

XPEDITER/TSO — Setting Breakpoints

General Questions

Question	Page(s)
What is a breakpoint?	2
What is the primary command syntax to set BEFORE and AFTER breakpoints?	2
What is the line command syntax to set breakpoints?	2
Can I set multiple BEFORE and AFTER breakpoints with one command?	3
Can I see a list of all the breakpoints I have set?	3
Can I skip the execution of statement that has a breakpoint set on it?	3

Functionality / Scenario Questions

Question	Page(s)
How do I set breakpoints?	5

NOTE: The following examples apply to COBOL, PL/I, C and Assembler unless otherwise indicated.

What is a breakpoint?

A breakpoint is a location in your program where you want to suspend normal program execution and gain control.

- Default breakpoints are set in the following places in the main program:
 - before the Procedure Division
 - after every STOP RUN, GOBACK, or EXIT PROGRAM
- Breakpoints can be set on the following:
 - statement number
 - paragraph name
 - module name
- Breakpoint indicators are located between the statement number and the source code and are as follows:
 - BEFORE breakpoints are designated by the letter B
 - AFTER breakpoints are designated by the letter A
 - BEFORE and AFTER breakpoints set on the same statement are designated by the '@'
- Breakpoints can also be set conditionally using the following commands:
 - WHEN
 - COUNT
 - PAUSE

For more information on how to use the WHEN, COUNT, and PAUSE commands, see the Rapid Response section, *Gaining Control Conditionally*.

What is the primary command syntax to set BEFORE and AFTER breakpoints?

You can specify breakpoints using a primary command in a number of ways. To set a BEFORE breakpoint, use the primary commands BEFORE, BEF, or B followed by a statement number, paragraph name, or module name.

Examples: B 338
 BEFORE CWXTSUBC:
 BEF 0000-MAINLINE

To set an AFTER breakpoint, use the primary commands AFTER, AFT, or A followed by a statement number, paragraph name, or module name.

Examples: AFTER 353
 A CWXTDATE:
 AFT 1000-PROCESS-DATA

For an example of using a primary command to set breakpoints, see page 9.

What is the line command syntax to set breakpoints?

To specify a breakpoint using a line command, enter a B in the line command area to set a BEFORE breakpoint or an A in the line command area to set an AFTER breakpoint.

For an example of using line commands to set breakpoints, see page 10.

Can I set multiple BEFORE and AFTER breakpoints with one command?

There are a number of ways you can set multiple breakpoints with both primary and line commands.

Note: Most of the following examples use the **BEFORE** breakpoint; the **AFTER** breakpoint can be used in the same way.

Primary Commands:

You can enter the following primary commands to set generic or global breakpoints.

Examples:

BEFORE ALL STATE	This sets a breakpoint at every statement. This can also be issued by using the command BEFORE ALL.
BEFORE ALL PARA	This sets a breakpoint at every paragraph.
BEFORE ALL PROG (<i>COBOL only</i>)	This sets a breakpoint at every procedure division and entry statement in the current source.
BEFORE ALL PROC (<i>PL/I only</i>)	This sets a breakpoint at every procedure within the current procedure.

You can also issue a string of BEFORE and AFTER breakpoints using primary commands.

Examples: B 338 362 349
 B 367 428;A 367 389 427
 B 380 THRU 392

Line Commands:

You can enter block line commands to set up multiple breakpoints using the BB line command on the first and last lines for which you want to issue the breakpoints.

You can also enter multiple single breakpoints using the B and A line commands.

For an example of how to use multiple breakpoints, see page 7.

Can I see a list of all the breakpoints I have set?

To see a list of breakpoints you have set, use the SHOW BREAKS command. For an example of how to use the SHOW BREAKS command, see page 11.

Can I skip the execution of a statement that has a breakpoint set on it?

You can skip the execution of a statement that has a breakpoint set on it by using the SKIP command.

To use the SKIP primary command, use the command SKIP or S followed by a statement number, paragraph name, or module name. You can skip individual or multiple lines.

Examples: S 556
 S 529 THRU 536
 SKIP 6000-PRINT-EMPLOYEE-REPORT

To use the SKIP line command, enter an S on the line(s) you want to skip during program execution.

How do I set breakpoints?

This example shows you how to set BEFORE and AFTER breakpoints using primary and line commands. You will also view and delete the breakpoints you have set.

1. Begin a new test of your program.

```
Profile: DEFAULT ----- XPEDITER/TSO - STANDARD (2.1) -----
COMMAND ==> [ENTER]

COMMANDS:  Setup (Display Setup Menu)

TEST SELECTION CRITERIA:

          Program ==> CWXTCOB
          Entry Point ==>
          Load Module ==>

          Initial Script ==>
          Post Script ==>

          PARM String ==> 00003

File List/JCL Member ==> 'USERID.XPEDITER.V6R5M0.SAMPLIB(CWXTJCLC)'

Code Coverage Test? ==> NO
Is This a DB2 Test? ==> NO   Plan ==>           System ==>
Test with XPEDITER+ ==> NO   Qualified LU name ==> .
          Press ENTER to process or enter END command to terminate
```

Figure 1: Test Standard Screen

2. Scroll down.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> [PF8]                                SCROLL ==> CSR
                BEFORE BREAKPOINT ENCOUNTERED
                -----+-----
000347  01 PARMINFO                                >  ..00003
      ** END **

-----
----- Before CWXTCOB <>
===== B  PROCEDURE DIVISION USING PARMINFO.
000351      0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
000359          PERFORM 1000-PROCESS-DATA
000360              UNTIL END-OF-FILE
000361          PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362          MOVE 1 TO REGION-SUB
000363          IF END-OF-MONTH
000364          PERFORM 7000-PRINT-REGION-REPORT

```

Figure 2: XPEDITER Default Before Breakpoint

3. Scroll up.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> [PF7]                                SCROLL ==> CSR
PROGRAM: CWXTCOB      MODULE: CWXTCOB      COMP DATE: 02/25/99  COMP TIME: 09:47:36
                -----+-----
000347  01 PARMINFO                                >  ..00003
      ** END **

-----
----- Before CWXTCOB <>
000365      UNTIL REGION-SUB > 4.
000366      PERFORM 9900-CLOSE.
000367 A  GOBACK.
000368      ***
000369      *** DIFFERENT PROCESSING OCCURS BASED ON EMPLOYEE TYPE.  THERE
000370      *** ARE 3 VALID EMPLOYEE TYPES.  IF A RECORD IS READ CONTAINING
000371      *** A RECORD TYPE OTHER THAN H, S OR M, AN ERROR MESSAGE IS
000372      *** WRITTEN AND PROCESSING CONTINUES.
000373      ***
000374      1000-PROCESS-DATA.
000375      IF HOURLY
000376          PERFORM 2000-PROCESS-HOURLY
000377      ELSE
000378          IF SALES
000379          PERFORM 3000-PROCESS-SALES

```

Figure 3: XPEDITER Default After Breakpoint

4. Stop program execution before each paragraph. For more information on how to set multiple breakpoints, see *“Can I set multiple BEFORE and AFTER breakpoints with one command?”* on page 3.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> before all para                                SCROLL ==> CSR
PROGRAM: CWXTCOB      MODULE: CWXTCOB    COMP DATE: 02/25/99  COMP TIME: 09:47:36
                                -----+-----
000347    01 PARMINFO                                >  ..00003
          ** END **

-----
===== B  PROCEDURE DIVISION USING PARMINFO.
000351    0000-MAINLINE.
000352        PERFORM 9000-OPEN.
000353        PERFORM 9100-CHECK-PARM.
000354        IF GOOD-PARM
000355            PERFORM 9200-INIT
000356            PERFORM 8000-READ-INPUT
000357                UNTIL RECORDS-READ = START-NUMBER
000358                OR END-OF-FILE
000359            PERFORM 1000-PROCESS-DATA
000360                UNTIL END-OF-FILE
000361            PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362            MOVE 1 TO REGION-SUB
000363            IF END-OF-MONTH
000364            PERFORM 7000-PRINT-REGION-REPORT
----- Before CWXTCOB <>

```

Figure 4: Globally Setting Breakpoints Before All Paragraphs

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
                                1 COMMAND(S) COMPLETED
                                -----+-----
000347    01 PARMINFO                                >  ..00003
          ** END **

-----
===== B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352        PERFORM 9000-OPEN.
000353        PERFORM 9100-CHECK-PARM.
000354        IF GOOD-PARM
000355            PERFORM 9200-INIT
000356            PERFORM 8000-READ-INPUT
000357                UNTIL RECORDS-READ = START-NUMBER
000358                OR END-OF-FILE
000359            PERFORM 1000-PROCESS-DATA
000360                UNTIL END-OF-FILE
000361            PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362            MOVE 1 TO REGION-SUB
000363            IF END-OF-MONTH
000364            PERFORM 7000-PRINT-REGION-REPORT
----- Before CWXTCOB <>

```

Figure 5: Results of Globally Setting Breakpoints

5. Access additional information about the BEFORE command.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> help before                                SCROLL ==> CSR
                                1 COMMAND(S) COMPLETED
                                -----
000347    01 PARMINFO                                >  ..00003
    ** END **

----- Before CWXTCOB <>
====> B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
000359      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361      PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362      MOVE 1 TO REGION-SUB
000363      IF END-OF-MONTH
000364          PERFORM 7000-PRINT-REGION-REPORT

```

Figure 6: HELP Command

6. Exit the help screen.

```

----- XPEDITER/TSO - HELP -----
COMMAND ==> [PF3]                                SCROLL ==> CSR
PROGRAM: CWXTCOB    MODULE: CWXTCOB    COMP DATE: 02/25/99    COMP TIME: 09:47:36
----- Before CWXTCOB -----
***** TOP OF DATA *****

BEFORE

The BEFORE command sets a breakpoint to pause before execution of
an instruction. The breakpoint is indicated by a "B" in column 9
on the Source screen. If the line command area is expanded for
31-bit mode addressing, the breakpoint indicator is displayed in
column 11 on the Source screen.

"Before breakpoints" are automatically set at PROCEDURE DIVISION
(COBOL) or PROCEDURE (PL/I) and at one or more ENTRY statements in
the main module before a debugging session starts. These automatic
"before breakpoints" are interpreted as "before module" breakpoints.
All other "before breakpoints" are interpreted as "before
statement" breakpoints.

When the "before breakpoint" is encountered, XPEDITER/TSO temporarily
suspends execution of your program, issues a message, and returns
control to you at the terminal. Use the GO command when you are ready

```

Figure 7: Results of HELP Command

7. Stop program execution before statement 359. For more information on setting breakpoints using a primary command, see *“What is the primary command syntax to set BEFORE and AFTER breakpoints?”* on page 2.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> b 359                                SCROLL ==> CSR
PROGRAM: CWXTCOB      MODULE: CWXTCOB      COMP DATE: 02/25/99  COMP TIME: 09:47:36
                                -----+-----
000347      01 PARMINFO                                >  ..00003
          ** END **

----- Before CWXTCOB <>
===== B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
000359      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361      PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362      MOVE 1 TO REGION-SUB
000363      IF END-OF-MONTH
000364      PERFORM 7000-PRINT-REGION-REPORT

```

Figure 8: BEFORE Command

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
                                1 COMMAND(S) COMPLETED
                                -----+-----
000347      01 PARMINFO                                >  ..00003
          ** END **

----- Before CWXTCOB <>
===== B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
000359 B      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361      PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362      MOVE 1 TO REGION-SUB
000363      IF END-OF-MONTH
000364      PERFORM 7000-PRINT-REGION-REPORT

```

Figure 9: Results of BEFORE Command

8. Stop execution after statements 354 and 359 and before statement 361. For more information on using line commands to set breakpoints, see *“What is the line command syntax to set breakpoints?”* on page 2.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> [ENTER]                                SCROLL ==> CSR
                1 COMMAND(S) COMPLETED
                -----
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
=====> B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
a 354      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
a 359 B      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
b 361      PERFORM 6000-PRINT-EMPLOYEE-REPORT

```

Figure 10: BEFORE and AFTER Line Commands

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
                3 COMMAND(S) COMPLETED
                -----
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
=====> B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354 A      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357              UNTIL RECORDS-READ = START-NUMBER
000358              OR END-OF-FILE
000359 @      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT

```

Figure 11: Results of BEFORE and AFTER Line Commands

9. Display all breakpoints set.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> show breaks                                SCROLL ==> CSR
                                3 COMMAND(S) COMPLETED
                                -----+---
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
===== B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354 A      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357          UNTIL RECORDS-READ = START-NUMBER
000358          OR END-OF-FILE
000359 @      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362          MOVE 1 TO REGION-SUB
000363          IF END-OF-MONTH
000364          PERFORM 7000-PRINT-REGION-REPORT

```

Figure 12: SHOW BREAKS Command

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
      SPECIFIED STATEMENTS ARE SHOWN - RESTORE SOURCE WITH 'END'
                                -----+---
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
----- 349 LINES NOT DISPLAYED
===== B  PROCEDURE DIVISION USING PARMINFO.
000351 B  0000-MAINLINE.
----- 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
----- 4 LINES NOT DISPLAYED
000359 @      PERFORM 1000-PROCESS-DATA
----- 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
----- 5 LINES NOT DISPLAYED
000367 A      GOBACK.
----- 6 LINES NOT DISPLAYED
000374 B  1000-PROCESS-DATA.

```

Figure 13: Results of SHOW BREAKS Command

10. Delete the before breakpoint on statement 351 and after the after breakpoint on statement 359.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> [ENTER]                                SCROLL ==> CSR
          SPECIFIED STATEMENTS ARE SHOWN - RESTORE SOURCE WITH 'END'
          -----
000347    01 PARMINFO                                >  ..00003
          ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
----- 349 LINES NOT DISPLAYED
====> B  PROCEDURE DIVISION USING PARMINFO.
d 351 B  0000-MAINLINE.
----- 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
----- 4 LINES NOT DISPLAYED
da 359 @      PERFORM 1000-PROCESS-DATA
----- 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
----- 5 LINES NOT DISPLAYED
000367 A      GOBACK.
----- 6 LINES NOT DISPLAYED
000374 B  1000-PROCESS-DATA.
----- 20 LINES NOT DISPLAYED

```

Figure 14: DELETE Command

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
          DELETE COMMAND PROCESSED
          -----
000347    01 PARMINFO                                >  ..00003
          ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
----- 349 LINES NOT DISPLAYED
====> B  PROCEDURE DIVISION USING PARMINFO.
000351    0000-MAINLINE.
----- 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
----- 4 LINES NOT DISPLAYED
000359 B      PERFORM 1000-PROCESS-DATA
----- 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
----- 5 LINES NOT DISPLAYED
000367 A      GOBACK.
----- 6 LINES NOT DISPLAYED
000374 B  1000-PROCESS-DATA.
----- 20 LINES NOT DISPLAYED

```

Figure 15: Results of DELETE Command

11. Delete the before breakpoints on all paragraphs.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> delete before all para                                SCROLL ==> CSR
                DELETE COMMAND PROCESSED
                -----+---
000347  01 PARMINFO                                >  ..00003
    ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
- - - - - 349 LINES NOT DISPLAYED
====> B  PROCEDURE DIVISION USING PARMINFO.
000351  0000-MAINLINE.
- - - - - 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
- - - - - 4 LINES NOT DISPLAYED
000359 B      PERFORM 1000-PROCESS-DATA
- - - - - 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
- - - - - 5 LINES NOT DISPLAYED
000367 A      GOBACK.
- - - - - 6 LINES NOT DISPLAYED
000374 B  1000-PROCESS-DATA.
- - - - - 20 LINES NOT DISPLAYED

```

Figure 16: DELETE BEFORE Command

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
                DELETE COMMAND PROCESSED
                -----+---
000347  01 PARMINFO                                >  ..00003
    ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
- - - - - 349 LINES NOT DISPLAYED
====> B  PROCEDURE DIVISION USING PARMINFO.
000351  0000-MAINLINE.
- - - - - 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
- - - - - 4 LINES NOT DISPLAYED
000359 B      PERFORM 1000-PROCESS-DATA
- - - - - 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
- - - - - 5 LINES NOT DISPLAYED
000367 A      GOBACK.
- - - - - 6 LINES NOT DISPLAYED
000374  1000-PROCESS-DATA.
- - - - - 20 LINES NOT DISPLAYED

```

Figure 17: Results of DELETE BEFORE Command

12. Reset the excluded lines.

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==> [PF3]                                SCROLL ==> CSR
                DELETE COMMAND PROCESSED
                -----
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
***** TOP OF MODULE *****
----- 349 LINES NOT DISPLAYED
====> B  PROCEDURE DIVISION USING PARMINFO.
000351  0000-MAINLINE.
----- 2 LINES NOT DISPLAYED
000354 A      IF GOOD-PARM
----- 4 LINES NOT DISPLAYED
000359 B      PERFORM 1000-PROCESS-DATA
----- 1 LINE NOT DISPLAYED
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
----- 5 LINES NOT DISPLAYED
000367 A      GOBACK.
----- 6 LINES NOT DISPLAYED
000374  1000-PROCESS-DATA.
----- 20 LINES NOT DISPLAYED

```

Figure 18: Reset Excluded Lines

```

----- XPEDITER/TSO - SOURCE -----
COMMAND ==>                                SCROLL ==> CSR
PROGRAM: CWXTCOB    MODULE: CWXTCOB    COMP DATE: 02/25/99    COMP TIME: 09:47:36
                -----
000347  01 PARMINFO                                >  ..00003
      ** END **

----- Before CWXTCOB <>
====> B  PROCEDURE DIVISION USING PARMINFO.
000351  0000-MAINLINE.
000352      PERFORM 9000-OPEN.
000353      PERFORM 9100-CHECK-PARM.
000354 A      IF GOOD-PARM
000355          PERFORM 9200-INIT
000356          PERFORM 8000-READ-INPUT
000357          UNTIL RECORDS-READ = START-NUMBER
000358          OR END-OF-FILE
000359 B      PERFORM 1000-PROCESS-DATA
000360          UNTIL END-OF-FILE
000361 B      PERFORM 6000-PRINT-EMPLOYEE-REPORT
000362          MOVE 1 TO REGION-SUB
000363          IF END-OF-MONTH
000364          PERFORM 7000-PRINT-REGION-REPORT

```

Figure 19: Results of Resetting Excluded Lines