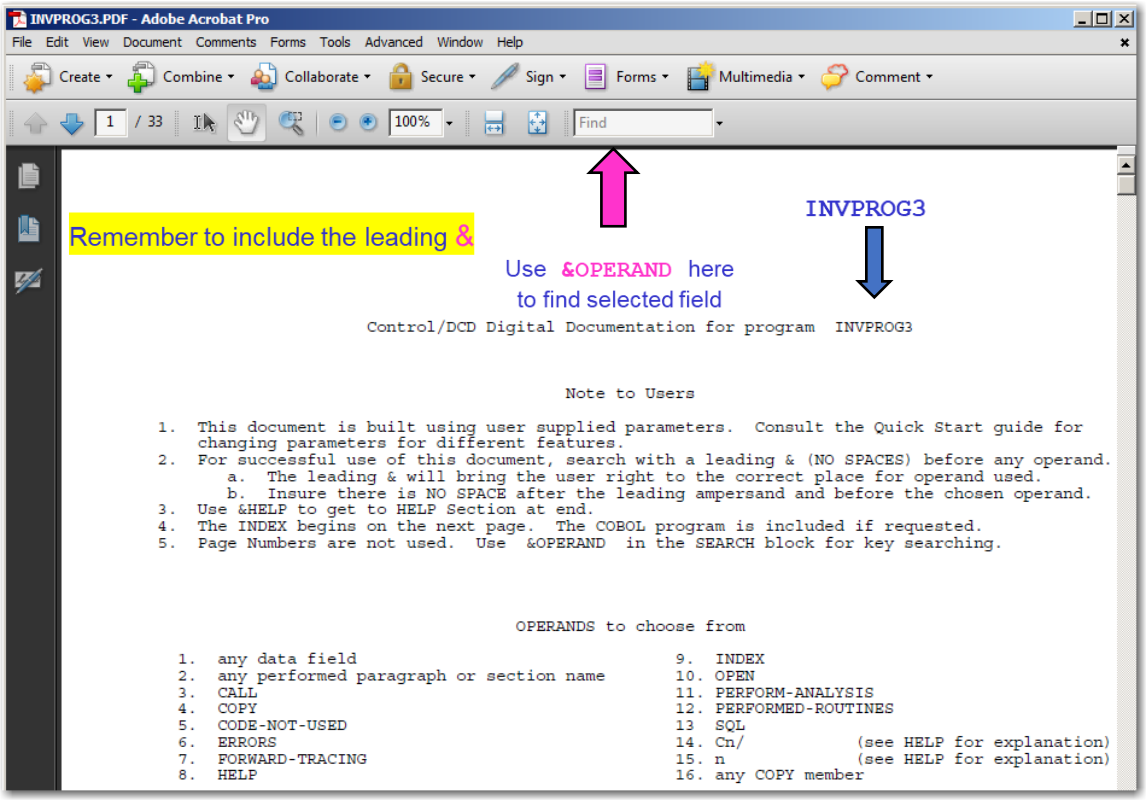
The DDM is created on the mainframe in PDF format and the user transfers file off the mainframe via a *simple* BINARY transfer

Documentation is now available off the mainframe

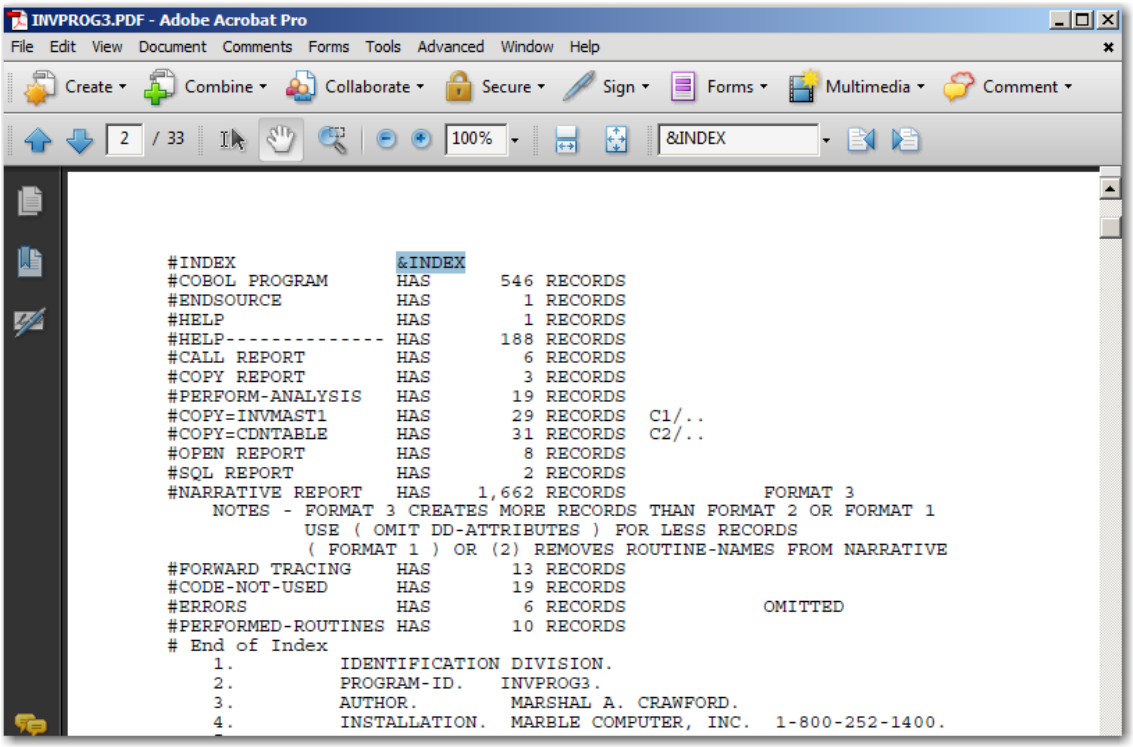
A PDF file prefixed with your USERID and a final suffix of PDF has been created on the mainframe and it is ready to download to the workstation. Use your installation's FTP product to transfer the file to the workstation.
The download type must be set to BINARY!



Many benefits to using Adobe or any other PDF reader



&INDEX points to DDM Index



For examples see Search Operands

Click Here to Return to Index

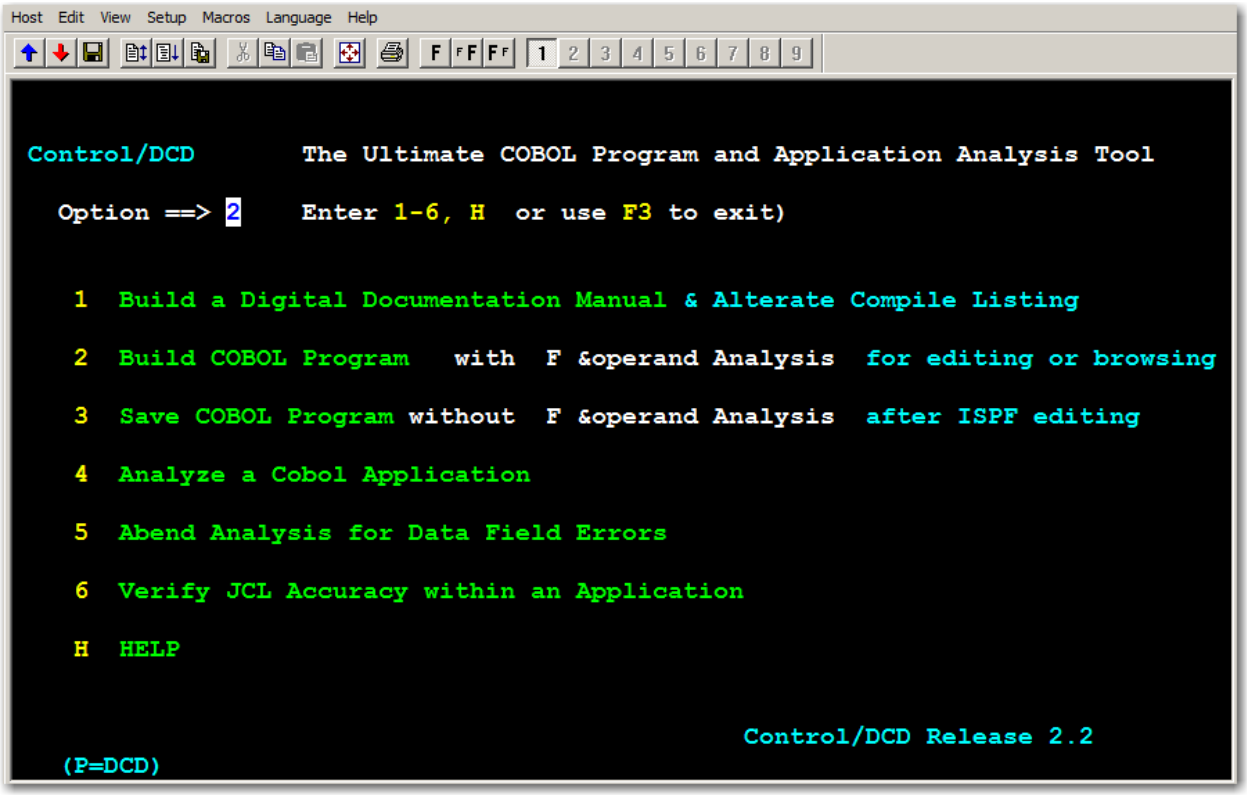
Using the Analysis on the Mainframe

Introducing the COBOL Source Editor

COBOL Source Editor in ISPF

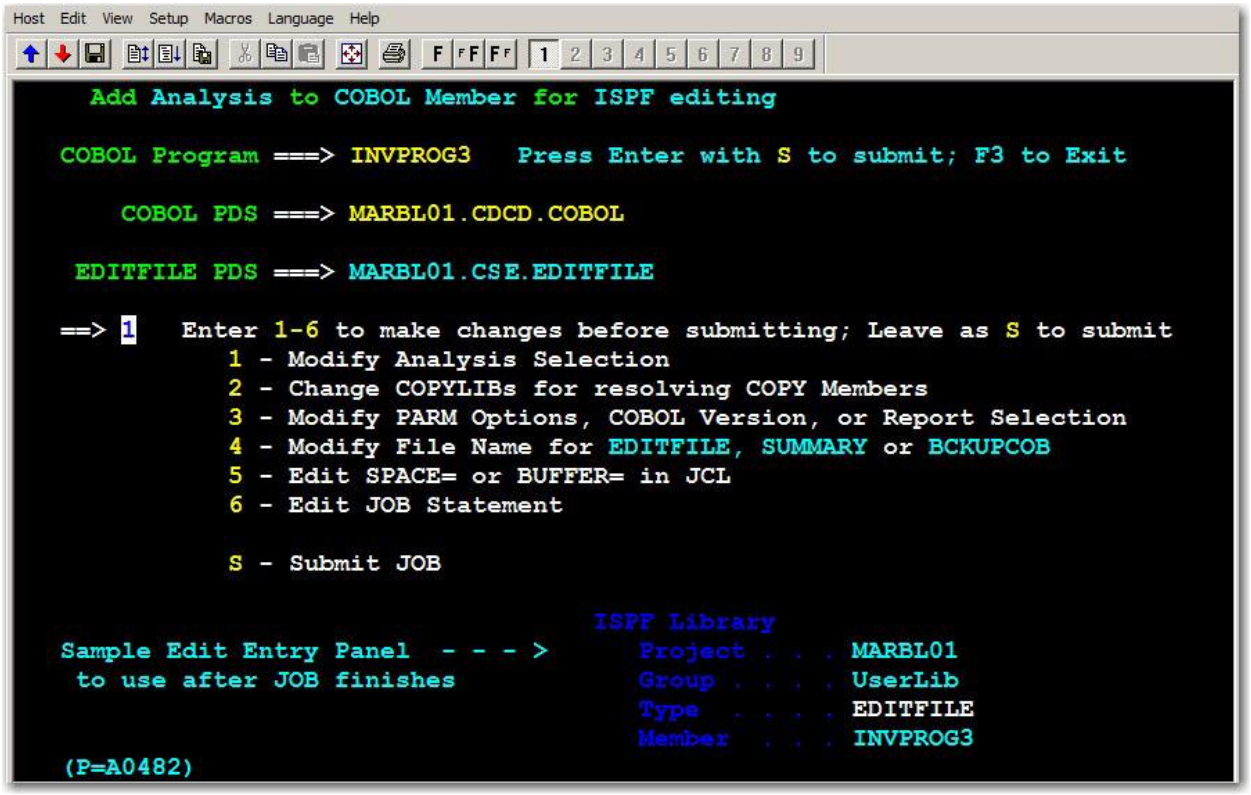
A Control/DCD analytical PDS report is produced as a member on the EDITFILE PDS that can be browsed or edited using ISPF using option 2.

In this feature, ALL code, analytics, and reports remain on the mainframe.

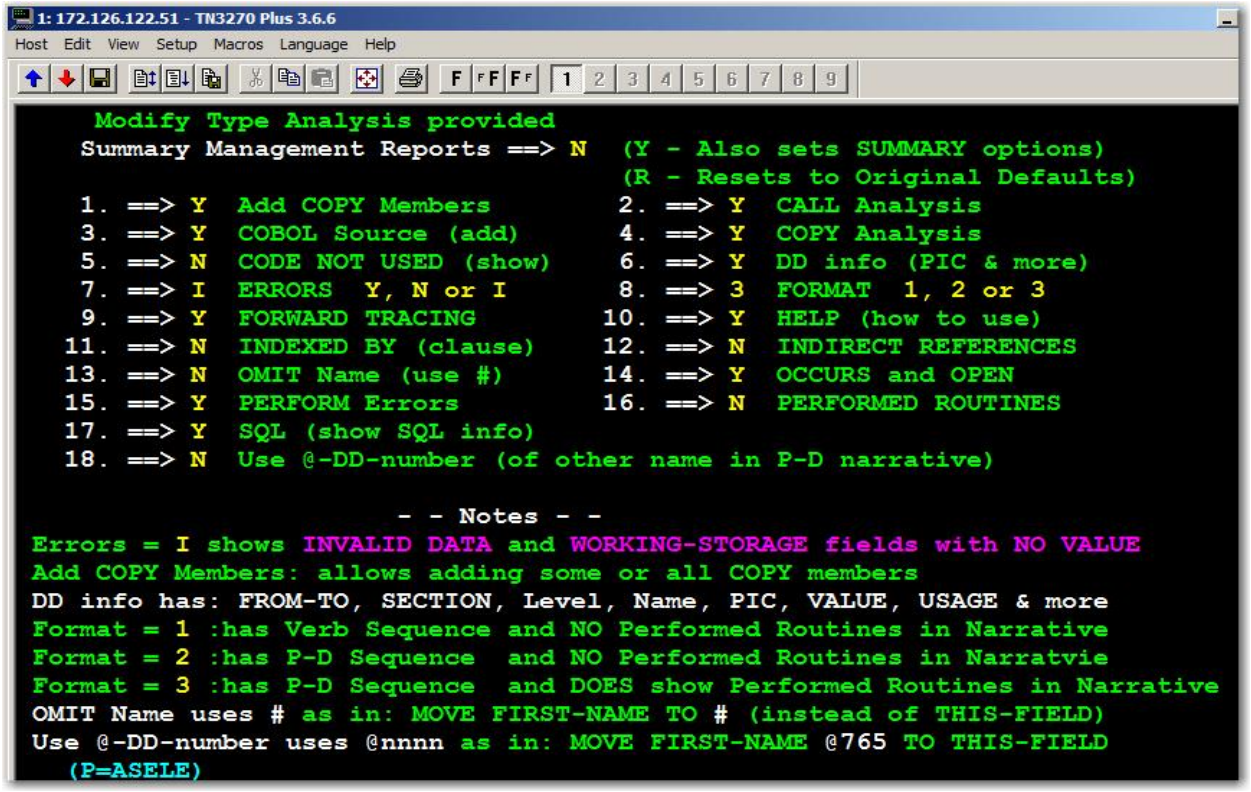


Select 2, and then press enter

Option 1 used to Modify Analysis Selection
Option 2 used to Change COPYLIBs for resolving COPYs

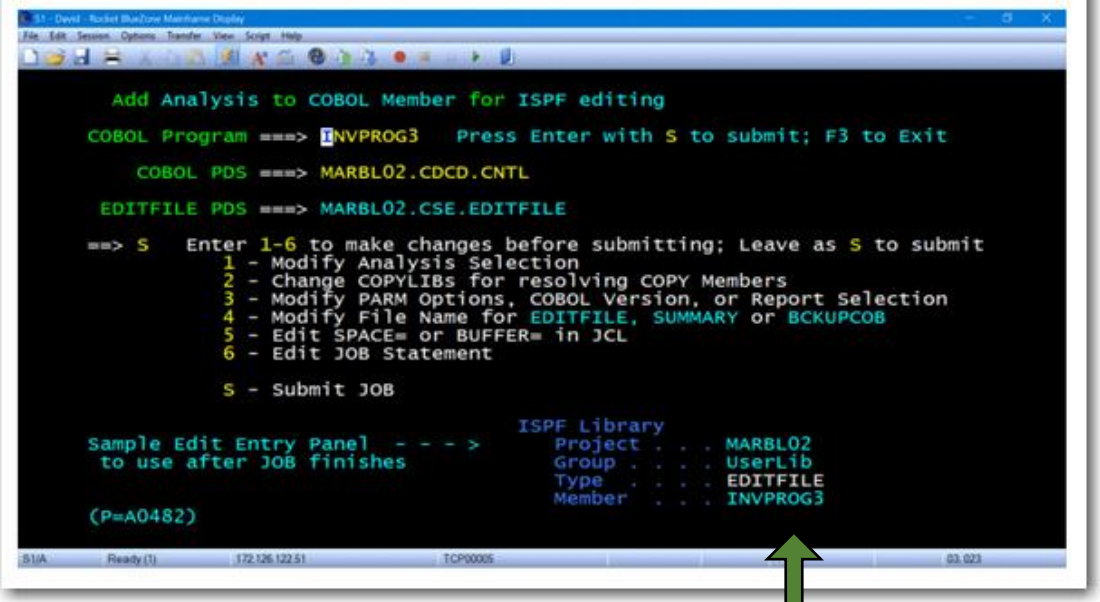


The user may choose what type of **ANALYSIS** information will be included in the ISPF EDITFILE



Examples for each type of ANALYSIS follow

This returns you to the Add Analysis to Cobol Member for ISPF editing sub menu.



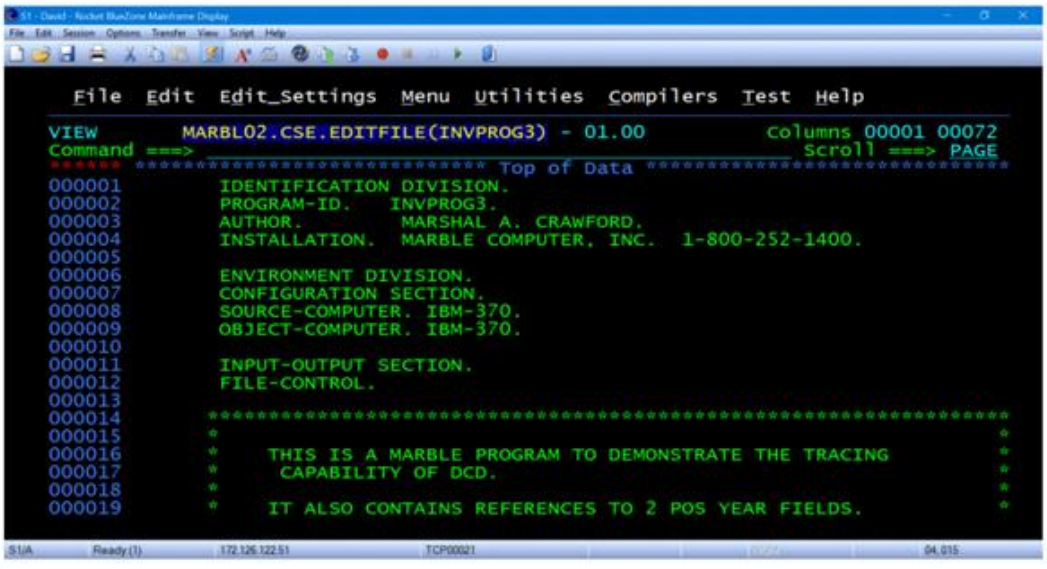
Use S option to submit job, then check return code.

Then go to the Member built in EDITFILE to browse or edit.

Go to EDIT or BROWSE panel to see CODE with Narrative.
Use EDIT panel if user wants to edit source code for later saving.

In this example;

MARBL02.CSE.EDITFILE(INVPROG3)



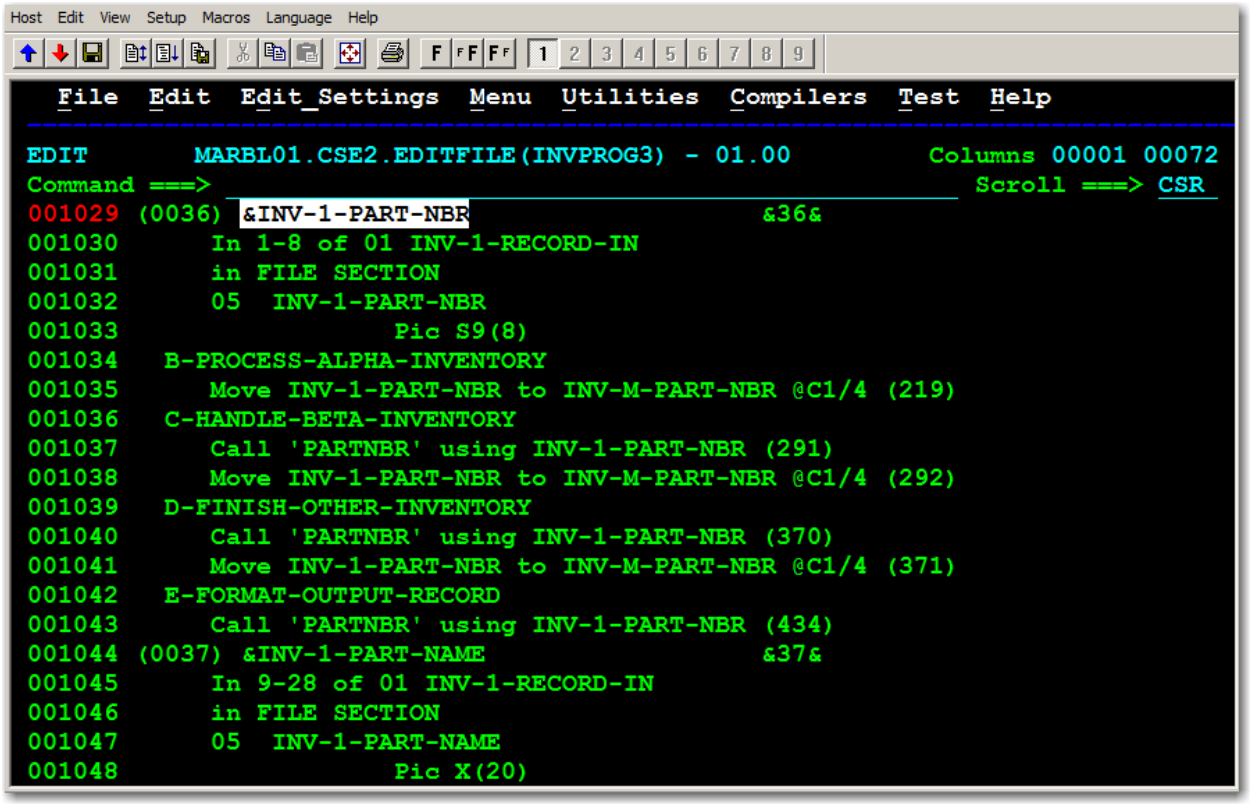
Using FIND command to get documentation

Unique analysis is available – Leading & is required.

Example of a simple FIND: **FIND &COPY**
List of operands available using a leading &.

Field Name	**INDEX**
Performed Routine	**OPEN**
CALL	**PERFORM-ERRORS**
COPY	**PERFORMED-ROUTINES**
CODE-NOT-USED	**SQL**
ERRORS	**Cn/**
FORWARD-TRACING	**n FIRST**
HELP	**Copy-Member-Name**

FIND &Data-Name Example – **F &INV-1-PART-NBR**



For examples see Search Operands

Click Here to Return to Index

Using the Alternate Compile Listing

- **Main** source listing with **Narrative**
- many small supporting reports
- **Condensed Data Division** cross reference
- Alternate Compile Listing is still available and can be useful, though newer features are highly recommended!

Condensed Procedure Division Lines

put NARRATIVE on the 133-character source listing

DCD Narrative Description

Part I

DCD Narrative on right:

70 01 **INV-2-RECORD-IN**. > **Move SPACES to # (444,555)**

indicates PROCEDURE DIVISION lines:

444 **MOVE SPACES TO INV-2-RECORD-IN.**

555 **MOVE SPACES TO INV-2-RECORD-IN.**

DCD Narrative Description

Part II

DCD Narrative:

71 **05 INV-2-PART-NBR** > **Move INV-M-PART @103 to # (447)**

indicates the presence of:

103 **05 INV-M-PART**

and indicates PROCEDURE DIVISION line:

447 **MOVE INV-M-PART TO INV-2-PART-NBR.**

Alternate Compile Listing

DATA DIVISION

```
SQ-NBR  1--6     PROGRAM-ID YEAR2POS   mm/dd/yyyy  …  73—80      # REFERS TO DATA-NAME AT THIS LINE

     66        FD INV-MAST-FILE-2                              > Close #  (214)
                                                                 Open Output  #  (197)
     67             RECORD CONTAINS    94  CHARACTERS
     68             BLOCK CONTAINS      0  RECORDS
     69             LABEL RECORDS ARE  STANDARD.
     70        01  INV-2-RECORD-IN.                            > Move  SPACES  to  #   (444)
                                                                 Write #  (231)
     71             05  INV-2-PART-NBR          PIC S9(8).     > Move INV-M-PART-NBR @103 to # (447)
     72             05  INV-2-PART-NAME         PIC X(20).        > If  'NON-UNIFORM-PART'  =  #  (450)
                                                                 Move 'MARK-TO-REMOVE' to # (451)
                                                                    INV-M-PART-NM @104 to # (448)
     73             05  INV-2-ORIG-DATE.                        > Move INV-M-ORIG-DT @105 to # (453)
     74                 10  INV-2-ORIG-YEAR     PIC 99.
     75                 10  INV-2-ORIG-MONTH    PIC 99.
                76                                                  05          INV-2-ORIG-CONTACT-NAM  PIC X(20).
                                                                > Move INV-M-ORIG-NAM @108 to # (455)
```

Quick description of Alternate Compile Listing

Tracing between DIVISIONs

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

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

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

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

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

Alternate Compile Listing

PROCEDURE DIVISION

```
SQ-NBR  1—6    PROGRAM-ID   YEAR2POS   mm/dd/yyyy …  73—80        SQ-NBRS    LOGIC FLOW

   442         E-FORMAT-OUTPUT-RECORD     SECTION.                              PERFORMED BY  300   373
   443         E010-CLEAR-OUTPUT-RECORD.
   444              MOVE   SPACES              TO  INV-2-RECORD-IN.      70
   445
   446         E020-FORMAT-RECORD.
   447              MOVE  INV-M-PART-NBR  TO  INV-2-PART-NBR.      103   71
   448              MOVE  INV-M-PART-NM   TO  INV-2-PART-NAME.     104   72
   449
   450              IF  'NON-UNIFORM-PART'  =  INV-2-PART-NAME      72
   451                   MOVE 'MRK-REM'     TO  INV-2-PART-NAME.    72
   452                   PERFORM       P-TEST-PARM-FIELD.                      PERFORM  523  THRU  538
   453
   454         E999-EXIT.  EXIT.                                               RETURN TO PERFORM AT 300  373
```

Recap for Alternate Compile Listing

- Original way DCD created **Narrative**
- Still available for user's comfortable with it
- Control/DCD expanded on this Narrative to create the previously discussed DDM and yet to be discussed COBOL Source Editor
- Require more paging than newer features

Click Here to Return to Index

Comparison Chart

Features	Control/DCD	DCD/IV
Abend Analysis	✓	✗
Alternate Compile Listing	✓	✓
Analysis of Multiple COBOL programs	✓	✓
Assist in Conversion to Cobol 6	✓	✗
Supports COBOL 6 code	✓	✗
COBOL Source Editor	✓	✗
Compile Mode	✗	✓
Dead Code Analysis	✓	✓
Digital Documentation Manual	✓	✗
• Abbreviated Forward Tracing	✓	✗
Forward Tracing	✓	✗
INDEX	✓	✗
JCL Analysis	✓	✓
PERFORM Analysis and listing Errors	✓	✗
Procedure Division Narrative	✓	✓
• Performed Routine with activity	✓	✗
Search by & character before operand	✓	✗
Simplified ISPF Panels	✓	✗
System Record Analysis across programs	✓	✓

Click Here to Return to Index

Support

Marble Computer, Inc

1-800-252-1400

www.marblecomputer.com

Training@marblecomputer.com

Tech@marblecomputer.com

Sales@marblecomputer.com

Click Here to Return to Index