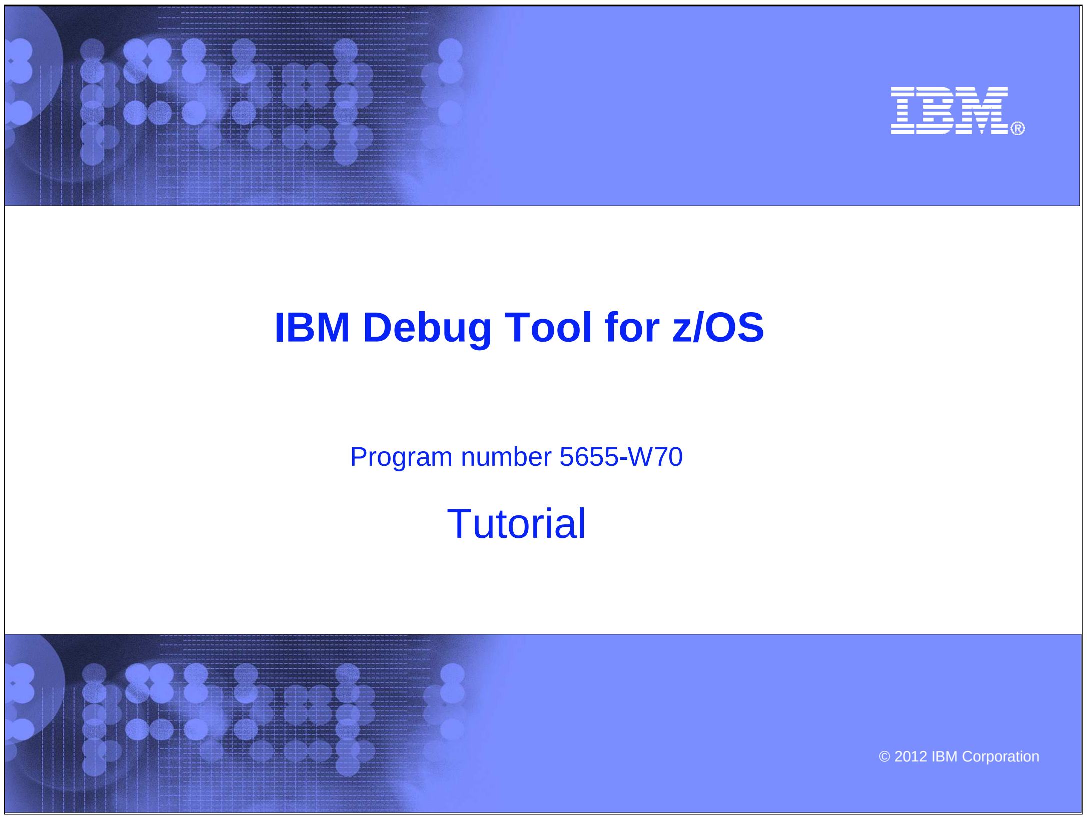
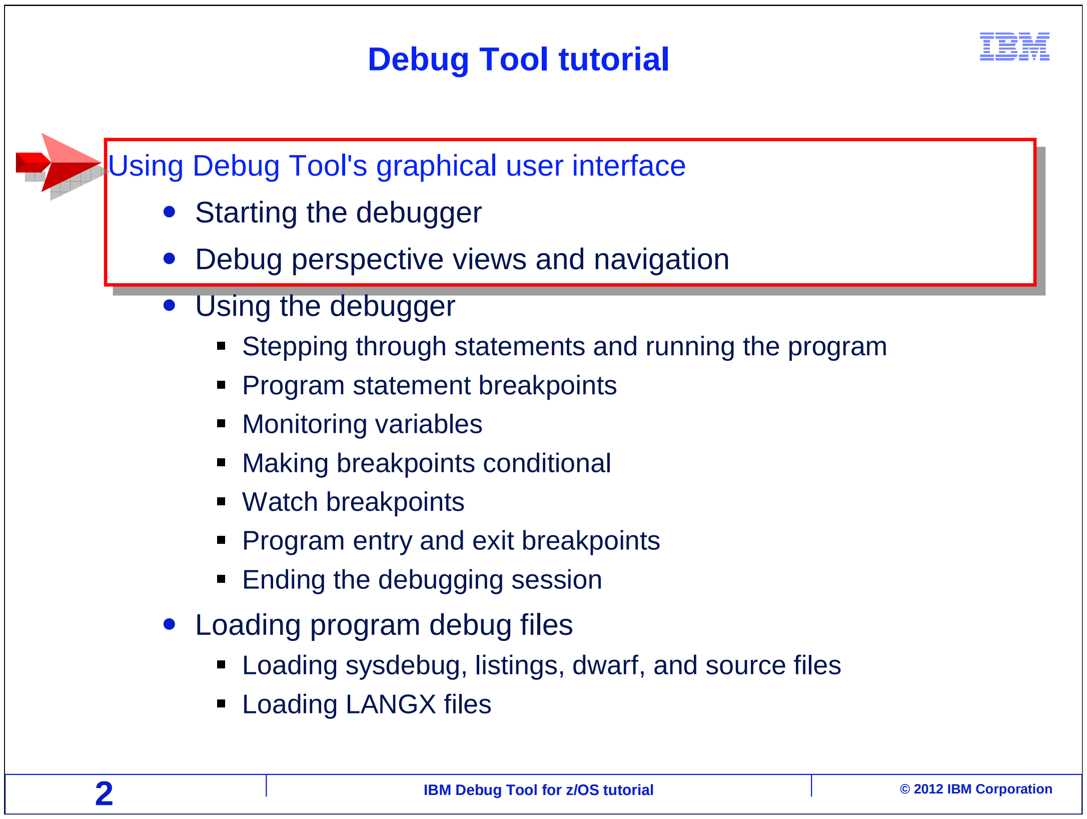


IBM Debug Tool for z/OS

Program number 5655-W70

Tutorial

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

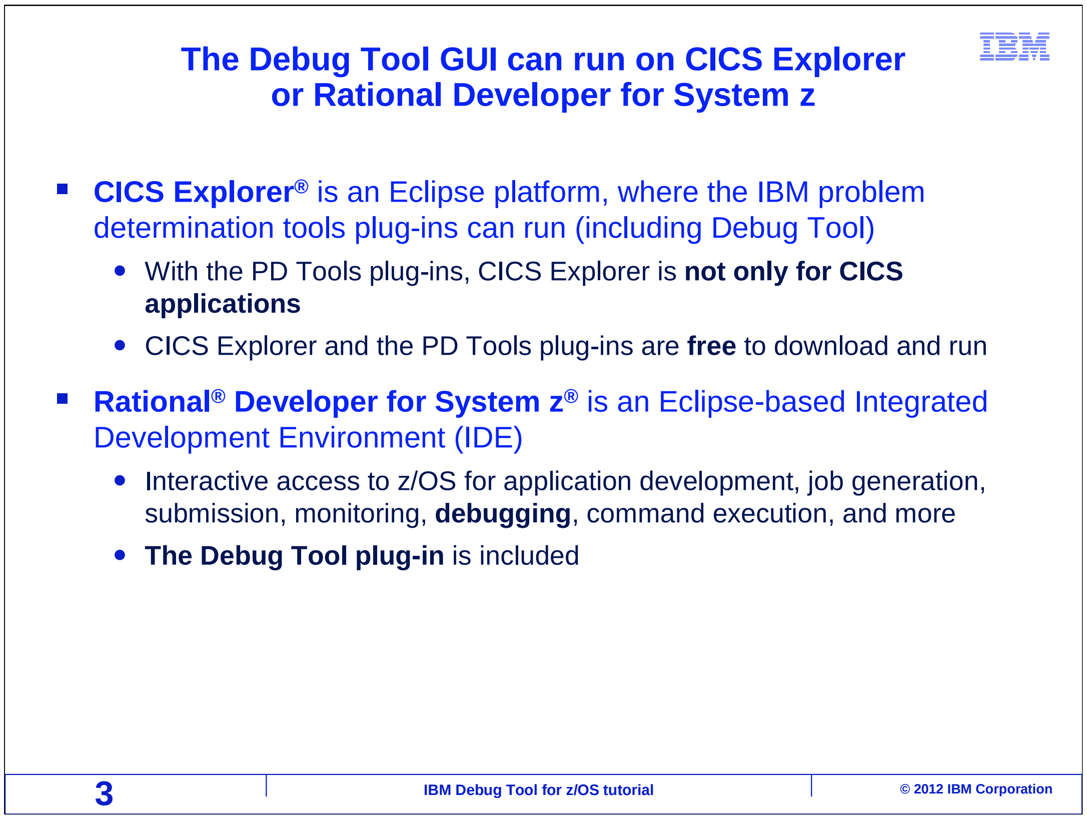


Debug Tool tutorial

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

This section introduces you to Debug Tool's graphical user interface. You will see an overview of starting the debugger, learn about the windows shown in the interface, and how to navigate them.


The Debug Tool GUI can run on CICS Explorer or Rational Developer for System z

- **CICS Explorer®** is an Eclipse platform, where the IBM problem determination tools plug-ins can run (including Debug Tool)
  - With the PD Tools plug-ins, CICS Explorer is **not only for CICS applications**
  - CICS Explorer and the PD Tools plug-ins are **free** to download and run
- **Rational® Developer for System z®** is an Eclipse-based Integrated Development Environment (IDE)
  - Interactive access to z/OS for application development, job generation, submission, monitoring, **debugging**, command execution, and more
  - **The Debug Tool plug-in** is included

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

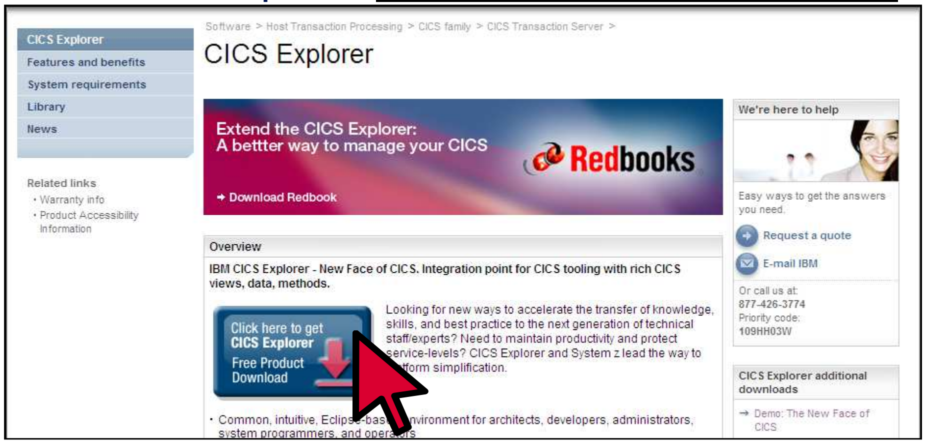
The Debug Tool plug-in for eclipse enables debugging of C, C++, COBOL, PL/I, and assembler programs. The program being debugged runs on the host z/OS system, but the debugger is displayed on your workstation.

The GUI debugger is an eclipse plug-in, so it needs an eclipse platform to run. Eclipse is a popular integrated development environment standard, and there are several eclipse platforms available. CICS Explorer is an eclipse platform where the Debug Tool plug-in can run. It is a good choice as the eclipse platform, since it can be downloaded from the IBM web site and used at no charge, and is maintained and supported. By the way, do not let the name fool you. It is not just for CICS applications. With the plug-in you can debug any program supported by Debug Tool, including batch, CICS, IMS™, DB2®, and others.

Rational Developer for System z is a complete integrated development environment, providing advanced features for z/OS developers. The debug tool plug-in in included.

Download CICS Explorer and the plug-ins from the IBM web site

Download CICS Explorer at www.ibm.com/software/htp/cics/explorer


Software > Host Transaction Processing > CICS family > CICS Transaction Server >

CICS Explorer

- CICS Explorer
- Features and benefits
- System requirements
- Library
- News

Related links
- Warranty info
- Product Accessibility Information

Extend the CICS Explorer: A bettter way to manage your CICS

Redbooks

→ Download Redbook

We're here to help

Easy ways to get the answers you need.

Request a quote

E-mail IBM

Or call us at:
877-426-3774
Priority code:
109HH03W

Overview

IBM CICS Explorer - New Face of CICS. Integration point for CICS tooling with rich CICS views, data, methods.

Click here to get CICS Explorer Free Product Download

Looking for new ways to accelerate the transfer of knowledge, skills, and best practice to the next generation of technical staff/experts? Need to maintain productivity and protect service-levels? CICS Explorer and System z lead the way to platform simplification.

CICS Explorer additional downloads

→ Demo: The New Face of CICS

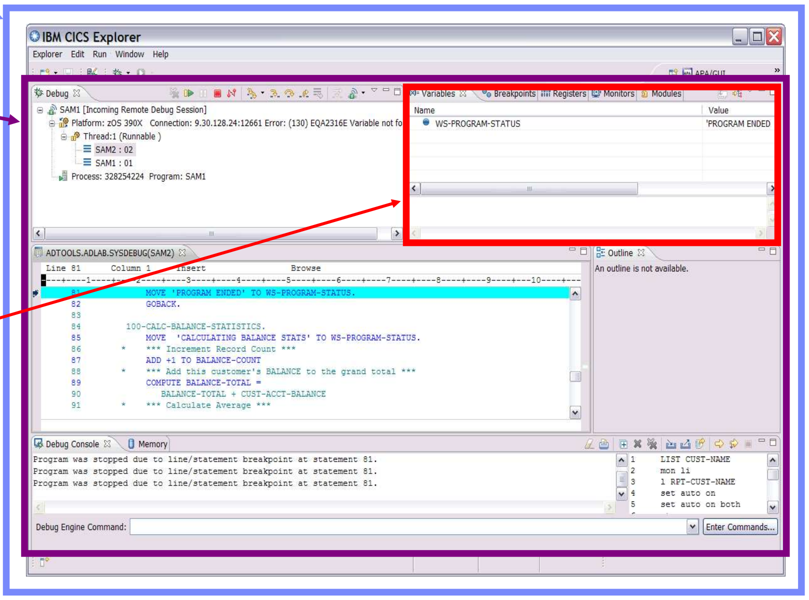
- Common, intuitive, Eclipse-based environment for architects, developers, administrators, system programmers, and operators

Then download the PD Tools plug-ins at www.ibm.com/software/awdtools/deployment/pdtplugin

CICS Explorer can be downloaded at no charge from the IBM web site. Go to www.ibm.com/software/htp/cics/explorer, click the download button, and follow the instructions to install it. Then download and install the IBM PD Tools plug-ins, including the Debug Tool plug-in, at www.ibm.com/software/awdtools/deployment/pdtplugins.

Definition of terms
for the Debug Tool GUI

- **Workbench**
  - The desktop environmen
- **Perspective**
  - A set of functions an capabilities
  - The Debug Tool GUI is perspective
- **View**
  - A window within a perspective providing a specific capability



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12661 Error: (130) EQA2316E Variable not fo
    Thread:1 (Runnable )
      SAM2 : 02
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
Line 81      Column 1      Insert                Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---
    81              MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
    82              GOBACK.
    83
    84          100-CALC-BALANCE-STATISTICS.
    85              MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
    86         *    *** Increment Record Count ***
    87              ADD +1 TO BALANCE-COUNT
    88         *    *** Add this customer's BALANCE to the grand total ***
    89              COMPUTE BALANCE-TOTAL =
    90                 BALANCE-TOTAL + CUST-ACCT-BALANCE
    91         *    *** Calculate Average ***
```

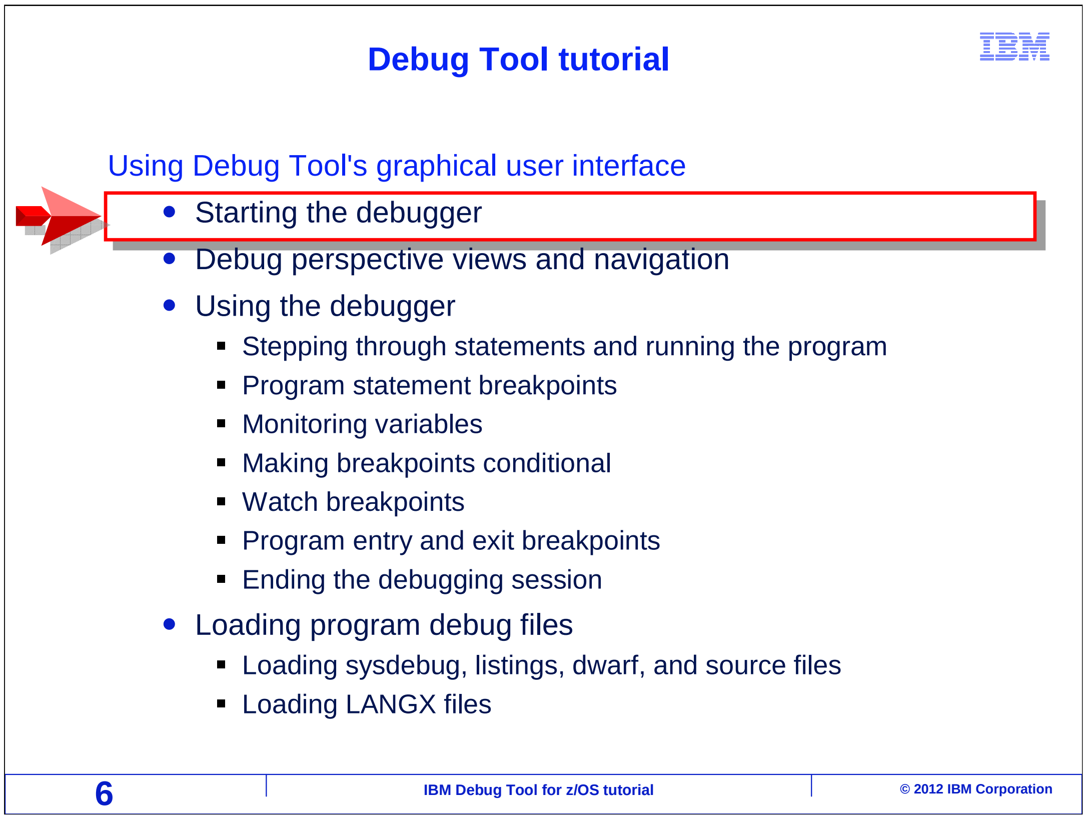
```
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
```

```
1      LIST CUST-NAME
2      mon li
3      1 RPT-CUST-NAME
4      set auto on
5      set auto on both
```

Here are a few eclipse terms you will be hearing in this tutorial. First, the workbench is what you see when you start the eclipse platform. It consists of *perspectives*.

A perspective provides a set of functions and capabilities. The debugging plug-in is an example of a perspective. It contains one or more views.

A view is a window in a perspective that provides a particular capability, such as an editor or status window.



Debug Tool tutorial

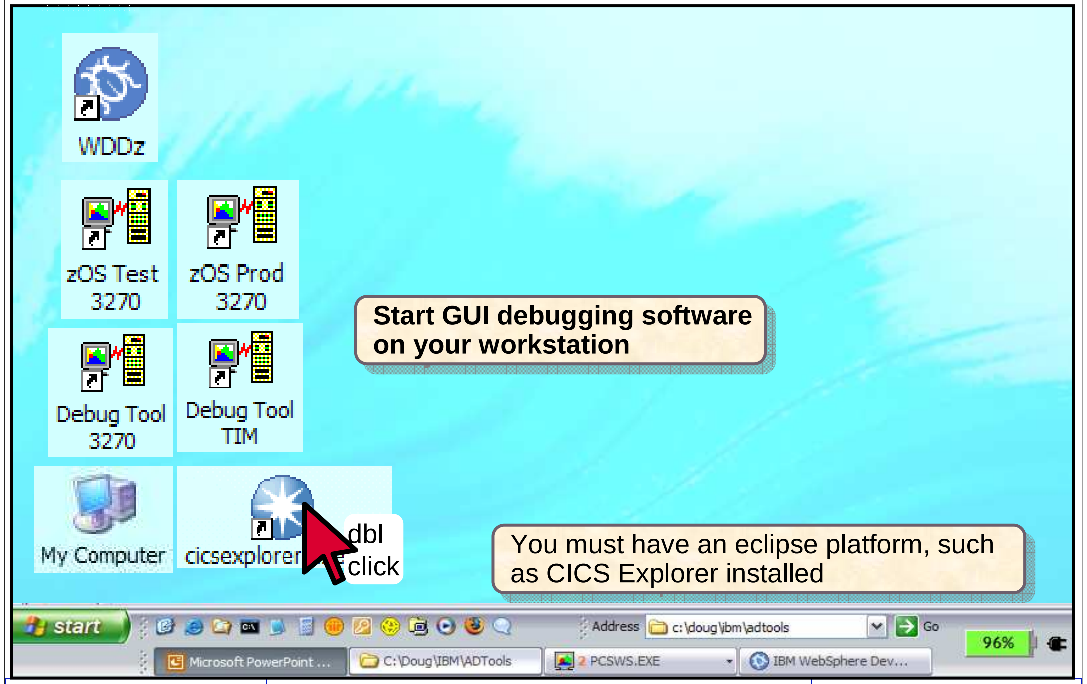
Using Debug Tool's graphical user interface

- 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

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

Next, you will see an overview of starting the GUI debugger.

Starting the remote graphical user interface

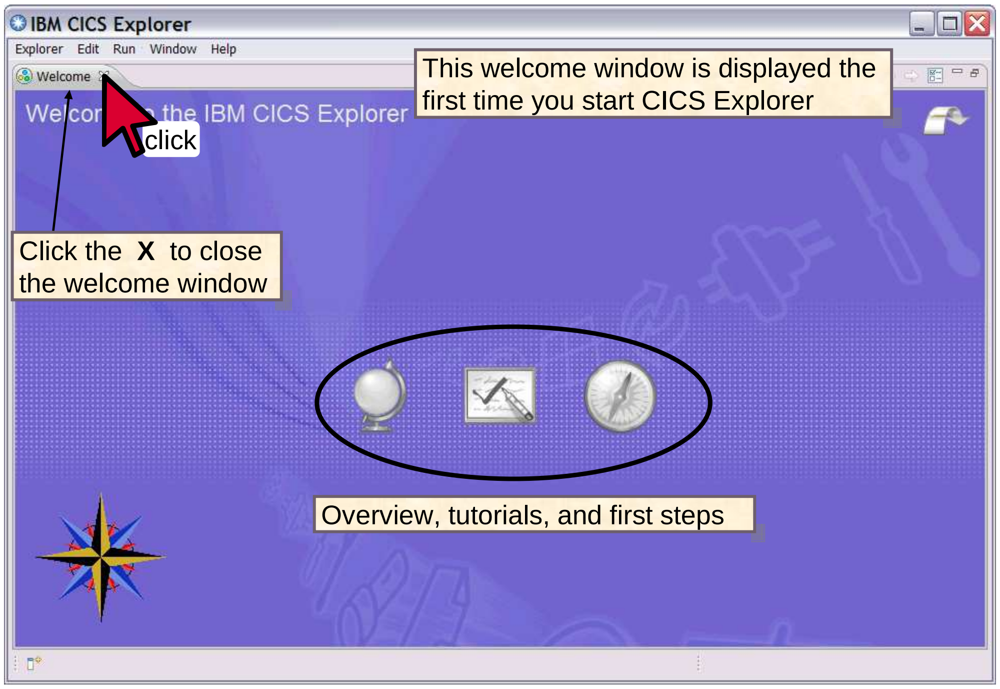


Start GUI debugging software on your workstation

You must have an eclipse platform, such as CICS Explorer installed

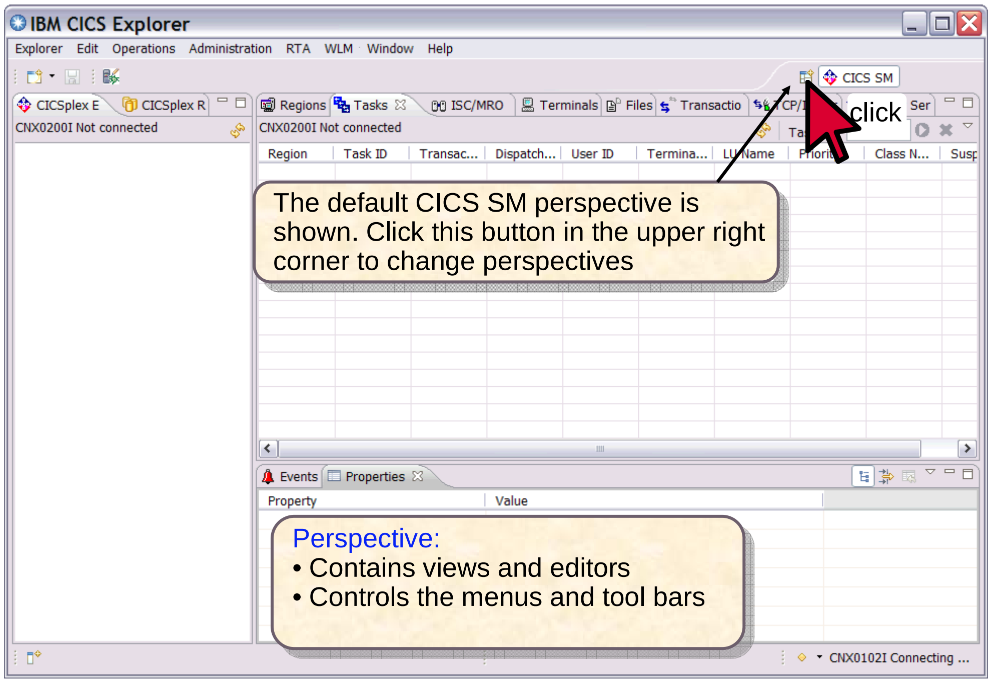
First, an eclipse platform such as CICS Explorer with the Debug Tool plug-in must be installed on your work station. Double click the icon to start it.

CICS Explorer welcome panel


IBM CICS Explorer
Explorer Edit Run Window Help
Welcome
Welcome to the IBM CICS Explorer
click
This welcome window is displayed the first time you start CICS Explorer
Click the X to close the welcome window
Overview, tutorials, and first steps

The first time you start CICS Explorer, the workbench opens and the welcome view is displayed. There are icons for items such as the overview, tutorials, and the "first steps" guide. Close the welcome view by clicking the “x” on its tab. You can re-open it later if needed by clicking **Help > Welcome** from the menu.

Default CICS Explorer perspective

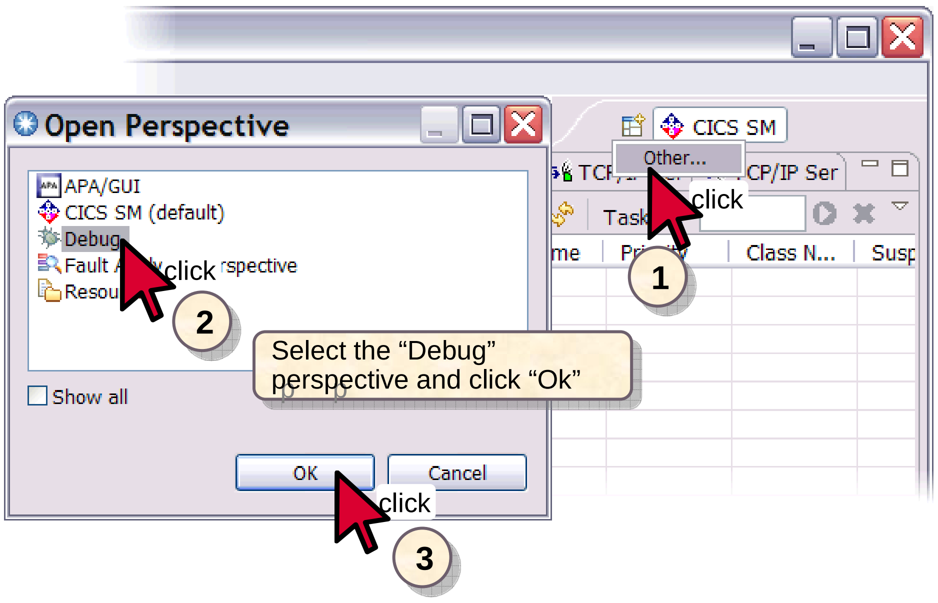


The default CICS SM perspective is shown. Click this button in the upper right corner to change perspectives

Perspective:
- Contains views and editors
- Controls the menus and tool bars

In CICS Explorer, the default perspective is CICS SM. A perspective contains associated views and editors. Switch to the Debug Tool perspective. There are several ways to change perspectives. In this example the “Open Perspective” button is clicked.

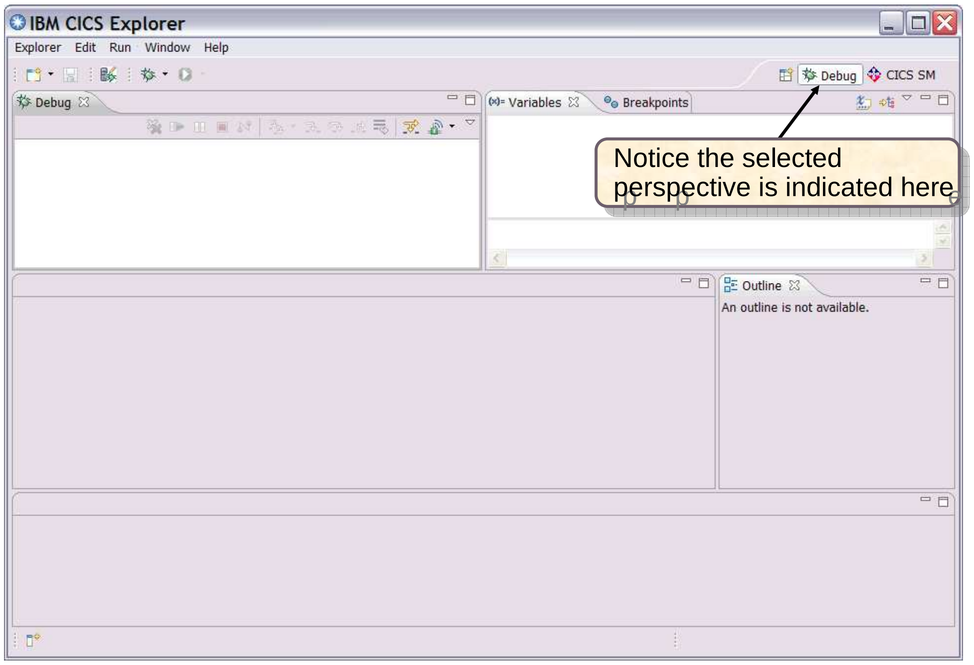
Switching to the Debug perspective



Select the "Debug" perspective and click "Ok"

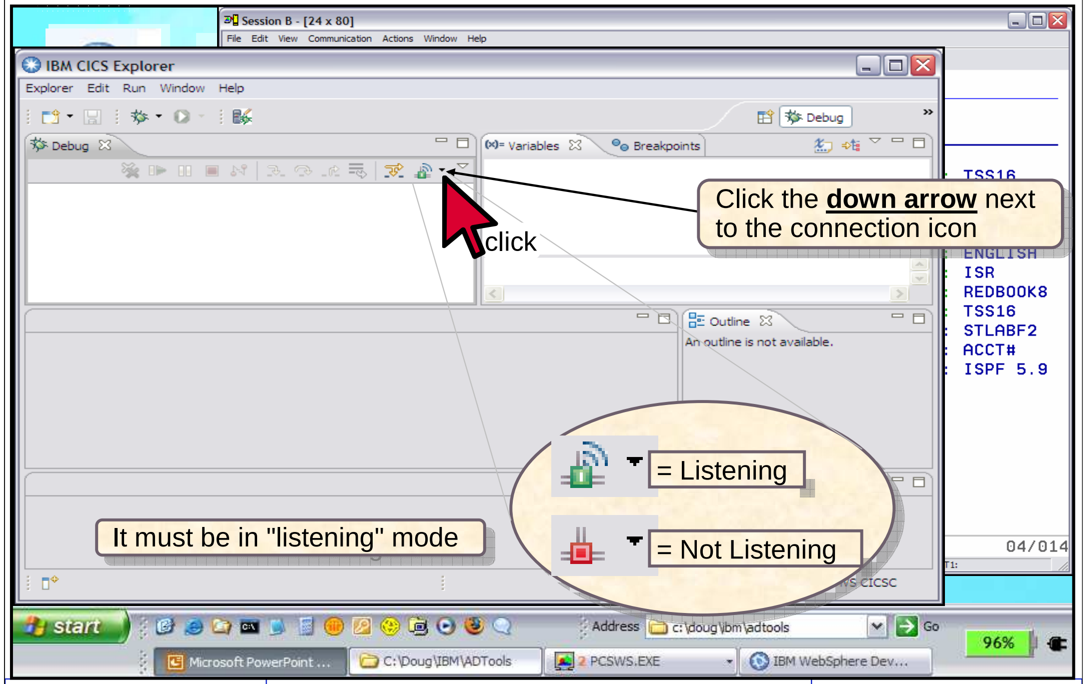
After clicking the "Open Perspective" button, click "Other…". A pop-up window lists the available perspectives. Select "Debug" to open the Debug perspective, and click OK.

The Debug perspective


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
CICS SM
Debug
Variables
Breakpoints
Outline
An outline is not available.
Notice the selected perspective is indicated here

The Debug perspective is opened.

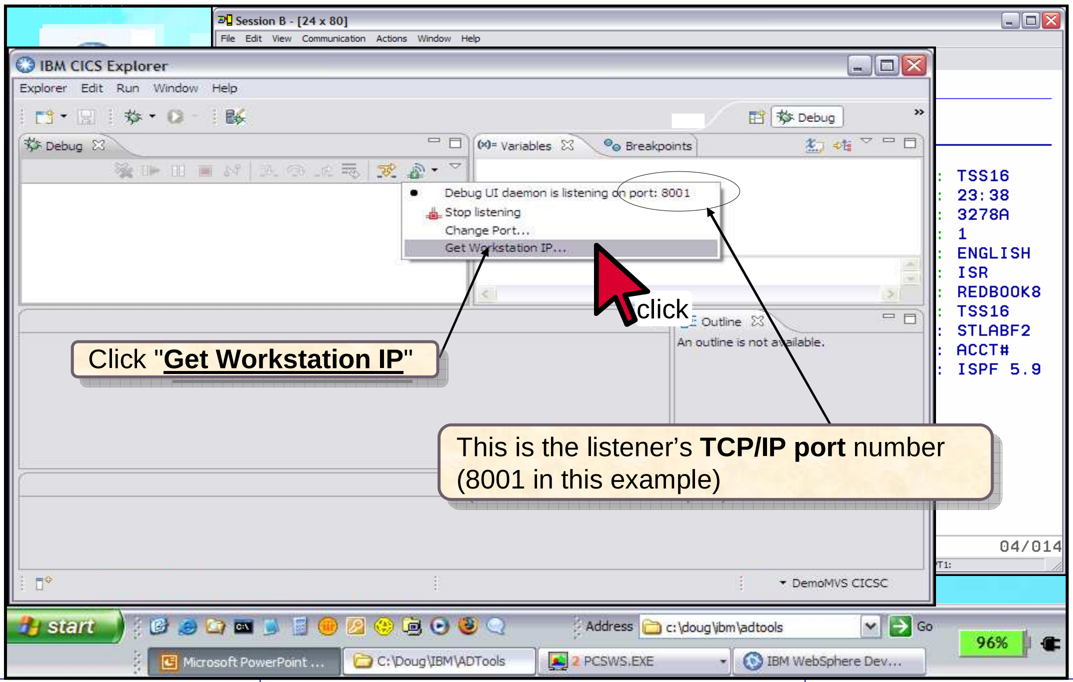
To get the workstation TCP/IP address, click the down arrow next to the connection icon



The debugger must be in “listening” mode. There is an icon on the toolbar that shows the listening status. If it is not listening, the listener icon is red, and you can click the icon to switch it on. The icon is green when the listener is on.

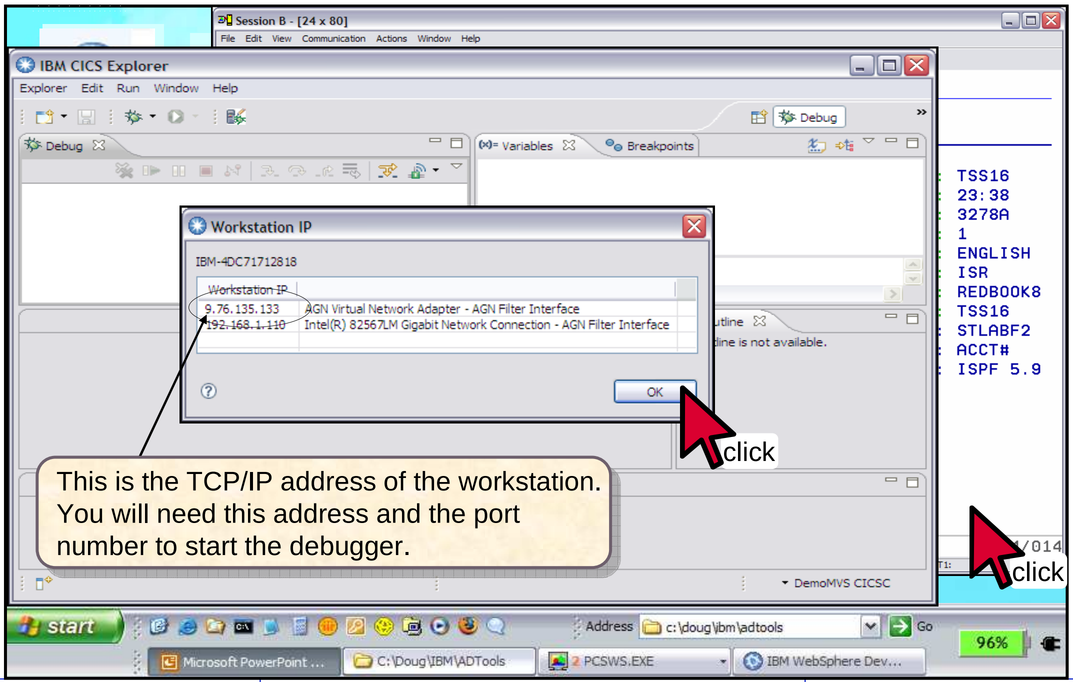
You need to determine your workstation's IP address. Click the small black down arrow next to the listening icon.

Note the port number, and click Get Workstation IP…


Debug UI daemon is listening on port: 8001
Stop listening
Change Port…
Get Workstation IP…
click
Click "Get Workstation IP"
This is the listener's TCP/IP port number (8001 in this example)

That displays a menu. Make a note of the listener's IP port. In this example, it is 8001. Then, click "Get Workstation IP".

Note the TCP/IP address of your workstation


Workstation IP

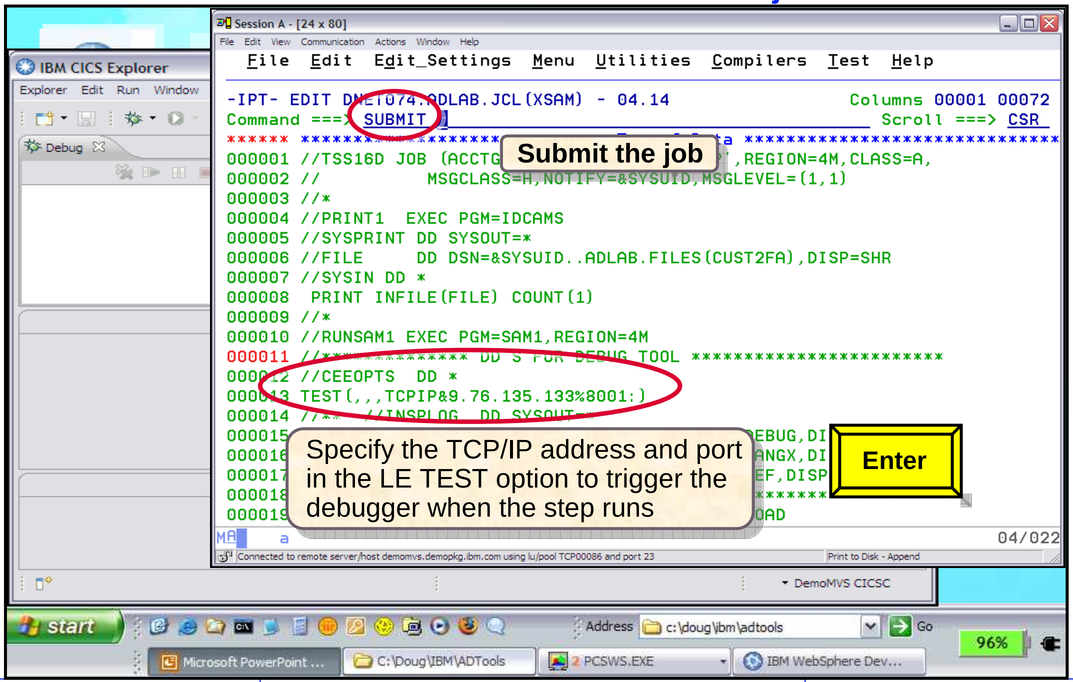
IBM-4DC71712818

Workstation IP	
9.76.135.133	AGN Virtual Network Adapter - AGN Filter Interface
~~192.168.1.110~~	Intel(R) 82567LM Gigabit Network Connection - AGN Filter Interface

This is the TCP/IP address of the workstation. You will need this address and the port number to start the debugger.

Make a note of your workstation’s IP address. At this point, the GUI debugger is ready to receive a debugging session. Click OK to close the pop-up window. Next, a terminal session is displayed.

Batch example: set a debugging trigger, and submit the JCL to run the batch job

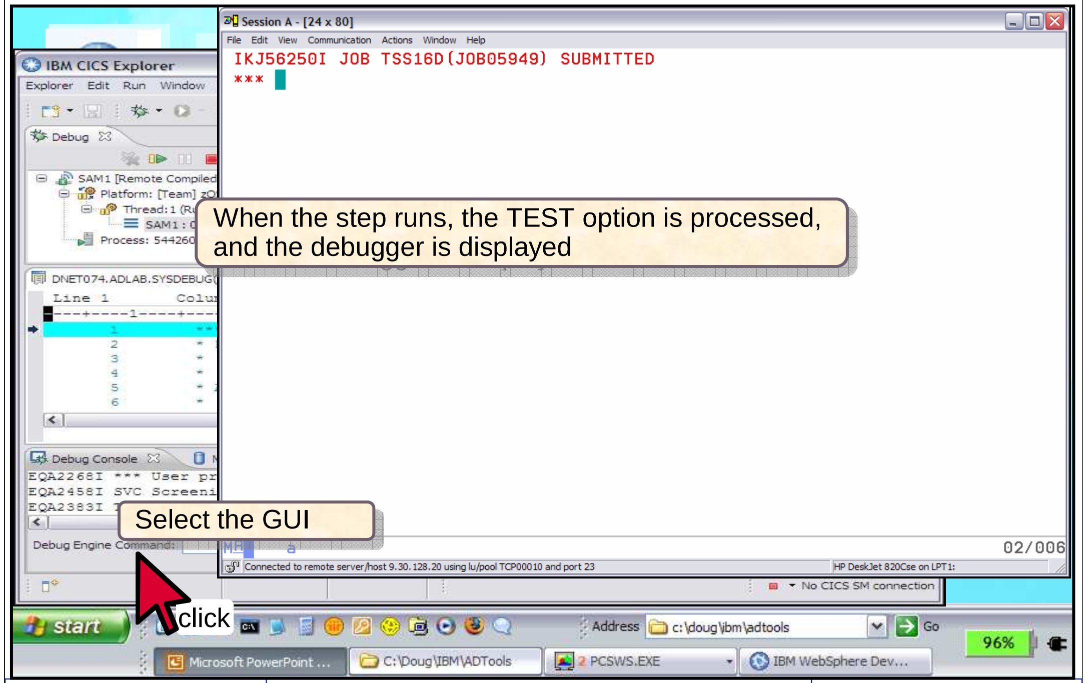

Submit the job
Specify the TCP/IP address and port in the LE TEST option to trigger the debugger when the step runs
Enter
IBM CICS Explorer
Debug
DemoMVS CICSC

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

-IPT- EDIT DNET074.ADLAB.JCL(XSAM) - 04.14                Columns 00001 00072
Command ===> SUBMIT                                           Scroll ===> CSR
****** *****************                 ta *******************************
000001 //TSS16D JOB (ACCTG                    ,REGION=4M,CLASS=A,
000002 //          MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
000003 //*
000004 //PRINT1  EXEC PGM=IDCAMS
000005 //SYSPRINT DD SYSOUT=*
000006 //FILE     DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000007 //SYSIN DD *
000008  PRINT INFILE(FILE) COUNT(1)
000009 //*
000010 //RUNSAM1 EXEC PGM=SAM1,REGION=4M
000011 //*************** DD S FOR DEBUG TOOL ************************
000012 //CEEOPTS  DD *
000013 TEST(,,,TCPIP&9.76.135.133%8001:)
000014 //**   //INSPLOG  DD SYSOUT=*
000015                                            EBUG,DI
000016                                            ANGX,DI
000017                                            EF,DISP
000018                                            *******
000019                                            OAD
MA      a                                                              04/022
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00086 and port 23    Print to Disk - Append
```

In this example, a batch program will be debugged. In TSO, which is accessed in it's own terminal session, edit the JCL used to run the program. One easy way to debug a batch program is to code a CEEOPTS DD statement. In the CEEOPTS DD, a TEST option is coded. Specify your workstation's TCP address and port number in the TEST option. The JCL is submitted to batch.

View other sections of the tutorial to learn how to start a remote debugging session. There are different tutorial sections that describe starting the debugger for programs running in various environments such as batch, CICS, and others.

When the job step runs, the debugger is triggered



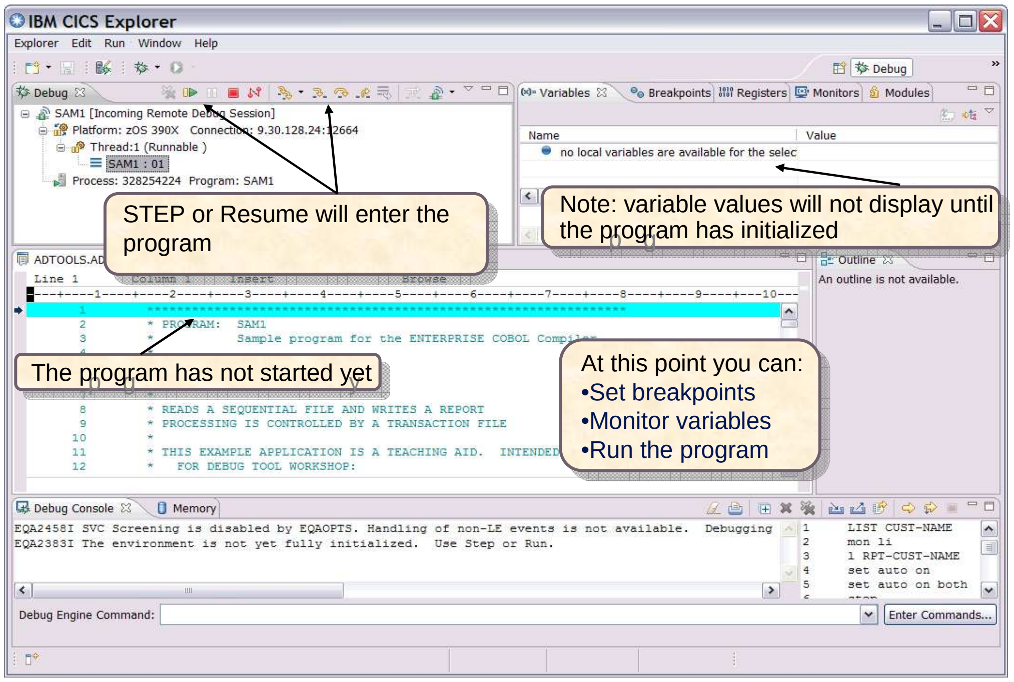
```
IKJ56250I JOB TSS16D(JOB05949) SUBMITTED
***
```

When the step runs, the TEST option is processed, and the debugger is displayed

Select the GUI

When the job step with the TEST option runs, the debugger starts and is displayed by the GUI debugging software. Select the debugger window.

Active debugging session



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12664
    Thread:1 (Runnable )
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

no local variables are available for the selec

An outline is not available.

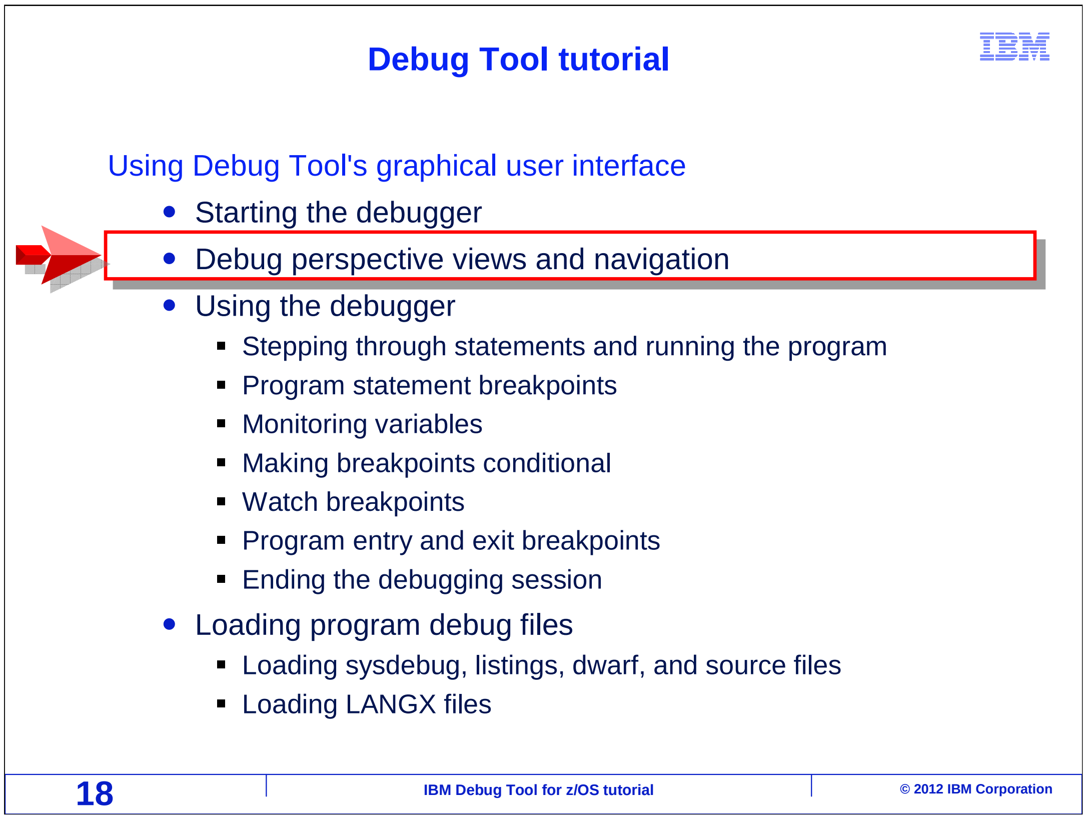
```
Line 1      Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10---
           *****************************************************************
           * PROGRAM:   SAM1
           *            Sample program for the ENTERPRISE COBOL Compiler
           * READS A SEQUENTIAL FILE AND WRITES A REPORT
           * PROCESSING IS CONTROLLED BY A TRANSACTION FILE
           *
           * THIS EXAMPLE APPLICATION IS A TEACHING AID.  INTENDED
           *   FOR DEBUG TOOL WORKSHOP:
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging
EQA2383I The environment is not yet fully initialized.  Use Step or Run.
```

```
1      LIST CUST-NAME
2      mon li
3      1 RPT-CUST-NAME
4      set auto on
5      set auto on both
```

Debug Engine Command:

This is an example of a session that just started. Most applications are paused before the initialization of the main program. The main program has not yet started. Be aware that variables do not exist until the program initializes, which happens automatically when you run or step into the program. At this point, you can set breakpoints, add variables to the monitor, change settings, and run the program.



Debug Tool tutorial

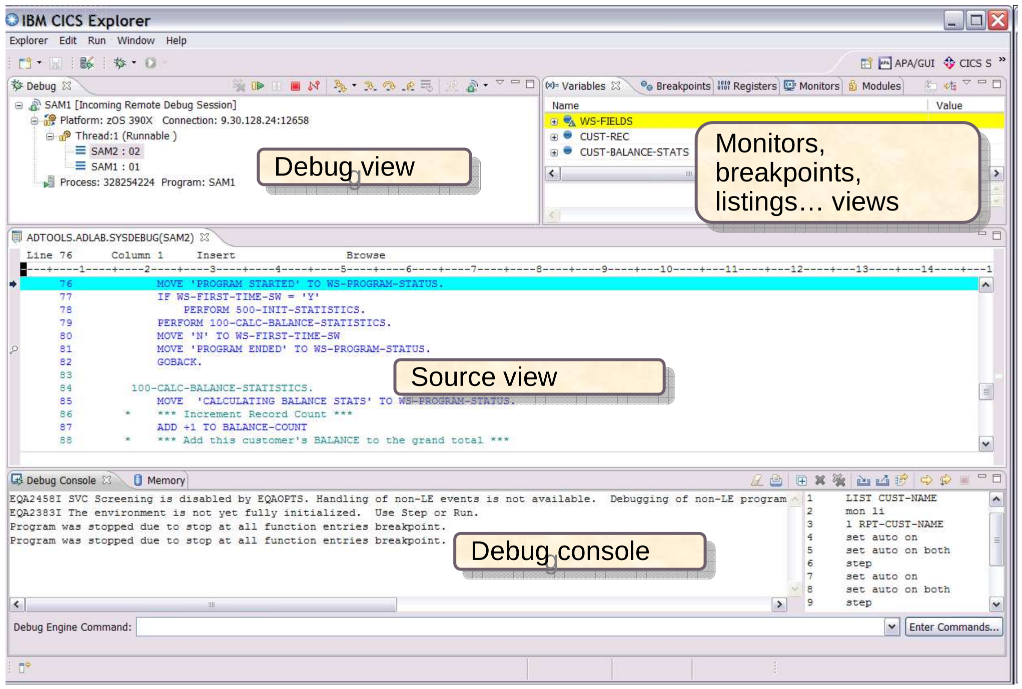
Using Debug Tool's graphical user interface

- 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

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

Next you will learn about the various views shown in the debug perspective.

Debug perspective



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12658
    Thread:1 (Runnable )
      SAM2 : 02
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

```
WS-FIELDS
CUST-REC
CUST-BALANCE-STATS
```

ADTOOLS.ADLAB.SYSDEBUG(SAM2)

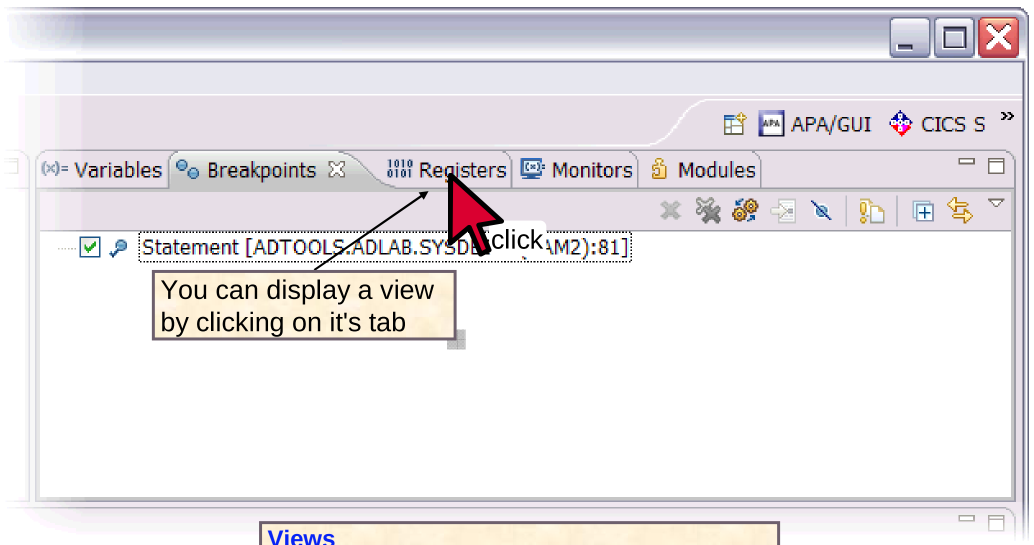
```
Line 76      Column 1      Insert                   Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---11----+---12----+---13----+---14----+---1
     76            MOVE 'PROGRAM STARTED' TO WS-PROGRAM-STATUS.
     77            IF WS-FIRST-TIME-SW = 'Y'
     78                PERFORM 500-INIT-STATISTICS.
     79            PERFORM 100-CALC-BALANCE-STATISTICS.
     80            MOVE 'N' TO WS-FIRST-TIME-SW
     81            MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
     82            GOBACK.
     83
     84        100-CALC-BALANCE-STATISTICS.
     85            MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
     86       *    *** Increment Record Count ***
     87            ADD +1 TO BALANCE-COUNT
     88       *    *** Add this customer's BALANCE to the grand total ***
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging of non-LE program
EQA2383I The environment is not yet fully initialized.  Use Step or Run.
Program was stopped due to stop at all function entries breakpoint.
Program was stopped due to stop at all function entries breakpoint.
```

```
1      LIST CUST-NAME
2      mon li
3      1 RPT-CUST-NAME
4      set auto on
5      set auto on both
6      step
7      set auto on
8      set auto on both
9      step
```

Notice that the debug perspective has several panes. Each pane can have one or more "views" in it, and each view has a tab.

Display a view

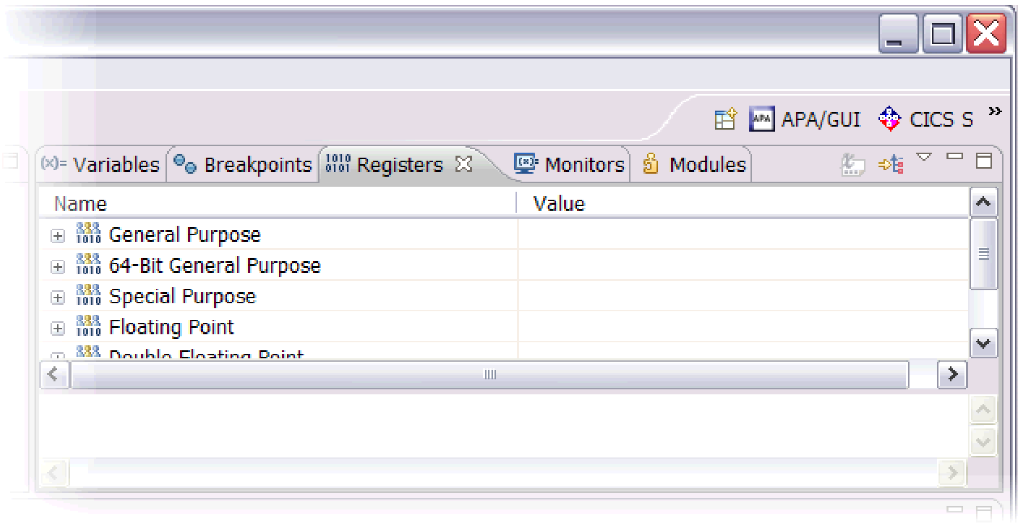


You can display a view by clicking on it's tab

Views
- Support editors and provide ways to navigate your Workbench
- Can appear by itself, or stacked with other views in a tabbed notebook

The pane in the upper right contains five views: variables, breakpoints, registers, monitors, and modules. Select a view by clicking it’s tab. Here, the breakpoints view is in the forefront, and the registers tab is clicked.

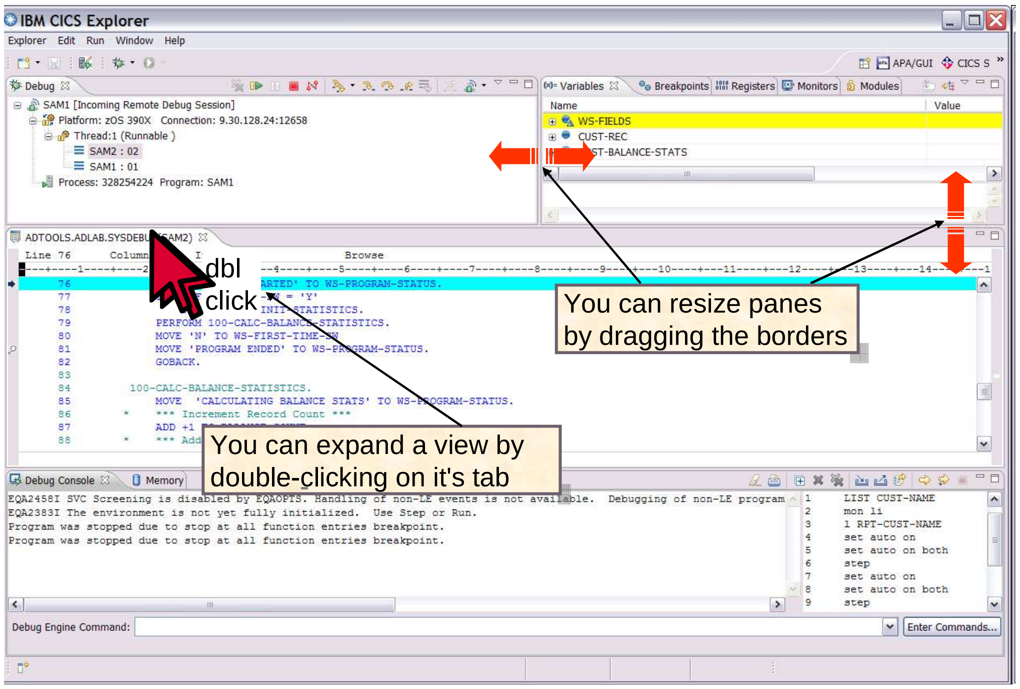
Result of clicking a view’s tab



Name	Value
General Purpose	
64-Bit General Purpose	
Special Purpose	
Floating Point	
Double Floating Point	

That brought the registers view to the front.

Resizing views


IBM CICS Explorer
Explorer Edit Run Window Help
APA/GUI
CICS S
Debug
Variables
Breakpoints
Registers
Monitors
Modules
Name
Value
WS-FIELDS
CUST-REC
ST-BALANCE-STATS
SAM1 [Incoming Remote Debug Session]
Platform: zOS 390X Connection: 9.30.128.24:12658
Thread:1 (Runnable)
SAM2 : 02
SAM1 : 01
Process: 328254224 Program: SAM1
ADTOOLS.ADLAB.SYSDEBU(SAM2)
Line 76 Column
Browse
dbl click
You can resize panes by dragging the borders
You can expand a view by double-clicking on it's tab
Debug Console
Memory
Debug Engine Command:
Enter Commands...

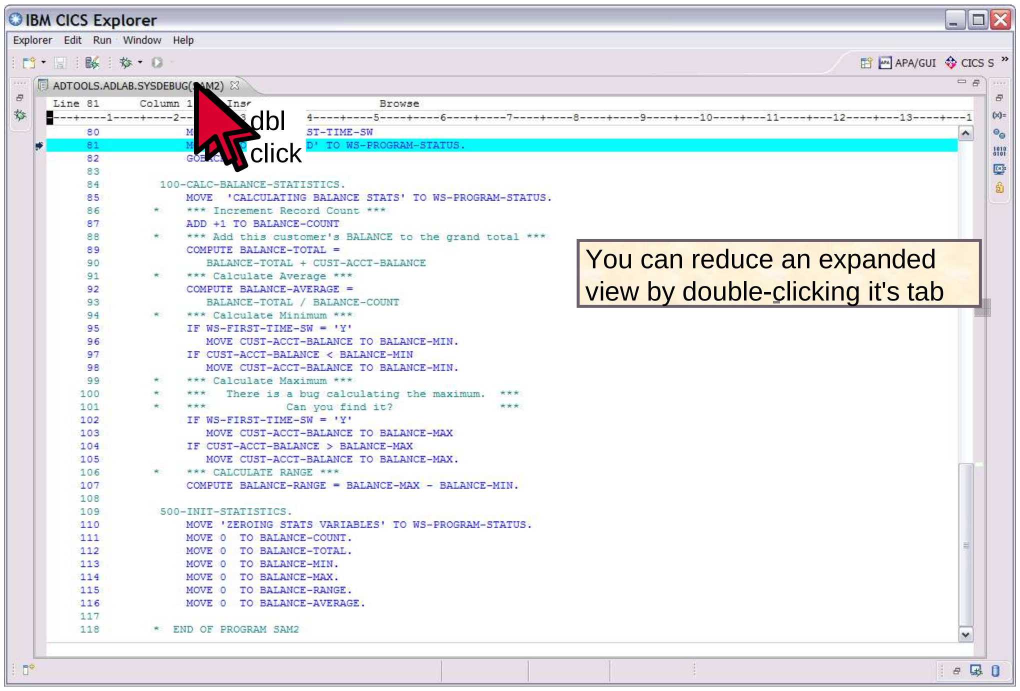
```
ARTED' TO WS-PROGRAM-STATUS.
-W = 'Y'
INIT-STATISTICS.
      PERFORM 100-CALC-BALANCE-STATISTICS.
      MOVE 'N' TO WS-FIRST-TIME-SW
      MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
      GOBACK.

  100-CALC-BALANCE-STATISTICS.
      MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
 *    *** Increment Record Count ***
      ADD +1
 *    *** Add
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging of non-LE program
EQA2383I The environment is not yet fully initialized.  Use Step or Run.
Program was stopped due to stop at all function entries breakpoint.
Program was stopped due to stop at all function entries breakpoint.
```

```
1      LIST CUST-NAME
2      mon li
3      l RPT-CUST-NAME
4      set auto on
5      set auto on both
6      step
7      set auto on
8      set auto on both
9      step
```

It is easy to change the size of a pane, just drag-and-drop it's borders. You can expand a view by double-clicking it’s tab. The tab of the source view is double-clicked.

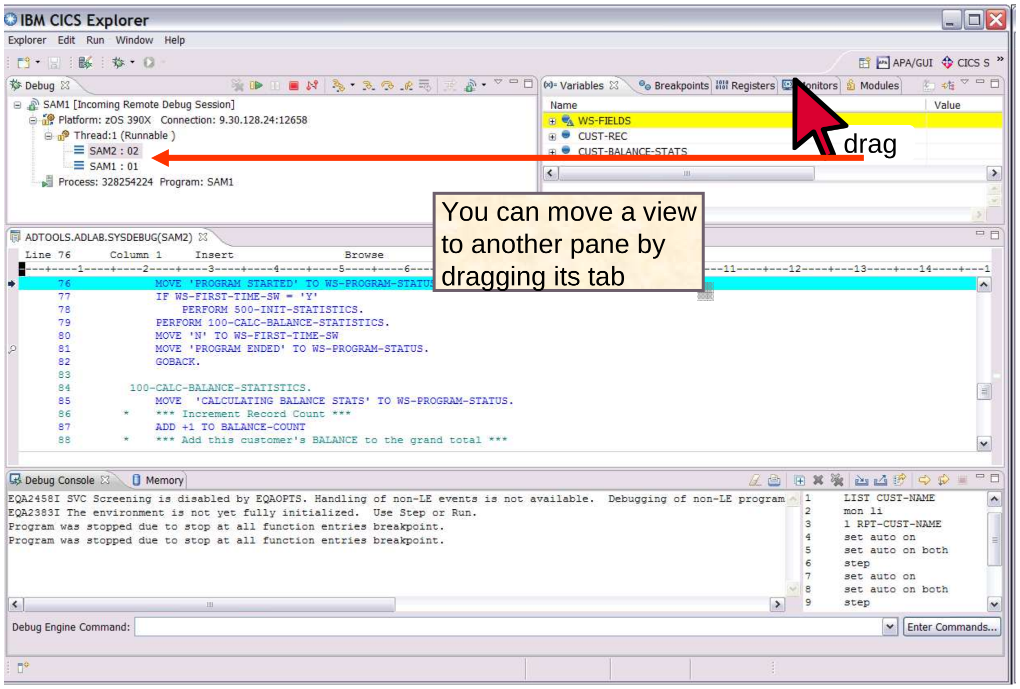
Full screen view window


IBM CICS Explorer
Explorer Edit Run Window Help
APA/GUI
CICS S
ADTOOLS.ADLAB.SYSDEBUG(SAM2)
Line 81 Column 1 Inse Browse
dbl click
You can reduce an expanded view by double-clicking it's tab

```
           M                        ST-TIME-SW
           M                        D' TO WS-PROGRAM-STATUS.
           GOBAC

       100-CALC-BALANCE-STATISTICS.
           MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
      *    *** Increment Record Count ***
           ADD +1 TO BALANCE-COUNT
      *    *** Add this customer's BALANCE to the grand total ***
           COMPUTE BALANCE-TOTAL =
              BALANCE-TOTAL + CUST-ACCT-BALANCE
      *    *** Calculate Average ***
           COMPUTE BALANCE-AVERAGE =
              BALANCE-TOTAL / BALANCE-COUNT
      *    *** Calculate Minimum ***
           IF WS-FIRST-TIME-SW = 'Y'
              MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
           IF CUST-ACCT-BALANCE < BALANCE-MIN
              MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
      *    *** Calculate Maximum ***
      *    ***   There is a bug calculating the maximum.  ***
      *    ***            Can you find it?                ***
           IF WS-FIRST-TIME-SW = 'Y'
              MOVE CUST-ACCT-BALANCE TO BALANCE-MAX
           IF CUST-ACCT-BALANCE > BALANCE-MAX
              MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
      *    *** CALCULATE RANGE ***
           COMPUTE BALANCE-RANGE = BALANCE-MAX - BALANCE-MIN.

       500-INIT-STATISTICS.
           MOVE 'ZEROING STATS VARIABLES' TO WS-PROGRAM-STATUS.
           MOVE 0  TO BALANCE-COUNT.
           MOVE 0  TO BALANCE-TOTAL.
           MOVE 0  TO BALANCE-MIN.
           MOVE 0  TO BALANCE-MAX.
           MOVE 0  TO BALANCE-RANGE.
           MOVE 0  TO BALANCE-AVERAGE.

      *  END OF PROGRAM SAM2
```

That expanded it to fill the entire workbench. Sometimes it is helpful to expand a view so you can see more information. Reduce it to it's original size by double-clicking it's tab again.

Drag a view to another pane



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12658
    Thread:1 (Runnable )
      SAM2 : 02
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

You can move a view to another pane by dragging its tab

ADTOOLS.ADLAB.SYSDEBUG(SAM2)

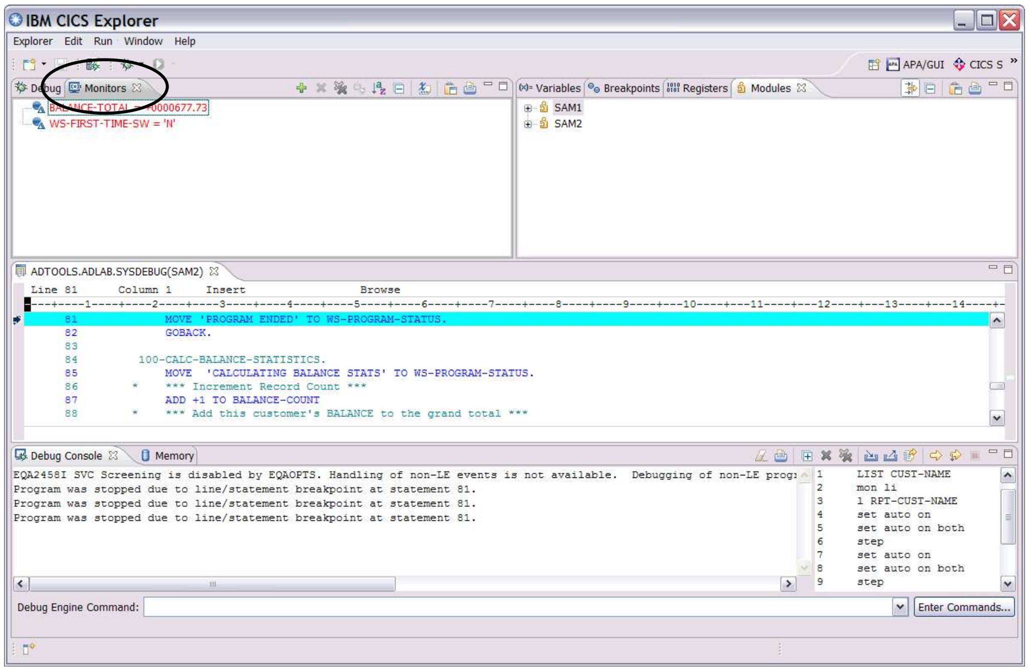
```
Line 76       Column 1      Insert                    Browse
----+----1----+----2----+----3----+----4----+----5----+----6----
     76            MOVE 'PROGRAM STARTED' TO WS-PROGRAM-STATUS
     77            IF WS-FIRST-TIME-SW = 'Y'
     78                PERFORM 500-INIT-STATISTICS.
     79            PERFORM 100-CALC-BALANCE-STATISTICS.
     80            MOVE 'N' TO WS-FIRST-TIME-SW
     81            MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
     82            GOBACK.
     83
     84        100-CALC-BALANCE-STATISTICS.
     85            MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
     86       *    *** Increment Record Count ***
     87            ADD +1 TO BALANCE-COUNT
     88       *    *** Add this customer's BALANCE to the grand total ***
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging of non-LE program
EQA2383I The environment is not yet fully initialized.  Use Step or Run.
Program was stopped due to stop at all function entries breakpoint.
Program was stopped due to stop at all function entries breakpoint.
```

```
1     LIST CUST-NAME
2     mon li
3     l RPT-CUST-NAME
4     set auto on
5     set auto on both
6     step
7     set auto on
8     set auto on both
9     step
```

You can move a view by dragging and dropping it's tab into a different pane. Here, the monitors view is moved.

Relocated tab



```
BALANCE-TOTAL = +0000677.73
WS-FIRST-TIME-SW = 'N'
```

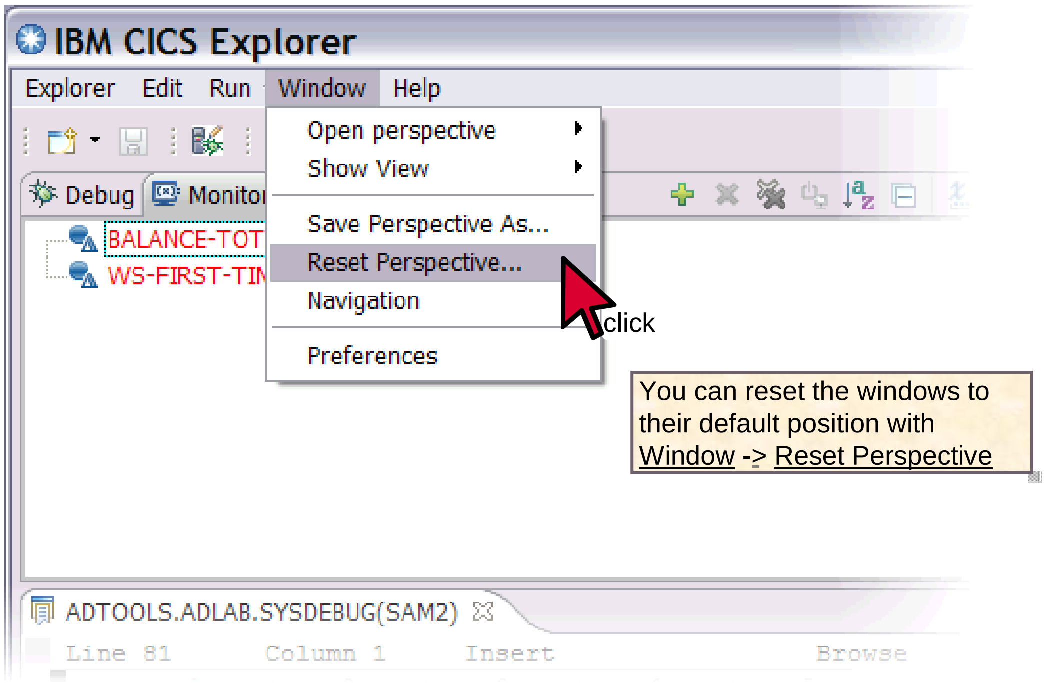
```
Line 81      Column 1     Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---11----+---12----+---13----+---14----+-
   81            MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
   82            GOBACK.
   83
   84        100-CALC-BALANCE-STATISTICS.
   85            MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
   86       *    *** Increment Record Count ***
   87            ADD +1 TO BALANCE-COUNT
   88       *    *** Add this customer's BALANCE to the grand total ***
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging of non-LE progı
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
```

```
1      LIST CUST-NAME
2      mon li
3      1 RPT-CUST-NAME
4      set auto on
5      set auto on both
6      step
7      set auto on
8      set auto on both
9      step
```

Now the monitors view is located in the same pane as the debug view. This lets you control which views are visible at the same time.

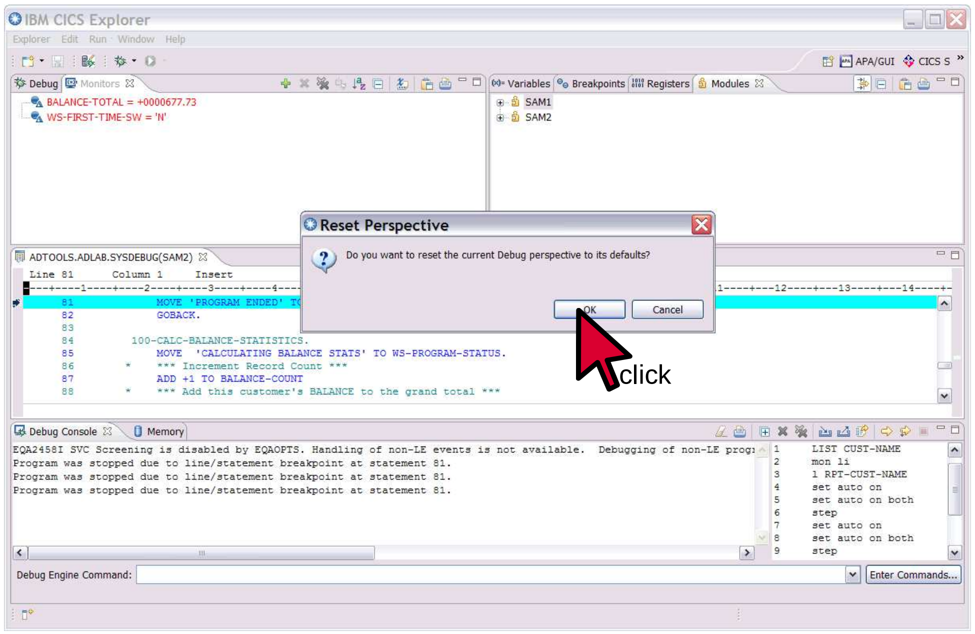
Restore the default view configuration



You can reset the windows to their default position with Window -> Reset Perspective

Do not be too concerned about moving, resizing, or closing views. It is easy to get them back to their original places. To reset the views click the “window” menu, and select “reset perspective”.

Reset perspective result



BALANCE-TOTAL = +0000677.73
WS-FIRST-TIME-SW = 'N'

Do you want to reset the current Debug perspective to its defaults?

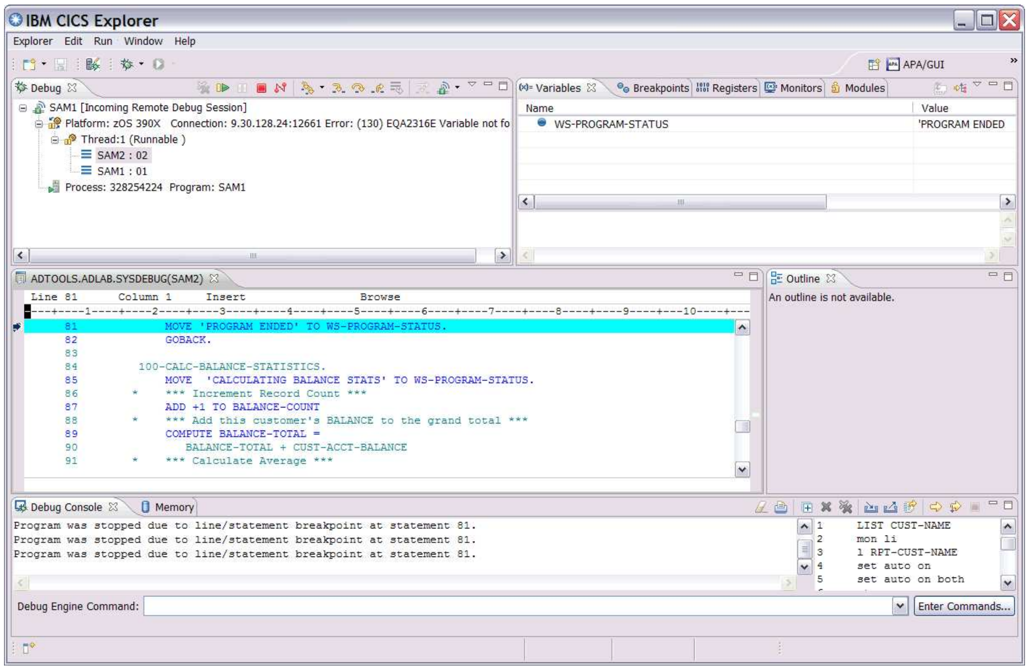
```
 Line 81       Column 1      Insert
      81             MOVE 'PROGRAM ENDED' TO
      82             GOBACK.
      83
      84         100-CALC-BALANCE-STATISTICS.
      85             MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
      86        *    *** Increment Record Count ***
      87             ADD +1 TO BALANCE-COUNT
      88        *    *** Add this customer's BALANCE to the grand total ***
```

```
EQA2458I SVC Screening is disabled by EQAOPTS. Handling of non-LE events is not available.  Debugging of non-LE prog
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
Program was stopped due to line/statement breakpoint at statement 81.
```

```
1      LIST CUST-NAME
2      mon li
3      l RPT-CUST-NAME
4      set auto on
5      set auto on both
6      step
7      set auto on
8      set auto on both
9      step
```

Then click “Ok" in the “reset perspective” pop-up.

Reset perspective result



```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:12661 Error: (130) EQA2316E Variable not fo
    Thread:1 (Runnable )
      SAM2 : 02
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

Name	Value
WS-PROGRAM-STATUS	'PROGRAM ENDED

ADTOOLS.ADLAB.SYSDEBUG(SAM2)

```
Line 81       Column 1      Insert                  Browse
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+---10----+---
      81             MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
      82             GOBACK.
      83
      84         100-CALC-BALANCE-STATISTICS.
      85             MOVE  'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
      86        *    *** Increment Record Count ***
      87             ADD +1 TO BALANCE-COUNT
      88        *    *** Add this customer's BALANCE to the grand total ***
      89             COMPUTE BALANCE-TOTAL =
      90                BALANCE-TOTAL + CUST-ACCT-BALANCE
      91        *    *** Calculate Average ***
```

An outline is not available.

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

```
1      LIST CUST-NAME
2      mon li
3      1 RPT-CUST-NAME
4      set auto on
5      set auto on both
```

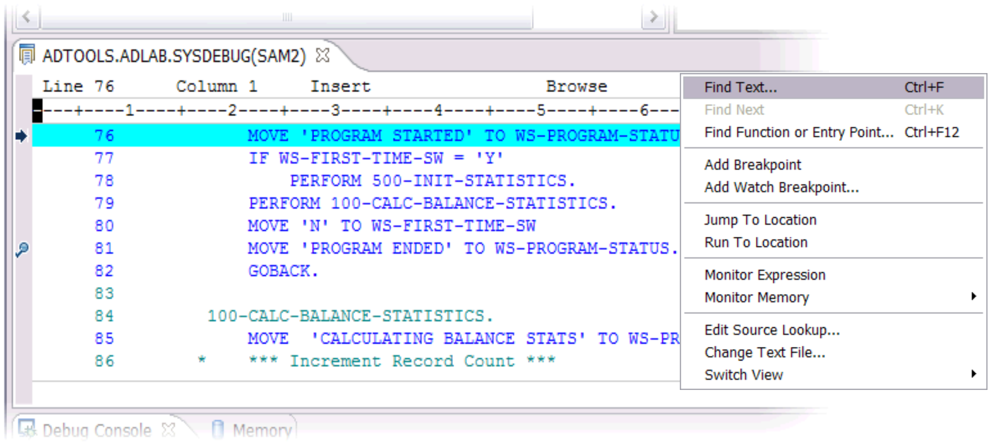
Debug Engine Command:

The debug perspective was reset, with the default views shown and put back in their original positions.

The Source view

Source view description

- Displays program source
- The next statement that will run is highlighted
- Set and remove statement breakpoints
- Right clicking reveals view specific actions



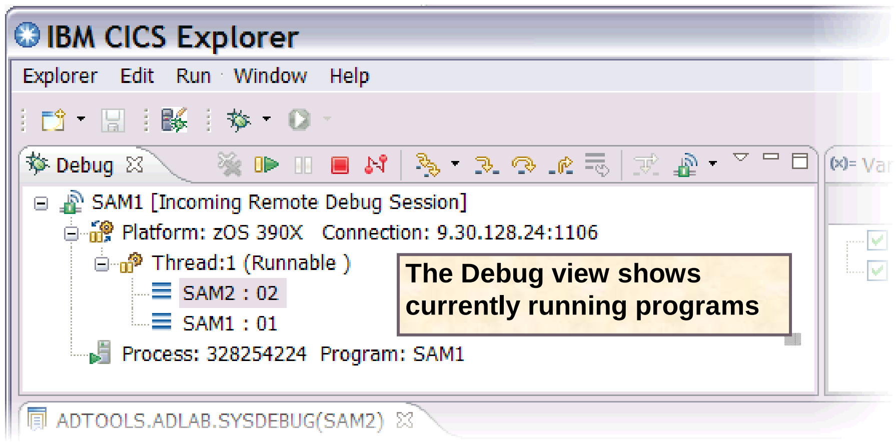
```
ADTOOLS.ADLAB.SYSDEBUG(SAM2)
Line 76       Column 1      Insert                       Browse
----+----1----+----2----+----3----+----4----+----5----+----6---
    76             MOVE 'PROGRAM STARTED' TO WS-PROGRAM-STATU
    77             IF WS-FIRST-TIME-SW = 'Y'
    78                 PERFORM 500-INIT-STATISTICS.
    79             PERFORM 100-CALC-BALANCE-STATISTICS.
    80             MOVE 'N' TO WS-FIRST-TIME-SW
    81             MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
    82             GOBACK.
    83
    84         100-CALC-BALANCE-STATISTICS.
    85             MOVE  'CALCULATING BALANCE STATS' TO WS-PR
    86        *    *** Increment Record Count ***
```

The source view displays the program. Notice that one of the statements is highlighted. That is the current statement, which is the next statement that will run. The grey area to the left of the program statements can be used to set or remove statement breakpoints. Right clicking in the source view displays an action menu, which can be used to add breakpoints, jump or run to a statement, and take other actions.

The Debug view

Debug view description

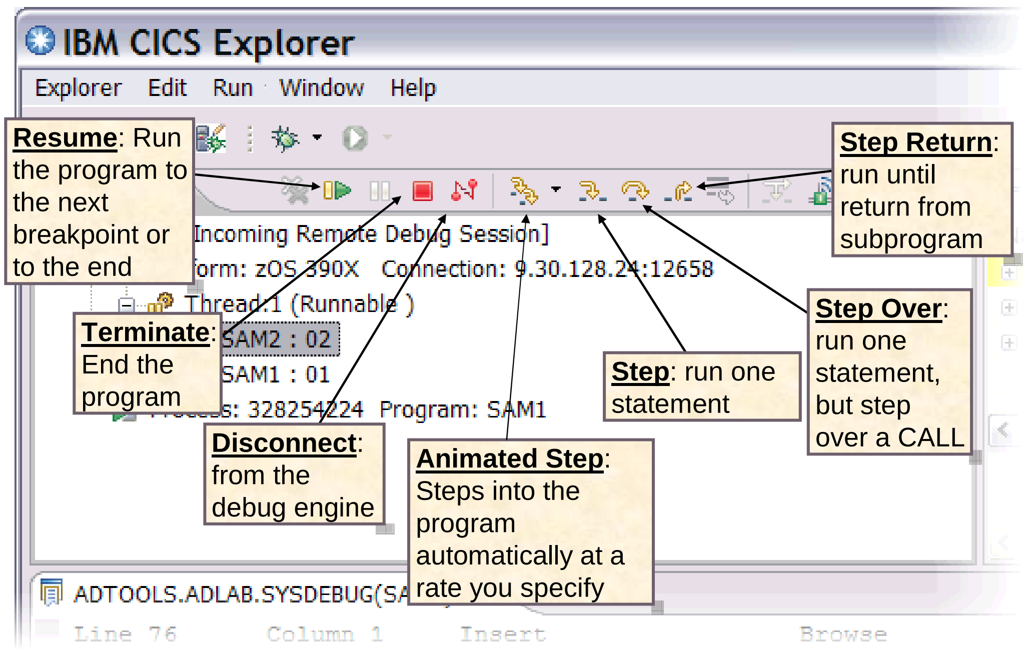
- Controls running and terminating the program
- Step controls to step through the program line by line
- Retrieve the workstation IP address


IBM CICS Explorer
Explorer Edit Run Window Help
Debug
(x)= Var
ADTOOLS.ADLAB.SYSDEBUG(SAM2)
The Debug view shows currently running programs

```
SAM1 [Incoming Remote Debug Session]
  Platform: zOS 390X   Connection: 9.30.128.24:1106
    Thread:1 (Runnable )
      SAM2 : 02
      SAM1 : 01
  Process: 328254224  Program: SAM1
```

The debug view displays the current call chain, or program stack.

Debug view action bar buttons



IBM CICS Explorer
Explorer Edit Run Window Help
[Incoming Remote Debug Session]
form: zOS 390X Connection: 9.30.128.24:12658
Thread:1 (Runnable)
SAM2 : 02
SAM1 : 01
Process: 328254224 Program: SAM1
ADTOOLS.ADLAB.SYSDEBUG(SA
Line 76 Column 1 Insert Browse

Resume: Run the program to the next breakpoint or to the end

Terminate: End the program

Disconnect: from the debug engine

Animated Step: Steps into the program automatically at a rate you specify

Step: run one statement

Step Over: run one statement, but step over a CALL

Step Return: run until return from subprogram

Notice the icons in the debug view. Click them to make the program take certain actions.

The green triangle is resume. It causes the program to run until it encounters a breakpoint. If no breakpoints are set, or if the program does not reach a breakpoints as it runs, it runs all the way to the termination of the application.

The red box is terminate. It forces the immediate termination of the application with a zero return code. If you click it, no more statements will run.

The disconnect icon disconnects the debugger from the application, but allows the application to continue running.

The yellow down arrow is “step”, or “step into”, which runs a single program statement. You can watch a program run statement by statement by clicking the step icon repeatedly. If the current statement calls a sub-program, in many cases it will step directly into the subroutine or procedure.

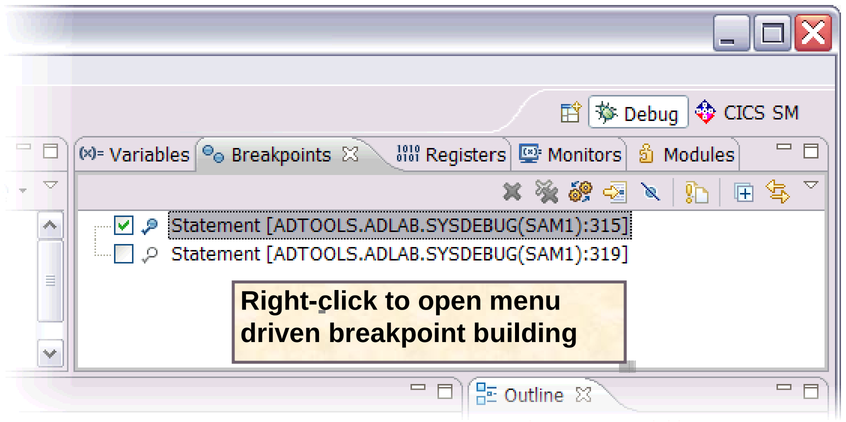
The step over icon allows you to step over a sub-program. Use this when you are on a statement that calls a sub-program, and you want the subroutine to run, but do want to see it in the debugger.

The step return button is used to run until the next program return. Use it to easily run to the end of a program or sub-program and then stop.

The Breakpoints view

Breakpoints view description

- Lists breakpoints
- Set address, entry, statement, load, and watch breakpoints
- Delete breakpoints
- Enable and disable breakpoints



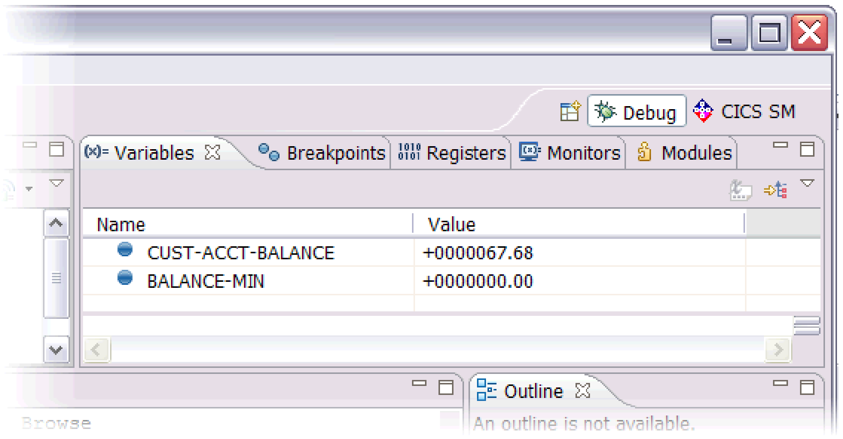
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):315]
Statement [ADTOOLS.ADLAB.SYSDEBUG(SAM1):319]

The breakpoints view displays a list of breakpoints that have been set. With pop-up menu options, you can add, change, and remove breakpoints, and enable or disable breakpoints. You will see more about the different types of breakpoints, and how to set breakpoints in a minute.

The Variables view

Variables view description

- Displays variables associated with the stack frame selected in the debug view
- Monitor local variables and memory
- Change representation (hex, decimal)
- Filter locals (all, current, previous, language based criteria)



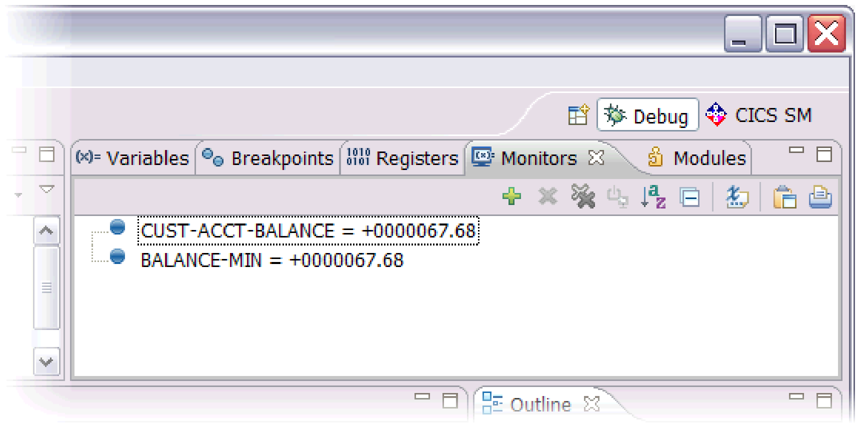
Name	Value
CUST-ACCT-BALANCE	+0000067.68
BALANCE-MIN	+0000000.00

The variables view is a place where you can see and change the values of variables in your program. You can display the auto monitor, which is a dynamic display that shows which ever variables are referenced by the current statement. Or you can display all local variables in the program, or optionally for COBOL programs, you can display all variables in the working-storage, linkage, or file section.

The Monitors view

Monitors view description

- Shows variables and expressions that you have selected
- Add or remove a variable or expression to the monitor
- Change representation (hex, decimal)
- Change a variable value



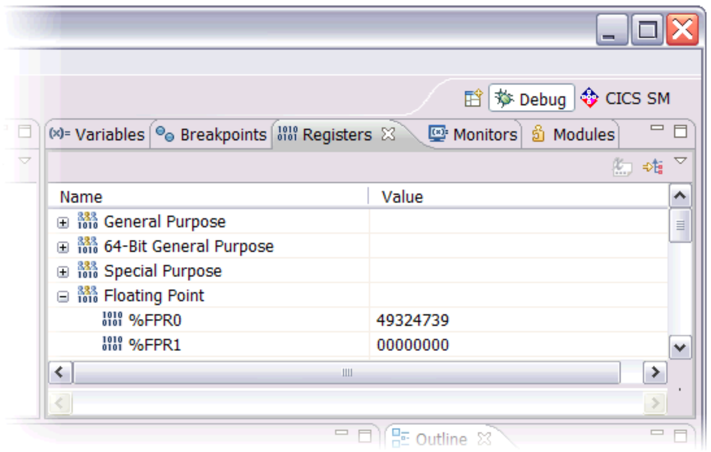
```
CUST-ACCT-BALANCE = +0000067.68
BALANCE-MIN = +0000067.68
```

The monitors view is another place where you can display and change variables. Having two views to display variables, the monitors and the variables view, gives you a lot of flexibility, and makes it easy to always see the variables you need. You can add as many individual variables as you need to the monitor, and remove them individually when you do not need to see them any more. You can optionally auto monitor variables in the monitor, so you always automatically see referenced variables as you step through a program.

The Registers view

Registers view description

- Lists register values
- Change a register value
- Add a register group



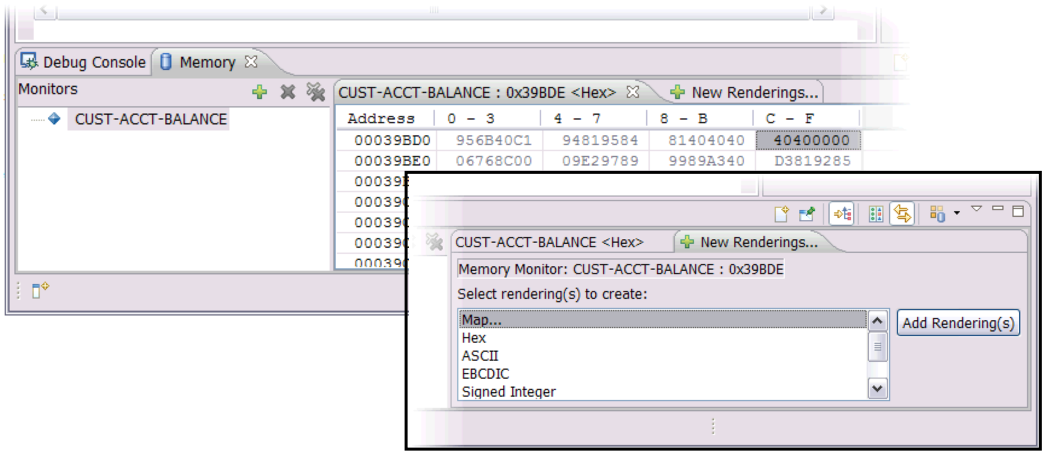
Name	Value
General Purpose	
64-Bit General Purpose	
Special Purpose	
Floating Point	
%FPR0	49324739
%FPR1	00000000

The registers view lets you view and change the contents of machine registers. The different sets of registers can be seen, including the general purpose, 64-bit general purpose, floating point registers, and others.

The Memory view

Memory view description

- View and modify memory
- Add / remove memory monitor
- Contains two panes -- the Memory Monitors pane and the Memory Renderings pane



```
Address   0 - 3     4 - 7     8 - B     C - F
00039BD0  956B40C1  94819584  81404040  40400000
00039BE0  06768C00  09E29789  9989A340  D3819285
```

Memory Monitor: CUST-ACCT-BALANCE : 0x39BDE

Select rendering(s) to create:

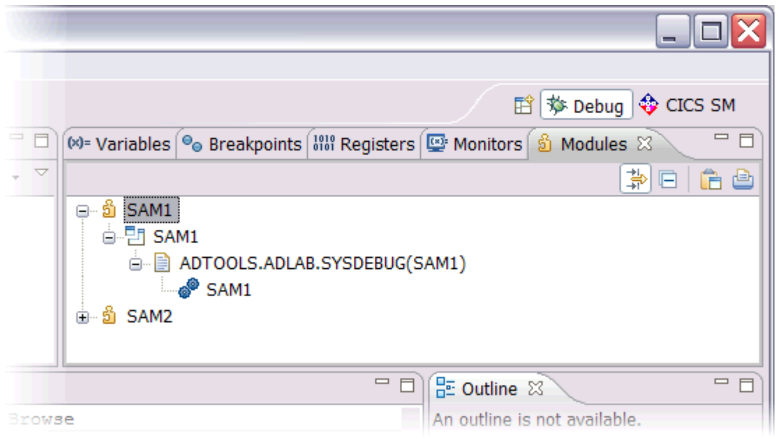
Map...
Hex
ASCII
EBCDIC
Signed Integer

The memory view is used to display and change the contents of memory. You can see memory at the address of a variable, based on a register, or at an address you specify. There are several rendering formats available, including hexadecimal, EBCDIC, ASCII, signed integer, and unsigned integer.

The Modules view

Modules view description

- Displays a list of modules loaded while running your application
- Expand items to see compile units, files, and functions
- Double-click source file nodes to open source in the editor



```
SAM1
  SAM1
    ADTOOLS.ADLAB.SYSDEBUG(SAM1)
      SAM1
SAM2
```

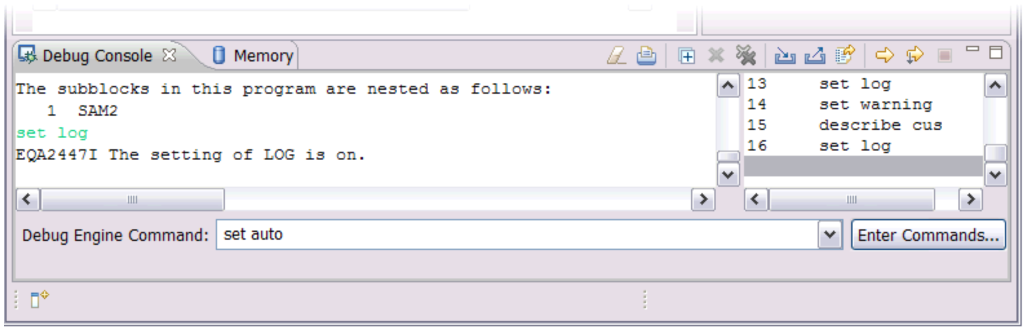
An outline is not available.

The modules view displays information about program modules. A module can be expanded to show compile units, related debug files, and functions. When debugging an application with multiple modules, you can double-click a debug file to have the program displayed in the source window. This gives you an easy way to see the source of a program other than the program currently running. For example, if currently running in a sub-program, you can refer to the source of the main program.

The Debug Console

Debug Console description

- Issue commands to the debug engine
- View messages from the engine and commands



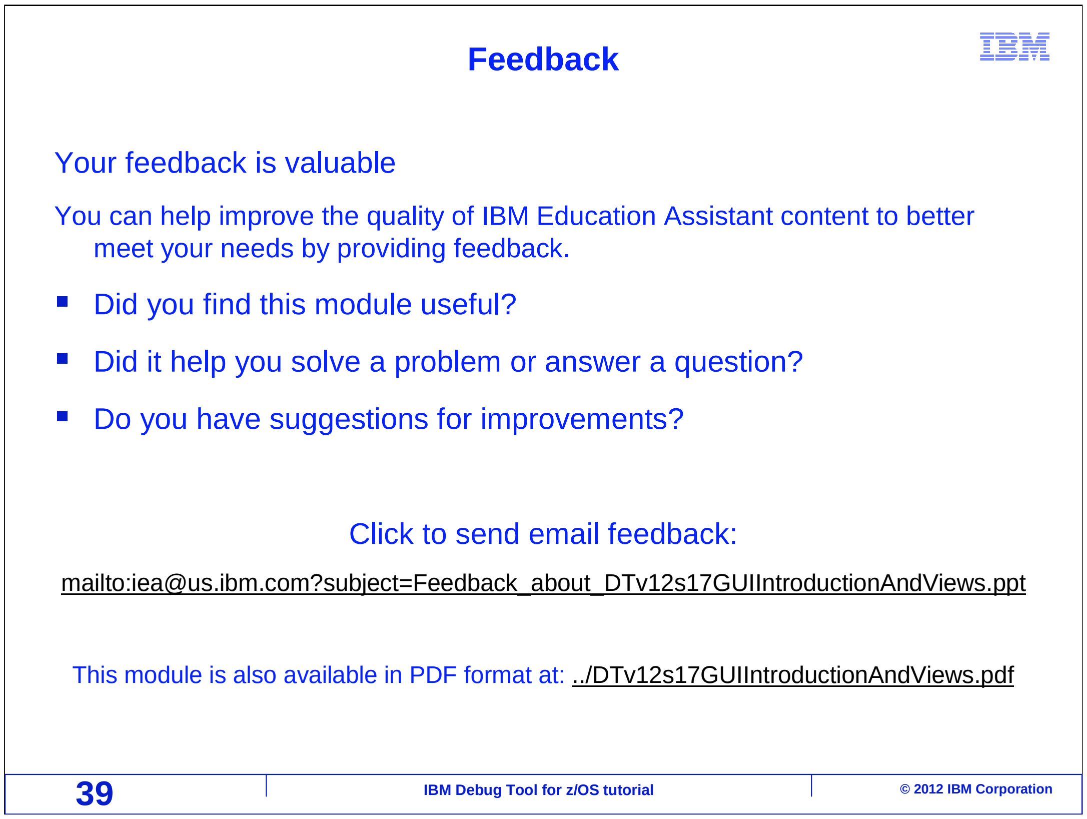
```
The subblocks in this program are nested as follows:
   1  SAM2
set log
EQA2447I The setting of LOG is on.
```

```
13      set log
14      set warning
15      describe cus
16      set log
```

Debug Engine Command: set auto

The Debug Console allows you to issue commands to the debugging engine, which is controlling the program that is running on the host system. The commands available here are a subset of the commands available in the 3270 terminal debugging interface, and include commands to take actions such as turn the auto monitor on or off, load debugging source files, change settings, and many others.

That is the end of this section. Please continue with the next section, using the GUI debugger.


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_DTv12s17GUIIntroductionAndViews.ppt

This module is also available in PDF format at: ../DTv12s17GUIIntroductionAndViews.pdf

39 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