



# Control/SE

## Quick Start Guide

September, 2016 Release 2.1

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

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

Support@marblecomputer.com  
www.marblecomputer.com

# Table of Contents

<b>I.</b>	<b>Control/SE Overview .....</b>	<b>2</b>
	The Need for Control/SE	
	Overview of Control/SE	
	The Control/SE Command Structure	
	More notes on the FIND Command	
<b>II.</b>	<b>Using Control/SE .....</b>	<b>6</b>
	JCL Information	
	Controlling the EDITFILE information created	
	Modifying the Control/SE Defaults	
	Editing the COBOL Program	
	Sample ISOLATE Program JCL	
<b>III.</b>	<b>Examples .....</b>	<b>10</b>
	F &IPT-NBR-ENTRIES	
	F &M-MERGE-THREE-FILES-TO-ONE	
	F &CODE-NOT-USED	
	F &PERFORM-ANALYSIS	
	F &FORWARD-TRACING	
	F &OPEN	
	F &INDEX	
	OTHER Finds	
<b>IV.</b>	<b>Summary Management Reports .....</b>	<b>17</b>
	How to create and access	
<b>V.</b>	<b>Alternate Compile Listing .....</b>	<b>17</b>
	How to access and more information	

## Control/SE Overview

---

### The Need for Control/SE

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 with a severe shortage of experienced COBOL programmers. Some organizations have opted to develop in-house COBOL programming talent via training courses at their internal “COBOL University”.

This creates new problems in the maintenance of the code, the predominance of which is COBOL although Assembler and PL/1 as well as Java are routinely found on today’s Mainframes.

While IBM and other Independent Software Vendors (ISVs) have pushed efforts to remove the coding off the mainframe through new Development Environments (IDEs) like Rational, Topaz and other products continues to grow slowly. The reason, the preferred platform for developing and maintaining mainframe code is still the Interactive System Productivity Facility (ISPF). 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 as long as there are mainframes.

Marble Computer has designed Control/SE 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 the majority of mainframe organizations. We’ve taken our 38 years of supporting COBOL and COBOL programmers and put that logic and structure to work in Control/SE.

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.

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

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

Thank you,

The Marble Development Team

## Overview of Control/SE

Control/SE is software for COBOL programs. It has three parts.

1. An Editable Source File with viable ANALYSIS info added to it.
2. An 'Alternate Compile Listing' with special added ANALYSIS.
3. A Summary Management System that maintains Analysis Information by program.

### Outline for using the Editable Source File:

- a. A job is run using the CSE PROC to build an Editable Source File with which contain the Cobol Source Code with added ANALYSIS information after the end of the COBOL source. This filename for this file is found within the //EDITFILE DD.
- b. The user edits this modified and editable COBOL source file within ISPF using the ANALYSIS information at the end of the source to greatly speed and help the analysis of the COBOL program.
- c. When the program is successfully modified, the user gets out of the modified and editable COBOL source file and then runs ISOLATE JCL to isolate and save just the COBOL part of this file to a normal COBOL Partitioned Data Set.

First look at sample JCL for running the CSE PROC follows:

```
//STEPB EXEC CSE, PROGRAM=USERPROG
//CSE.COBOLIN DD DSN=USERID.PREFIX.COBOL (&PROGRAM) , DISP=SHR
//CSE.COPYLIB DD DSN=USERID.COPYLIB.COBOL, DISP=SHR
//CSE.EDITFILE DD DSN=USERID.CSE.EDITFILE (&PROGRAM) , DISP=SHR
//CSE.USERSEL DD *
FORMAT 3
ADD CODE-NOT-USED
```

*Notes – The COBOLIN and COPYLIB DD' s are not needed if they are correctly defined within the CSE PROC. The USERSEL DD is optional, ALTHOUGH useful to modify the options controlling what type of additional information is provided or what information to omit in generating the Editable Source File.*

See other JCL information later in this Quick Start Guide that show the use of other DD's.

### The Control/SE Command Structure (for use in editing the COBOL program in ISPF)

The user modifies the COBOL program using ISPF in the same manner as before with two exceptions as follows:

- Use the FIND command with an & preceding the operand for special Control/SE finds.
- Consider using a special format for a global CHANGE command with ALL at the end to prohibit making global changes within the ANALYSIS area of the Editable file, unless the user also wants global changes for the ANALYSIS information. See next page.

## **FIND Command with & preceding the operand** (with no space between& an operand)

Below are ways to use Control/SE's Operands:

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

The next three example are explained under '**More notes on the FIND command**' on the next page.

13. F &Cn/
14. F &n FIRST
15. F &Copy-Member-Name

## **Special ISPF format of CHANGE Command** (suggested for Global changes follows)

C operand-1 operand-2 **range-begin** **range-end** [ALL]

For **range-begin**, use ISPF convention **.ZF** to indicate starting from program begin.

For **range-end**, set up **.E** as a label as follows:

1. Within the editable source file, Do a FIND &ENDSOURCE to find end of source as shown below:

```
001024 006512 LAST-RTN-EXIT. EXIT.  
001025 #ENDSOURCE          &ENDSOURCE
```

2. Go into the label area of the &ENDSOURCE line and modify TWO consecutive characters anywhere in this label area with the two characters (.E) as shown below:

```
001024 006512 LAST-RTN-EXIT. EXIT.  
001.E5 #ENDSOURCE          &ENDSOURCE
```

Now substitute **.ZF** and **.E** for **range-begin** and **range-end** as follows:

```
C operand-1 operand-2 .ZF .E ALL
```

Notes - The **.E** label will stay there as long as the user stays within this editable file. When returning to ISPF after exiting, this label must be rebuilt. The range feature of ISPF should NOT be used when using **FIND &operand** as the FIND needs to go into the ANALYSIS area of the //EDITFILE DD file to find analysis information.

**More notes on the FIND command** (for #13, #14, and #15 above)

#13. **Find &Cn/** is useful when COPY members are included. To include COPY's use either ADD COPY=member OR ADD COPY=ALL as follows:

```
//USERSEL DD *  
ADD COPY=operand
```

The FIND &Cn/ takes the user to the beginning of this COPY member.

Optionally do a F &INDEX to see COPYs included within the INDEX.

#14. **Find &n FIRST-** *see example below*

Requires the following USERSEL command for building an @number in built narrative:

```
//USERSEL DD *  
ADD @
```

Narrative Line prompting the use of **F &98 FIRST** in example below:

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

The following FIND will bring up the Narrative for data name WS-CURR-NBR at line 98:

```
F &98 FIRST
```

The use of **FIRST** is needed as the data-name being branched to may be before or after this data-name within the added narrative.

#15. **FIND &Copy-Member-Name** is similar in function to the use of **Find &Cn/** in that it will branch to the beginning of that COPY member, provided that an ADD COPY= is available to bring that or all COPYs in.

### JCL information

```
//STEPB EXEC CSE,PROGRAM=USERPROG
//CSE.EDITFILE DD DSN=USERID.CSE.EDITFILE (&PROGRAM) , DISP=SHR
//CSE.VERSION DD *
VS-COBOL (or IBM-COBOL, VS-COBOL-II, ENTERPRISE)
//CSE.USERSEL DD *
FORMAT 3
ADD CODE-NOT-USED
OMIT DD-ATTRIBUTES
ADD COPY=ACCTMBR3
```

*Notes – The COBOLIN and COPYLIB DD's are omitted in this example indicating they are correctly defined within the CSE PROC. The VERSION DD is not needed for ENTERPRISE Cobol, but is needed for older COBOLs, except for COBOL 68, and 74 in which case the Control/SE user's manual must be consulted. The USERSEL DD is optional, however is useful to modify the options controlling what type of additional information is provided or what information to omit in generating the Editable Source File.*

*Additional DD's the user need to be aware of are BCKUPCOB, PRINT or SUMMARY .*

- BCKUPCOB is used to create a backup of the source code each time the CSE PROC is executed.
- PRINT contains the Alternate Compile Listing. See this topic in this Quick Start Guide.
- SUMMARY contains the Summary Management Reports file. See this topic in this Quick Start Guide.

### Controlling the EDITFILE information created

Consult the Examples later in this Quick Start Guide for a clearer guide on items discussed just below, such as 'Data Division Narrative', 'Procedure Division Narrative', and included 'Performed Routine Names'.

#### **Defaults established when no modifying control statements are provided.**

By default, the following reports are turned **OFF** (CODE-NOT-USED, ERRORS, & PERFORMED-ROUTINES). All other reports are turned **ON**.

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

Within the NARRATIVE report;

- Listing **INDEXED BY** references are turned **OFF**

- Listing **INDIRECT** references are turned **OFF**
- The presence of **@n** references following other names allowing ( **F &n FIRST** ) is turned **OFF**

## Modifying the Control/SE DEFAULTS

Use the **//CSE.USERSEL DD \*** to modify defaults

- All commands (**ADD**, **OMIT**, or **FORMAT**) must begin in column 1
- Leave only one space after the command **ADD**, **OMIT**, or **FORMAT**
- Any commands with an **\*** in column 1 will be omitted and not flagged as a command error

To **add** an omitted report, use one of the following **ADD** commands:

**ADD CODE-NOT-USED**  
**ADD ERRORS**  
**ADD PERFORMD-ROUTINES**

To **add** the Summary Management reports, use the following **ADD** command:

**ADD SUMMARY**

To **omit** a defaulted to report, use one of the following **OMIT** commands:

**OMIT COBOL**  
**OMIT CALL**  
**OMIT COPY**  
**OMIT FORWARD-TRACING**  
**OMIT HELP**  
**OMIT INDEX**  
**OMIT OPEN**  
**OMIT FORWARD-TRACING**  
**OMIT PERFORM-ANALYSIS**

To **change** **FORMAT 3** narrative for Data Division fields, use one of the following:

**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
  - 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
  - Performed Routine Names are **NOT** included



To modify other features in Control/SE, use one of the following:

<b>ADD @</b>	Adds @n (n=number) after other names in narrative allowing (F &n FIRST) to find the narrative for that specified name.
<b>ADD DD-ATTRIBUTES</b>	Use in FORMAT 1 to include Data Division attributes to Narrative.
<b>OMIT DD-ATTRIBUTES</b>	Use in FORMAT 2 or FORMAT 3 to omit Data Division attributes.
<b>ADD INDEXED</b>	Add INDEXED BY sub-clause in Narrative.
<b>ADD INDIRECT</b>	Add INDIRECT references into Narrative.
<b>OMIT NAME</b>	Replace the field name in the P-D Narrative with abbreviated # character.

To include one or more COPY members for ISPF viewing, do one of the following:

<b>ADD COPY=member</b>	Use to add an individual COPY member
<b>OR</b>	
<b>ADD COPY=ALL</b>	Use to add all COPY members, then consider OMIT commands, shown below.

If all COPYs have been selected with an **ADD COPY=ALL** command, then one or more **OMIT COPY=** commands may be used:

<b>OMIT COPY=member</b>	after selecting ALL COPYs, use to omit one or more members not wanted.
-------------------------	--

## Editing the COBOL Program

### More notes on the FIND command (for any of the following):

1. F &Data-Division-Field-Name
2. F &Performed-Paragraph (or Section)
3. F &CALL
4. F &COPY
5. F &CODE-NOT-USED
6. F &ERRORS
7. F &FORWARD-TRACING
8. F &HELP
9. F &INDEX
10. F &OPEN
11. F &PERFORM-ANALYSIS
12. F &PERFORMED-ROUTINES
13. F &Cn/
14. F &n FIRST
15. F &Copy-Member-Name

When using on any of the above, remember to include the preceding **&** ahead of the operand.

For numbers 1 & 2 above enter a data division field name or a procedure division performed paragraph or section name. You only need to enter enough characters for a unique find.

For numbers 3 through 12, enter the whole name shown **OR** enough characters to make a unique find.

For number 13, this refers to seeing a reference in CSE built Data Division Narrative to a field that resides within a COPY member as shown on the example on the next line.

```
MOVE IN-COPY-NAME @C3/24 TO THIS-FILE (1293)
```

As example, the letter n following the capital C refers to the sequential number of the COPY in this program. If the COPY member references has been included with an ADD COPY= under //USERSEL DD \*, doing a **FIND** on **&C1/** , **&C2/** etc. will bring up that COPY Member in the Analysis information.

For number 14, this refers to finding the narrative to another-name (e.g. MOVE THIS-NAME TO ANOTHER-NAME). This feature requires use of (**ADD @**) command in //USERSEL DD for inserting the @n after the other name in the current narrative. When (**FIND &n FIRST**) is used, (*where n = number following @ character*), the Narrative for the other name is brought up. See **ADD @** command above under the heading 'Modifying these DEFAULTS'.

For number 15, this is similar to number 13, in that an ADD COPY= must be present to bring in a COPY member before doing 'F &Copy-Member-Name'.

### Sample ISOLATE Program JCL (to save source when editing is complete)

```
//STEPB EXEC ISOLATE ,PROGRAM=TESTPROG
//STEP1.EDITFILE DD DSN=USERID.CSE.EDITFILE (&PROGRAM) ,DISP=SHR
//STEP1.OUTCOBOL DD DSN=USERID.PROD.COBOLE (&PROGRAM) ,DISP=SHR
```

## Examples

---

**ISPF Find on a data-name** shows **Data Division Information**,  
and shows **Procedure Division references**,  
along with the **Performed Routine** hosting each reference.

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==> F &IPT-NBR-ENTRIES
001460 (0163) &IPT-NBR-ENTRIES                &163&
001461      In 5-8 of IPT-INTERNAL-HOLD-TABLE
001462      in WORKING-STORAGE
001463      05 IPT-NBR-ENTRIES
001464          Pic S9(8) Value ZERO
001465          Usage is COMP
001466      Used in Depending on IPT-NBR-ENTRIES
001467 C-INITIAL-READ-FILE
001468      Move +45 to IPT-NBR-ENTRIES (260)
001469 N-SORT-INPUT-PROCEDURE
001470      Move WS-CURR-HOLD-NBR @125 to IPT-NBR-ENTRIES (342)
001471      Set IPT-INDEX @166 to IPT-NBR-ENTRIES (344)
001472 P-LOOK-FOR-MATCH
001473      Move IPT-NBR-ENTRIES to CPY-NBR-ENTRIES @131 (C2/32)
```

### Notes on Finding Data Name Analysis

- 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**
- **C2/32** in last **P-D activity** line refers to the 32nd line in the 2nd COPY member. If that COPY member is included with (**ADD COPY=member**), use (**F &C2/**) to go to code for 2nd COPY member.

## ISPF Find on a performed paragraph or section

shows **Forward Tracing** going forward from there.

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==>  F  &M-MERGE-THREE-FILES-TO-ONE
000653 4      276      &M-MERGE-THREE-FILES-TO-ONE
000654 5      560      &FILEIO-ERROR-ROUTINE --> (4 Performs)
000655 6      588      &FILEIO-RTN-2 --> (Perform/VARYING)
000656 7      318      &N-SORT-INPUT-PROCEDURE
000657 8      374      &N220-SEEK-MATCH THRU N299-EXIT -->
000658                                (Perform/UNTIL)
000659 9      395      &N410-GET-IPT-RANGE-SEQ THRU N499-EXIT -->
000660                                (Perform/VARYING)
000661 10     411      &N450-RELEASE-SORT-RECORD
000662 11     423      &N500-WRITE-SORT-REC THRU N599-EXIT -->
000663                                (2 Performs)
000664 (See #5)          FILEIO-ERROR-ROUTINE
000665 (See #10)        N500-WRITE-SORT-REC THRU N599-EXIT -->
000666                                (2 Performs)
000667 (See #10)        N500-WRITE-SORT-REC THRU N599-EXIT
000668 (See #10)        N500-WRITE-SORT-REC THRU N599-EXIT
000669 (See #8)        N410-GET-IPT-RANGE-SEQ THRU N499-EXIT -->
```

### Notes on Forward Tracing created

- Forward tracing shows the structure of PERFORMs with 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**. See the line **000664 (See #5)** five lines up from the bottom of above example.

## ISPF Find on CODE-NOT-USED

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

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==> F &CODE-NOT-USED
000574 #CODE-NOT-USED          &CODE-NOT-USED
000575          &Unreferenced PARAGRAPHS and SECTIONS for IPTPROG
000576 Before removing routine, check FALL THRU's not intended
000577          Unused SECTIONS
000578 Seq Number  Section/Paragraph Name      Message or Warning
000579 543          R-ENTRY-TO-SUB-ACCT-2  NO Direct Refs to SECTION
000580 Before removing routine, check FALL THRU's not intended
000581          Unused PARAGRAPHS
000582 Seq Number  Section/Paragraph Name      Message or Warning
000583 544          R010-BEGIN-SUB-ACCT-2  NO Direct Refs to PARAGRAPH
000584          &Unused 01 Records and 77 Entries for Prog: IPTPROG
000585 Seq Number  Lvl   Field Name
000586 146          01   WS-COUNT-SEQS
000587 175          01   S1-RECORD
000588          &Unused Data Names for Prog: IPTPROG
000589 Seq Number  Lvl   Field Name
000590 (39)***01**WS-AW-RECORD***** contains unused fields **
000591 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.

## ISPF Find on PERFORM-ANALYSIS

shows **Major Perform Errors**  
and **Moderate and Minor Perform Errors**

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==>  F &PERFORM-ANALYSIS (space after & to BYPASS FIND)
002657 PERFORM Warnings & Major Errors
002658 Count   Type & Seq Nbr(s)
002659
002660 01 GO TOs leaving range of PERFORM MAJOR PERFORM ERROR
002661 477
002662
002663 03 Backward GO TO's MINOR PERFORM WARNING
002664 753      833      912
002665
002666 PERFORM & GO TO activity
002667 Count   Type & Seq Nbr(s)
002668
002669 12 PERFORM SECTIONS
002670 242      248      258      304      306      310
002671 332      336      338      479      480      685
002672
002673 05 PERFORM Paragraphs
002674 356      400      414      428      452
002675
002676 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

## ISPF Find on FORWARD-TRACING

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==> F &FORWARD-TRACING
000511 #FORWARD TRACING      &FORWARD-TRACING
000512 1      229      PROGRAM-ENTRY
000513 2      265      C-BUILD-NARR-FILE-TO-MERGE
000514 3      487      MERGE-THREE-FILES-TO-ONE
000515 4      554      FILEIO-ERROR-ROUTINE --> (4 Performs)
000516 5      588      FILEIO-RTN-2 --> (Perform/VARYING)
000517 6      612      N-SORT-INPUT-PROCEDURE
000518 7      653      N220-LOOK-FOR-MATCH --> (Perform/UNTIL)
000519 (See #4)          FILEIO-ERROR-ROUTINE --> (2 Performs)
000520 8      892      P-SORT-OUTPUT-ROUTINE
000521 9      1021     G-BUILD-OPEN-NARR-TO-MERGE
```

### Notes on Forward Tracing

- Forward tracing shows the structure of PERFORMs with 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**.

## ISPF Find on OPEN

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==>F &OPEN
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.



## ISPF Find on INDEX available (shows all possible types of information available)

```
EDIT          USERL03.EDITFILE(IPTPROG) - 01.00
Command ==> F &INDEX
000851 #INDEX          &INDEX
000852 #COBOL PROGRAM  HAS          854 RECORDS
000853 #HELP           HAS          262 RECORDS
000854 #CALL REPORT   HAS           28 RECORDS
000855 #COPY REPORT   HAS           17 RECORDS
000856 #PERFORM-ANALYSIS HAS          34 RECORDS
000857 #COPY=FILEINFO HAS           18 RECORDS  C1/..
000858 #COPY=DDACLNBR HAS           50 RECORDS  C2/..  OMITTED
000859 #COPY=DCDPREC  HAS          139 RECORDS  C3/..
000860 #OPEN REPORT   HAS           16 RECORDS
000861 #NARRATIVE REPORT HAS        1,969 RECORDS
000862 #FORWARD-TRACING HAS           131 RECORDS
000863 #CODE-NOT-FOUND HAS           61 RECORDS          OMITTED
```

### Notes on INDEX

- The INDEX report shows a total of all possible records **available** by option in the //EDITFILE DD file.

## Other Finds

### Notes on Other Finds not shown

- FINDs on CALLs, COPYs show pertinent information on CALLs and COPYs.
- FINDs on ERRORs show syntax errors found when scanning the code, similar to Compile Errors.
- FINDs on HELP show information similar to what is in this Quick Start Guide.
- FINDs on PERFORM-ROUTINES show a list of all Paragraphs and Sections that are performed.
- FINDs on ( **Cn/** ) may be used when COPY members are included. See ADD COPY= under the heading ‘**Modifying the CSE Defaults**’ and see the heading ‘**More Notes on the FIND command**’.
- FINDs on ( **n FIRST** ) allow branching from the narrative on one name to the narrative on another name. See the heading ‘**More Notes on the FIND command**’.
- FINDs on ( **Copy-Member-Name** ) is similar in function to ( **Cn/** ). See the sub-heading ‘**More Notes on the FIND command**’.

## Summary Management Reports

---

### How to create and access

1. Use the //USERSEL DD \* with an ADD SUMMARY command as follows:

```
//USERSEL DD *  
ADD SUMMARY
```

2. Go to the //SUMMARY DD 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 information available there.

## Alternate Compile Listing

---

### How to access and more information

1. Go to the //PRINT DD to find the DSN used in that DD. The Alternate Compile listing is stored within that DD. Warning – When another CSE run is done on another program, that Alternate Compile Listing will be replaced by the other program if it is not save elsewhere first.
2. See the Control/SE User’s Manual for more information on the Alternate Compile Listing.