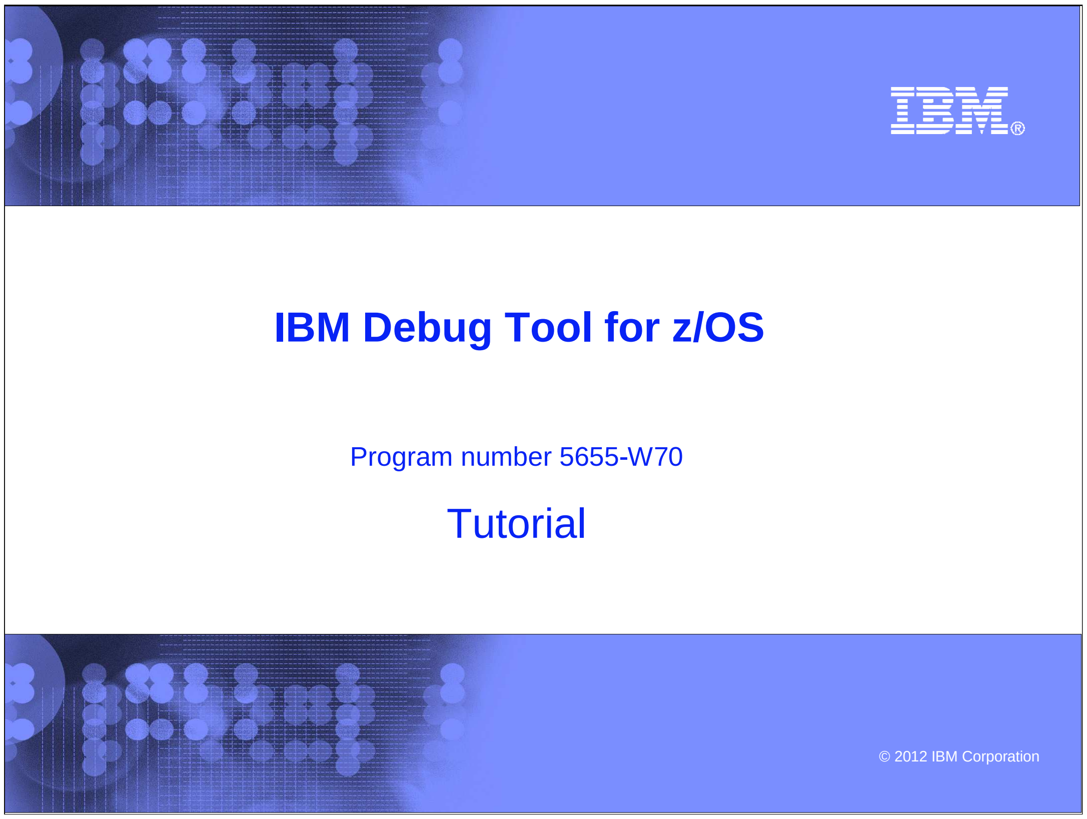
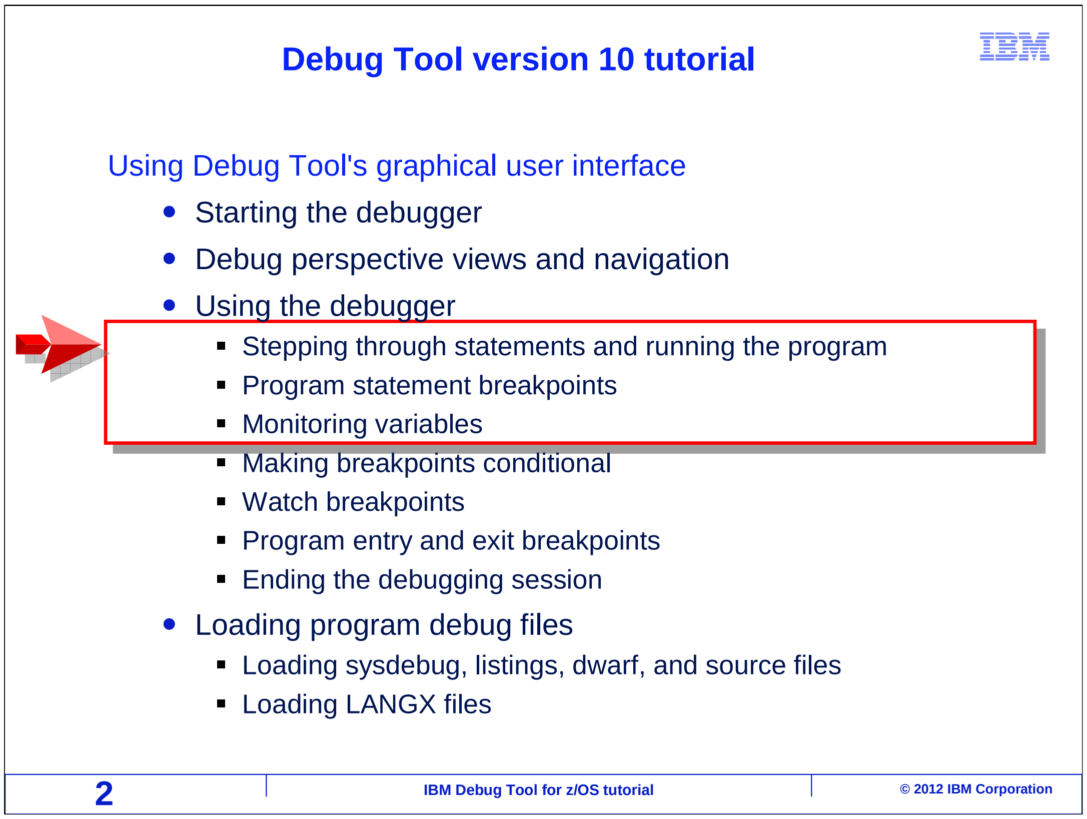


IBM Debug Tool for z/OS

Program number 5655-W70

Tutorial

This is the tutorial for IBM Debug Tool for z/OS®, one of the IBM zSeries® problem determination tools.



Debug Tool version 10 tutorial

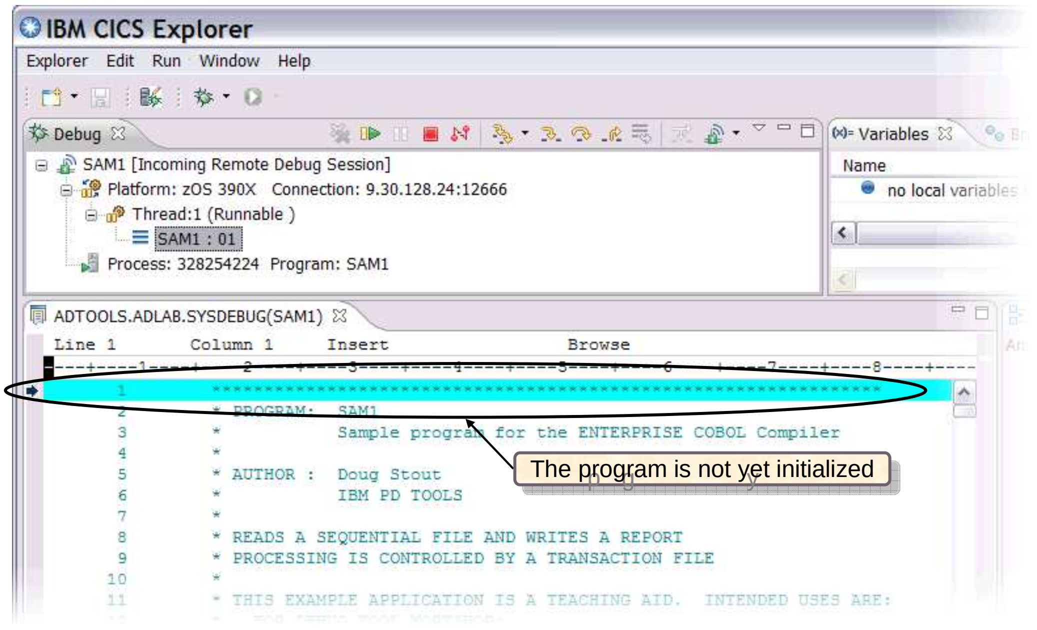
Using Debug Tool's graphical user interface

- Starting the debugger
- Debug perspective views and navigation
- Using the debugger
  - Stepping through statements and running the program
  - Program statement breakpoints
  - Monitoring variables
  - Making breakpoints conditional
  - Watch breakpoints
  - Program entry and exit breakpoints
  - Ending the debugging session
- Loading program debug files
  - Loading sysdebug, listings, dwarf, and source files
  - Loading LANGX files

2 | IBM Debug Tool for z/OS tutorial | © 2012 IBM Corporation

In this section, you will see how to step through a program, set and run to breakpoints at program statements, and how to display and monitor variables.

The debugger starts at the top of the program

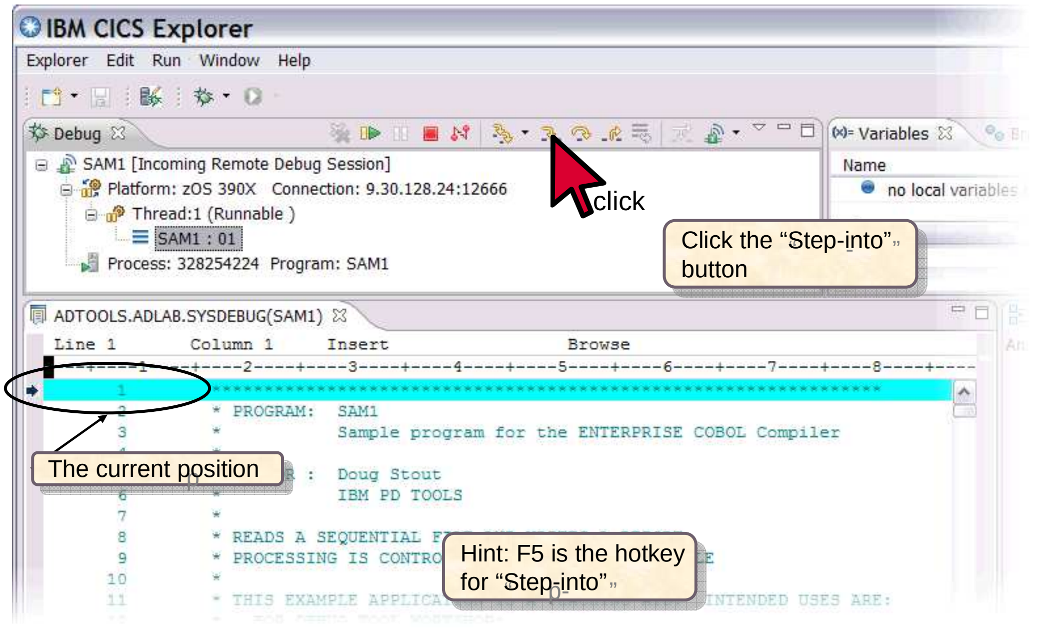

IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables
Name
no local variables
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
The program is not yet initialized

```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Line 1        Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
      ******************************************************************
      * PROGRAM:   SAM1
      *            Sample program for the ENTERPRISE COBOL Compiler
      *
      * AUTHOR :   Doug Stout
      *            IBM PD TOOLS
      *
      * READS A SEQUENTIAL FILE AND WRITES A REPORT
      * PROCESSING IS CONTROLLED BY A TRANSACTION FILE
      *
      * THIS EXAMPLE APPLICATION IS A TEACHING AID.  INTENDED USES ARE:
```

When a debug session starts, it is paused before the program has initialized.

Use the “Step-into” button to run one statement at a time

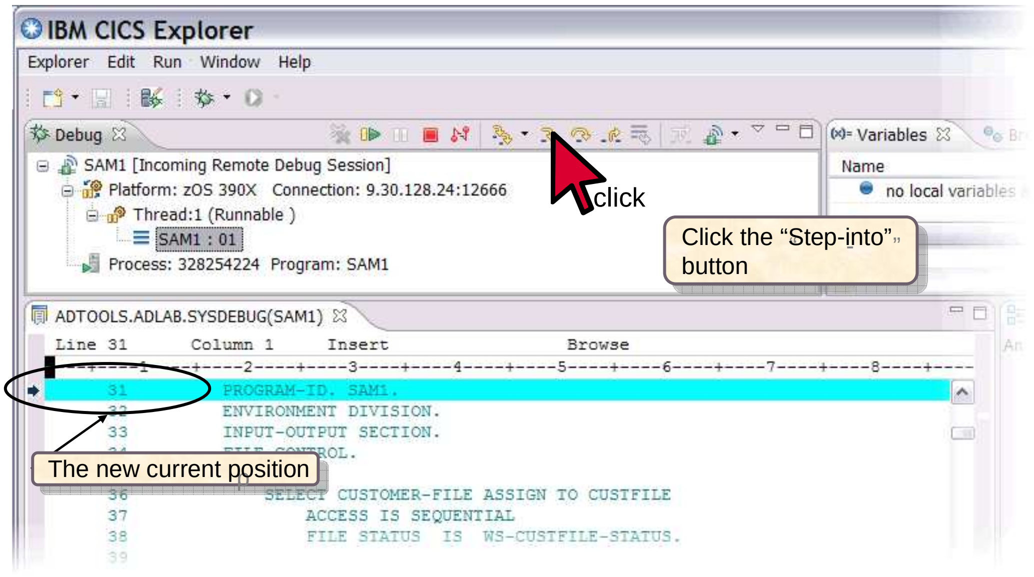

IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables
Name
no local variables
click
Click the “Step-into” button
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
The current position
Hint: F5 is the hotkey for “Step-into”

```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Line 1        Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
     1       *****************************************************************
     2        * PROGRAM:   SAM1
     3        *            Sample program for the ENTERPRISE COBOL Compiler
              R :          Doug Stout
     6        *            IBM PD TOOLS
     7        *
     8        * READS A SEQUENTIAL F
     9        * PROCESSING IS CONTRO                                    E
    10        *
    11        * THIS EXAMPLE APPLICA                      INTENDED USES ARE:
```

The current position is indicated by the highlighted source line. To begin the program, click the “resume” button to run the program or the “Step” button to run just one statement. In this example the step button is clicked to take one step.

“Step-into” again

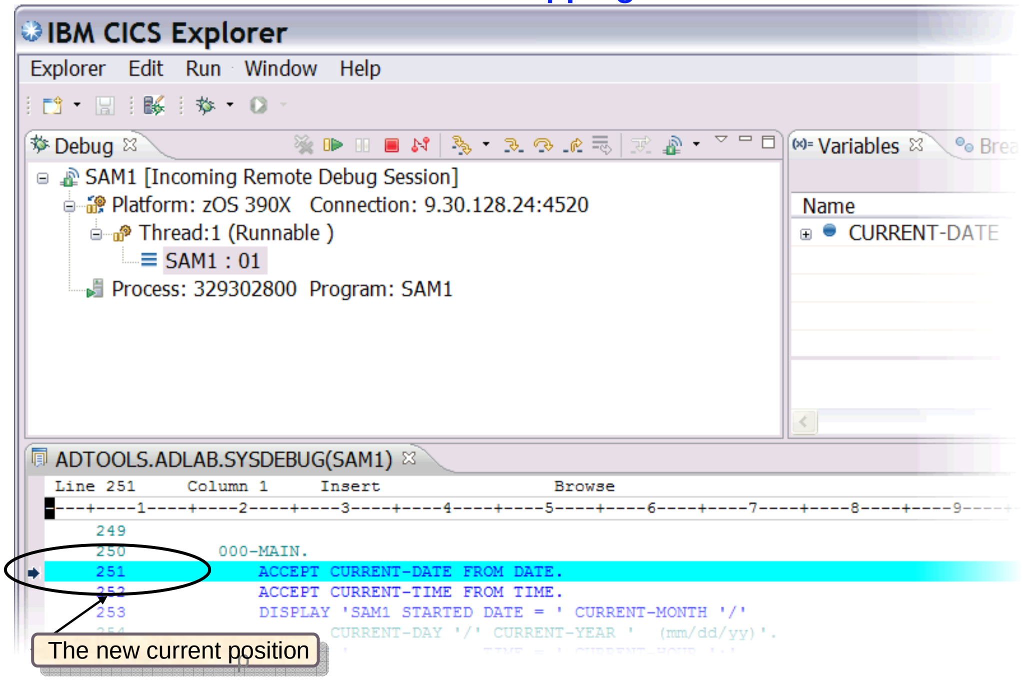

IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables
Name
no local variables
click
Click the “Step-into” button
The new current position
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Line 31       Column 1      Insert                     Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
          PROGRAM-ID. SAM1.
          ENVIRONMENT DIVISION.
          INPUT-OUTPUT SECTION.
          FILE-CONTROL.
              SELECT CUSTOMER-FILE ASSIGN TO CUSTFILE
                  ACCESS IS SEQUENTIAL
                  FILE STATUS  IS  WS-CUSTFILE-STATUS.
```

The program initializes, and the current line indicator moves to the next statement. The "step" button is clicked again.

After stepping

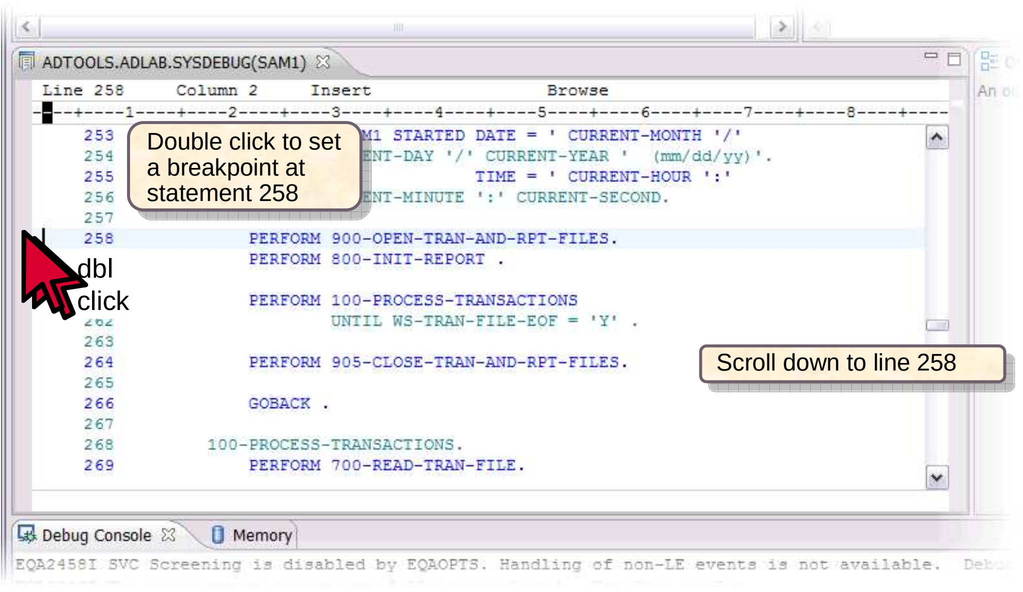


```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:4520
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 329302800  Program: SAM1
```

```
Line 251      Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----
   249
   250          000-MAIN.
   251              ACCEPT CURRENT-DATE FROM DATE.
   252              ACCEPT CURRENT-TIME FROM TIME.
   253              DISPLAY 'SAM1 STARTED DATE = ' CURRENT-MONTH '/'
   254                       CURRENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
```

You can continue to step through the program a statement at a time.

Set a breakpoint in the source view

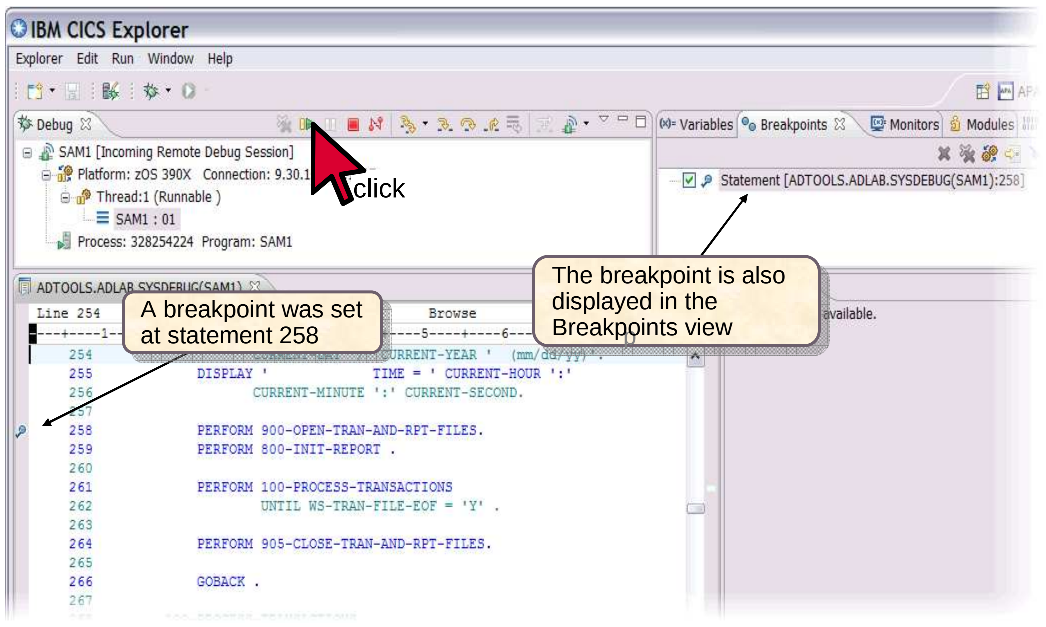


```
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
 Line 258      Column 2      Insert                    Browse
 ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
     253                              M1 STARTED DATE = ' CURRENT-MONTH '/'
     254                              ENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
     255                                         TIME = ' CURRENT-HOUR ':'
     256                              ENT-MINUTE ':' CURRENT-SECOND.
     257
     258            PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
                    PERFORM 800-INIT-REPORT .

                    PERFORM 100-PROCESS-TRANSACTIONS
     262                    UNTIL WS-TRAN-FILE-EOF = 'Y' .
     263
     264            PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.
     265
     266            GOBACK .
     267
     268        100-PROCESS-TRANSACTIONS.
     269            PERFORM 700-READ-TRAN-FILE.
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.
```

You can set breakpoints that make the program stop when it gets to specific statements. One way is to double click in the gray area to the left of a source statement. In this example the gray area next to line 258 is double-clicked.

Click the "Resume" button to run the program


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables Breakpoints Monitors Modules
click
A breakpoint was set at statement 258
The breakpoint is also displayed in the Breakpoints view
available.
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
Line 254
Browse

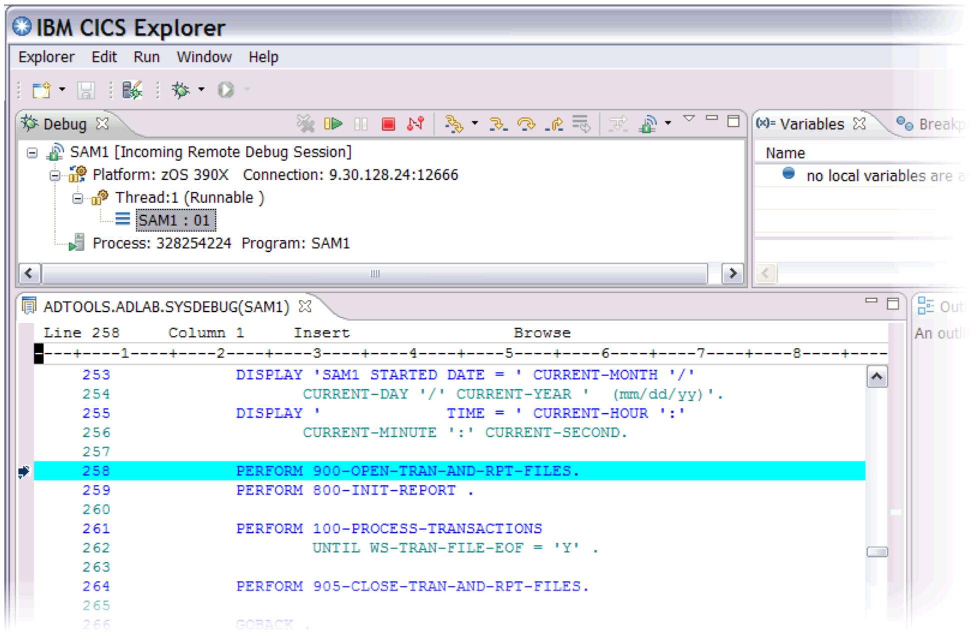
```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X  Connection: 9.30.1
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]
```

```
                     CURRENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
           DISPLAY '             TIME = ' CURRENT-HOUR ':'
                     CURRENT-MINUTE ':' CURRENT-SECOND.

           PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
           PERFORM 800-INIT-REPORT .

           PERFORM 100-PROCESS-TRANSACTIONS
                    UNTIL WS-TRAN-FILE-EOF = 'Y' .

           PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.

           GOBACK .
```

That set a breakpoint. Notice the breakpoint indicator icon next to line 258. With a breakpoint set, clicking the resume button will run the program until the breakpoint is reached. Or if the statement is never reached, it will continue to run to the end of the program.

Result of clicking the “Resume” button
The program ran until it reached a breakpoint



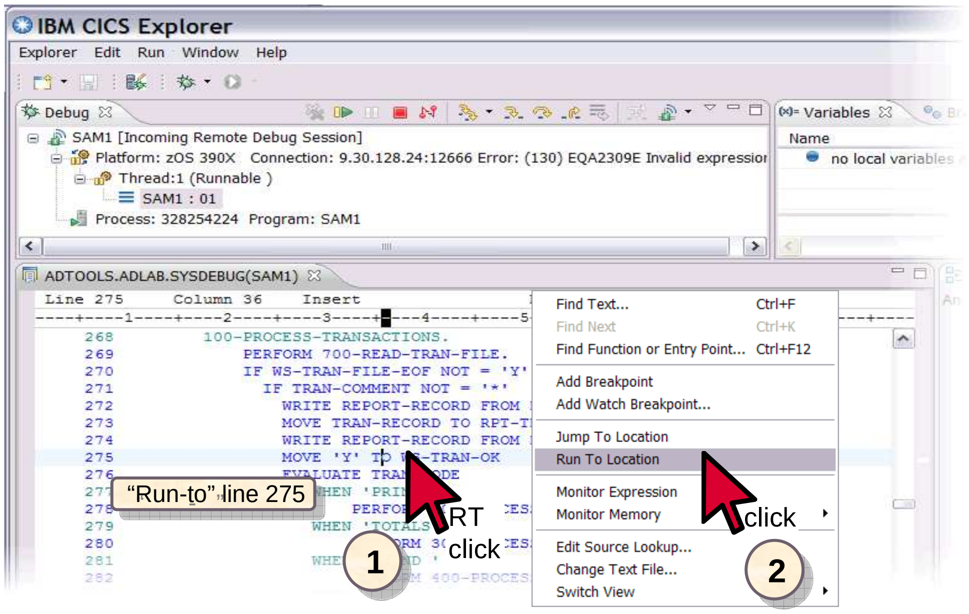
```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 258     Column 1      Insert                     Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
    253            DISPLAY 'SAM1 STARTED DATE = ' CURRENT-MONTH '/'
    254                   CURRENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
    255            DISPLAY '              TIME = ' CURRENT-HOUR ':'
    256                   CURRENT-MINUTE ':' CURRENT-SECOND.
    257
    258            PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
    259            PERFORM 800-INIT-REPORT .
    260
    261            PERFORM 100-PROCESS-TRANSACTIONS
    262                    UNTIL WS-TRAN-FILE-EOF = 'Y' .
    263
    264            PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.
    265
    266            GOBACK .
```

The program ran each statement until it reached the breakpoint. The program is suspended before running line 258. This statement will be the first to run the next time you click “Resume” or “Step”.

Right click on a line and select “Run To Location” to run the program until it reaches the statement

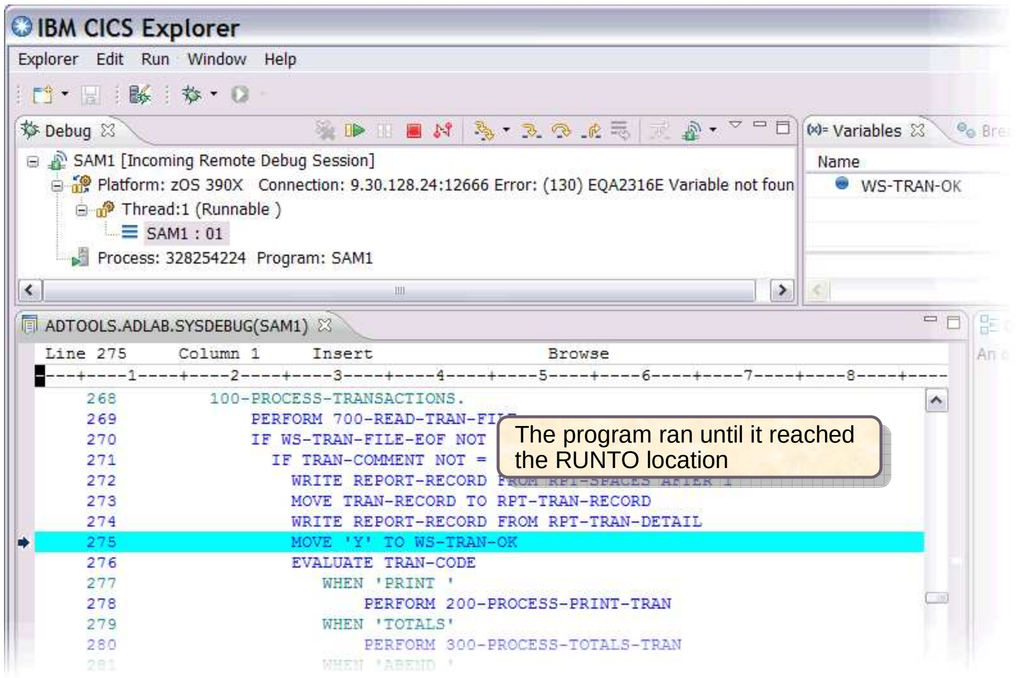


```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666 Error: (130) EQA2309E Invalid expressior
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
 Line 275      Column 36     Insert
----+----1----+----2----+----3----+----4----+----5
      268            100-PROCESS-TRANSACTIONS.
      269                PERFORM 700-READ-TRAN-FILE.
      270                IF WS-TRAN-FILE-EOF NOT = 'Y'
      271                  IF TRAN-COMMENT NOT = '*'
      272                    WRITE REPORT-RECORD FROM
      273                    MOVE TRAN-RECORD TO RPT-T
      274                    WRITE REPORT-RECORD FROM
      275                    MOVE 'Y' TO WS-TRAN-OK
      276                    EVALUATE TRAN-CODE
      277                      WHEN 'PRI
      278                        PERFO
      279                      WHEN 'TOTALS
      280                        PERFORM 3
      281                      WHEN 'END '
      282                        PERFORM 400-PROCES
```

That is one way to have the program run until it reaches a specific statement: set a breakpoint there and click resume. And there is another, even quicker way. Click once on a line to select it, in this example line 275, then right click the line and select “Run To Location”.

Result of the “Run To Location”



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12666 Error: (130) EQA2316E Variable not foun
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

WS-TRAN-OK

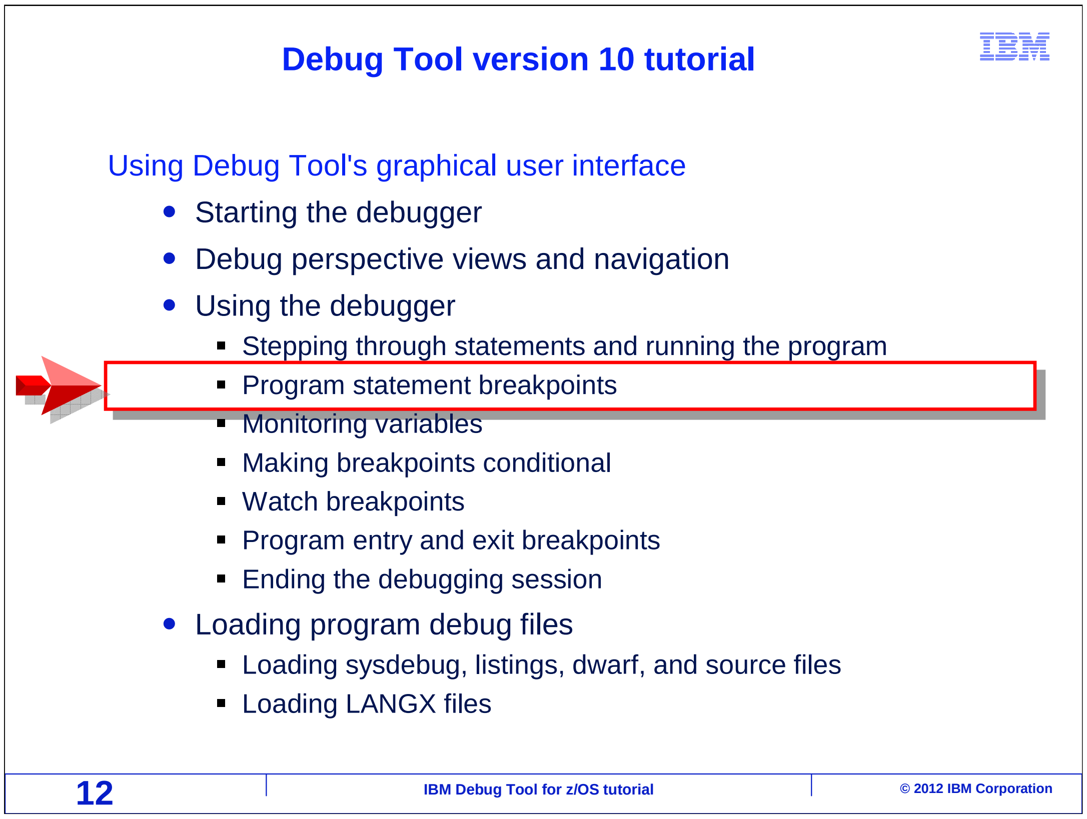
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
Line 275     Column 1     Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
 268          100-PROCESS-TRANSACTIONS.
 269              PERFORM 700-READ-TRAN-FI
 270              IF WS-TRAN-FILE-EOF NOT
 271                IF TRAN-COMMENT NOT =
 272                  WRITE REPORT-RECORD FROM RPT-SPACES AFTER 1
 273                  MOVE TRAN-RECORD TO RPT-TRAN-RECORD
 274                  WRITE REPORT-RECORD FROM RPT-TRAN-DETAIL
 275                  MOVE 'Y' TO WS-TRAN-OK
 276                  EVALUATE TRAN-CODE
 277                     WHEN 'PRINT '
 278                         PERFORM 200-PROCESS-PRINT-TRAN
 279                     WHEN 'TOTALS'
 280                         PERFORM 300-PROCESS-TOTALS-TRAN
 281                     WHEN 'ABEND '
```

The program ran until it reached the RUNTO location

11 | IBM Debug Tool for z/OS tutorial | © 2012 IBM Corporation

The program ran until it reached statement 275.



Debug Tool version 10 tutorial

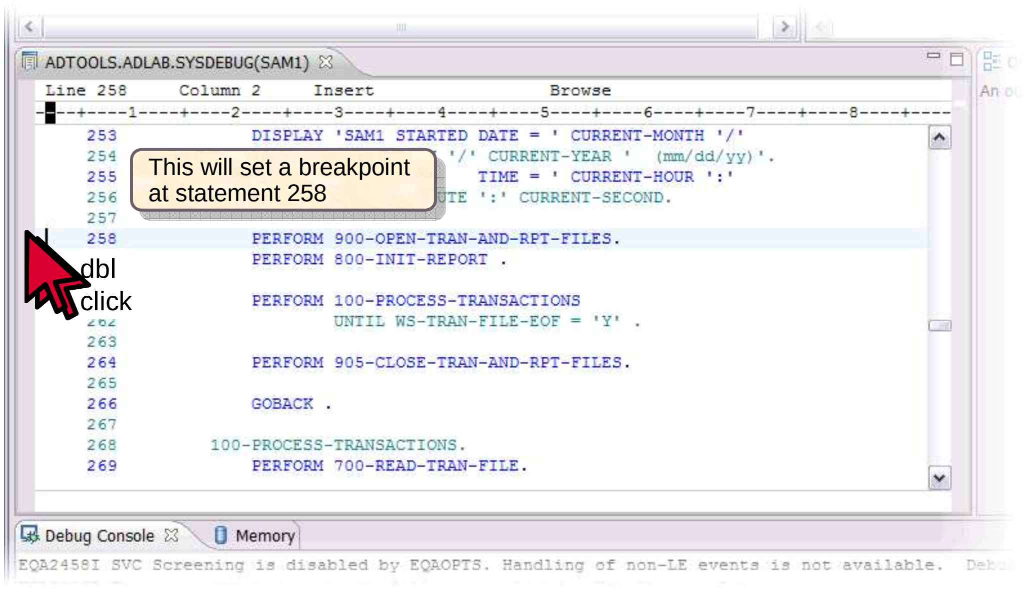
Using Debug Tool's graphical user interface

- Starting the debugger
- Debug perspective views and navigation
- Using the debugger
  - Stepping through statements and running the program
  - Program statement breakpoints
  - Monitoring variables
  - Making breakpoints conditional
  - Watch breakpoints
  - Program entry and exit breakpoints
  - Ending the debugging session
- Loading program debug files
  - Loading sysdebug, listings, dwarf, and source files
  - Loading LANGX files

12 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

Next, you will see different ways to set breakpoints at program statements.

Set a breakpoint in the source view

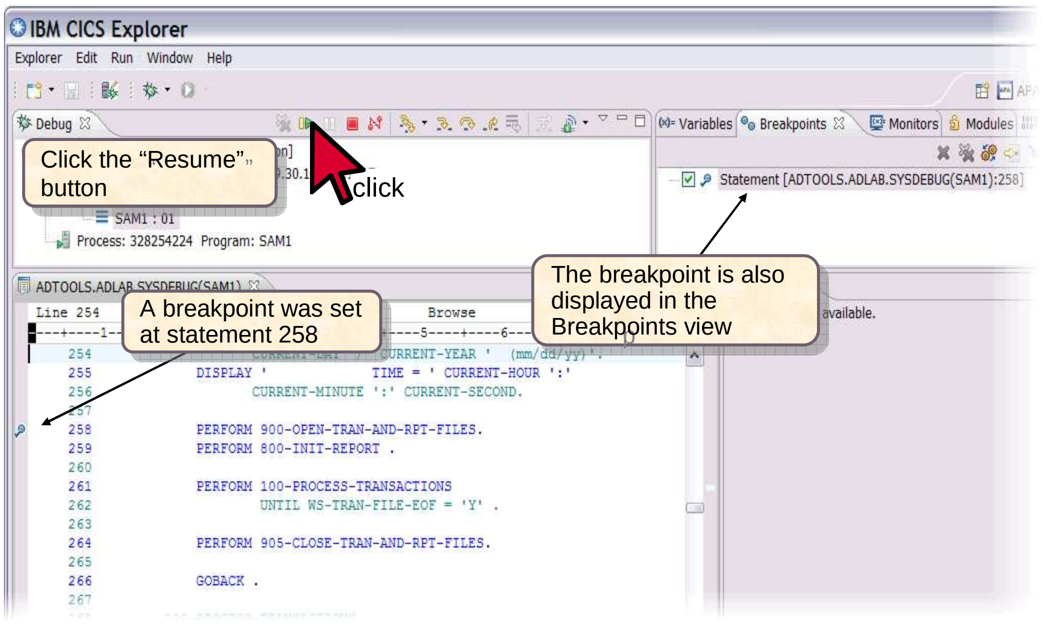

ADTOOLS.ADLAB.SYSDEBUG(SAM1)
This will set a breakpoint at statement 258
dbl click
Debug Console
Memory

```
Line 258      Column 2      Insert                       Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
   253             DISPLAY 'SAM1 STARTED DATE = ' CURRENT-MONTH '/'
   254                                   '/' CURRENT-YEAR '   (mm/dd/yy)'.
   255                                      TIME = ' CURRENT-HOUR ':'
   256                                 UTE ':' CURRENT-SECOND.
   257
   258             PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
                   PERFORM 800-INIT-REPORT .

                   PERFORM 100-PROCESS-TRANSACTIONS
   262                     UNTIL WS-TRAN-FILE-EOF = 'Y' .
   263
   264             PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.
   265
   266             GOBACK .
   267
   268         100-PROCESS-TRANSACTIONS.
   269             PERFORM 700-READ-TRAN-FILE.
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.
```

As you have already seen you can double-click in the gray area immediately to the left of a statement. This example shows setting a breakpoint on line 258.

Click the “Resume” button to run the program


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables Breakpoints Monitors Modules
Click the “Resume” button
click
A breakpoint was set at statement 258
The breakpoint is also displayed in the Breakpoints view
available.

SAM1 : 01
Process: 328254224 Program: SAM1

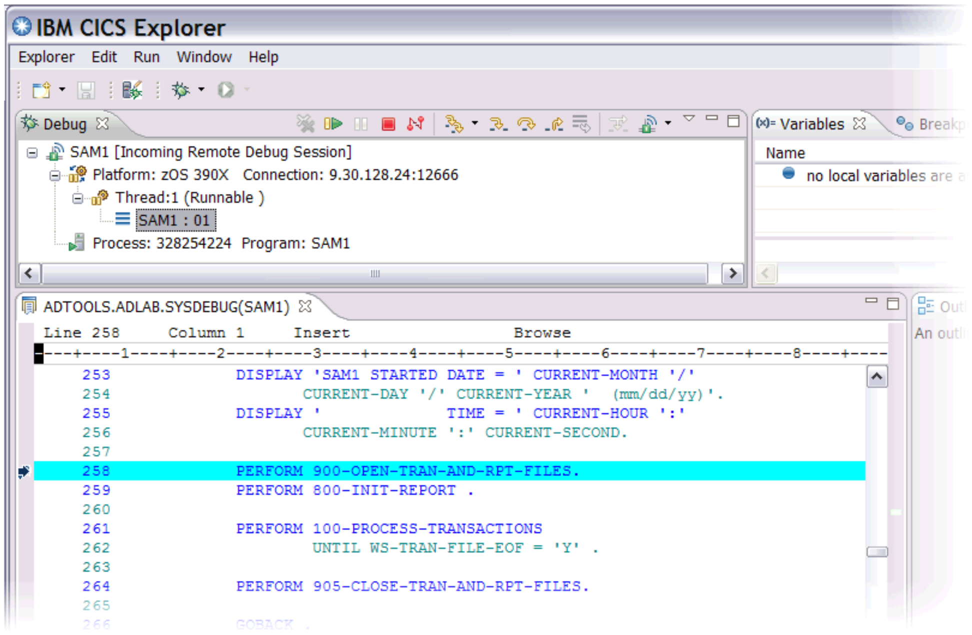
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]

ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
Line 254                                Browse
    254                CURRENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
    255          DISPLAY '            TIME = ' CURRENT-HOUR ':'
    256                 CURRENT-MINUTE ':' CURRENT-SECOND.
    257
    258          PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
    259          PERFORM 800-INIT-REPORT .
    260
    261          PERFORM 100-PROCESS-TRANSACTIONS
    262                  UNTIL WS-TRAN-FILE-EOF = 'Y' .
    263
    264          PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.
    265
    266          GOBACK .
    267
```

And that set a breakpoint. Notice that the breakpoint is displayed in the breakpoints view. The breakpoints view shows a list of breakpoints. Clicking “Resume” runs the program until the next breakpoint is reached.

Result of clicking the “Resume” button
The program ran until it reached a breakpoint



SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X   Connection: 9.30.128.24:12666
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224   Program: SAM1

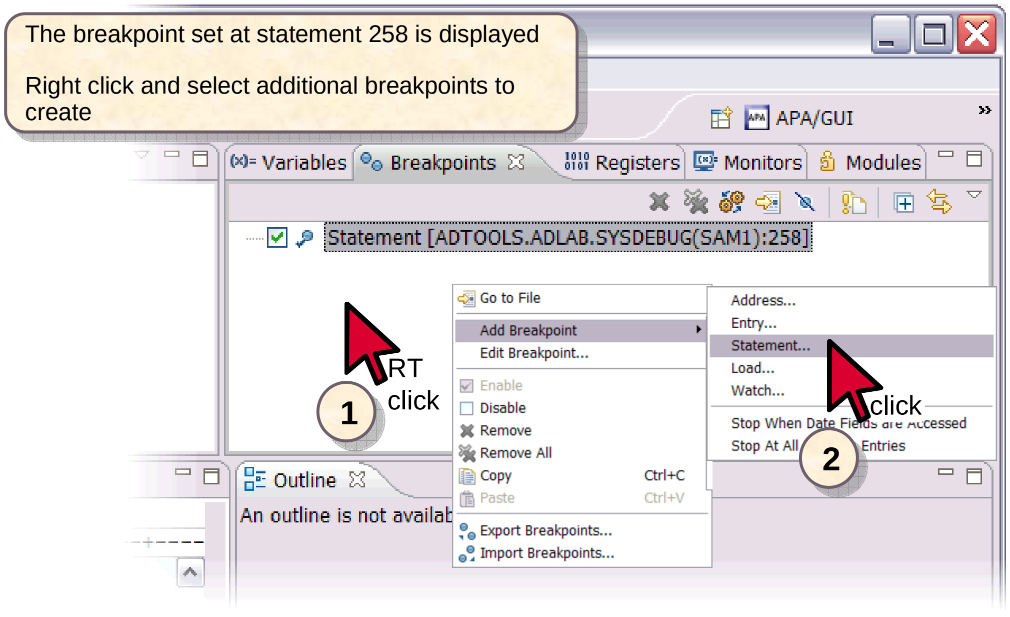
```
 Line 258     Column 1      Insert                     Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----
     253             DISPLAY 'SAM1 STARTED DATE = ' CURRENT-MONTH '/'
     254                    CURRENT-DAY '/' CURRENT-YEAR '   (mm/dd/yy)'.
     255             DISPLAY '                 TIME = ' CURRENT-HOUR ':'
     256                    CURRENT-MINUTE ':' CURRENT-SECOND.
     257
     258             PERFORM 900-OPEN-TRAN-AND-RPT-FILES.
     259             PERFORM 800-INIT-REPORT .
     260
     261             PERFORM 100-PROCESS-TRANSACTIONS
     262                     UNTIL WS-TRAN-FILE-EOF = 'Y' .
     263
     264             PERFORM 905-CLOSE-TRAN-AND-RPT-FILES.
     265
     266             GOBACK .
```

15 | IBM Debug Tool for z/OS tutorial | © 2012 IBM Corporation

The program ran until the breakpoint was triggered at line 258.

The breakpoints view pop-up menu gives options for setting breakpoints

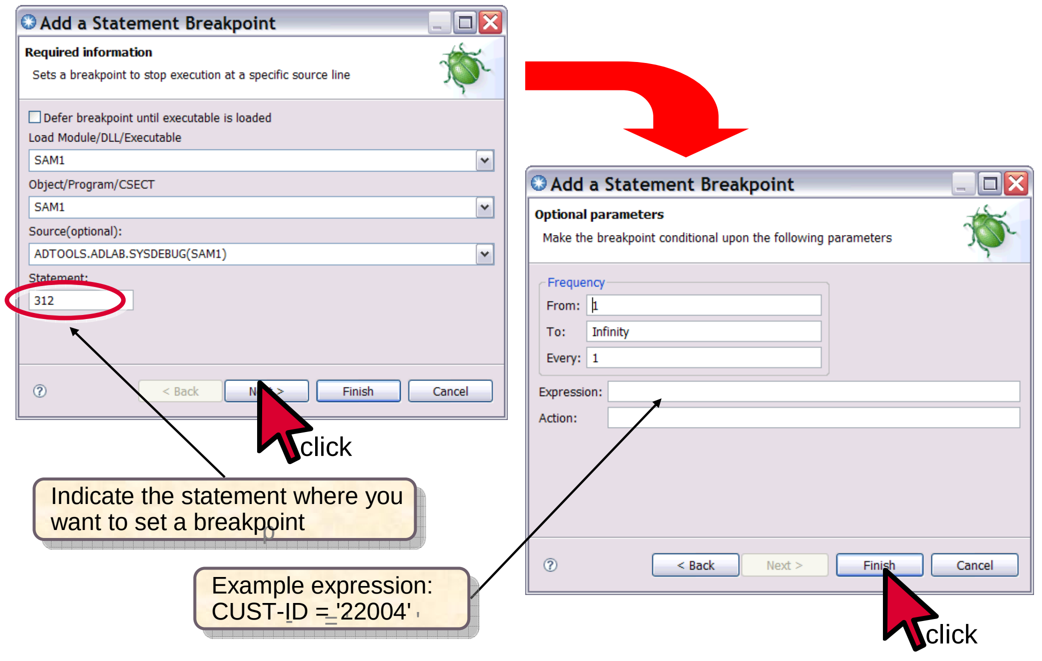
IBM


The breakpoint set at statement 258 is displayed

Right click and select additional breakpoints to create

You can also set a breakpoint from the breakpoints view. Right click in the white area of the breakpoints view and select “Add Breakpoint”. A menu is displayed with a list of available breakpoint types. Select “Statement”.

Specify *line-number* to set a statement breakpoint



click
Indicate the statement where you want to set a breakpoint
Example expression: CUST-ID = '22004'
click
< Back
Next >
Finish
Cancel

Add a Statement Breakpoint

Required information

Sets a breakpoint to stop execution at a specific source line

☐ Defer breakpoint until executable is loaded

Load Module/DLL/Executable: SAM1

Object/Program/CSECT: SAM1

Source(optional): ADTOOLS.ADLAB.SYSDEBUG(SAM1)

Statement: 312

Add a Statement Breakpoint

Optional parameters

Make the breakpoint conditional upon the following parameters

Frequency

From: 1

To: Infinity

Every: 1

Expression:

Action:

A dialog is displayed. On the first screen, the breakpoint information page, specify the statement number where you want the breakpoint to be set.

The other fields give you additional control and options. The “Defer breakpoint until executable is loaded” check box lets you define a breakpoint in a program that has not been loaded, such as subroutines that have not been called yet. Indicate the Load Module or DLL in the first box, and the program or CSECT in the second, and optionally the debug file. By default, the current program is assumed, so the load module and program names are already filled in.

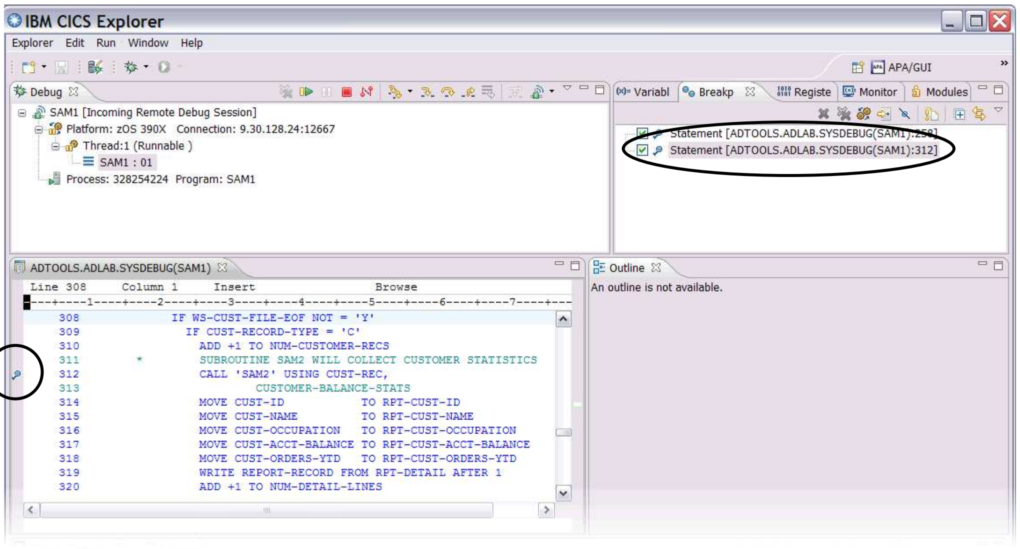
Click Next to continue to the “Optional parameters” page.

You can optionally set a frequency for the breakpoint. For example, if you enter "99" in the "from" field, the breakpoint will trigger the 99th time the statement runs, but not the first 98 times. This is useful to quickly run to a certain record in a file, prevent run-away situations, or debug loops.

You can also add an expression that must be true before the breakpoint will trigger, which lets you continue to run through the statement until some variable is equal to or exceeds a specified value.

Click “Finish”.

The breakpoint was set at the statement



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12667
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):312]
```

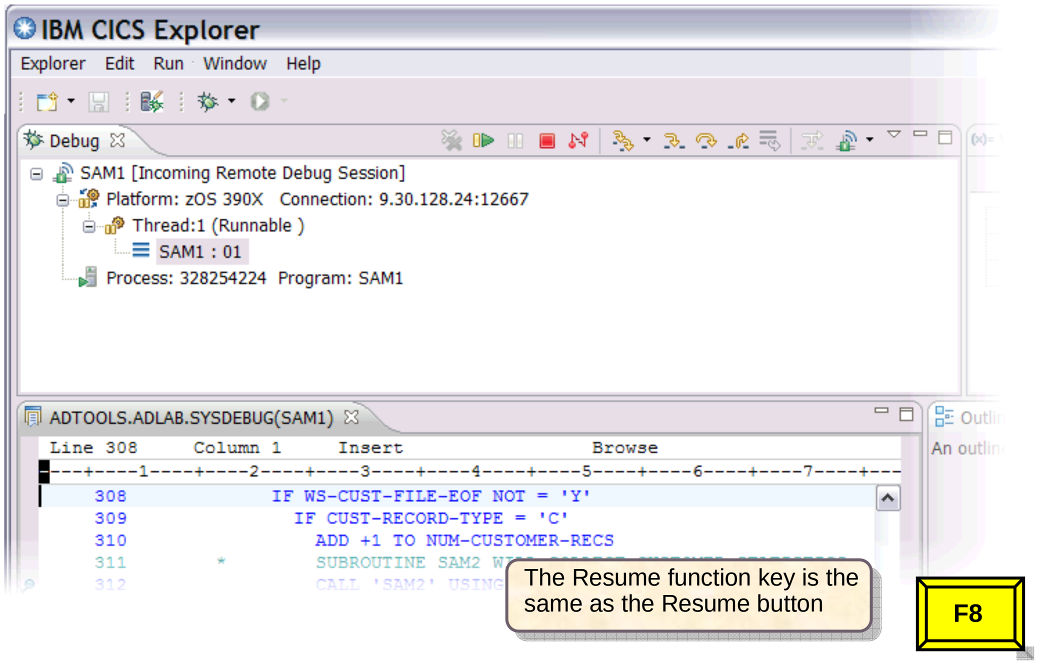
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 308      Column 1      Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
   308             IF WS-CUST-FILE-EOF NOT = 'Y'
   309               IF CUST-RECORD-TYPE = 'C'
   310                 ADD +1 TO NUM-CUSTOMER-RECS
   311        *        SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
   312                 CALL 'SAM2' USING CUST-REC,
   313                         CUSTOMER-BALANCE-STATS
   314                 MOVE CUST-ID             TO RPT-CUST-ID
   315                 MOVE CUST-NAME           TO RPT-CUST-NAME
   316                 MOVE CUST-OCCUPATION     TO RPT-CUST-OCCUPATION
   317                 MOVE CUST-ACCT-BALANCE   TO RPT-CUST-ACCT-BALANCE
   318                 MOVE CUST-ORDERS-YTD     TO RPT-CUST-ORDERS-YTD
   319                 WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
   320                 ADD +1 TO NUM-DETAIL-LINES
```

An outline is not available.

The new breakpoint is set at statement 312, and is shown in both the source and breakpoints views.

The Resume button or function key runs the program

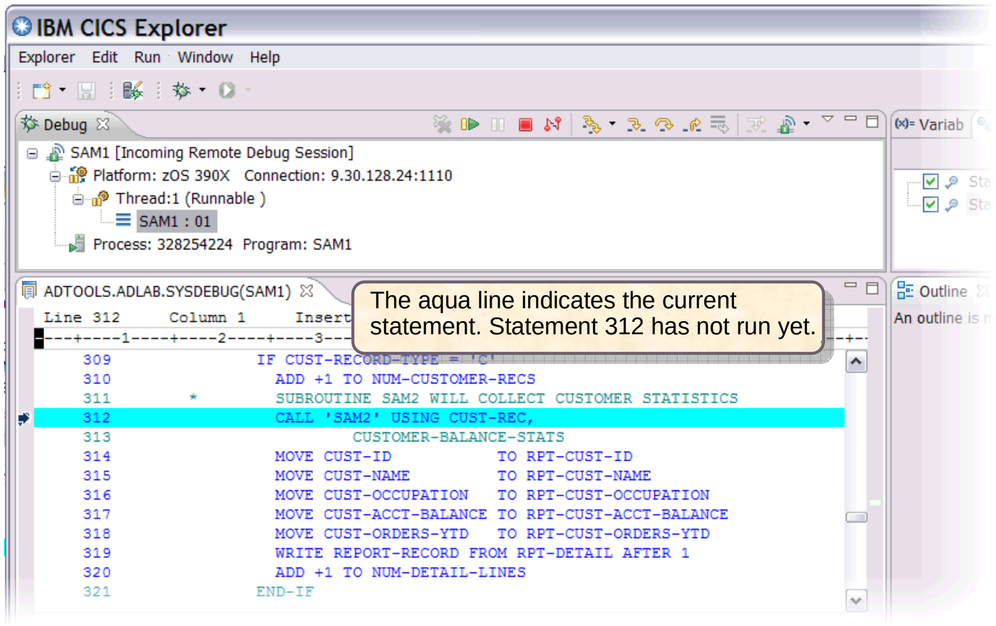

IBM CICS Explorer
Explorer Edit Run Window Help
Debug
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
Outline
An outline
The Resume function key is the same as the Resume button
F8

```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12667
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Line 308      Column 1       Insert                     Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---
  308              IF WS-CUST-FILE-EOF NOT = 'Y'
  309                IF CUST-RECORD-TYPE = 'C'
  310                  ADD +1 TO NUM-CUSTOMER-RECS
  311          *       SUBROUTINE SAM2 WI
  312                  CALL 'SAM2' USING
```

Now that the breakpoint is set, run the program. Another way to resume, other than the "resume" button, is to press the F8 key.

The program ran until it reached a breakpoint



SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:1110
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

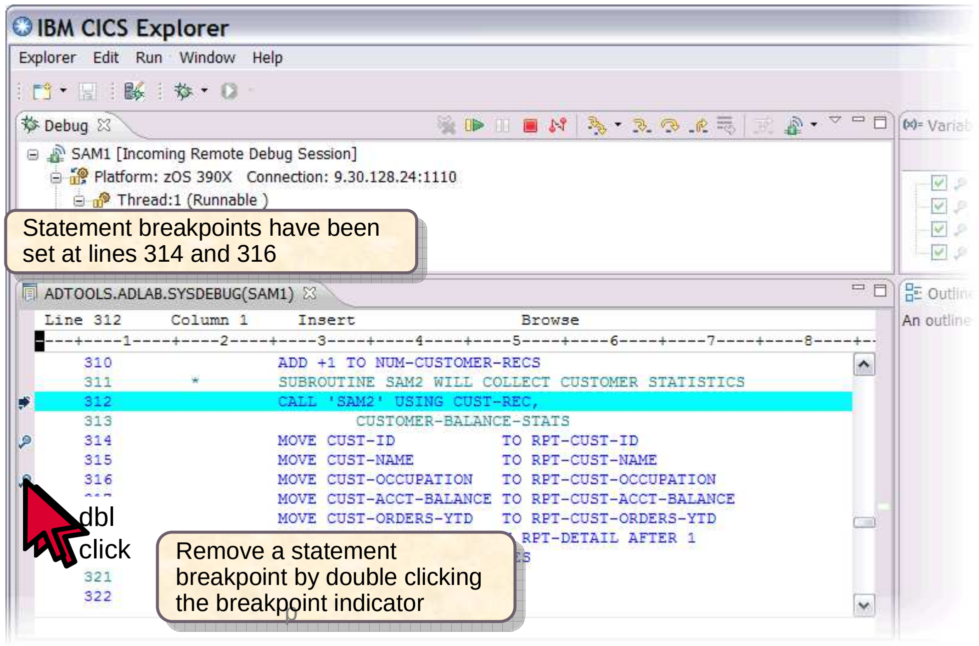
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

The aqua line indicates the current statement. Statement 312 has not run yet.

```
Line 312      Column 1      Insert
----+----1----+----2----+----3---
 309                 IF CUST-RECORD-TYPE = 'C'
 310                   ADD +1 TO NUM-CUSTOMER-RECS
 311          *        SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
 312                   CALL 'SAM2' USING CUST-REC,
 313                           CUSTOMER-BALANCE-STATS
 314                   MOVE CUST-ID           TO RPT-CUST-ID
 315                   MOVE CUST-NAME         TO RPT-CUST-NAME
 316                   MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
 317                   MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
 318                   MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
 319                   WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
 320                   ADD +1 TO NUM-DETAIL-LINES
 321                 END-IF
```

The program ran until it triggered the next breakpoint, at statement 312.

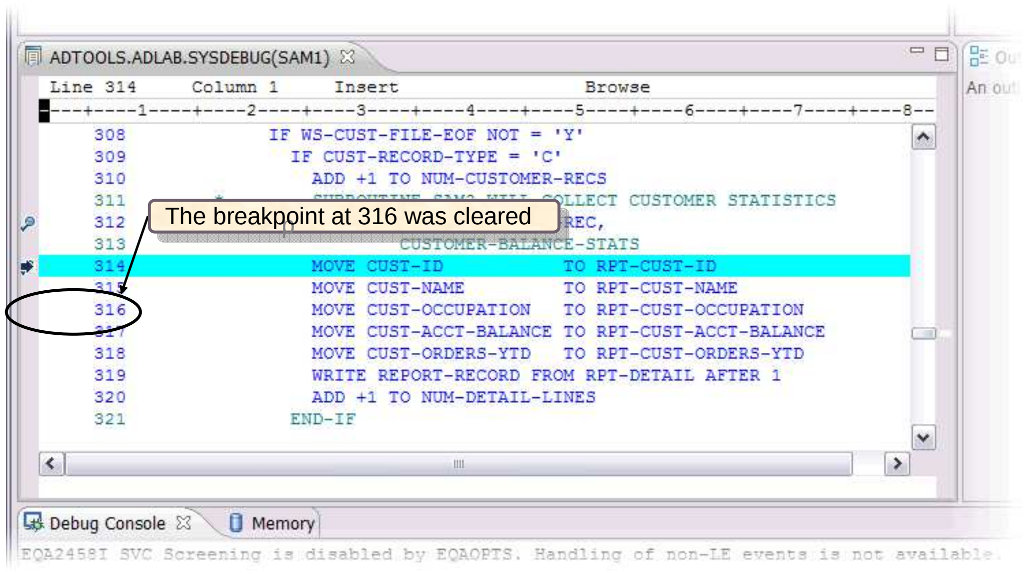
Double click a breakpoint marker to clear a statement breakpoint


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:1110
Thread:1 (Runnable)
Statement breakpoints have been set at lines 314 and 316
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
dbl click
Remove a statement breakpoint by double clicking the breakpoint indicator
Variab
Outline
An outline

```
 Line 312      Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+-
  310                ADD +1 TO NUM-CUSTOMER-RECS
  311        *       SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
  312                CALL 'SAM2' USING CUST-REC,
  313                        CUSTOMER-BALANCE-STATS
  314                MOVE CUST-ID            TO RPT-CUST-ID
  315                MOVE CUST-NAME          TO RPT-CUST-NAME
  316                MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
                     MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
                     MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
                                              RPT-DETAIL AFTER 1
  321
  322
```

You have seen that there are different ways to set statement breakpoints. There are also several ways to remove them. Here is one method, double click a breakpoint icon in the gray area next to a statement.

Result of double clicking a breakpoint marker

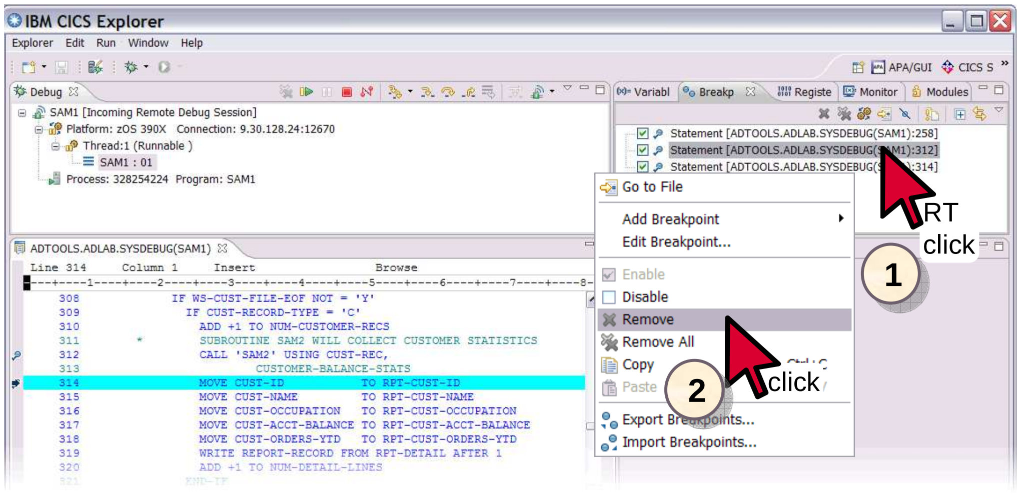

ADTOOLS.ADLAB.SYSDEBUG(SAM1)
The breakpoint at 316 was cleared
Debug Console
Memory

```
 Line 314      Column 1      Insert                   Browse
 ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8--
      308                IF WS-CUST-FILE-EOF NOT = 'Y'
      309                  IF CUST-RECORD-TYPE = 'C'
      310                    ADD +1 TO NUM-CUSTOMER-RECS
      311           *        SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
      312                                             REC,
      313                            CUSTOMER-BALANCE-STATS
      314                    MOVE CUST-ID              TO RPT-CUST-ID
      315                    MOVE CUST-NAME            TO RPT-CUST-NAME
      316                    MOVE CUST-OCCUPATION      TO RPT-CUST-OCCUPATION
      317                    MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
      318                    MOVE CUST-ORDERS-YTD      TO RPT-CUST-ORDERS-YTD
      319                    WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
      320                    ADD +1 TO NUM-DETAIL-LINES
      321                  END-IF
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.
```

That removed the breakpoint from line 316.

Right click and select "remove" to clear a statement breakpoint



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12670
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

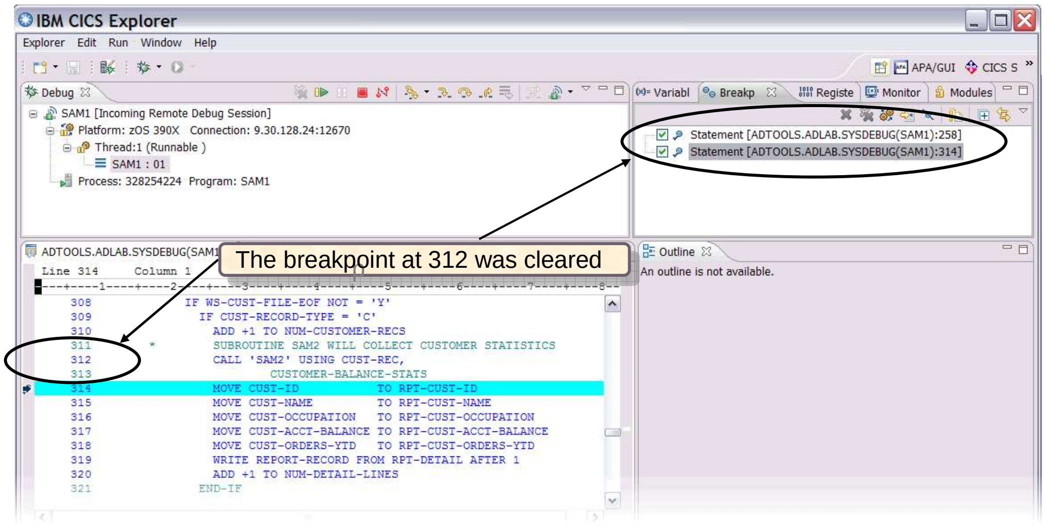
```
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):312]
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):314]
```

ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 314     Column 1     Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8-
   308              IF WS-CUST-FILE-EOF NOT = 'Y'
   309                IF CUST-RECORD-TYPE = 'C'
   310                  ADD +1 TO NUM-CUSTOMER-RECS
   311       *          SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
   312                  CALL 'SAM2' USING CUST-REC,
   313                          CUSTOMER-BALANCE-STATS
   314                  MOVE CUST-ID           TO RPT-CUST-ID
   315                  MOVE CUST-NAME         TO RPT-CUST-NAME
   316                  MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
   317                  MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
   318                  MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
   319                  WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
   320                  ADD +1 TO NUM-DETAIL-LINES
   321                END-IF
```

Another way to remove a breakpoint is to select the breakpoint to be removed in the breakpoints view, right click the breakpoint, and click "remove". Here that is done for the breakpoint at statement 312.

The statement breakpoint is cleared from the source and breakpoints views



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12670
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):314]
```

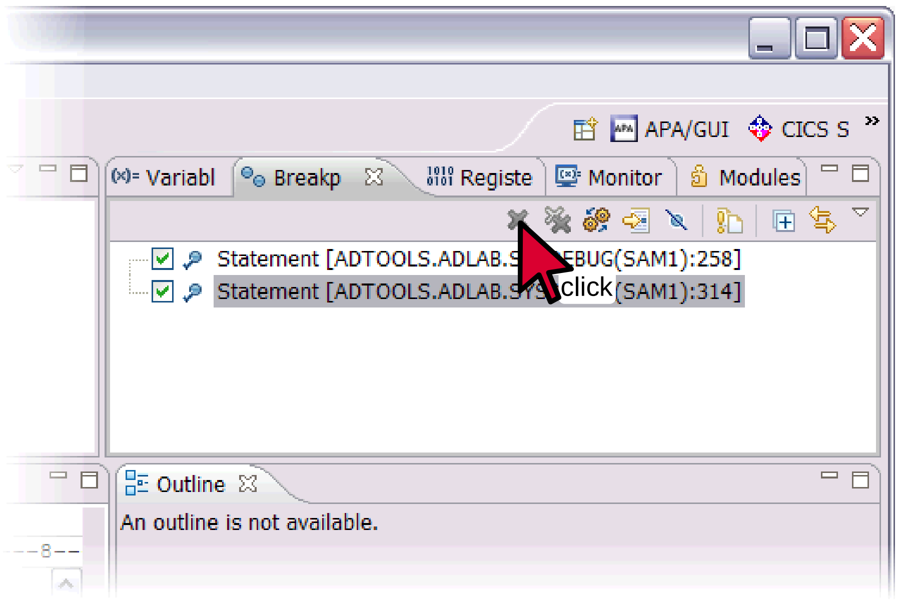
ADTOOLS.ADLAB.SYSDEBUG(SAM1

An outline is not available.

```
Line 314      Column 1
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8--
 308              IF WS-CUST-FILE-EOF NOT = 'Y'
 309                IF CUST-RECORD-TYPE = 'C'
 310                  ADD +1 TO NUM-CUSTOMER-RECS
 311         *        SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
 312                  CALL 'SAM2' USING CUST-REC,
 313                          CUSTOMER-BALANCE-STATS
 314                  MOVE CUST-ID             TO RPT-CUST-ID
 315                  MOVE CUST-NAME           TO RPT-CUST-NAME
 316                  MOVE CUST-OCCUPATION     TO RPT-CUST-OCCUPATION
 317                  MOVE CUST-ACCT-BALANCE   TO RPT-CUST-ACCT-BALANCE
 318                  MOVE CUST-ORDERS-YTD     TO RPT-CUST-ORDERS-YTD
 319                  WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
 320                  ADD +1 TO NUM-DETAIL-LINES
 321                END-IF
```

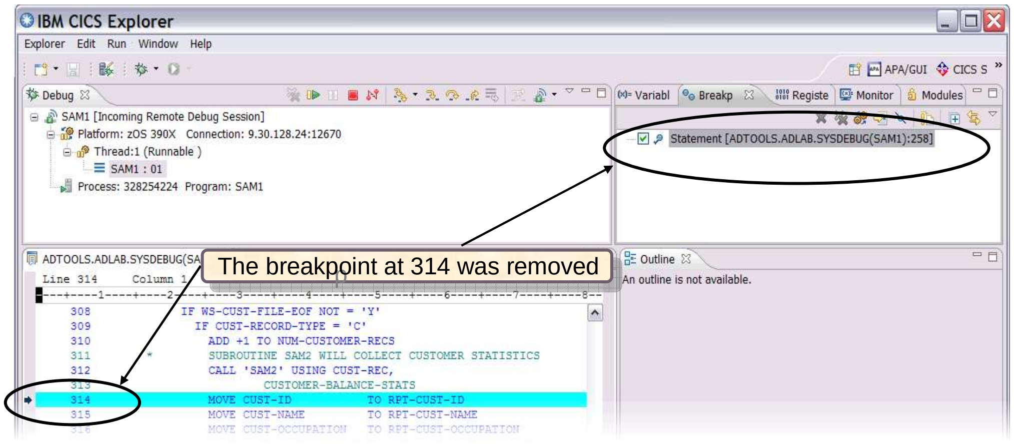
And the breakpoint is removed.

Select a breakpoint and click the “Remove” button


APA/GUI
CICS S
Variabl
Breakp
Registe
Monitor
Modules
Statement [ADTOOLS.ADLAB.S EBUG(SAM1):258]
Statement [ADTOOLS.ADLAB.SYS (SAM1):314]
click
Outline
An outline is not available.

Another way is to select a breakpoint in the breakpoints view, and then click the "X"-shaped “Remove” button. In this example the breakpoint at line 314 is selected, and remove is clicked.

Result of clicking the “Remove” button



SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:12670
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

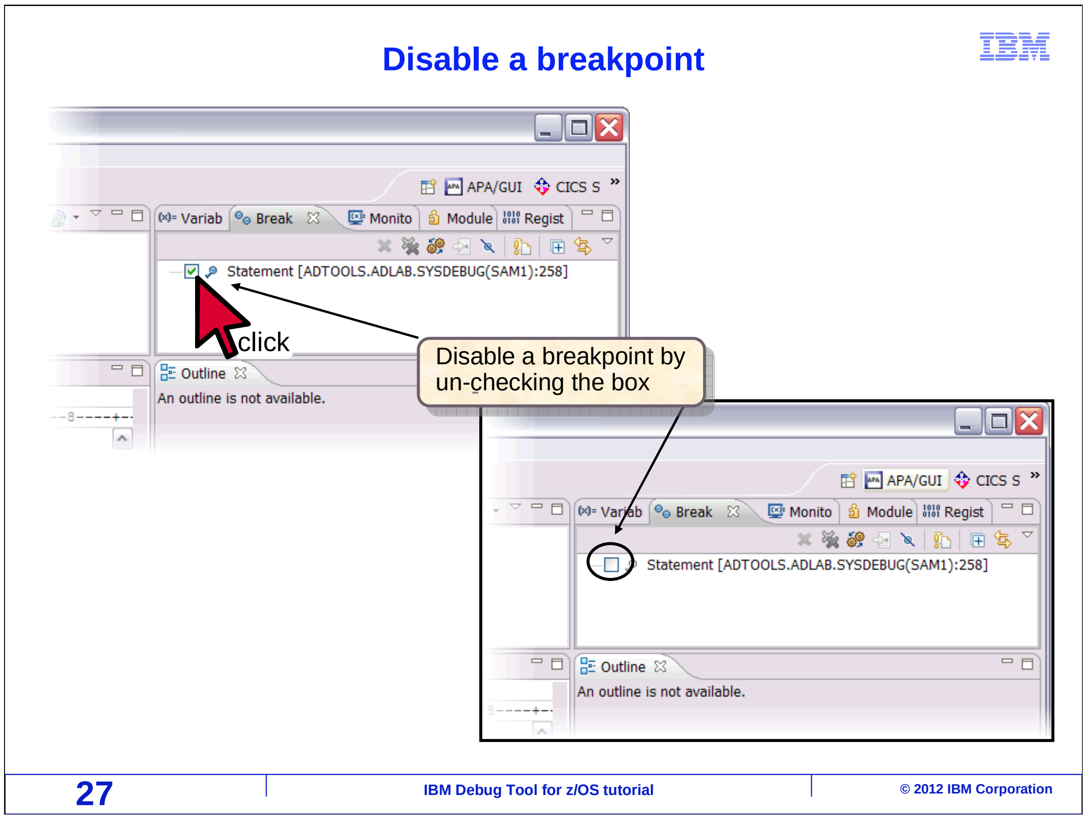
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]

An outline is not available.

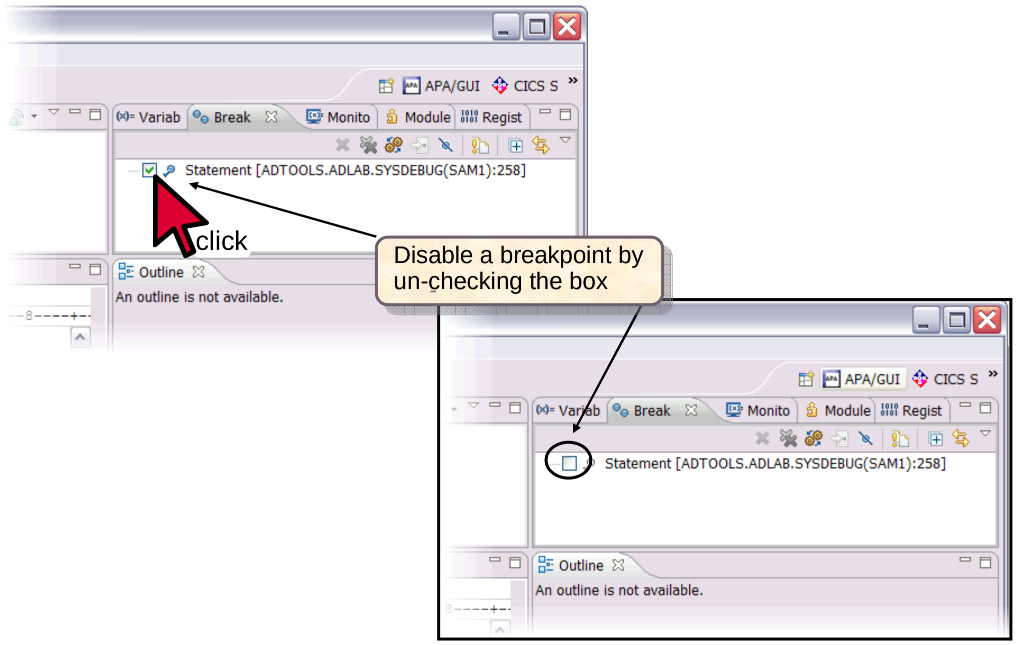
ADTOOLS.ADLAB.SYSDEBUG(SA

```
Line 314      Column 1
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8--
   308          IF WS-CUST-FILE-EOF NOT = 'Y'
   309            IF CUST-RECORD-TYPE = 'C'
   310              ADD +1 TO NUM-CUSTOMER-RECS
   311      *       SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
   312              CALL 'SAM2' USING CUST-REC,
   313                      CUSTOMER-BALANCE-STATS
   314              MOVE CUST-ID            TO RPT-CUST-ID
   315              MOVE CUST-NAME          TO RPT-CUST-NAME
   316              MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
```

Which removes the breakpoint.

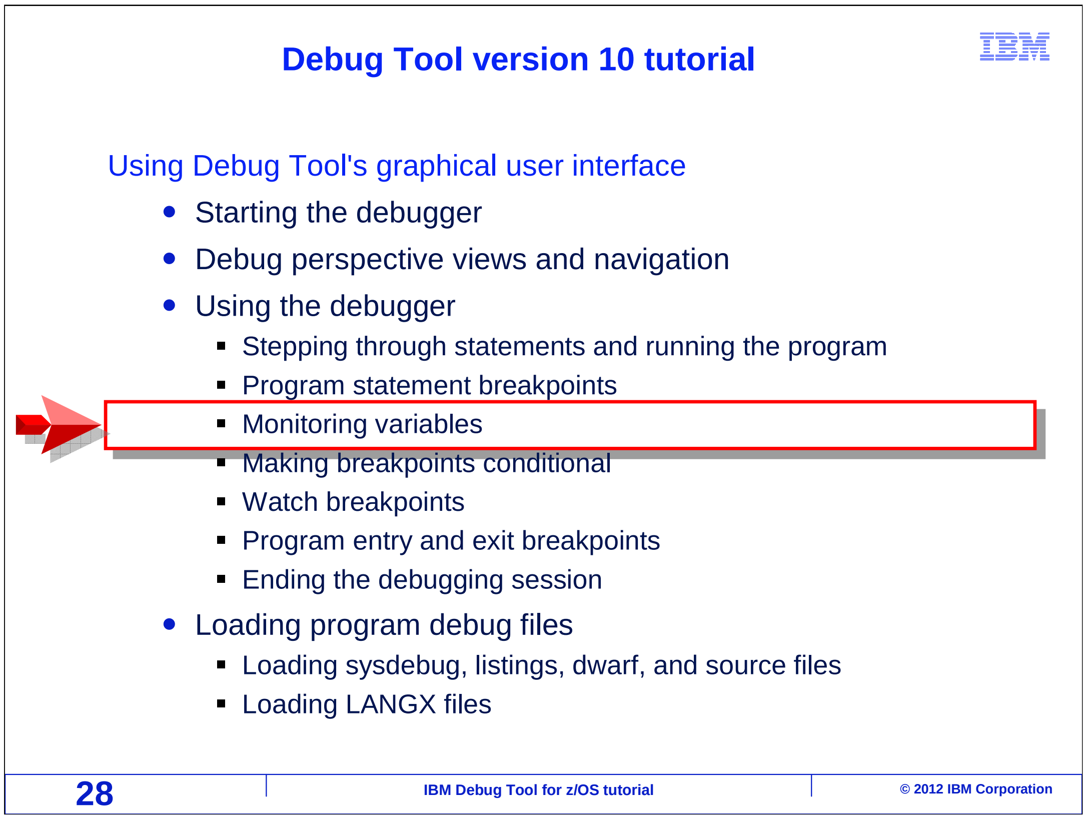


Disable a breakpoint


APA/GUI
CICS S
Variab
Break
Monito
Module
Regist
click
Disable a breakpoint by un-checking the box
Outline
An outline is not available.

Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):258]

Instead of removing a breakpoint, you can turn it off temporarily. Notice the check box to the left of breakpoints in the breakpoints view. A check mark in the box indicates the breakpoint is enabled; no checkmark means the breakpoint is disabled. To enable or disable, click the box to toggle the setting.



Debug Tool version 10 tutorial

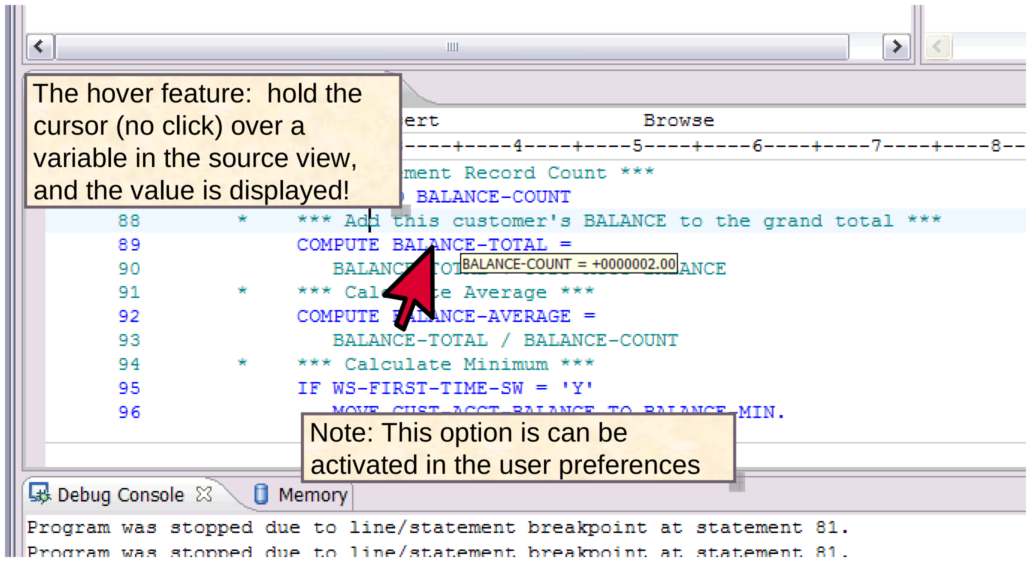
Using Debug Tool's graphical user interface

- Starting the debugger
- Debug perspective views and navigation
- Using the debugger
  - Stepping through statements and running the program
  - Program statement breakpoints
  - Monitoring variables
  - Making breakpoints conditional
  - Watch breakpoints
  - Program entry and exit breakpoints
  - Ending the debugging session
- Loading program debug files
  - Loading sysdebug, listings, dwarf, and source files
  - Loading LANGX files

28 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

Next, you will see how to view and update the values of program variables.

Hover over a variable to see it's value



The hover feature: hold the cursor (no click) over a variable in the source view, and the value is displayed!

Note: This option is can be activated in the user preferences

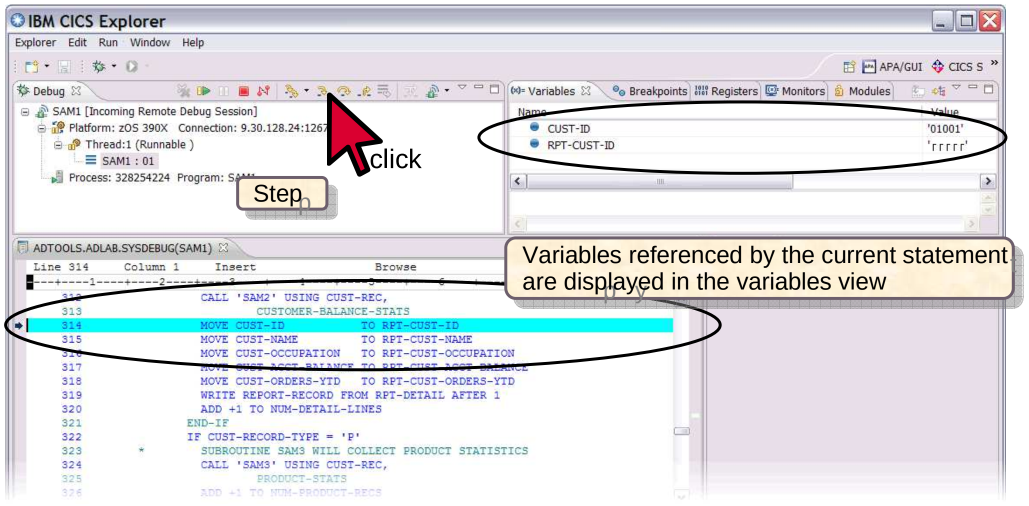
```
----+----4----+----5----+----6----+----7----+----8--
ment Record Count ***
 BALANCE-COUNT
      *    *** Add this customer's BALANCE to the grand total ***
           COMPUTE BALANCE-TOTAL =
              BALANCE-TOTAL                    ANCE
      *    *** Calculate Average ***
           COMPUTE BALANCE-AVERAGE =
              BALANCE-TOTAL / BALANCE-COUNT
      *    *** Calculate Minimum ***
           IF WS-FIRST-TIME-SW = 'Y'
              MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
```

```
BALANCE-COUNT = +0000002.00
```

```
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
```

One way to see the value of a variable is by “hovering”. First select the source view by clicking anywhere in the source view. Then place your mouse cursor over a variable, but do not click. The variable value pops up automatically. That's an easy way to see a value.

The Variables view shows program variables, and optionally variables referenced by the current statement

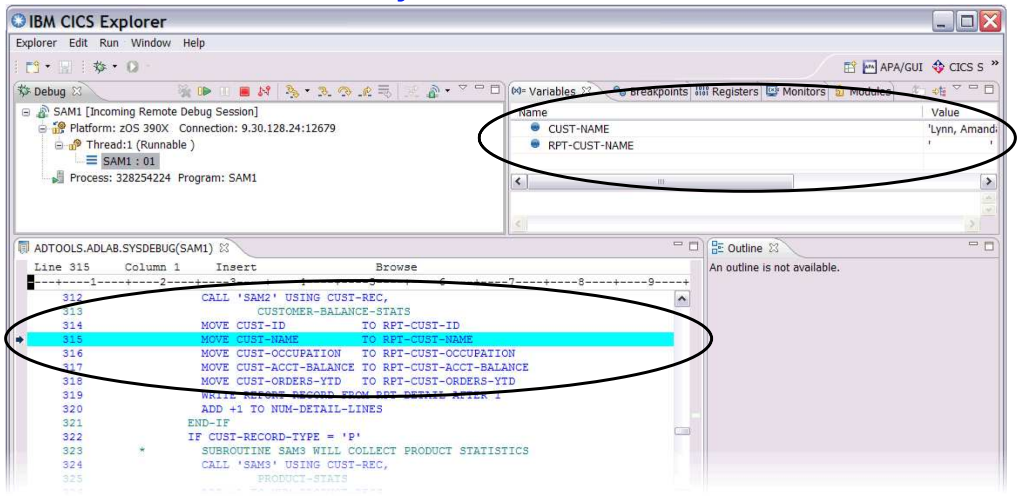

IBM CICS Explorer
Explorer Edit Run Window Help
APA/GUI CICS S
Debug
Variables Breakpoints Registers Monitors Modules
Name Value
SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:1267
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1
Step
click
ADTOOLS.ADLAB.SYSDEBUG(SAM1)
Variables referenced by the current statement are displayed in the variables view

Name	Value
CUST-ID	'01001'
RPT-CUST-ID	'rrrrr'

```
 Line 314      Column 1      Insert                  Browse
 ----+----1----+----2----+----3----+----4----+----5----+----6----+--
   312                 CALL 'SAM2' USING CUST-REC,
   313                         CUSTOMER-BALANCE-STATS
   314                 MOVE CUST-ID           TO RPT-CUST-ID
   315                 MOVE CUST-NAME         TO RPT-CUST-NAME
   316                 MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
   317                 MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
   318                 MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
   319                 WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
   320                 ADD +1 TO NUM-DETAIL-LINES
   321               END-IF
   322               IF CUST-RECORD-TYPE = 'P'
   323        *        SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
   324                 CALL 'SAM3' USING CUST-REC,
   325                         PRODUCT-STATS
   326                 ADD +1 TO NUM-PRODUCT-RECS
```

The variables view provides another easy way to see variables. You can control the scope of the variables shown in this view by changing the "filter" setting. The filter can be set to show the current statement's variables, all variables, or COBOL working-storage, linkage, or file section. By default this view automatically monitors variables referenced by the current statement. Notice in this example, line 314 is the current statement, and it references two variables: cust-id and rpt-cust-id. Those variables are displayed in the variables view automatically. The step button is clicked.

After a Step. The variables view displays variables referenced by the current statement



SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:12679
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

Name	Value
CUST-NAME	'Lynn, Amand
RPT-CUST-NAME	' '

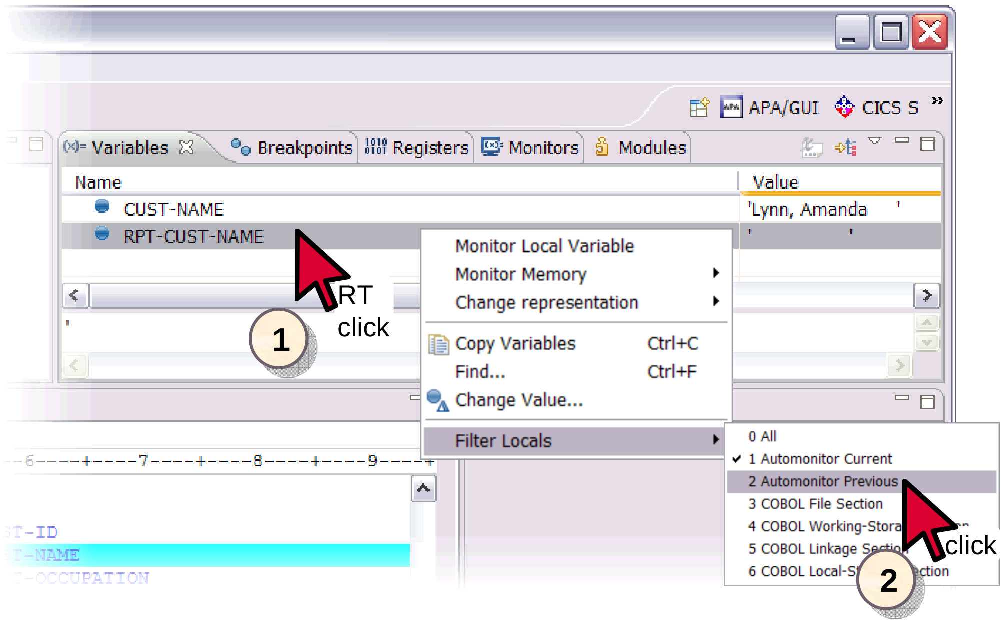
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 315      Column 1      Insert                   Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+
   312                CALL 'SAM2' USING CUST-REC,
   313                        CUSTOMER-BALANCE-STATS
   314                MOVE CUST-ID            TO RPT-CUST-ID
   315                MOVE CUST-NAME          TO RPT-CUST-NAME
   316                MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
   317                MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
   318                MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
   319                WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
   320                ADD +1 TO NUM-DETAIL-LINES
   321              END-IF
   322              IF CUST-RECORD-TYPE = 'P'
   323        *       SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
   324                CALL 'SAM3' USING CUST-REC,
   325                        PRODUCT-STATS
```

An outline is not available.

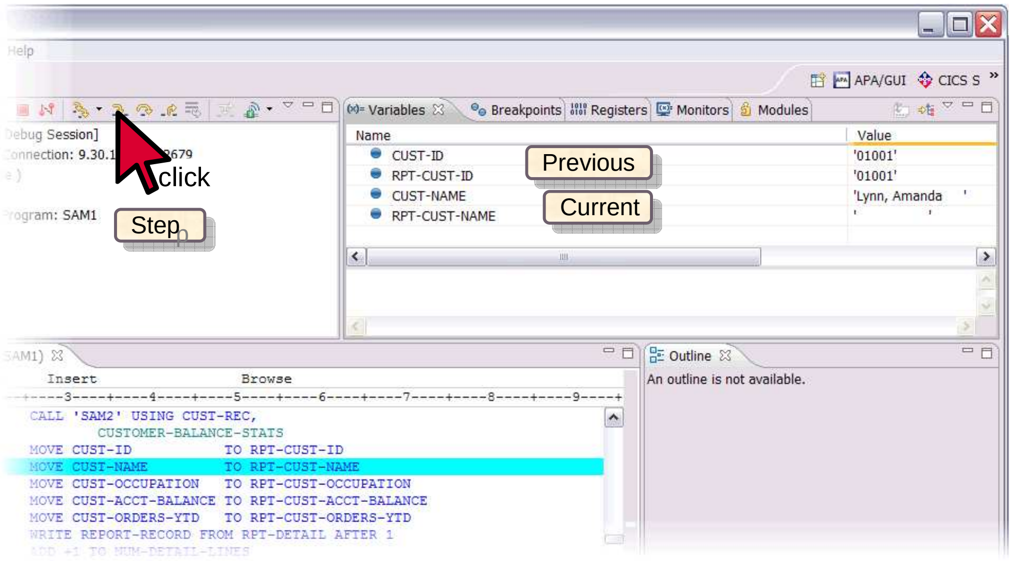
Now the next statement, line 315 is current, and it references different variables: cust-name and rpt-cust-name. Now these variables are shown instead. As you run or step through a program, the current variables are displayed.

Select “Automonitor Previous” to display variables from both the current and previous statements



To change the scope of the variables view, right click in the view and select “Filter Locals”. A menu shows the filtering options. "Automonitor current" is the default setting, and you just saw what that setting displays. The filter is changed to “Automonitor Previous”.

Automonitor Previous displays variables from both the current and previous statements

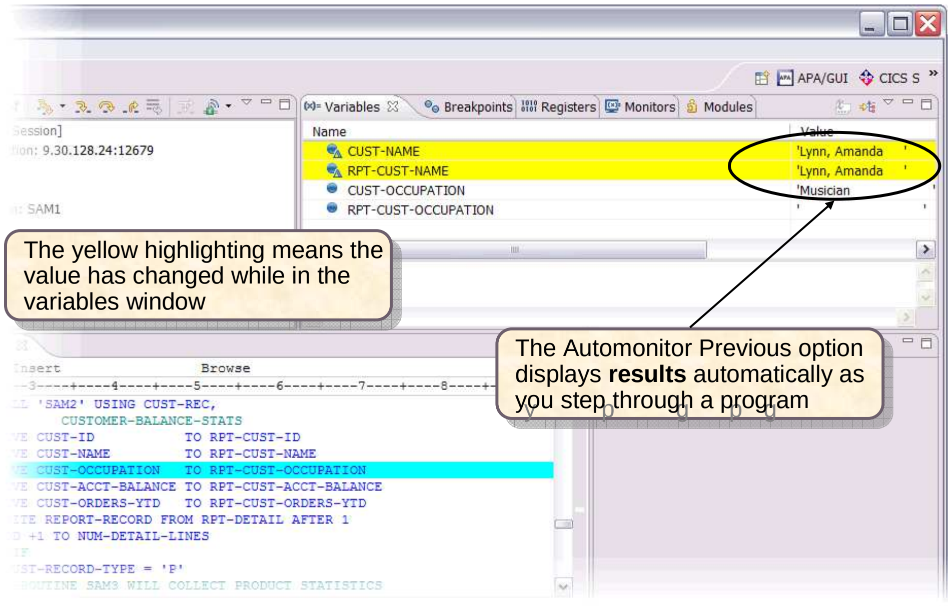


Name	Value
CUST-ID	'01001'
RPT-CUST-ID	'01001'
CUST-NAME	'Lynn, Amanda '
RPT-CUST-NAME	' '

```
CALL 'SAM2' USING CUST-REC,
        CUSTOMER-BALANCE-STATS
MOVE CUST-ID            TO RPT-CUST-ID
MOVE CUST-NAME          TO RPT-CUST-NAME
MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
ADD +1 TO NUM-DETAIL-LINES
```

“Automonitor Previous” is similar to the "Automonitor current" setting, in that it automatically shows variables referenced by the current statement. But in addition, it shows variables referenced by the previous statement shown. Notice that the current statement references two variables, and rpt-cust-name is one of them. It appears in the variables view, and it's value is blanks. The step button is clicked.

After a STEP. The Automonitor refreshes the display as you step and run



The yellow highlighting means the value has changed while in the variables window

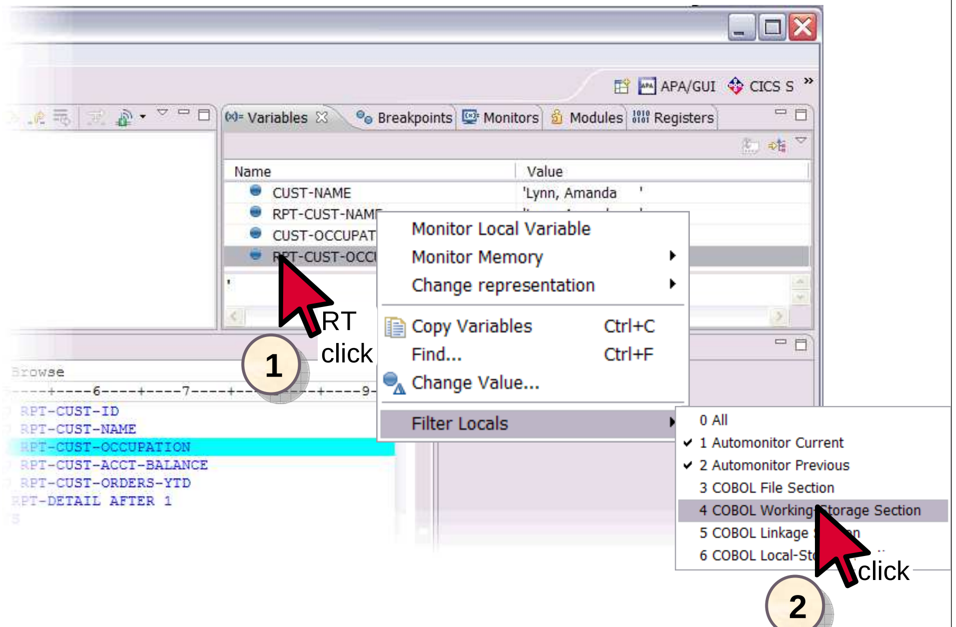
The Automonitor Previous option displays **results** automatically as you step through a program

```
--3----+----4----+----5----+----6----+----7----+----8----+-
LL 'SAM2' USING CUST-REC,
      CUSTOMER-BALANCE-STATS
VE CUST-ID            TO RPT-CUST-ID
VE CUST-NAME          TO RPT-CUST-NAME
VE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
VE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
VE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
ITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
D +1 TO NUM-DETAIL-LINES
IF
ST-RECORD-TYPE = 'P'
UTINE SAM3 WILL COLLECT PRODUCT STATISTICS
```

One statement ran, and now the next line is the current statement. Notice that in the variables view, in addition to the variables referenced by the new statement, the rpt-cust-name variable is still displayed. You can see that it's value changed.

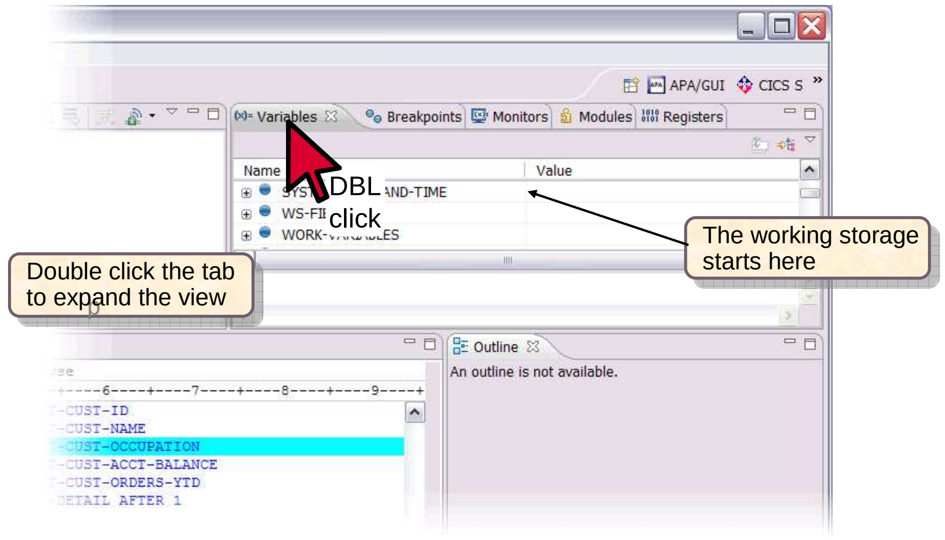
"Automonitor previous" can be a very helpful setting, because it automatically shows the results of changes to variables as you step through a program. This can save you a lot of time as you monitor variable values while debugging.

Select the COBOL Working-Storage Section to view all the Working-Storage variables



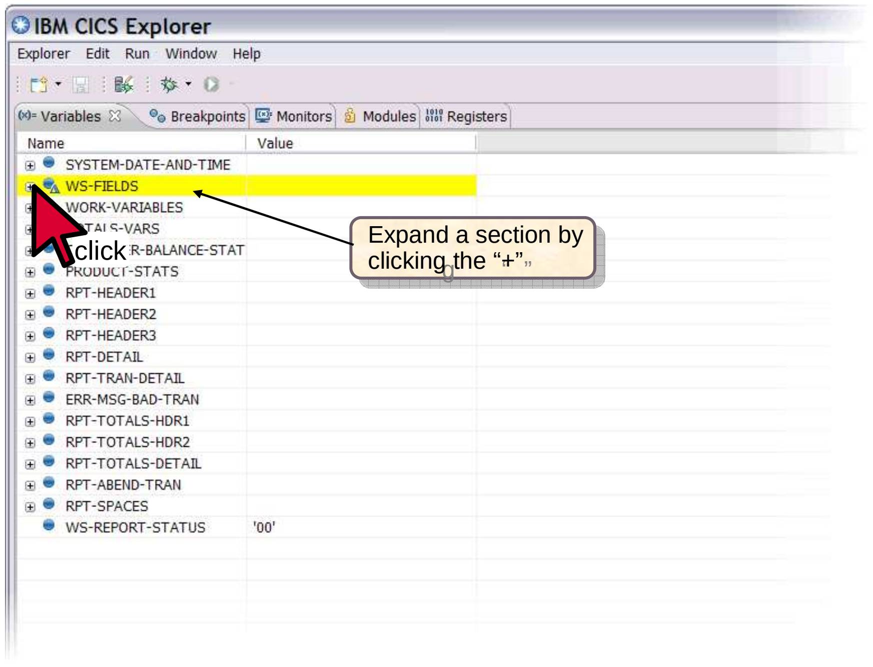
There are other filtering options. Again, to see the options, right click in the variables view and select "filter locals". Notice that there are options to display all variables, and options to display variables in the COBOL file section, working-storage, linkage, and local storage sections. This time, COBOL working-storage is selected.

Result of adding the COBOL working-storage section to view all the working-storage variables



All variables in the COBOL working-storage section are displayed. You can scroll the view using the scroll bar to locate variables. Here the variables view is expanded by double clicking its tab.

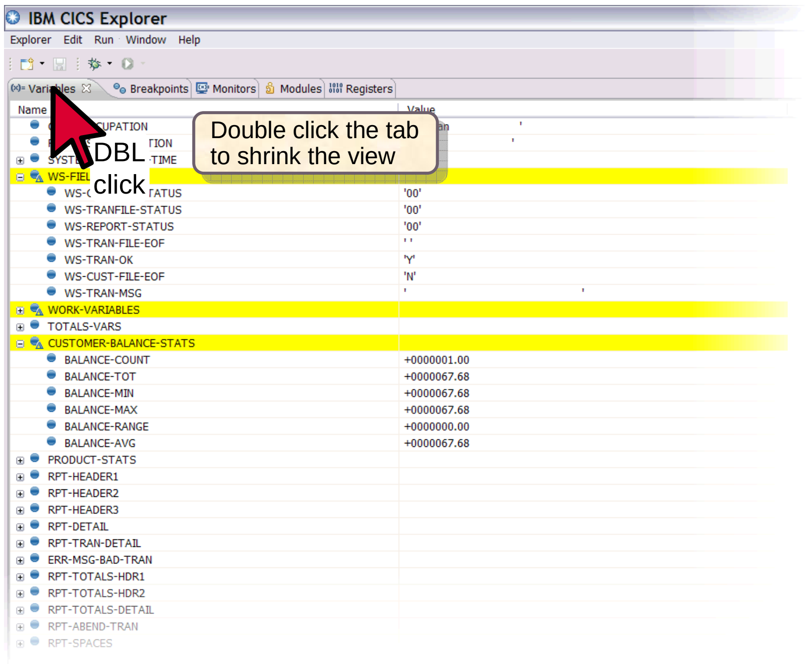
Showing all working-storage variables



Name	Value
⊞ SYSTEM-DATE-AND-TIME	
⊞ WS-FIELDS	
⊞ WORK-VARIABLES	
⊞ TALS-VARS	
⊞ R-BALANCE-STAT	
⊞ PRODUCT-STATS	
⊞ RPT-HEADER1	
⊞ RPT-HEADER2	
⊞ RPT-HEADER3	
⊞ RPT-DETAIL	
⊞ RPT-TRAN-DETAIL	
⊞ ERR-MSG-BAD-TRAN	
⊞ RPT-TOTALS-HDR1	
⊞ RPT-TOTALS-HDR2	
⊞ RPT-TOTALS-DETAIL	
⊞ RPT-ABEND-TRAN	
⊞ RPT-SPACES	
WS-REPORT-STATUS	'00'

Group variables are collapsed, and these are indicated by a "+" in front of the variable name. To expand an item to see it's subordinate variables, click the "+".

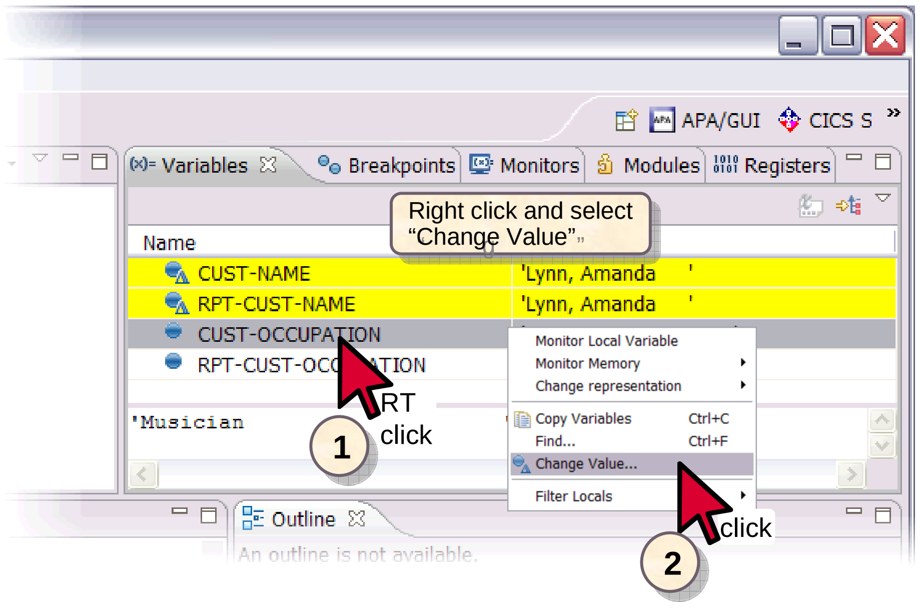
Showing all working-storage variables



Name	Value
WS-FIELDS	
WS-…TATUS	'00'
WS-TRANFILE-STATUS	'00'
WS-REPORT-STATUS	'00'
WS-TRAN-FILE-EOF	' '
WS-TRAN-OK	'Y'
WS-CUST-FILE-EOF	'N'
WS-TRAN-MSG	' '
WORK-VARIABLES	
TOTALS-VARS	
CUSTOMER-BALANCE-STATS	
BALANCE-COUNT	+0000001.00
BALANCE-TOT	+0000067.68
BALANCE-MIN	+0000067.68
BALANCE-MAX	+0000067.68
BALANCE-RANGE	+0000000.00
BALANCE-AVG	+0000067.68
PRODUCT-STATS	
RPT-HEADER1	
RPT-HEADER2	
RPT-HEADER3	
RPT-DETAIL	
RPT-TRAN-DETAIL	
ERR-MSG-BAD-TRAN	
RPT-TOTALS-HDR1	
RPT-TOTALS-HDR2	
RPT-TOTALS-DETAIL	
RPT-ABEND-TRAN	
RPT-SPACES	

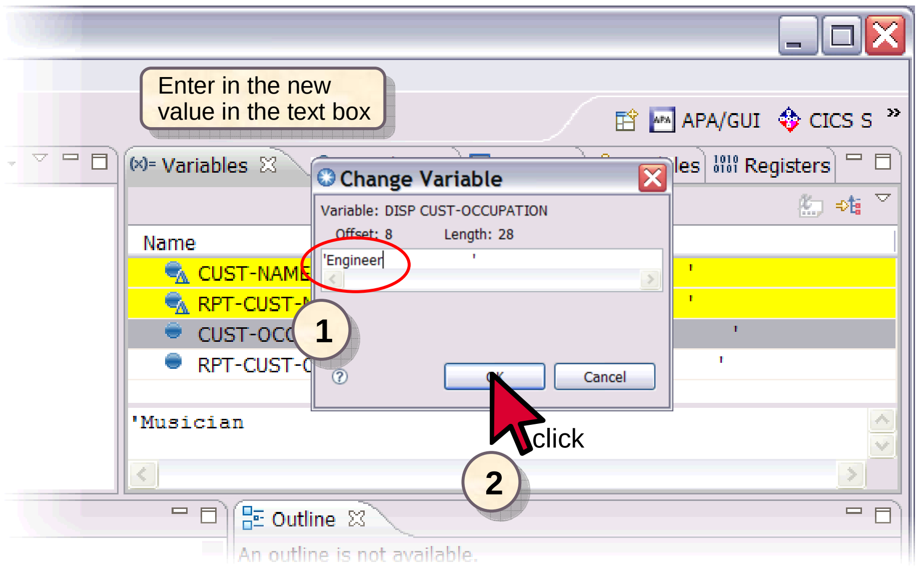
The group was expanded to show the lower level variables. To retract it again, you can click the “-”. To reduce the view back to it's normal size, the tab is double clicked.

Changing a variable value



You can change the value of a variable. Select and right click the variable that you want to change, then select "Change Value" from the pop-up menu.

Change the value

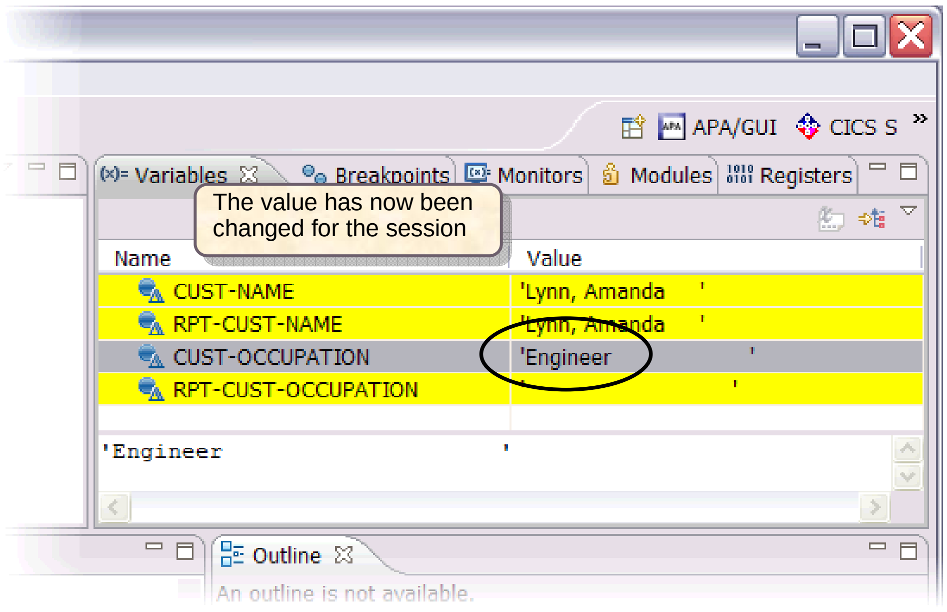


Variable: DISP CUST-OCCUPATION
Offset: 8 Length: 28
'Engineer '

'Musician

A “change variable” pop-up is displayed. Type in the new value, and click “OK”.

The value was changed

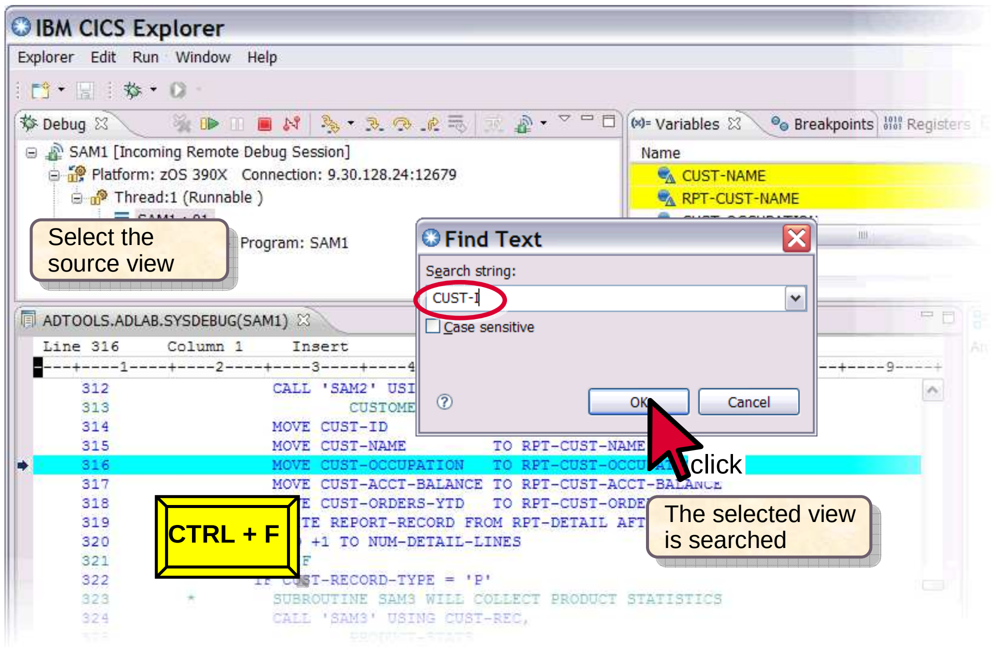

APA/GUI
CICS S
Variables
Breakpoints
Monitors
Modules
Registers
The value has now been changed for the session
Outline
An outline is not available.

Name	Value
CUST-NAME	'Lynn, Amanda '
RPT-CUST-NAME	'Lynn, Amanda '
CUST-OCCUPATION	'Engineer '
RPT-CUST-OCCUPATION	' '

```
'Engineer                  '
```

And the value is changed.

Control + F searches for text in a window

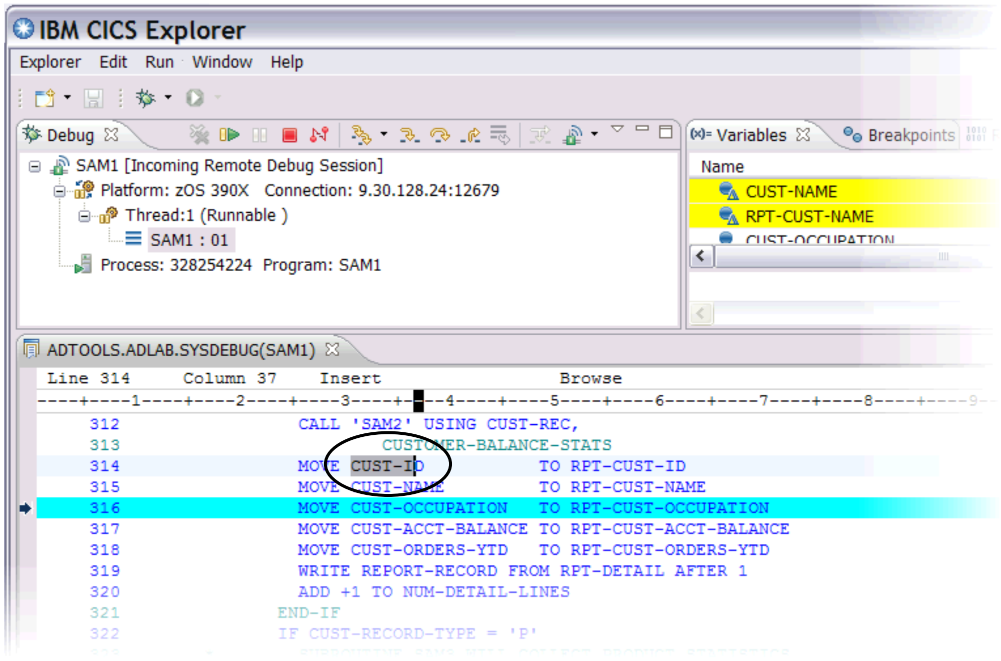


```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12679
    Thread:1 (Runnable )
```

```
Line 316      Column 1       Insert
312                    CALL 'SAM2' USI
313                            CUSTOME
314                    MOVE CUST-ID
315                    MOVE CUST-NAME           TO RPT-CUST-NAME
316                    MOVE CUST-OCCUPATION     TO RPT-CUST-OCCU
317                    MOVE CUST-ACCT-BALANCE   TO RPT-CUST-ACCT-BALANCE
318                         E CUST-ORDERS-YTD   TO RPT-CUST-ORDE
319                          TE REPORT-RECORD FROM RPT-DETAIL AFT
320                          +1 TO NUM-DETAIL-LINES
321                          F
322                 IF CUST-RECORD-TYPE = 'P'
323          *      SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
324                    CALL 'SAM3' USING CUST-REC,
```

Sometimes you need to find a variable in the source view. You can right-click in the source view and select "find text", or you can use the “CTRL + F” hot key to bring up the “Find Text” pop-up. Enter part of the variable name and click OK.

After repeated CTRL + Finds



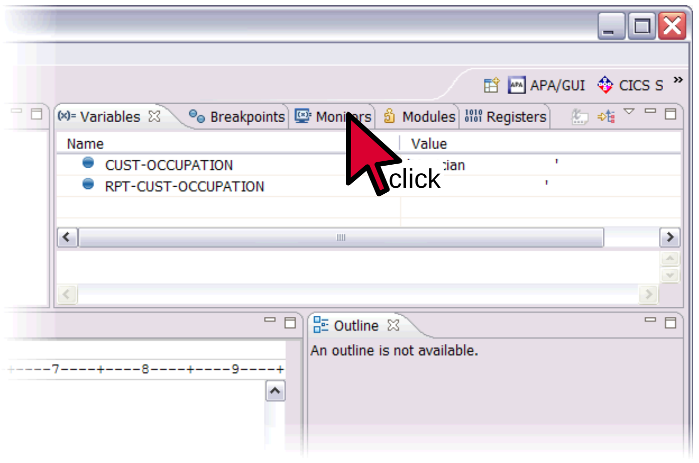
SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:12679
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

```
Line 314       Column 37     Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9--
                  CALL 'SAM2' USING CUST-REC,
                          CUSTOMER-BALANCE-STATS
                  MOVE CUST-ID           TO RPT-CUST-ID
                  MOVE CUST-NAME         TO RPT-CUST-NAME
                  MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
                  MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
                  MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
                  WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
                  ADD +1 TO NUM-DETAIL-LINES
                END-IF
                IF CUST-RECORD-TYPE = 'P'
```

43 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

The first occurrence of the string is highlighted. You can repeat the find to search forward in the program.

Display the monitors view

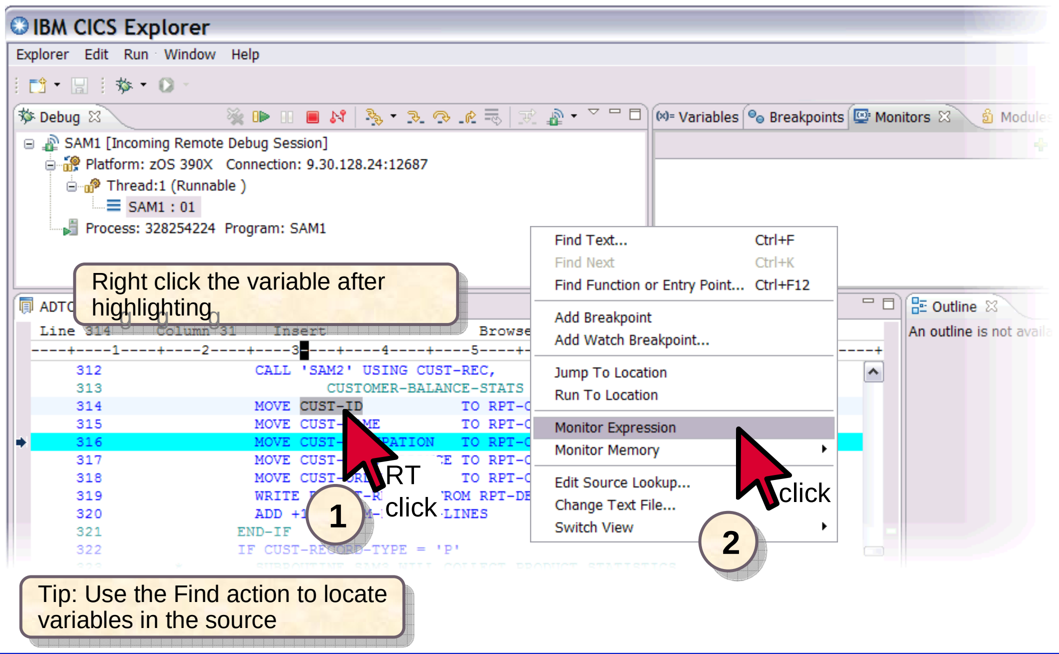


Name	Value
CUST-OCCUPATION	...cian '
RPT-CUST-OCCUPATION	'

An outline is not available.

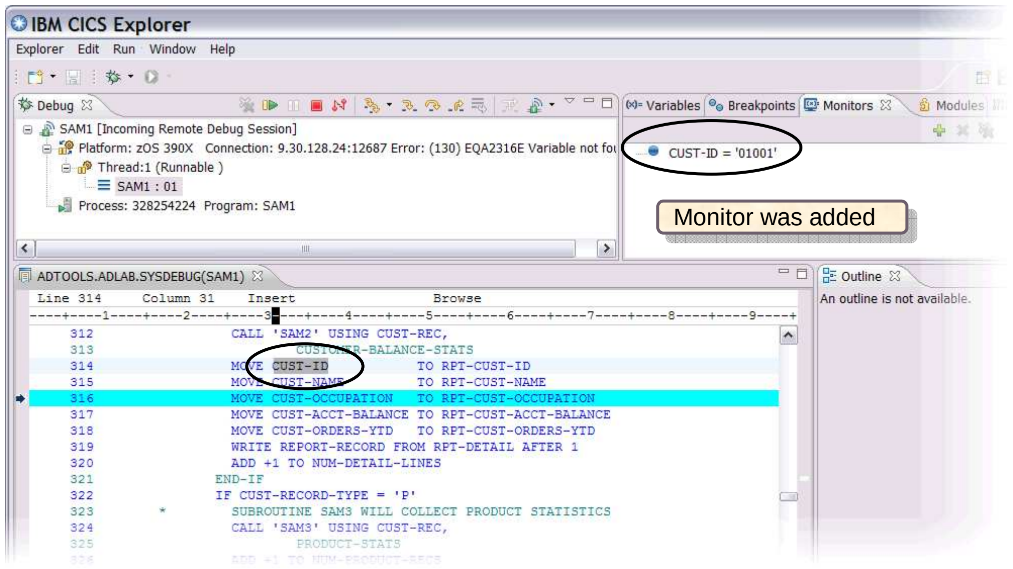
Next, you will learn about another view you can use to display variables, the monitor view. Click the "monitors" tab to select it.

Add a variable to the Monitors view



You can add individual variables to the monitors view. One way to add a variable is to highlight it in the source, right click it, and then select “monitor expression”. This example shows variable “CUST-ID” added to the monitor.

Result of the "Monitor Expression"



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12687 Error: (130) EQA2316E Variable not fo
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
CUST-ID = '01001'
```

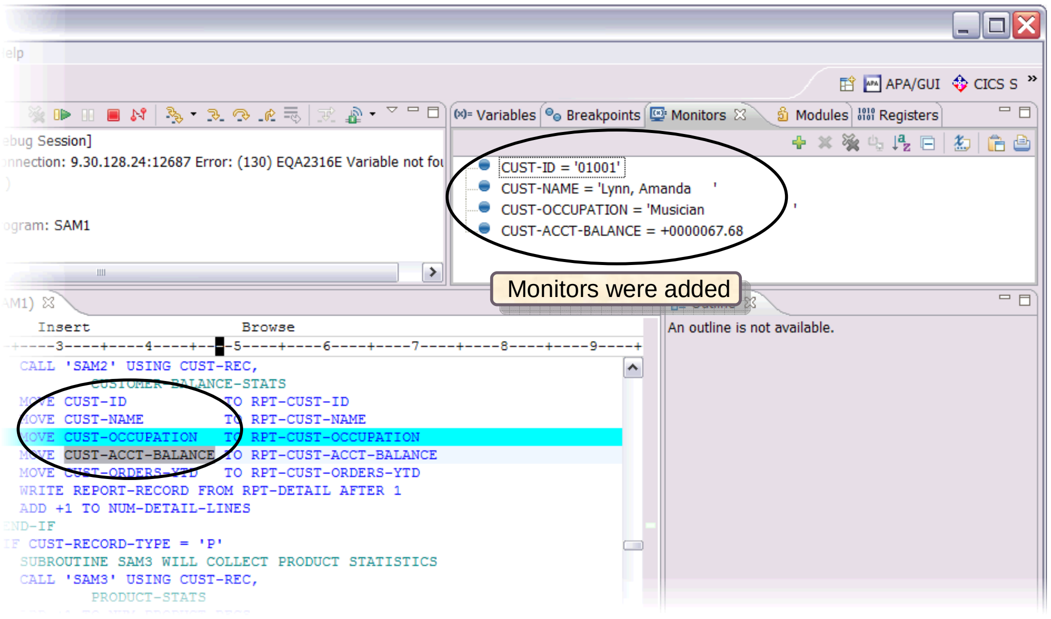
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

An outline is not available.

```
 Line 314     Column 31    Insert                 Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9-----+
                         CALL 'SAM2' USING CUST-REC,
                                 CUSTOMER-BALANCE-STATS
                         MOVE CUST-ID          TO RPT-CUST-ID
                         MOVE CUST-NAME        TO RPT-CUST-NAME
                         MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
                         MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
                         MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
                         WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
                         ADD +1 TO NUM-DETAIL-LINES
                       END-IF
                       IF CUST-RECORD-TYPE = 'P'
            *            SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
                         CALL 'SAM3' USING CUST-REC,
                                 PRODUCT-STATS
                         ADD +1 TO NUM-PRODUCT-RECS
```

CUST-ID is shown in the monitors view. A variable added to the monitor will remain there until you remove it. As you run and step through the program, you can watch as the program changes it's value.

After adding more variables to the monitor



```
ebug Session]
nnection: 9.30.128.24:12687 Error: (130) EQA2316E Variable not fou
ogram: SAM1
```

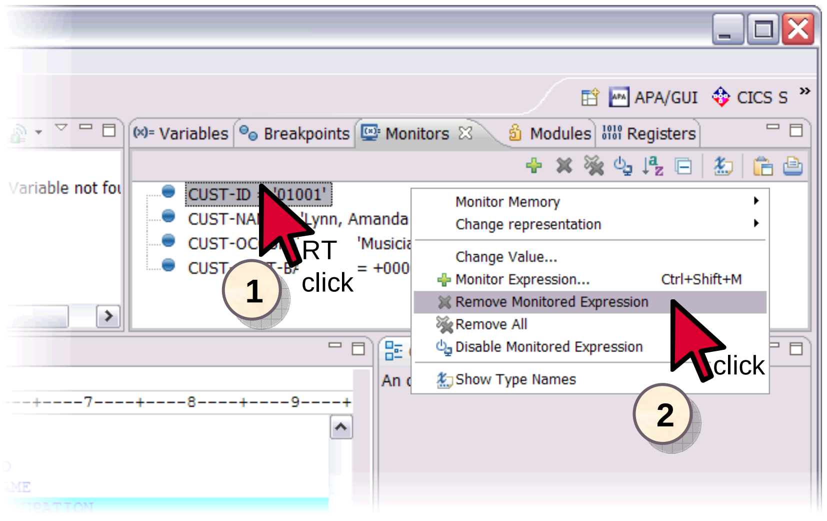
```
CUST-ID = '01001'
CUST-NAME = 'Lynn, Amanda      '
CUST-OCCUPATION = 'Musician                          '
CUST-ACCT-BALANCE = +0000067.68
```

```
An outline is not available.
```

```
-+----3----+----4----+---5----+----6----+----7----+----8----+----9----+
  CALL 'SAM2' USING CUST-REC,
            CUSTOMER-BALANCE-STATS
  MOVE CUST-ID           TO RPT-CUST-ID
  MOVE CUST-NAME         TO RPT-CUST-NAME
  MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
  MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
  MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
  WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
  ADD +1 TO NUM-DETAIL-LINES
END-IF
IF CUST-RECORD-TYPE = 'P'
  SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
  CALL 'SAM3' USING CUST-REC,
            PRODUCT-STATS
```

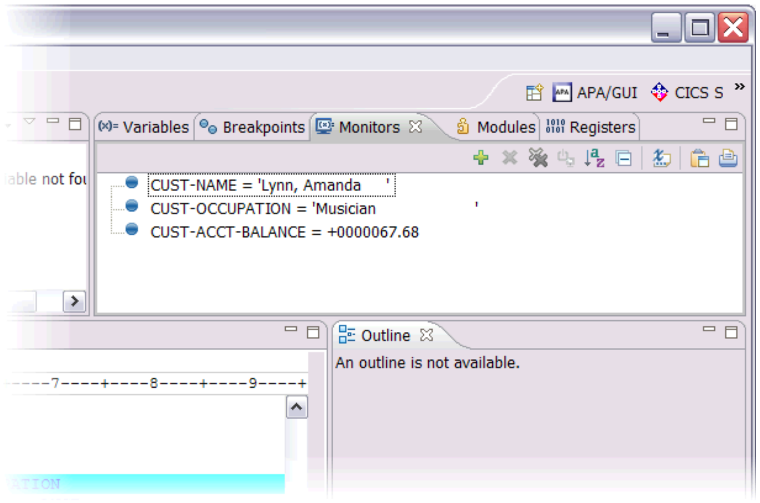
Add as many variables to the monitors view as you need. Typically, you use the monitor to display important variables that you always want to see. In this example, several other variables were also added.

Clearing a variable from the monitor



To remove a variable from the monitors view, highlight it and right click, and then select "Remove Monitored Expression".

Result of “Remove” action – the monitored variable was cleared

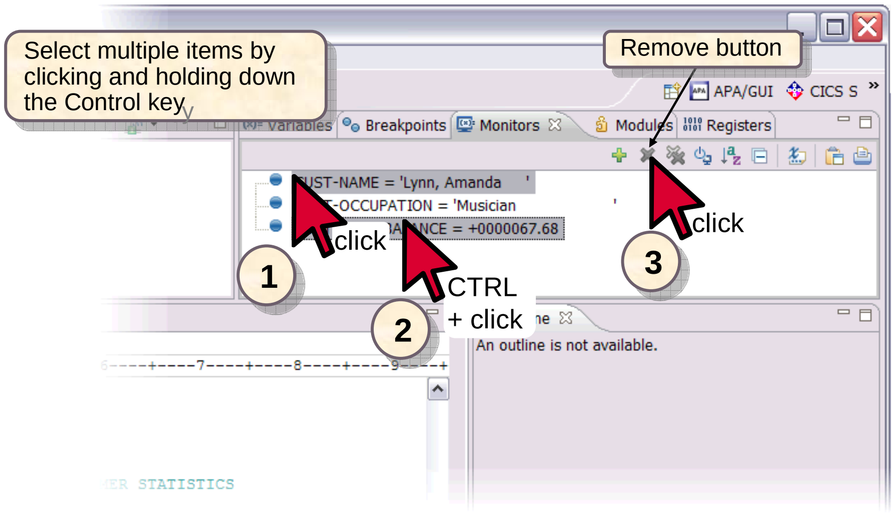


```
CUST-NAME = 'Lynn, Amanda     '
CUST-OCCUPATION = 'Musician              '
CUST-ACCT-BALANCE = +0000067.68
```

An outline is not available.

That removes the selected variable from the monitor.

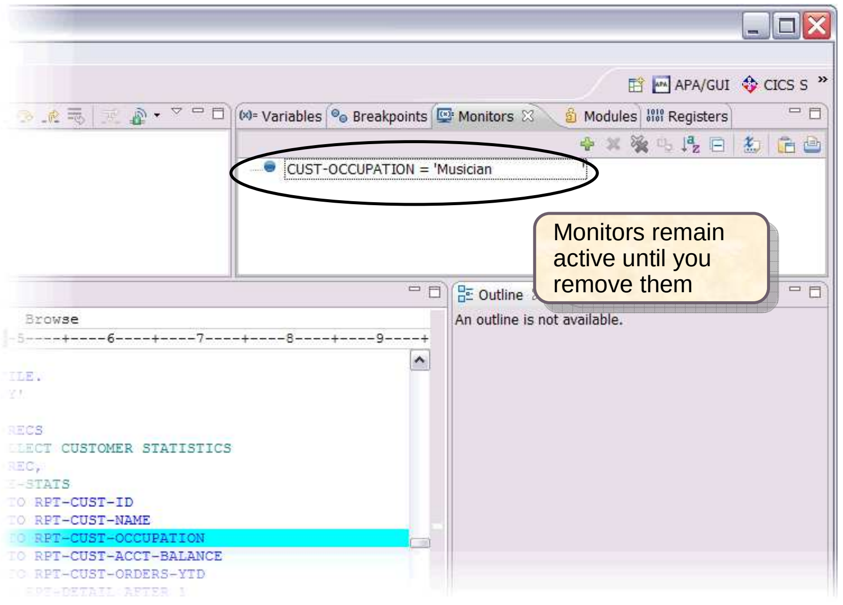
Remove multiple items from the monitor



CUST-NAME = 'Lynn, Amanda '
CUST-OCCUPATION = 'Musician '
BALANCE = +0000067.68

Another way to remove one or more variables is to select them and then click the "remove" button. You can remove several at a time if you hold down the “control” key, and multi-select by clicking on each variable. Once the variables are selected, click the “X”-shaped remove button.

Result of a clearing multiple items from the monitor

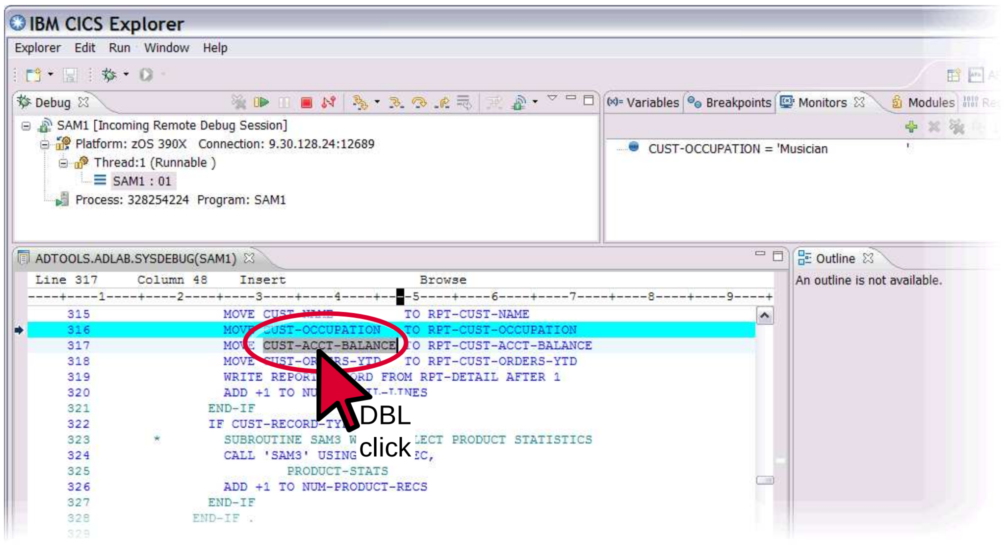

APA/GUI
CICS S
Variables
Breakpoints
Monitors
Modules
Registers
Outline
An outline is not available.
Browse
Monitors remain active until you remove them

```
CUST-OCCUPATION = 'Musician
```

```
-5----+----6----+----7----+----8----+----9----+
ILE.
Y'

RECS
LECT CUSTOMER STATISTICS
REC,
-STATS
TO RPT-CUST-ID
TO RPT-CUST-NAME
TO RPT-CUST-OCCUPATION
TO RPT-CUST-ACCT-BALANCE
TO RPT-CUST-ORDERS-YTD
 RPT-DETAIL AFTER 1
```

That removes the selected variables.

Select and double-click to add a variable to the monitor



SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X   Connection: 9.30.128.24:12689
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

CUST-OCCUPATION = 'Musician   '

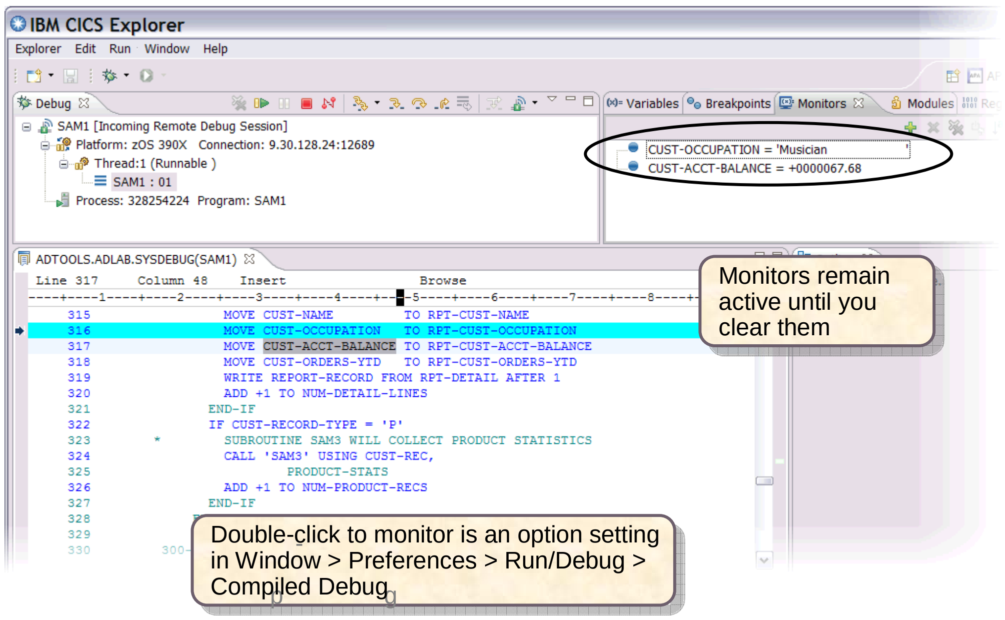
ADTOOLS.ADLAB.SYSDEBUG(SAM1)

An outline is not available.

```
Line 317      Column 48     Insert                 Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+
                 MOVE CUST-NAME         TO RPT-CUST-NAME
                 MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
                 MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
                 MOVE CUST-OR   RS-YTD  TO RPT-CUST-ORDERS-YTD
                 WRITE REPOR    ORD FROM RPT-DETAIL AFTER 1
                 ADD +1 TO NU      L-LINES
               END-IF
               IF CUST-RECORD-TY
      *          SUBROUTINE SAM3 W      LECT PRODUCT STATISTICS
                 CALL 'SAM3' USING      EC,
                         PRODUCT-STATS
                 ADD +1 TO NUM-PRODUCT-RECS
               END-IF
             END-IF .
```

An easy way to monitor a variable is to select and then double-click it.

Result of double-click a variable


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
Variables Breakpoints Monitors Modules

SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:12689
Thread:1 (Runnable)
SAM1 : 01
Process: 328254224 Program: SAM1

CUST-OCCUPATION = 'Musician '
CUST-ACCT-BALANCE = +0000067.68

ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 317     Column 48    Insert                   Browse
 ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+-
                   MOVE CUST-NAME          TO RPT-CUST-NAME
                   MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
                   MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
                   MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
                   WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
                   ADD +1 TO NUM-DETAIL-LINES
                 END-IF
                 IF CUST-RECORD-TYPE = 'P'
          *        SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
                   CALL 'SAM3' USING CUST-REC,
                           PRODUCT-STATS
                   ADD +1 TO NUM-PRODUCT-RECS
                 END-IF
```

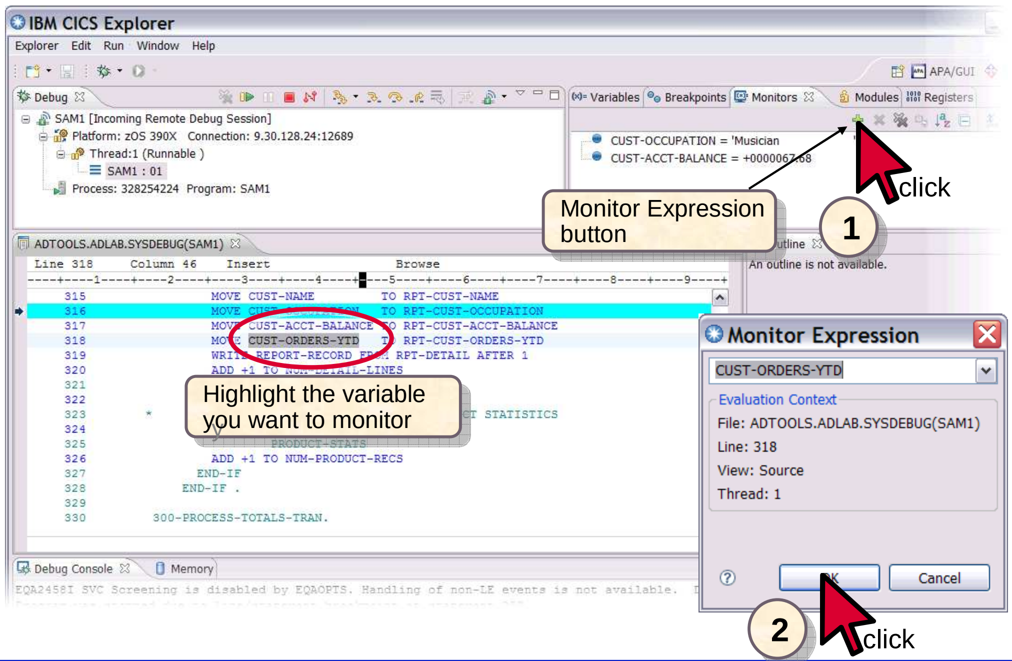
Monitors remain active until you clear them

Double-click to monitor is an option setting in Window > Preferences > Run/Debug > Compiled Debug

IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

That added the variable to the monitors view.

Add a variable to the monitor with the Monitor Expression button



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12689
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
CUST-OCCUPATION = 'Musician
CUST-ACCT-BALANCE = +000006768
```

An outline is not available.

ADTOOLS.ADLAB.SYSDEBUG(SAM1)

```
 Line 318      Column 46     Insert                Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+
                        MOVE CUST-NAME           TO RPT-CUST-NAME
                        MOVE CUST-OCCUPATION     TO RPT-CUST-OCCUPATION
                        MOVE CUST-ACCT-BALANCE   TO RPT-CUST-ACCT-BALANCE
                        MOVE CUST-ORDERS-YTD     TO RPT-CUST-ORDERS-YTD
                        WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
                        ADD +1 TO NUM-DETAIL-LINES


            *                                   CT STATISTICS

                                PRODUCT-STATS
                        ADD +1 TO NUM-PRODUCT-RECS
                      END-IF
                    END-IF .

                300-PROCESS-TOTALS-TRAN.
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.
```

CUST-ORDERS-YTD

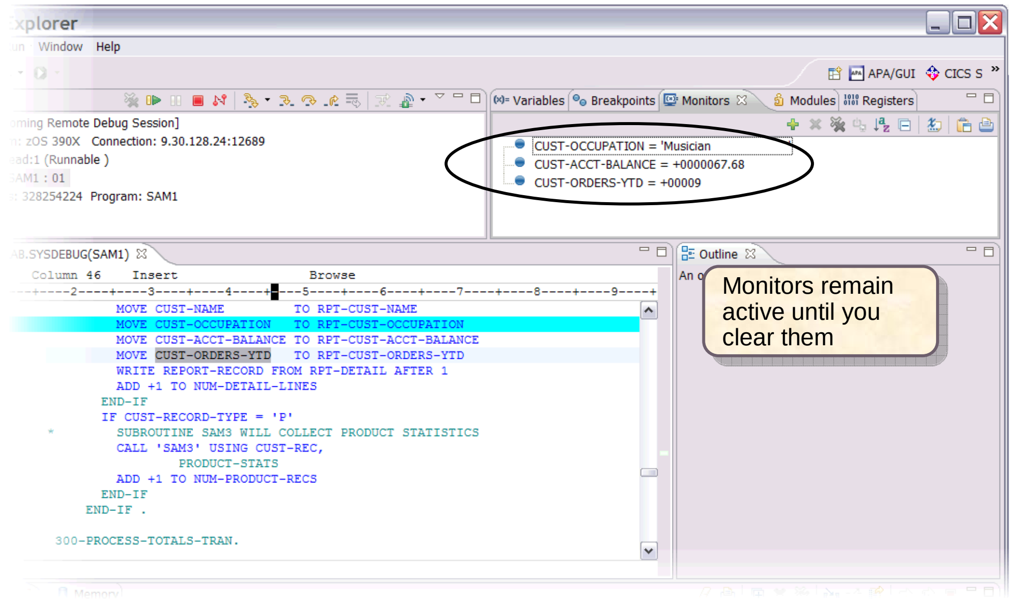
File: ADTOOLS.ADLAB.SYSDEBUG(SAM1)
Line: 318
View: Source
Thread: 1

54 | IBM Debug Tool for z/OS tutorial | © 2012 IBM Corporation

One other way to add a variable to the monitors view is to use the “Monitor Expression” button. In the source view, highlight the variable. Then in the monitors view, click the “+”-shaped Monitor Expression button. A pop-up is displayed, with the variable name already filled in. Click “OK”.

This method can also be used to add a variable that is not highlighted, by typing in the name of the variable.

Result of using the Monitor Expression button

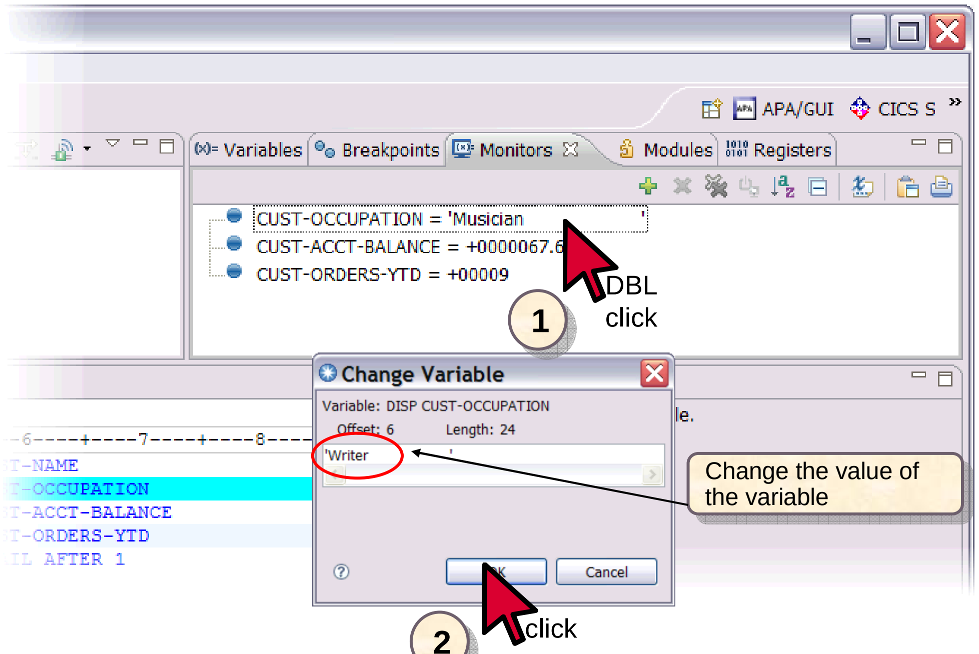


```
CUST-OCCUPATION = 'Musician
CUST-ACCT-BALANCE = +0000067.68
CUST-ORDERS-YTD = +00009
```

```
  Column 46     Insert                    Browse
           MOVE CUST-NAME          TO RPT-CUST-NAME
           MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
           MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
           MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
           WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
           ADD +1 TO NUM-DETAIL-LINES
         END-IF
         IF CUST-RECORD-TYPE = 'P'
 *         SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
           CALL 'SAM3' USING CUST-REC,
                   PRODUCT-STATS
           ADD +1 TO NUM-PRODUCT-RECS
         END-IF
       END-IF .

   300-PROCESS-TOTALS-TRAN.
```

The variable was added to the monitors view.

Overtype the value of a monitored item to change it

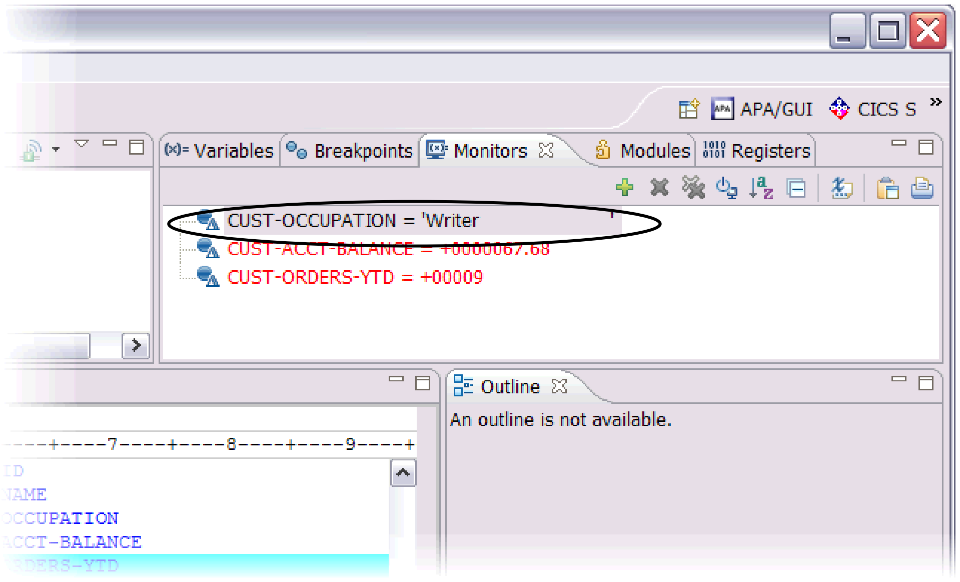


CUST-OCCUPATION = 'Musician '
CUST-ACCT-BALANCE = +0000067.6
CUST-ORDERS-YTD = +00009

Variable: DISP CUST-OCCUPATION
Offset: 6 Length: 24
'Writer '

To change the value of a monitored variable, double click it. Overtype with the new value in the pop-up, and click OK.

The value of a variable was changed



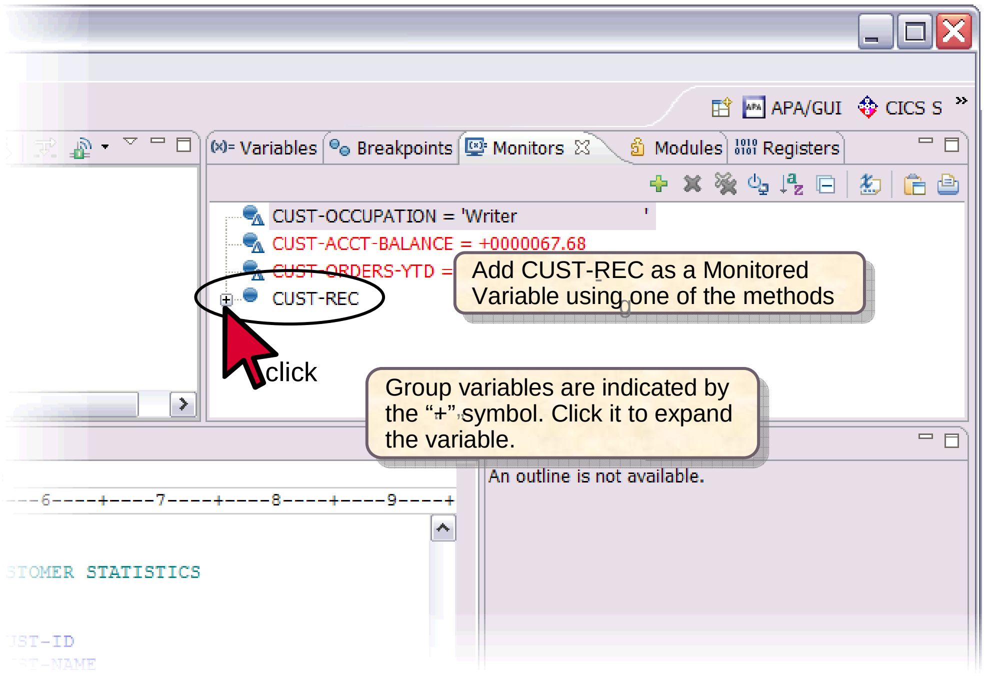
CUST-OCCUPATION = 'Writer '
CUST-ACCT-BALANCE = +0000067.68
CUST-ORDERS-YTD = +00009

An outline is not available.

```
----+----7----+----8----+----9----+
ID
NAME
OCCUPATION
ACCT-BALANCE
ORDERS-YTD
```

That changed the value.

Monitoring group variables



CUST-OCCUPATION = 'Writer '
CUST-ACCT-BALANCE = +0000067.68
CUST-ORDERS-YTD =
CUST-REC

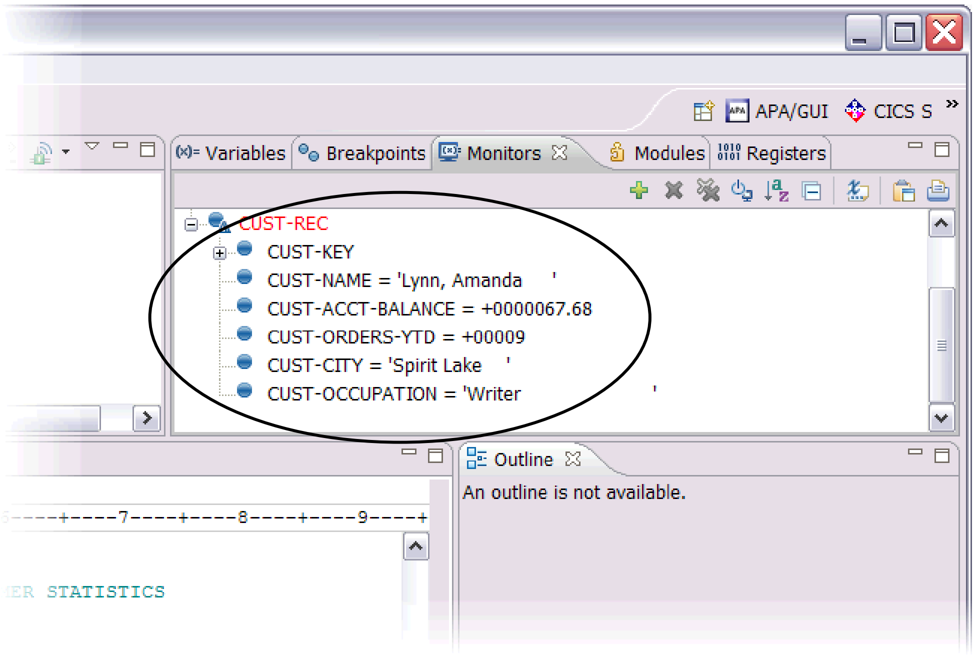
Add CUST-REC as a Monitored Variable using one of the methods

Group variables are indicated by the "+" symbol. Click it to expand the variable.

```
---6----+----7----+----8----+----9----+
STOMER STATISTICS
JST-ID
```

Group variables are indicated by a "+" symbol next to a variable. You can expand a group variable by clicking the "+".

Expanded group variable



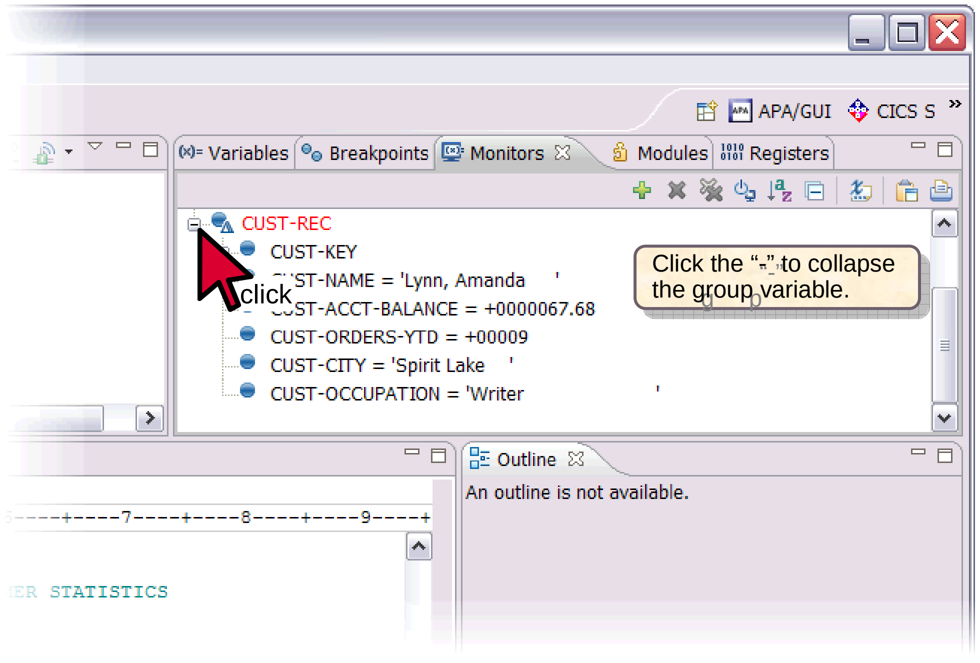
```
CUST-REC
  CUST-KEY
  CUST-NAME = 'Lynn, Amanda       '
  CUST-ACCT-BALANCE = +0000067.68
  CUST-ORDERS-YTD = +00009
  CUST-CITY = 'Spirit Lake    '
  CUST-OCCUPATION = 'Writer                 '
```

An outline is not available.

```
5----+----7----+----8----+----9----+
ER STATISTICS
```

That shows the subordinate variables in the group.

Collapse the group variable



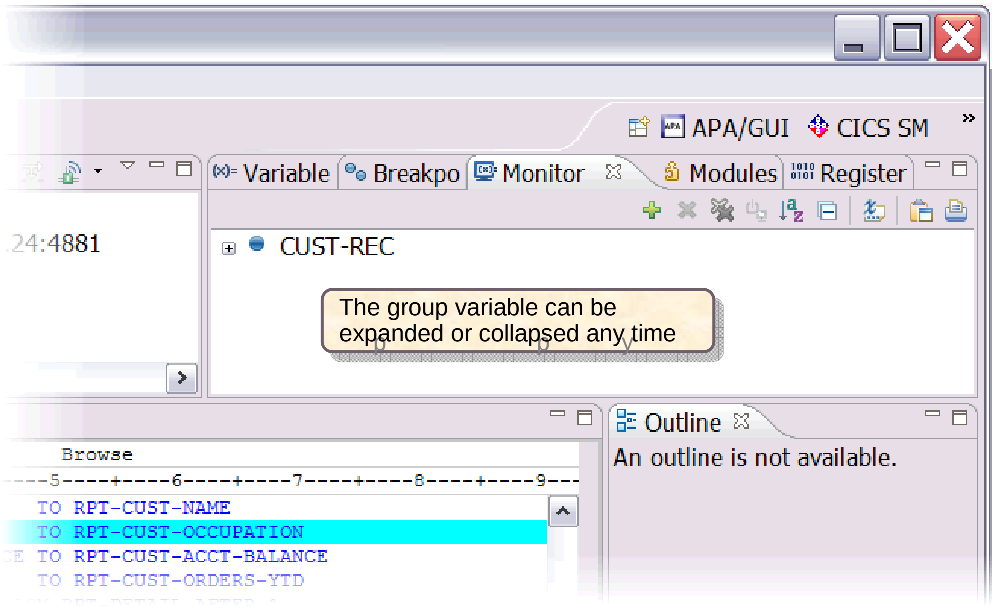
CUST-REC
CUST-KEY
CUST-NAME = 'Lynn, Amanda '
CUST-ACCT-BALANCE = +0000067.68
CUST-ORDERS-YTD = +00009
CUST-CITY = 'Spirit Lake '
CUST-OCCUPATION = 'Writer '

An outline is not available.

```
-----+----7----+----8----+----9----+
ER STATISTICS
```

Collapse a group variable by clicking the "-".

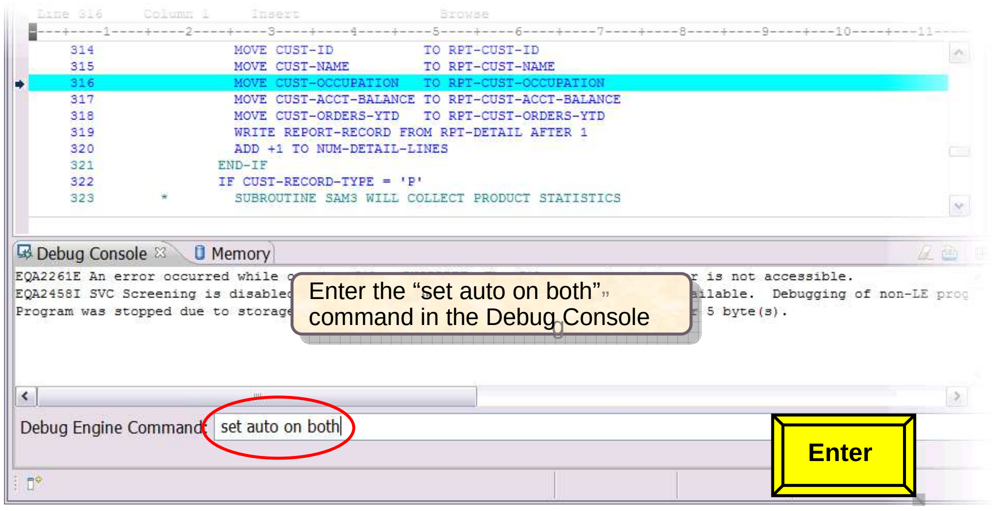
Collapsed the group variable


APA/GUI
CICS SM
Variable
Breakpo
Monitor
Modules
Register
24:4881
CUST-REC
The group variable can be expanded or collapsed any time
Outline
An outline is not available.
Browse

```
----5----+----6----+----7----+----8----+----9---
    TO RPT-CUST-NAME
    TO RPT-CUST-OCCUPATION
 E  TO RPT-CUST-ACCT-BALANCE
    TO RPT-CUST-ORDERS-YTD
```

The group variable is collapsed again.

“Set Auto On Both” command



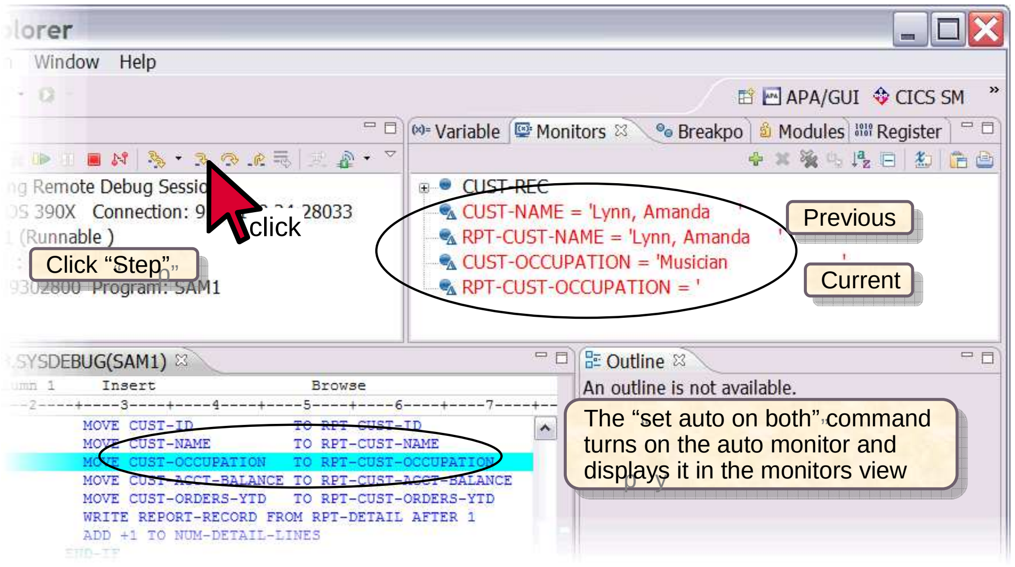
```
Line 316      Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---11---
                        MOVE CUST-ID            TO RPT-CUST-ID
                        MOVE CUST-NAME          TO RPT-CUST-NAME
                        MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
                        MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
                        MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
                        WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
                        ADD +1 TO NUM-DETAIL-LINES
                      END-IF
                      IF CUST-RECORD-TYPE = 'P'
          *              SUBROUTINE SAM3 WILL COLLECT PRODUCT STATISTICS
```

```
EQA2261E An error occurred while c                                    r is not accessible.
EQA2458I SVC Screening is disabled                                    ailable.  Debugging of non-LE prog
Program was stopped due to storage                                    5 byte(s).
```

Debug Engine Command: set auto on both

You have seen that you can auto monitor variables in the variables view. But you can optionally display the auto monitor in the monitors view instead. Enter a "set auto on" or "set auto on both" command in the debug console.

"Auto monitor" variables display in the monitors view

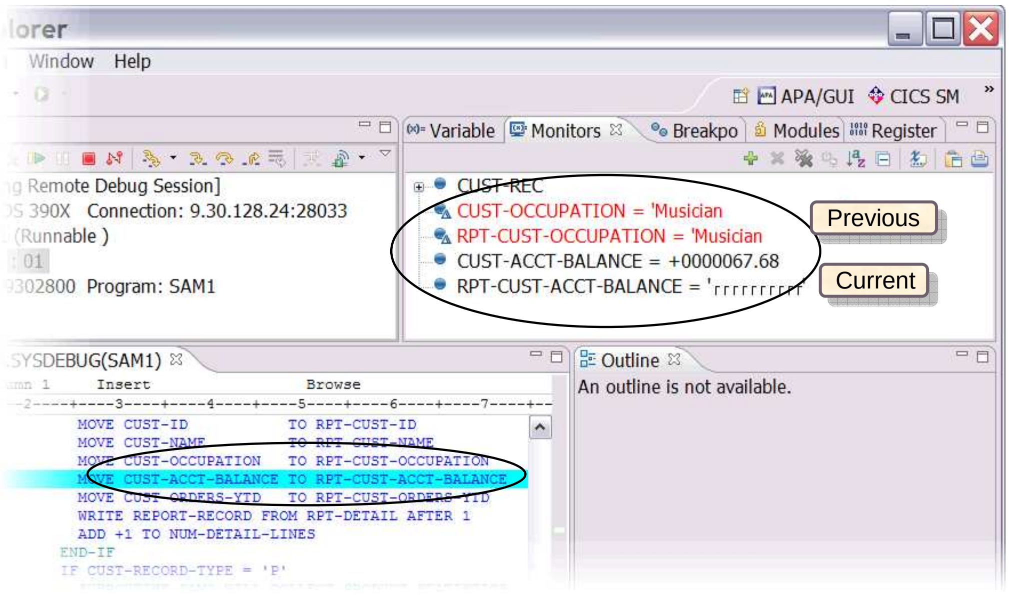

Window Help
APA/GUI CICS SM
Variable Monitors Breakpo Modules Register
Remote Debug Session
S 390X Connection: 9 28033
(Runnable)
9302800 Program: SAM1
click
Click "Step"
Previous
Current
SYSDEBUG(SAM1)
Insert Browse
Outline
An outline is not available.
The "set auto on both" command turns on the auto monitor and displays it in the monitors view

```
CUST-REC
  CUST-NAME = 'Lynn, Amanda          '
  RPT-CUST-NAME = 'Lynn, Amanda          '
  CUST-OCCUPATION = 'Musician
  RPT-CUST-OCCUPATION = '
```

```
MOVE CUST-ID          TO RPT-CUST-ID
MOVE CUST-NAME        TO RPT-CUST-NAME
MOVE CUST-OCCUPATION  TO RPT-CUST-OCCUPATION
MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
MOVE CUST-ORDERS-YTD  TO RPT-CUST-ORDERS-YTD
WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
ADD +1 TO NUM-DETAIL-LINES
END-IF
```

That turns on the auto monitor in the monitors view. Other variables you have added to the monitor also continue to be displayed. Notice that the variables from the current and previous lines are shown. "Step" is clicked.

Result of step with “Set Auto On Both”



```
g Remote Debug Session]
S 390X  Connection: 9.30.128.24:28033
(Runnable )
: 01
9302800  Program: SAM1
```

```
CUST-REC
CUST-OCCUPATION = 'Musician
RPT-CUST-OCCUPATION = 'Musician
CUST-ACCT-BALANCE = +0000067.68
RPT-CUST-ACCT-BALANCE = 'rrrrrrrrr'
```

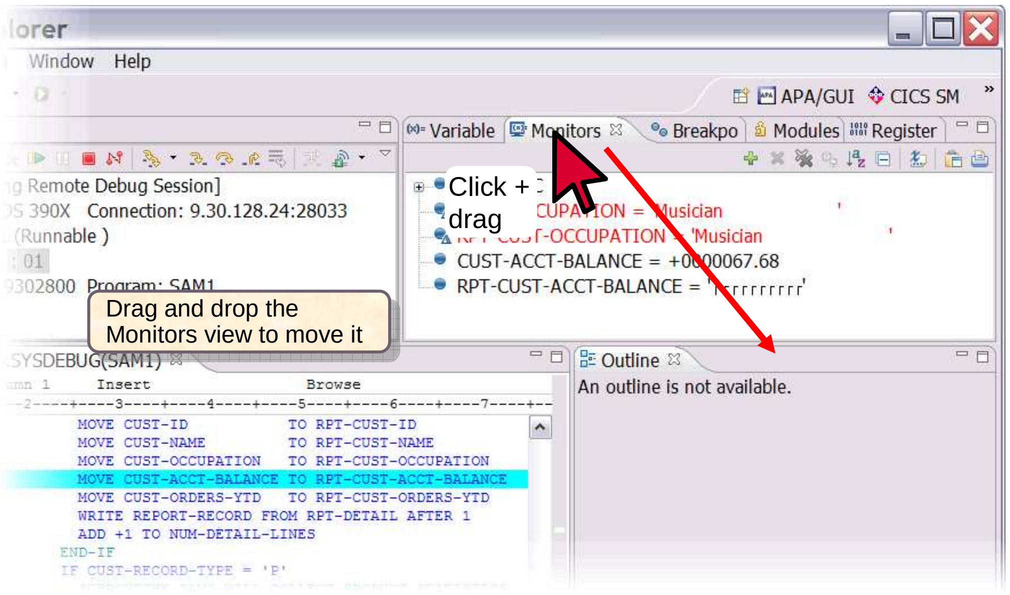
SYSDEBUG(SAM1)

```
mn 1        Insert                   Browse
-2----+----3----+----4----+----5----+----6----+----7----+--
         MOVE CUST-ID             TO RPT-CUST-ID
         MOVE CUST-NAME           TO RPT-CUST-NAME
         MOVE CUST-OCCUPATION     TO RPT-CUST-OCCUPATION
         MOVE CUST-ACCT-BALANCE   TO RPT-CUST-ACCT-BALANCE
         MOVE CUST-ORDERS-YTD     TO RPT-CUST-ORDERS-YTD
         WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
         ADD +1 TO NUM-DETAIL-LINES
       END-IF
       IF CUST-RECORD-TYPE = 'P'
```

An outline is not available.

After stepping, the variables in the auto monitor changed based on what are now the current and previous statements.

Move the monitors view



Window Help
APA/GUI CICS SM
Variable Monitors Breakpo Modules Register
Click + drag
Drag and drop the Monitors view to move it
Outline
An outline is not available.

Remote Debug Session]
S 390X Connection: 9.30.128.24:28033
(Runnable)
01
302800 Program: SAM1

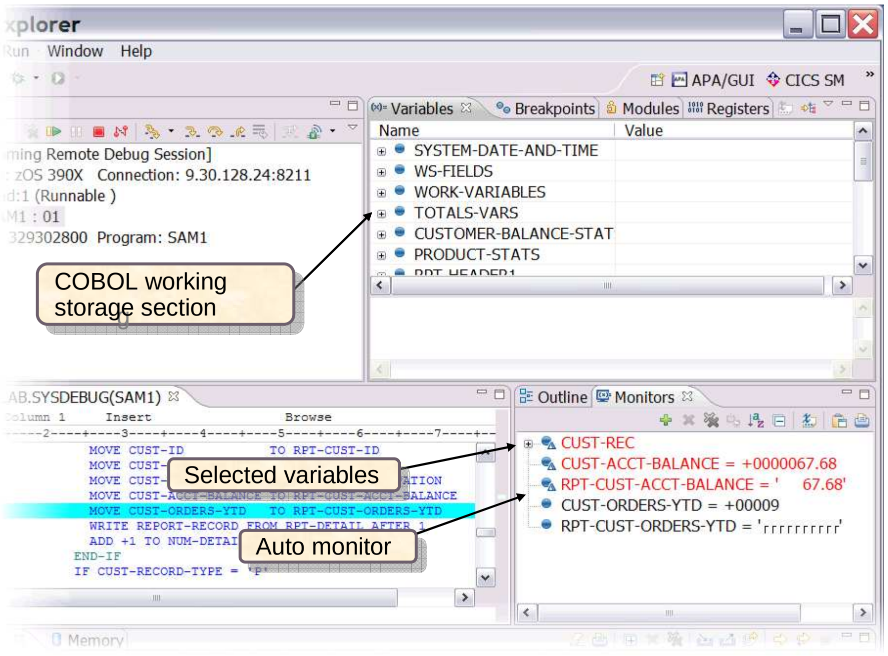
CUPATION = 'Musician '
RPT-CUST-OCCUPATION = 'Musician '
CUST-ACCT-BALANCE = +0000067.68
RPT-CUST-ACCT-BALANCE = 'rrrrrrrrrr'

SYSDEBUG(SAM1)

```
mn 1       Insert                     Browse
--2----+----3----+----4----+----5----+----6----+----7----+--
        MOVE CUST-ID            TO RPT-CUST-ID
        MOVE CUST-NAME          TO RPT-CUST-NAME
        MOVE CUST-OCCUPATION    TO RPT-CUST-OCCUPATION
        MOVE CUST-ACCT-BALANCE  TO RPT-CUST-ACCT-BALANCE
        MOVE CUST-ORDERS-YTD    TO RPT-CUST-ORDERS-YTD
        WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
        ADD +1 TO NUM-DETAIL-LINES
      END-IF
      IF CUST-RECORD-TYPE = 'P'
```

The monitors and variables views can be used in tandem. If you have them in different panes, you can see both of them at the same time. The monitors view is moved by dragging it's tab to a different pane.

An example of using the variables and monitors views together



```
ming Remote Debug Session]
zOS 390X   Connection: 9.30.128.24:8211
d:1 (Runnable )
M1 : 01
329302800  Program: SAM1
```

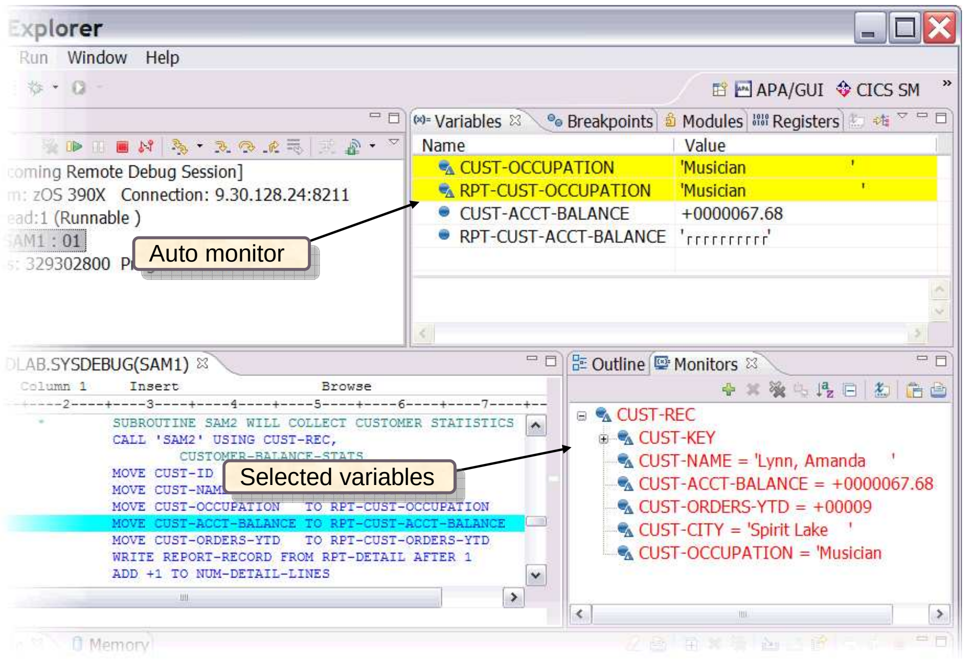
Name	Value
SYSTEM-DATE-AND-TIME	
WS-FIELDS	
WORK-VARIABLES	
TOTALS-VARS	
CUSTOMER-BALANCE-STAT	
PRODUCT-STATS	
RPT HEADER1	

```
AB.SYSDEBUG(SAM1)
Column 1        Insert                      Browse
-----2----+----3----+----4----+----5----+----6----+----7----+--
          MOVE CUST-ID           TO RPT-CUST-ID
          MOVE CUST-
          MOVE CUST-                                   ATION
          MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
          MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
          WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
          ADD +1 TO NUM-DETAI
        END-IF
        IF CUST-RECORD-TYPE = 'P'
```

```
CUST-REC
CUST-ACCT-BALANCE = +0000067.68
RPT-CUST-ACCT-BALANCE = '     67.68'
CUST-ORDERS-YTD = +00009
RPT-CUST-ORDERS-YTD = 'rrrrrrrrrr'
```

Here is an example of using both views together. In this case, the filter in the variables view is set to show COBOL working storage variables, and the monitors view is displaying selected variables and the auto monitor.

Another example of using the monitors and variables view in tandem



```
coming Remote Debug Session]
m: zOS 390X   Connection: 9.30.128.24:8211
ead:1 (Runnable )
AM1 : 01
s: 329302800
```

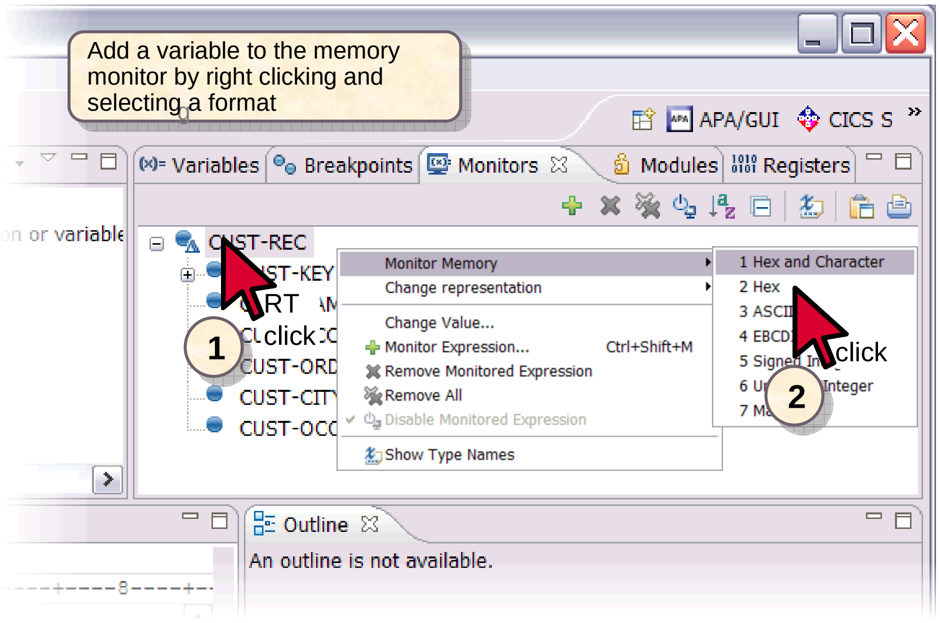
Name	Value
CUST-OCCUPATION	'Musician '
RPT-CUST-OCCUPATION	'Musician '
CUST-ACCT-BALANCE	+0000067.68
RPT-CUST-ACCT-BALANCE	'rrrrrrrrrr'

```
DLAB.SYSDEBUG(SAM1)
 Column 1      Insert                 Browse
-+----2----+----3----+----4----+----5----+----6----+----7----+--
    *       SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATISTICS
            CALL 'SAM2' USING CUST-REC,
                    CUSTOMER-BALANCE-STATS
            MOVE CUST-ID
            MOVE CUST-NAM
            MOVE CUST-OCCUPATION   TO RPT-CUST-OCCUPATION
            MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALANCE
            MOVE CUST-ORDERS-YTD   TO RPT-CUST-ORDERS-YTD
            WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
            ADD +1 TO NUM-DETAIL-LINES
```

```
⊟ CUST-REC
  ⊞ CUST-KEY
    CUST-NAME = 'Lynn, Amanda    '
    CUST-ACCT-BALANCE = +0000067.68
    CUST-ORDERS-YTD = +00009
    CUST-CITY = 'Spirit Lake    '
    CUST-OCCUPATION = 'Musician
```

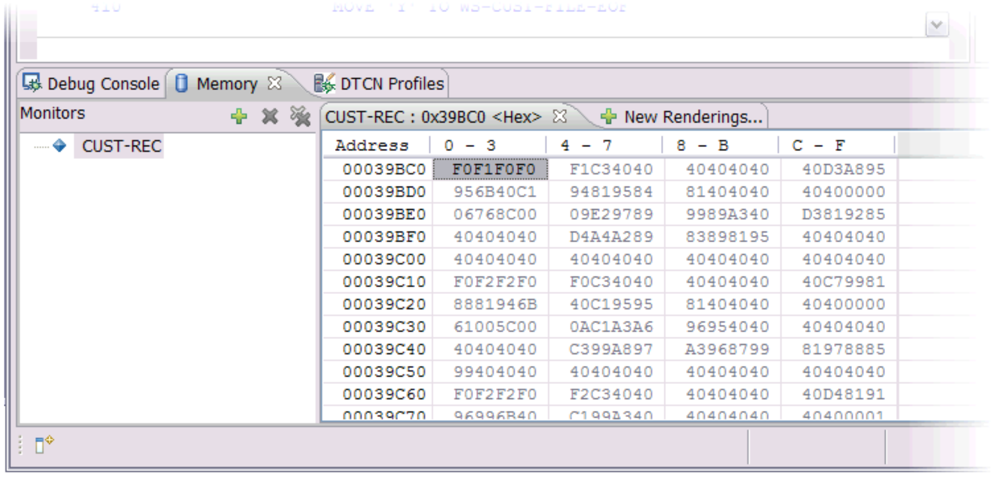
Here is another example of using both views together. This time, the filter in the variables view is set to show the auto monitor, and the monitors view is displaying only selected variables. Using these two views together gives you a lot of control over which variables display automatically as you debug your program.

Add a monitored variable to the memory monitor



Now you have seen how to use the monitors and variables views to work with variables. You can also display memory in the memory view. In this example, the cust-rec variable in the monitor is right clicked, “monitor memory” is selected, and then the representation.

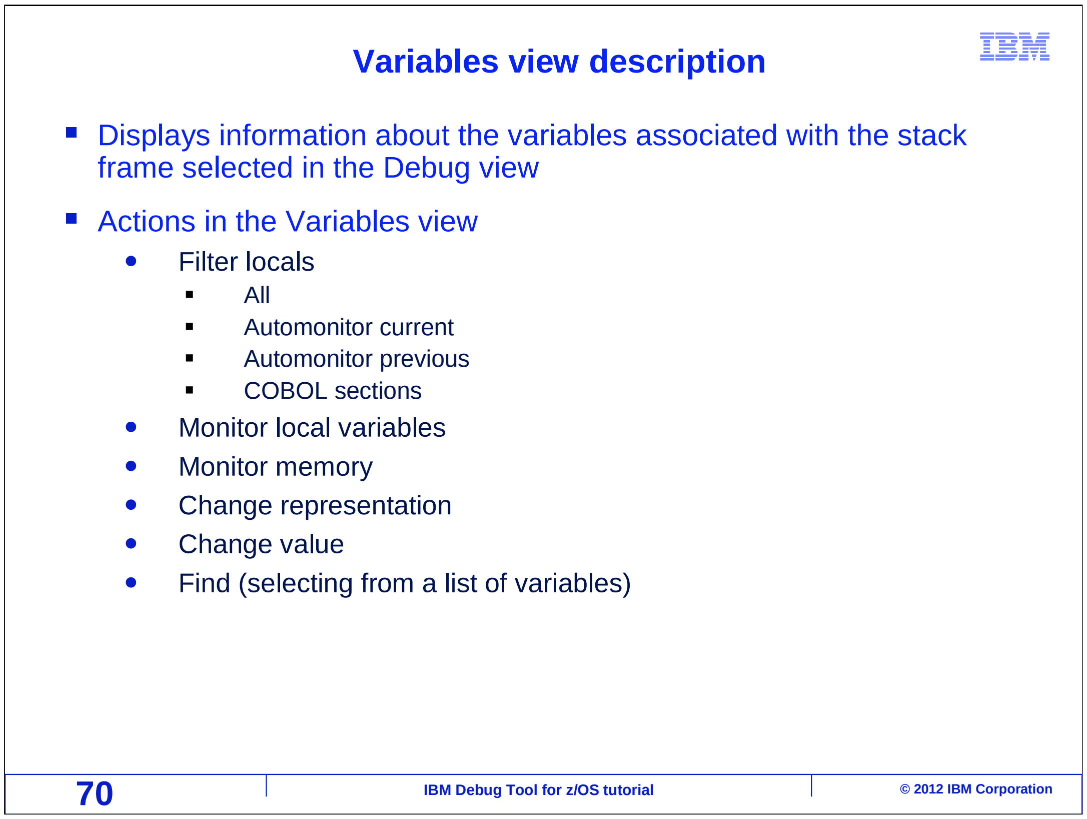
The variable is displayed in the selected format in the Memory view



```
Address    0 - 3     4 - 7     8 - B     C - F
00039BC0   F0F1F0F0  F1C34040  40404040  40D3A895
00039BD0   956B40C1  94819584  81404040  40400000
00039BE0   06768C00  09E29789  9989A340  D3819285
00039BF0   40404040  D4A4A289  83898195  40404040
00039C00   40404040  40404040  40404040  40404040
00039C10   F0F2F2F0  F0C34040  40404040  40C79981
00039C20   8881946B  40C19595  81404040  40400000
00039C30   61005C00  0AC1A3A6  96954040  40404040
00039C40   40404040  C399A897  A3968799  81978885
00039C50   99404040  40404040  40404040  40404040
00039C60   F0F2F2F0  F2C34040  40404040  40D48191
00039C70   96996B40  C199A340  40404040  40400001
```

That added a rendering to the memory view. It displays storage beginning at the address of the selected variable. You can scroll up and down through memory and overtype data values. The representation can be changed to character or hexadecimal.

In this example, a memory rendering was added based on a variable's location, but you can also add them based on addresses or register values. You can add as many memory renderings as you need.

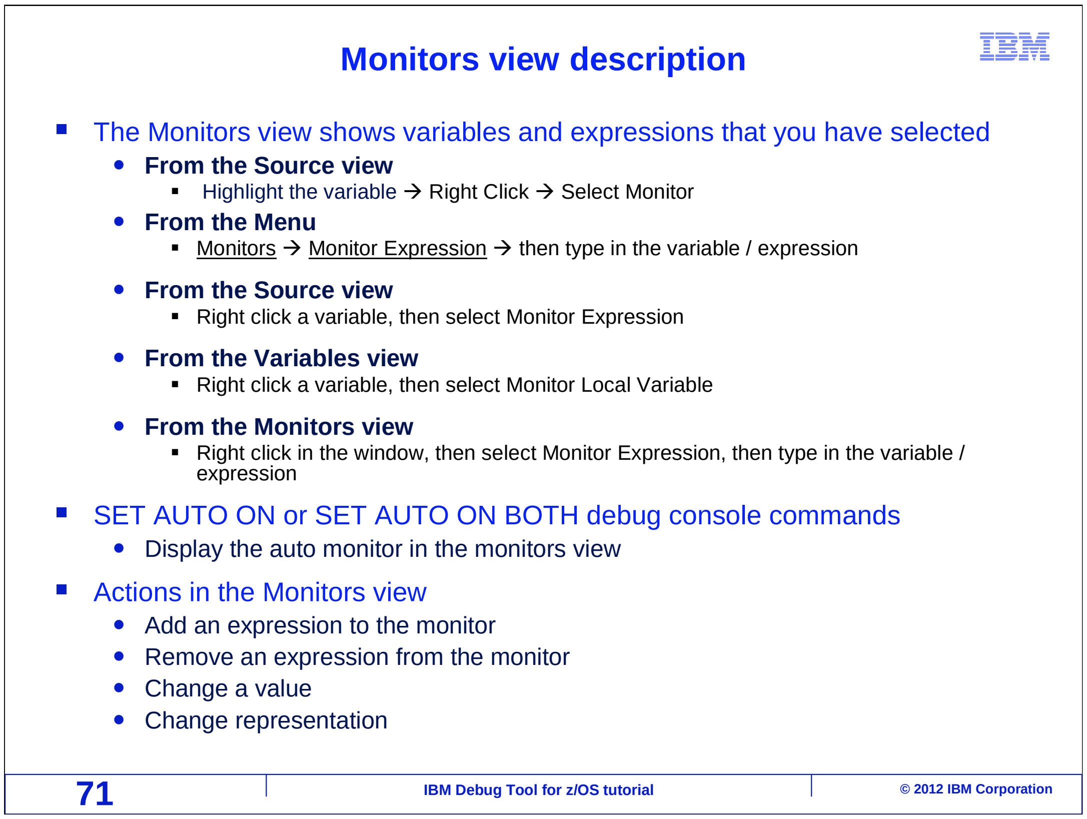


Variables view description

- Displays information about the variables associated with the stack frame selected in the Debug view
- Actions in the Variables view
  - Filter locals
    - All
    - Automonitor current
    - Automonitor previous
    - COBOL sections
  - Monitor local variables
  - Monitor memory
  - Change representation
  - Change value
  - Find (selecting from a list of variables)

70 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

To recap, the variables view displays variables in a program. The view has a filter that you use to change the scope of variables displayed.

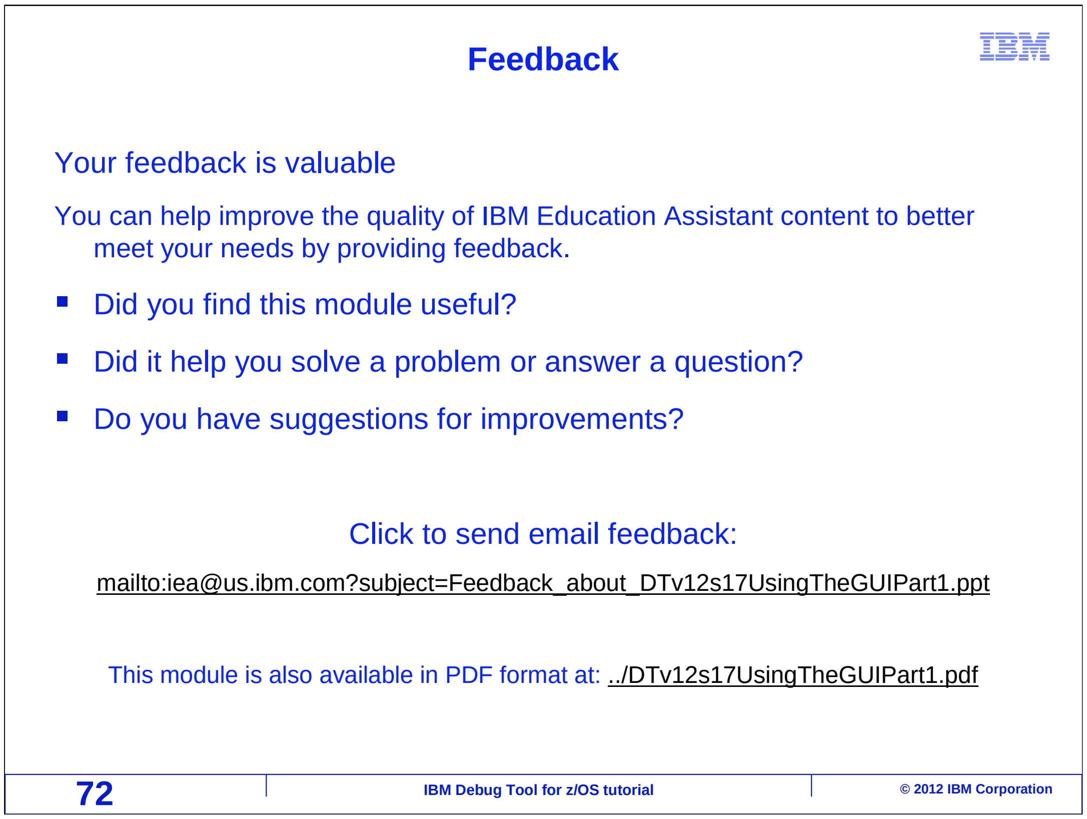


Monitors view description

- The Monitors view shows variables and expressions that you have selected
  - **From the Source view**
    - Highlight the variable → Right Click → Select Monitor
  - **From the Menu**
    - Monitors → Monitor Expression → then type in the variable / expression
  - **From the Source view**
    - Right click a variable, then select Monitor Expression
  - **From the Variables view**
    - Right click a variable, then select Monitor Local Variable
  - **From the Monitors view**
    - Right click in the window, then select Monitor Expression, then type in the variable / expression
- SET AUTO ON or SET AUTO ON BOTH debug console commands
  - Display the auto monitor in the monitors view
- Actions in the Monitors view
  - Add an expression to the monitor
  - Remove an expression from the monitor
  - Change a value
  - Change representation

71 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

The monitors view shows variables and expressions. Add as many variables as you want. You can optionally enter SET AUTO ON or SET AUTO ON BOTH commands to show the auto monitor.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_DTv12s17UsingTheGUIPart1.ppt

This module is also available in PDF format at: ../DTv12s17UsingTheGUIPart1.pdf

72 IBM Debug Tool for z/OS tutorial © 2012 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers