



IBM Software Group

# IBM Developer for z Systems – for ISPF Developers

## Module 9 – Debugging COBOL with IDz



DevOps

Jon Sayles, IBM z Products - [jsayles@us.ibm.com](mailto:jsayles@us.ibm.com)

© 2019 IBM Corporation

@Copyright March, 2019

# IBM Trademarks and Copyrights

© Copyright IBM Corporation 2008 through 2019.

**All rights reserved – including the right to use these materials for IDz instruction.**

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

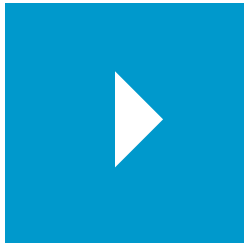
This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM Rational products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.



UNIT

# The IDz Workbench



## Topics:

- Debugging z/OS COBOL Batch Applications
- **Debugging z/OS COBOL Online Applications**

# Topic Objectives

---

After completing this unit, you should be able to:

- ▶ Using the Problem Determination Tools, Debug Option and IDz:
  - Debug a mainframe online transaction
- ▶ Describe the online transaction features for configuring your 3270 sessions with Debug Option
- ▶ Debug a CICS 3270 Application

# Online Debugging Overview

- Guess what?  
**No one gets it right the first time coding online programs either 😊**

```
ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19520101  <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

  B  <=== ENTER REQUEST
      B : SEE YOUR BIRTHDAY          (LINK TO PROGRAM CDAT2)
      R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
      C : CLEAR AND START OVER
      @ : ABEND WITH S0C7

HERE IS YOUR BIRTHDATE AND # OF DAYS ELAPSED

YOUR BIRTHDATE AND DAY: Tuesday 01 January 1952
HOW LONG AGO WAS THIS?      21,039 DAYS

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS
```

- Lucky for you:
  - ▶ Debug tool handles:
    - CICS 3270 online transactions
    - IMS TM online transactions
  - ▶ Without any different debugging techniques
    - The only difference from batch is the debug setup procedure for the online environment

# Steps for **Online** (CICS) Application Debug Session

1. Ensure that your compile proc has the necessary TEST parameter, and Compile/Link to create load module – and that your CICS application is setup for Debug Option testing
2. Discover workstation TCP/IP parameters:
  - IP Address
  - Listener port#
3. Access and login to your CICS region – Green Screen
4. Use the DTCN view, or execute the DTCN transaction and specify:
  - Terminal ID
  - Transaction code and programs – to put under Debug control
  - User-ID
  - TCP/IP parameters:
    - IP Address
    - Port#
  - Save the DTCN transaction specification
5. Debug your CICS application

# 1. Compile JCL Requirements for Using Debug Tool

- Use the **TEST** compiler option to prepare your executable COBOL program for use with the debugger.
  - ▶ The **TEST** option is *required* for remote debugging. It produces symbol and statement information that enables the debugger to perform symbolic source-level debugging
    - **Enterprise COBOL 3.4:** TEST(NONE,SYM,SEP)
    - **Enterprise COBOL V4.1 +:** TEST(NOHOOK,SEP,EJPD)
  - ▶ Include the DD card for your **SYSDEBUG** dataset in the COBOL Compile step
    - In traditional compile JCL – this would be in the **IGYCTRL** step
- If you are not using the IBM/IDz compile PROCs for building your applications, be sure to override the compiler option and add TEST - as shown

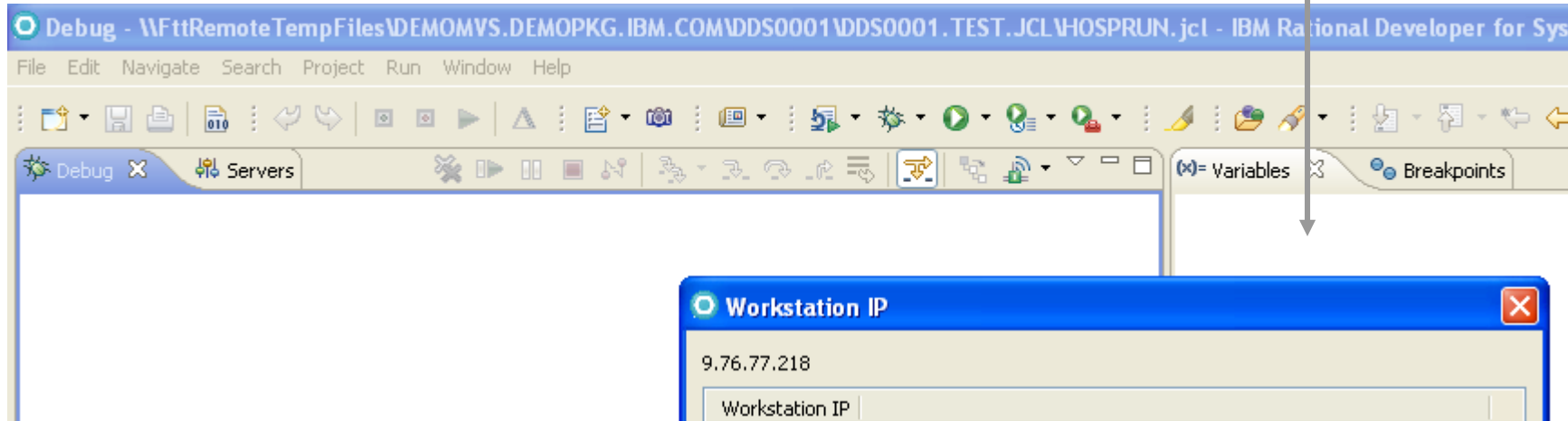
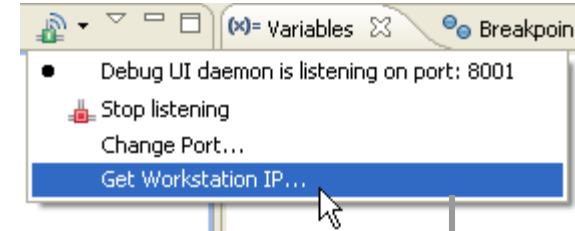
```
//STP0000 EXEC PROC=ELAXFCOC,  
// CICS=,  
// DB2=,  
// COMP=,  
// PARM.COBOL=('SQL',  
// 'LIB','TEST(NONE,SYM,SEP)')  
// COBOL.SYSPRINT DD DISP=SHR,  
// DSN=DDSO001.TEST.LISTING (TRTMNT)  
// COBOL.SYSDEBUG DD DISP=SHR,  
// DSN=DDSO001.TEST.SYSDEBUG (TRTMNT)  
// COBOL.SYSLIN DD DISP=SHR,  
// DSN=DDSO001.TEST.OBJ (TRTMNT)  
// COBOL.DBRMLIB DD DISP=SHR,  
// DSN=DDSO001.TEST.DBRMLIB (TRTMNT)  
// COBOL.SYSLIB DD DISP=SHR,  
// DSN=DDSO001.TEST.COPYLIB  
// DD DISP=SHR  
// COBOL.SYSXMLSD DD DUMMY  
// COBOL.SYSIN DD DISP=SHR,  
// DSN=DDSO001.TEST.COBOL (TRTMNT)  
// *  
//LKED EXEC PROC=ELAXFLNK  
//LINK.SYSLIB DD DSN=DDSO001.TEST.OBJ,  
// DISP=SHR  
// DD DSN=CEE.SCEELKED,  
// DISP=SHR  
//LINK.OBJ0000 DD DISP=SHR,  
// DSN=DDSO001.TEST.OBJ (TRTMNT)  
//LINK.SYSLIN DD *  
// INCLUDE OBJ0000  
/*
```

## 2. Discover TCP/IP address and IDz Port - Review

### ■ Open the Debug Perspective

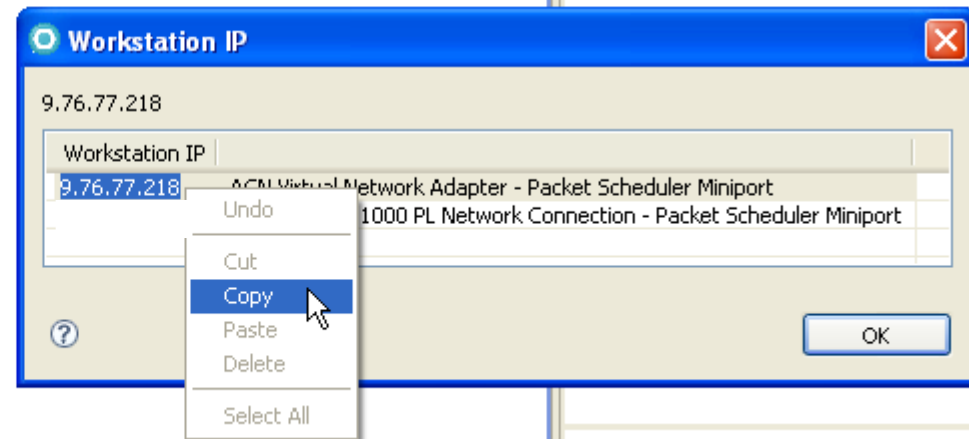
Click the small downward pointing triangle next to the debug-daemon icon

- ▶ **Note the Port#**
- ▶ **Select: Get Workstation IP...**
- ▶ **Copy the IP address**
- ▶ **Either paste the IP address into Notepad, or write it down**



**Note:** Your IDz Port# will most likely be set once, and will change infrequently.

However, depending on your installation's setup, your workstation's TCP/IP address could change - often

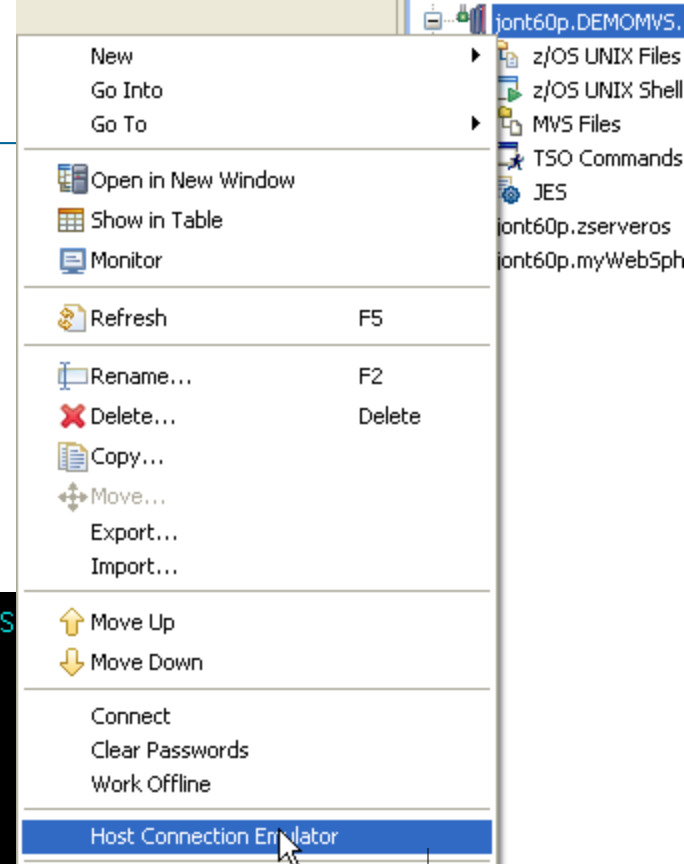




# 3. Login to your CICS Region

From Remote Systems Explorer:

- ▶ Right-click
- ▶ Select: Host Connection Emulator
- ▶ Select your CICS application
- ▶ Enter your Userid and Password and sign in



```
Signon to CICS                                APPLID CICS
WELCOME TO CICS

Type your userid and password, then press ENTER:

  Userid . . . . dds0001   Groupid . . . . _____
  Password . . .
  Language . . . . _____
  New Password . . .

DFHCE3520 Please type your userid.
F3=Exit
```

### 3. Setup the Debug Option Parameters using DTCN Transaction – 1 of 3

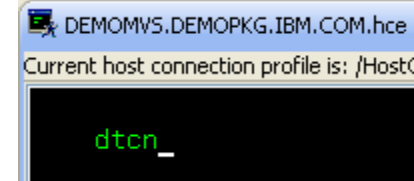
If you did NOT use the DTCN view to enter your DTCN properties you can do so using a CICS Transaction (green screen)

From CICS (after signing in):

- ▶ Clear the screen, Enter: **DTCN** – and press **↵Enter**

From the DTCN screen

- ▶ **Press F10** – this will fill in the Terminal Id for your workstation



Note that you can also  
type an asterisk: \*  
...as the Terminal Id

```
DTCN                                Debug Tool CICS Control - Primary Menu                                CICSACB1
                                * VSAM storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id      ==> 0096
Transaction Id   ==>
LoadMod::>CU(s) ==> CD*      ::> *      ==>      ::>
                                ==>      ::>      ==>      ::>
                                ==>      ::>      ==>      ::>
                                ==>      ::>      ==>      ::>
User Id          ==> DDS0001
NetName          ==>
IP Name/Address  ==>
Select type and ID of debug display device
Session Type     ==> TCP      MFI, TCP
Port Number      ==> 8001     TCP Port
Display Id       ==> 9.65.190.169

Generated String: TEST(ALL,'*',PROMPT,'TCPIP&9.65.190.169%8001:*')
Repository String: TEST(ALL,'*',PROMPT,'TCPIP&9.65.190.169%8001:*')

Profile Status:   Active. Press PF5 to Inactivate.
EQA2520W Terminal mismatch. Press PF10 to set to current terminal.
PF1=HELP 2=GHELP 3=EXIT 4=SAVE 5=ACT/INACT 6=DEL 7=SHOW 8=ADV 9=OPT 10=CUR TRM
```



**Note:** You would  
only use the DTCN  
transaction to specify  
your Debug Option  
properties if you could  
not use the DTCN view  
(prior slides)

### 3. Setup the Debug Option Parameters using DTCN Transaction – 2 of 3

#### DTCN transaction data entry screen

- ▶ **Enter the Tran-code**

- Transaction ID

- ▶ **Enter up to eight specific Program Id(s) you wish to debug through ...or...**

```
Select the combination of resources to debug (see Help for more information)
Terminal Id      ==> 0903
Transaction Id   ==>
LoadMod::>CU(s) ==> CD*      ::> *      ==>      ::>
                  ==>      ::>      ==>      ::>
                  ==>      ::>      ==>      ::>
```

- ▶ **Enter wildcard text for the Program Id(s)**

- Ex. CD\*

- ▶ Enter your User-ID

- ▶ Session Type: **TCP**

- ▶ **Port Number:**

from your Debugger look-up

- ▶ **Display ID:**

Your TCP/IP address,

from your Debugger look-up (note that you can not paste into this 3270, screen)

```
Select type and ID of debug display device
Session Type    ==> TCP          MFI, TCP
Port Number     ==> 8001        TCP Port
Display Id      ==> 9.76.89.169 _
```

- **Press F4** to save your debug profile
- **Press F3** to clear the screen

### 3. Setup the Debug Option Parameters using DTC Session Type – 3 of 3

**DTC Session Type** – Assumes that IDz's DBGMGR started task is running

```

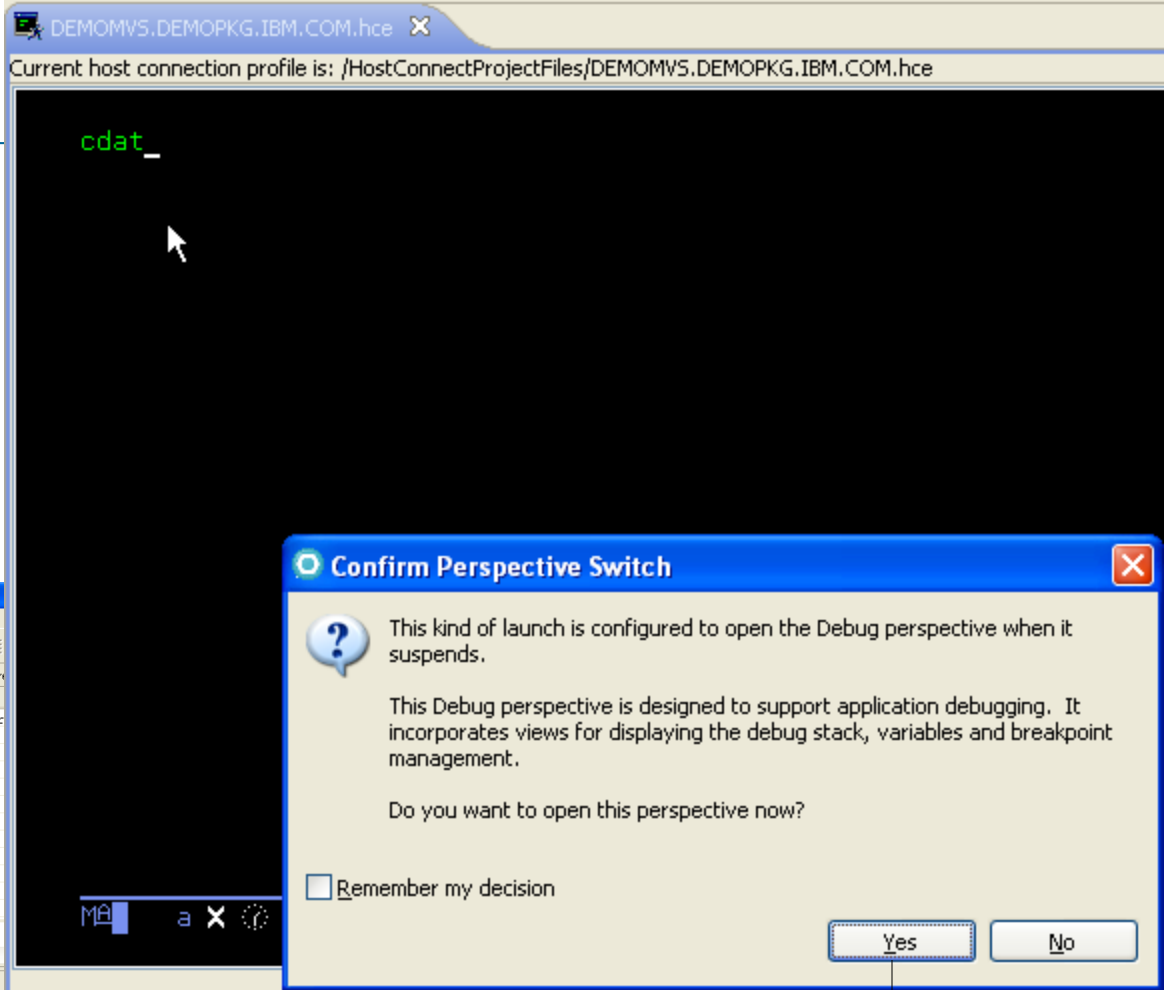
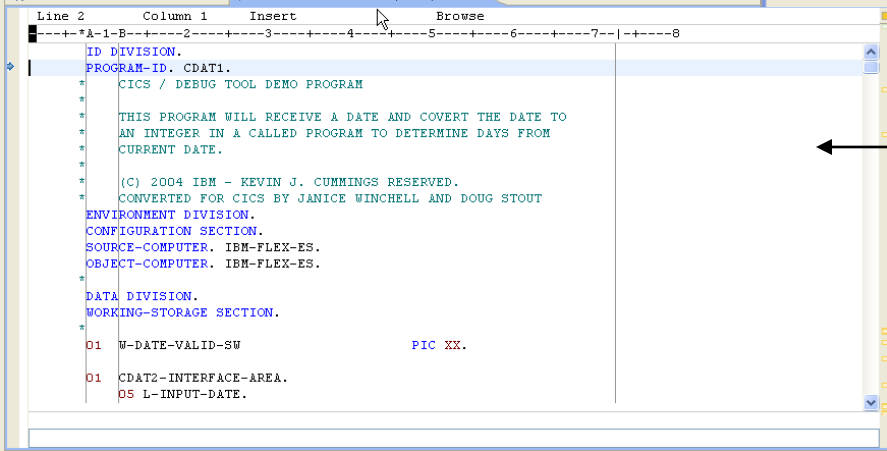
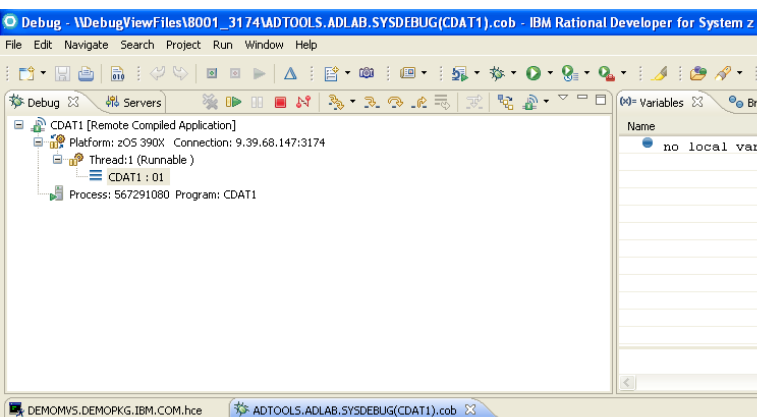
* VSAM storage method *
Select the combination of resources to debug (see Help for more in
Terminal Id      ==> 0088
Transaction Id   ==>
LoadMod::>CU(s)  ==> CD*      ::> *      ==>      ::>
                  ==>      ::>      ==>      ::>
                  ==>      ::>      ==>      ::>
                  ==>      ::>      ==>      ::>
User Id          ==> DDS0001
NetName          ==>
IP Name/Address  ==>
Select type and ID of debug display device
Session Type     ==> DTC      MFI, TCP, DIR, DTC, DBM
Port Number      ==>      TCP Port
Display Id       ==> DDS0001
```

- ▶ Press F10 to generate a unique Terminal ID
  - ▶ (Optionally) Enter a Tran-code
  - ▶ Enter up to eight specific load modules you'd like to debug
  - ▶ User Id → TSO-ID
  - ▶ Session Type → DTC
  - ▶ Display Id → TSO-ID
- 
- Press F4 to save your debug profile
  - Press F3 to clear the screen

## 4. Start Debugging

## From the CICS region

- ▶ **Enter the Tran-code**
- ▶ **Press ↵Enter**
- ▶ **Click: Yes at the Confirm Perspective Switch**



Debug as previously learned in the batch/remote and Local debug units.

14

# What Happens during CICS calls and screen-io – 1 of 2

You will be prompted, and presented with debug-run-time options

The screenshot displays the IBM ADTOOL.SYSDEBUG(CDAT1).c interface. The top pane shows the application structure for CDAT1, including the platform (zOS 390X), connection (9.39.68.147:3174), thread (Thread:1 (Runnable)), and process (Process: 567291080 Program: CDAT1). The bottom pane shows the source code for DEMOMVS.DEMOPKG.IBM.COM.hce, with line 560 highlighted. The code includes a CICS call and a call to 'DFHEI1'.

The 'Event Occurred' dialog box is open, displaying the message: "CEE0199W The termination of a thread was signaled due to a STOP statement. Select which action to take:". The options are:

- ☒ Step into handler
- ☐ Run
- ☐ Examine

The 'OK' button is visible at the bottom right of the dialog box.

The source code snippet shown is:

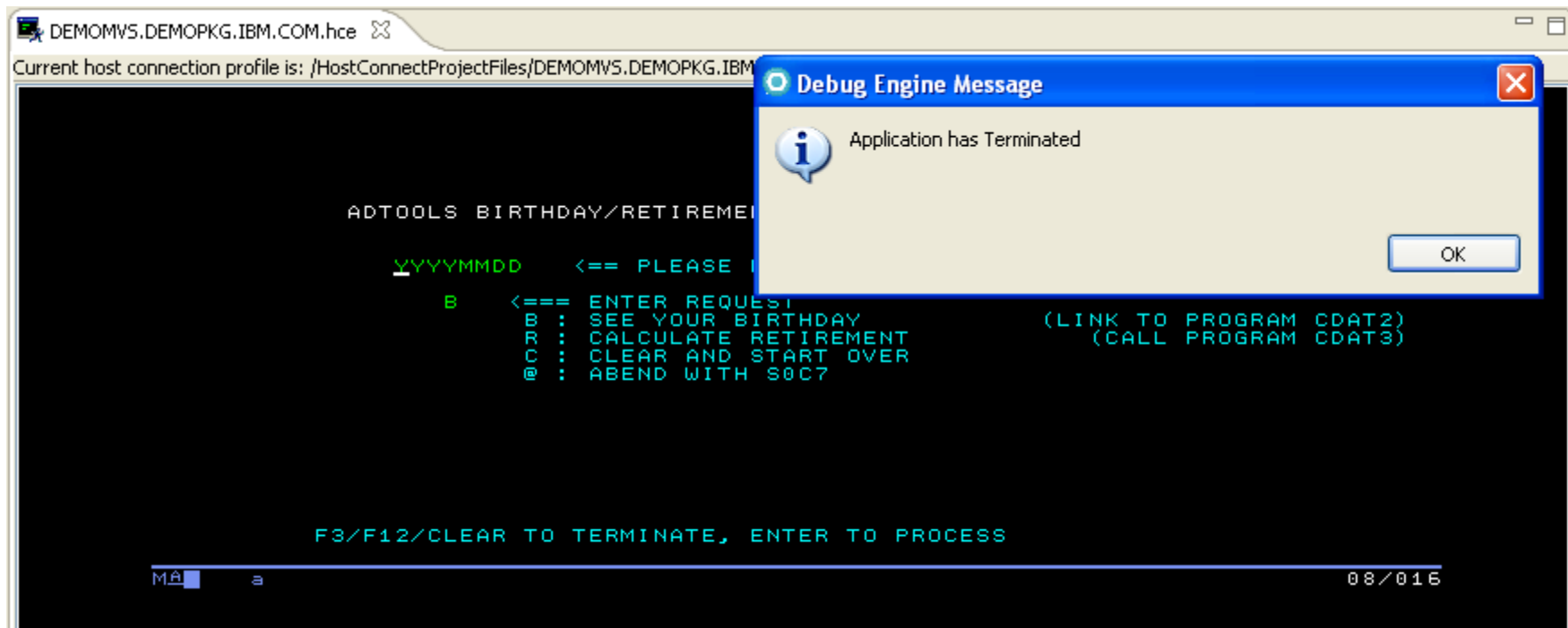
```
Line 560      Column 1      Insert      B
-----+*A-1-B--+-----2-----+-----3-----+-----4-----+-----5-----
      MOVE W-CDAT3-RETIRE-DATE TO W-COM-RET
      MOVE 'Y' TO W-COM-RETIRE-DATE-OK
      ELSE
      MOVE 'PROBLEMS IN CALCULATING RETIREM
      PERFORM 0350-ADD-MESSAGE-LINE
      END-IF.

0900-RETURN-PSEUDO.
*EXEC CICS
*  RETURN TRANSID('CDAT')
*  COMMAREA(WS-COMM-AREA)
*END-EXEC.
Move length of WS-COMM-AREA to dfhb0020
Call 'DFHEI1' using by content x'0e08e0000700001000f0f0f2f9f5
'404040' by content 'CDAT' by reference WS-COMM-AREA by
reference dfhb0020 end-call.
```

# What Happens during CICS calls and screen-io – 2 of 2

If your current transaction ends, and a BMS or 3270 screen is sent:

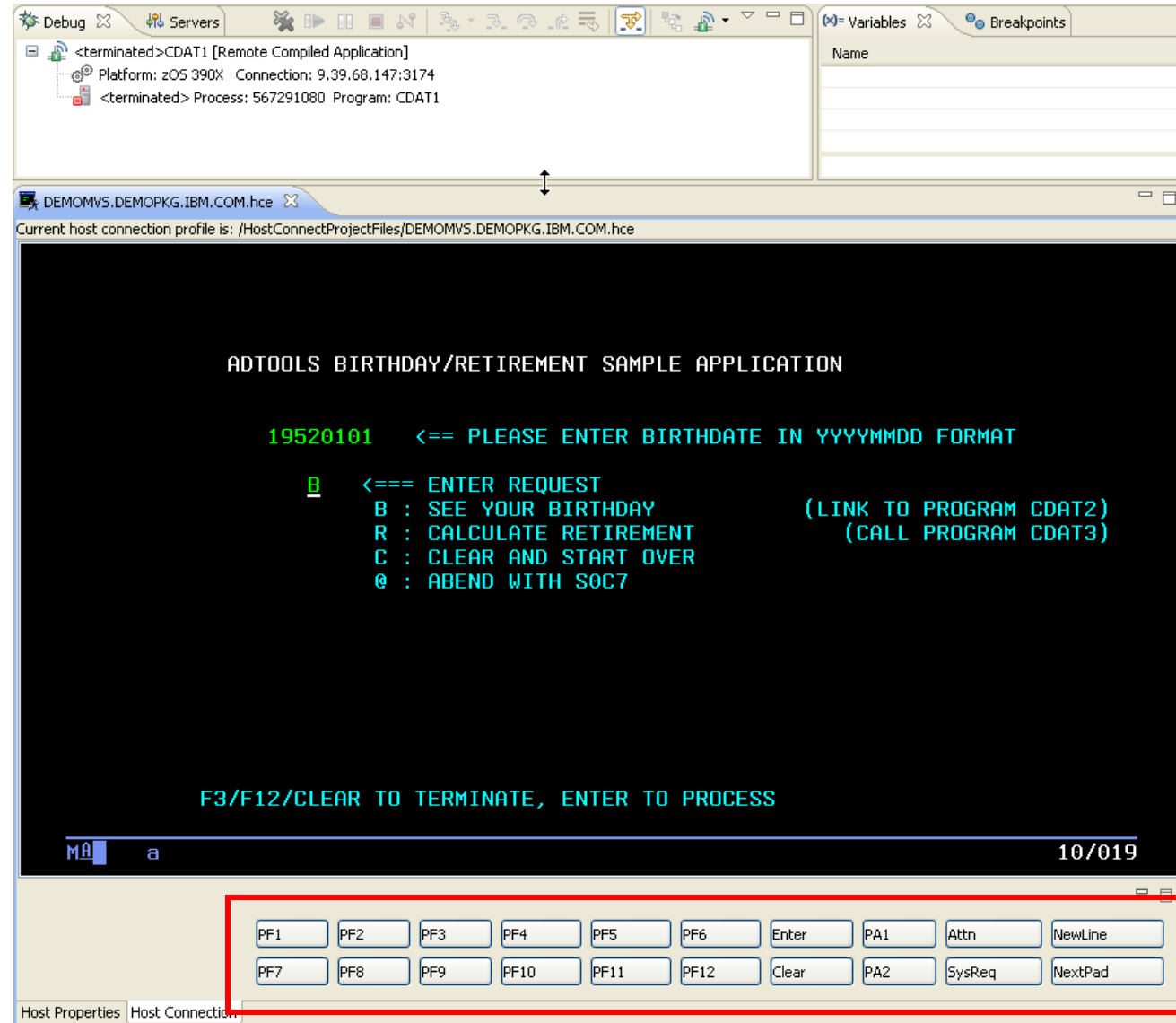
- You will be notified (prompted) by the debug engine
- If a screen is sent, the 3270 will display in the content area





# What About PF-Keys and Other Data Entry?

- You can resize the screen portion of the debugger
- And use the PF-Key emulation options in the Host Connection



# Same Debug Option Functionality!

All of the debugging features and functions are available when debugging online CICS or IMS TM applications.

The screenshot displays the IBM Debug Option interface for debugging the CDAT1 program. The interface is divided into several panes:

- Debug Window:** Shows the program structure with a tree view. The selected item is CDAT1:01. The status bar indicates the platform is zOS 390X, the thread is 1 (Runnable), and the process is 567291080.
- Variables Window:** Displays the current state of variables. The table below shows the variables and their values.
- Source Window:** Shows the program code for CDAT1. The current line is 379, and the column is 1. The code includes comments and program logic.
- Outline Window:** Shows the program structure, including the ID DIVISION, ENVIRONMENT DIVISION, DATA DIVISION, WORKING-STORAGE SECTION, LINKAGE SECTION, and PROCEDURE DIVISION.

Name	Value
W-DATE-VALID-SW	'00'
CDAT2-INTERFACE-AREA	
L-INPUT-DATE	
L-DAY-DIFFERENCE	00000000
L-DATE-FORMATTED	' } v\$0000000000000000 } v{C
L-PROGRAM-RETCODE	0000
WS-COMM-AREA	
W-COM-TRAN-EXEC-COUNT	0001
W-COM-PROGRAM-AREA	
W-COM-USER-REQUEST	' '
W-COM-DATE-INDICATOR	' '
W-COM-DATE-DIFFERENCE	00000000
W-COM-DATE-FORMATTED	' '
W-COM-RETIRE-DATE-OK	' '
W-COM-RETIRE-DATE	' '

```
Line 379      Column 1      Insert      Browse
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
          MOVE DFHCOMMAREA TO WS-COMM-AREA
          MOVE ERR-MSG-BAD-KEY TO MSG-LINE
          PERFORM 0350-ADD-MESSAGE-LINE
          PERFORM 0310-SEND-DATAONLY
          PERFORM 0900-RETURN-PSEUDO
          END-EVALUATE
          .
0100-PROCESS-REQUEST.
          PERFORM 0400-RECEIVE-MAP.
          MOVE FUNCTION UPPER-CASE(MAPREQI) TO W-COM-USER-REQUEST
          *
          *      B = BIRTHDAY INFO (LINK TO PROGRAM CDAT2)
          *      R = RETIREMENT      (CALL PROGRAM CDAT3)
          *      C = CLEAR
          *      @ = ABEND 0C7
          EVALUATE W-COM-USER-REQUEST
          WHEN 'C'
```

**Outline:** PROGRAM: CDAT1  
ID DIVISION.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
LINKAGE SECTION.  
01 dfheiblk.  
01 DFHCOMMAREA.  
05 COM-TRAN-EXEC-COUNT  
05 COM-AREA-TRAN-PROGR  
PROCEDURE DIVISION  
0000-MAINLINE.  
0100-PROCESS-REQUEST.  
0300-SEND-ERASE.  
0310-SEND-DATAONLY.  
0350-ADD-MESSAGE-LINE.  
0400-RECEIVE-MAP.  
0500-VERIFY-INPUT-DATE.  
0600-CALC-DAY-DIFFERENCE.  
0650-CALCULATE-RETIREMENT.  
0900-RETURN-PSEUDO.

# Topic Objectives

---

After having completed this unit, you now should be able to:

- ▶ Using the Problem Determination Tools, Debug Option and IDz:
  - Debug a mainframe online transaction
- ▶ Describe the online transaction features for configuring your 3270 sessions with Debug Option
- ▶ Debug a CICS 3270 Application