



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

Scenarios for starting the debugger for LE batch programs

- Trigger the debugger with an LE TEST option in JCL, and
 1. Display the debugger on a graphical user interface
 2. Display the debugger on a TIM (Debug Tool terminal interface manager) terminal through a session manager
 3. Display the debugger on a dedicated TIM terminal
 4. Display the debugger on a dedicated non-TIM terminal
- Trigger the debugger with the LE 'user exit data set' facility, and
 5. Display the debugger on a graphical user interface
 6. Display the debugger on a TIM terminal through a session manager
 7. Display the debugger on a dedicated TIM terminal
 8. Display the debugger on a dedicated non-TIM terminal



Running and debugging an LE batch program under TSO

- Use the 'Debug Tool setup file' online panels to run the program and display the debugger on the TSO terminal

In this section, you will see a way to run and debug a batch application under TSO, using the 'Debug Tool setup file' online panels. You can skip this section if you do not plan to use Debug Tool this way on your system.

- **Description**
 - The Debug Tool Setup file online panels are used to create a 'setup file'
 - A setup file performs a function similar to JCL
 - It names a program to run and has DD statements
 - When you run a setup file, the application runs in TSO
 - An LE TEST option can be used to set a trigger for the debugger
 - The debugger displays on the TSO terminal
- **Consider using this method:**
 - When batch initiators are not available to run batch jobs on your system
 - If dedicated Debug Tool terminals, or GUI debugging software, are not available on your system
- **When not to use this method:**
 - If the batch-oriented methods are available. They may be quicker to set up.
 - If you need to debug a program and use TSO services at the same time

This is an optional way to debug batch programs. A set of online panels in TSO are used, called the 'Debug Tool setup file' utility. The panels are used to create something called a 'setup file', which serves a similar purpose as JCL. When you think of it, JCL is used to specify the name of a load module to run, provide parameters for the program, and specify a list of files that the program will use in the form of DD statements. A setup file has similar information. After you create a setup file with the right information, you run it. The program runs in your TSO session, and has access to the files in the DD list.

If you set a debugging trap in the setup file, the debugger displays on the user's TSO terminal, or on GUI debugging software on your workstation. As with programs running in batch, an LE TEST option can be used to start the debugger.

Consider using this method if batch initiators are not available on your system to run batch jobs, or if dedicated Debug Tool terminals or GUI debugging software are not installed on your system.

Typically, this method would not be used if one of the batch-oriented methods is available. The batch methods may be quicker to set up. Also, you may prefer to use one of the batch methods if you need to debug a program and use TSO services (such as the editor or ISPF utilities) at the same time.

Navigate to the Debug Tool Utilities menu Select option 2 for Debug Tool Setup File



```
----- Debug Tool Utilities -----  
Option ==> 2 █  
More: +  
0 Job Card  
  Create Job Card image.  
1 Program Preparation  
  Convert, compile, assemble or link edit program.  
2 Debug Tool Setup File  
  Manage setup files and start debug session in TSO foreground or batch.  
3 Code Coverage  
  Measure code coverage in programs.  
4 IMS TM Setup  
  Update Language Environment run-time options in IMS. Create message region.  
5 Load Module Analyzer  
  Analyze load modules and each CSECT in the load module.  
6 Debug Tool User Exit Data Set  
  Modify the data set used by user exit during program initializa
```



First, navigate to the Debug Tool Utilities panel in TSO. Typically, you can get there using the menus on your system or with a command. Type 2 on the command line to select the 'Debug Tool setup file' option, and press Enter.

Specify the name of a new or existing setup file



```
----- Debug Tool Foreground - Edit Setup File -----
Command ==> _____

Setup File Library:
Project . . . . ISS16
Group . . . . ADLAB
Type . . . . DTSF
Member . . . . SAM1          (Blank or pattern for member selection list)
                               (or existing or new member name)

Other Data Set Name:
Data Set Name . . . . ADLAB.DTSF(SAM1)
Volume Serial . . . . _____ (If not cataloged)

_ Initialize New setup file for DB2 (/)
```

If the library does not exist, a panel is displayed where you can specify allocation parameters, and it will be created

Enter

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

You are prompted for the name of a setup file. If you specify a file name, or a PDS member name that does not exist yet, then you are creating a new setup file. If you have setup files, you can specify the name here to use it again. If you specify the name of a file or PDS or library that does not exist, you will be prompted automatically to allocate the new data set.

In this example, a PDS already exists where setup files can be saved. The name of the PDS is entered, but with a new member name, so this setup file will be stored as a new member. Enter is pressed.

A new setup file An I (insert) line command is entered



EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 1 of 1
 Command ==> Scroll ==> PAGE

Load Module Name _____
 Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_____ Enter / to modify parameters _____

Cmd	DD Name	Seq	C	DD Information (DSN/Sysin/Sysout/Dummy)	DISP
i	_____	_____	_____	***** Top of Data *****	_____
				***** Bottom of data *****	

The load module that will run and program parameters. Similar to JCL: //STEPNAME EXEC PGM=name, PARM=parms

DDs. Similar to JCL DD statements

Enter

A new setup file is shown. Notice the three main parts of a setup file: load module name, parameters, and DDs. The load module name can be entered the field near the top. This is the module that will run when the setup file runs. The 'load module name' field serves the same function of an EXEC statement in JCL. Parameters are optional. If your program requires parameters when it runs, type them into the field to the right of the text 'enter / to modify parameters'. Parameters are the same as the PARM option of an EXEC statement in JCL.

A new setup file does not have any DDs, and they can be entered into the list. To insert a new DD, type an "I" line command in the 'Cmd' field as in this example, and press Enter.

Result of I (insert) line command: a new line for a DD was added



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'          Row 1 to 2 of 2
Command ==> |                                           Scroll ==> PAGE

Load Module Name _____
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy)  DISP
-----
_____ 1 - ***** Top of Data *****
_____ 1 - ** Select Detail DD Information **
***** Bottom of data *****
```

The "I" line command added a new line for a DD.

DDs can be entered this way



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 2 of 2
Command ==> | Scroll ==> PAGE

Load Module Name _____
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
custfile 1 'tss16.adlab.cust2' shr
***** Top of Data *****
***** Bottom of data *****
```

This is equivalent to the JCL statement:
//CUSTFILE DD DSN=TSS16.ADLAB.CUST2,DISP=SHR

Enter

8

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

Here, a new DD for CUSTFILE is entered, referencing file 'TSS16.ADLAB.CUST2' with DISP=SHR. It serves the same function as a DD statement in JCL.

Use a COPY command to copy from JCL or another setup file



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 2 of 2
Command ==> copy Scroll ==> PAGE

Load Module Name _____
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
_____ - ***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
***** Bottom of data *****
```



You can continue to insert DD statements manually, one at a time, until all the DDs that your program needs are specified. However, if JCL already exists with the information needed to run the program, there is an easier way. You can copy information from the JCL into the setup file. To do it, type COPY on the command line, and press enter.

You can copy from JCL or another setup file



```
----- Debug Tool Foreground - Copy from Setup File or JCL -----
Command ==> █
Select data to copy into 'TSS16.ADLAB.DTSF(SAM1)'
Setup File or JCL Library:
Project . . . TSS16
Group . . . ADLAB
Type . . . JCL
Member . . . XSAMDTU (Blank or pattern for member selection list)
                    (or existing or new member name)
Other Data Set Name:
Data Set Name . . .
Volume Serial . . . (If not cataloged)
Note: When you copy from another setup file the entire contents are copied.
      When copying from JCL you can select the information you want to copy.
Enter
```

10

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

The COPY command displays the 'Copy from setup file or JCL' panel. From here you specify the name of an existing setup file or JCL that you want to copy. In this example, the name of a PDS member containing JCL is specified, and Enter is pressed.

When copying from JCL, select the statements to be copied



```
----- Debug Tool Foreground - Copy from JCL Datas Row 1 to 13 of 13
Command ==> _____ Scroll ==> PAGE

Enter S* on the command line or on a Sel line to select all JCL statements.
Enter S on a Sel line to select that JCL statement.
Enter RESET to deselect all JCL statements.

Sel  JCL Image
____ //TSS16T JOB
____ /* from 'TSS16.ADLAB.JCL (XSAMDTU)'
s    //RUNSAM1 EXEC PGM=SAM1
s    //STEPLIB DD DSN=TSS16.ADLAB.LOAD,DISP=SHR
____ //CUSTFILE DD DSN=TSS16.ADLAB.CUST3,DISP=SHR
s    //SYSPRINT DD SYSOUT=*
s    //SYSOUT DD SYSOUT=*
s    //CUSTRPT DD SYSOUT=*
s    //TRANFILE DD *
____ *TRAN (* IN COL 1 IS A COMMENT)
____ *-----
____ PRINT <== PRINT CUSTOMER LIST
____ TOTALS <== PRINT TOTALS
***** Bottom of data *****
```



The JCL was read, and a list of JCL statements are displayed. Use S line commands to select each of the JCL statements that you want to copy into the setup file.

Notice that the EXEC statement is selected. The name of the load module on the EXEC statement will be copied. If there are parameters on the EXEC statement, they will be copied also.

If you want to copy all of the statements in the JCL, you can use an S* command to select everything. After making the selections, press F3.

The selected JCL statements were copied into the setup file



```

EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 7 of 7
Command ==> █

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSTRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - SYSOUT=*
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
  
```

The load module name was copied from the EXEC statement

DD statements were copied

The selected JCL statements were copied into the setup file. Notice that the load module name, SAM1, was copied along with several DDs. The DD that had been entered manually is still there.

Notice the SYSOUT DD statement. In the JCL, it refers to JES output. If you have DD statements directed to SYSOUT, the output will still go to SYSOUT later when you run the setup file in TSO.

Overtyping DD information to change it



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 7 of 7
Command ==> _____ Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```

Tip: Specify * in the 'DD information' field to direct output to the TSO terminal

In this example, the DD information for the SYSOUT DD is changed to asterisk (*). This output will be sent to the TSO terminal as the program runs. This can be a good option in some cases if you want to see messages produced by the program automatically on your terminal. However, it is best to avoid this option if the DD will receive a large amount of output.

The RUN command runs the setup file



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 7 of 7
Command ==> run
Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments
_ Enter / to modify parameters
Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSTRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```



At this point, the setup file is ready to run. A load module name, and all the DDs it needs are specified. This particular program does not need any run-time parameters, so none were entered.

The RUN command runs the setup file.

Messages from the program are displayed



```
SAM1 STARTED DATE = 12/08/09 (mm/dd/yy)
          TIME = 19:29:02
```

```
*** █
```

Note: These messages are from the SYSOUT DD,
and were generated by the program used in this
example

Enter

15

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

In this example, the program is writing messages to a DD named SYSOUT as it runs. Remember that the SYSOUT DD was specified so that these messages would be displayed on the TSO terminal. The messages shown here are produced by the program, and are unique to this example. Press Enter to continue.

The program ran



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'
Command ==> _____ Scroll ==> PAGE RC 0
Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments
_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - SYSOUT=*
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```

The program return code is displayed

The program ran all the way through to it's termination. Notice that the program's return code is shown.

Triggering the debugger



- In this example, the program ran, but the debugger did not start
 - A debugging trigger was not set
- You can use the same methods to start the debugger that are available for debugging batch jobs:
 - A CEEOPTS DD with an LE TEST option
 - The Debug Tool user exit data set facility
 - An LE TEST option coded in the parameter string
 - Tip: Type a slash (/) in the field marked 'Enter / to modify parameters' to display a dialog that will generate a TEST option automatically
- In this example, a CEEOPTS DD will be used

So far, you have seen how to create a setup file to run a batch program in TSO. Running a setup file is a way to run a program, similar to running a program in batch. However, you may have noticed that the debugger did not start when the program ran.

Just as when a program runs in batch, a trigger for the debugger must be set so the debugger will start when the program runs. And the same methods that are available to trigger the debugger for a program running in batch are available when using a setup file. The most common methods include coding a CEEOPTS DD statement with an LE TEST option, using the Debug Tool 'user exit data set' facility, and coding an LE TEST option in the program parameter string.

In this example, a CEEOPTS DD statement will be used.

Insert a new DD statement



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' RC 0
Command ==> _____ Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
i 1 CUSTFILE 1 - ***** Top of Data ***** SHR
  CSTRPT 1 - SYSOUT=*
  STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
  SYSOUT 1 - *
  SYSPRINT 1 - SYSOUT=*
  TRANFILE 1 - SYSIN
***** Bottom of data *****
```

An I (insert) line command is entered

Enter

18

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

An I line command is entered to insert a new DD statement.

A CEEOPTS DD will be used to pass an LE TEST option and trigger the debugger



```

EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'          Row 1 to 8 of 8
Command ==> _____                               Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy)  DISP
***** Top of Data *****
e | ceeopts 1 - sysin
  CUSTFILE 1 - 'TSS16.ADLAB.CUSTZ'                          SHR
  CUSRPT 1 - SYSOUT=*
  STEPLIB 1 - 'TSS16.ADLAB.LOAD'                             SHR
  SYSOUT 1 - *
  SYSPRINT 1 - SYSOUT=*
  TRANFILE 1 - SYSIN
***** Bottom of data *****

```

'SYSIN' in the 'DD information' specifies that in-stream data will be used. The E (edit) line command is entered to edit the in-stream data.



The new line is overtyped to specify a CEEOPTS DD. 'SYSIN' is specified in the 'DD information' field. That specifies that in-stream data will be used.

An E (for edit) line command is entered for the CEEOPTS DD. This will display the in-stream data in the editor.

The editor is displayed



```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      TSS16.DTSUTEMP.XIN1D007          Columns 00001 00072
Command ==>                           Scroll ==> PAGE
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>         your edit profile using the command RECOVERY ON.
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
***** ***** Bottom of Data *****
```

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

The E line command displayed the editor. Since this is a new DD statement, no data has been entered yet.

The RUN command runs the setup file



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 8 of 8
Command ==> run
Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments
_ Enter / to modify parameters
Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CEEOPTS 1 - SYSIN
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```



Since the CEEOPTS DD contains SYSIN data, the actual data will be saved within the setup file itself. A RUN command is entered to run the setup file.

The debugger is displayed on the TSO terminal



```
COBOL LOCATION: SAM1 initialization
Command ==> Scroll ==> PAGE
MONITOR +-----1-----2-----3-----4-----5-----6- LINE: 0 OF 0
***** TOP OF MONITOR *****
***** BOTTOM OF MONITOR *****

SOURCE: SAM1 +-----1-----2-----3-----4-----5----- LINE: 1 OF 467
1 *****
2 * PROGRAM: SAM1
3 * Sample program for the ENTERPRISE COBOL Compiler
4 *
5 * AUTHOR : Doug Stout
6 * IBM PD TOOLS

LOG 0-----1-----2-----3-----4-----5----- LINE: 32 OF 34
0032 IBM Debug Tool Version 10 Release 1 Mod 0
0033 12/31/2009 00:44:02 AM
0034 5655-V50: Copyright IBM Corp. 1992, 2009
PF 1:? 2:STEP 3:QUIT 4:LIST 5:FIND 6:AT/CLEAR
PF 7:UP 8:DOWN 9:GO 10:ZOOM 11:ZOOM LOG 12:RETRIEVE
```

When the first LE program runs, Language Environment checks to see if a CEEOPTS DD statement is present. If it is, it is read for options. The TEST option is a trigger that directs LE to start the debugger. The debugger is displayed on your TSO terminal.

A GO command runs the program



```
COBOL  LOCATION: SAM1 initialization
Command ==> GO                               Scroll ==> PAGE
MONITOR +-----2-----3-----4-----5-----6- LINE: 0 OF 0
***** TOP OF MONITOR *****
***** BOTTOM OF MONITOR *****

SOURCE: SAM1 +---1---+---2---+---3---+---4---+---5---+ LINE: 1 OF 467
1 *****
2 * PROGRAM: SAM1
3 * Sample program for the ENTERPRISE COBOL Compiler
4 *
5 * AUTHOR : Doug Stout
6 * IBM PD TOOLS

LOG 0-----1-----2-----3-----4-----5----- LINE: 32 OF 34
0032 IBM Debug Tool Version 10 Release 1 Mod 0
0033 12/31/2009 00:44:02 AM
0034 5655-V50: Copyright IBM Corp. 1992, 2009
PF 1:?      2:STEP    3:QUIT     4:LIST     5:FIND
PF 7:UP     8:DOWN    9:GO      10:ZOOM    11:ZOOM LOG 12:RETRIEVE
```

In this example, the program will run to completion, since there are no breakpoints set



In this example, no breakpoints are set in the program, and a GO command is entered to run the program.

Messages from the program are displayed



```
SAM1 STARTED DATE = 12/08/09 (mm/dd/yy)
          TIME = 19:35:33
```

```
*** █
```

Note: These messages were generated by the program used in this example

Enter

25

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

Messages from the application program may be displayed on the TSO terminal. Press Enter to continue.

The program ran to completion, and the debugger ended



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'  
Command ==> _____ Scroll ==> PAGE RC 0  
  
Load Module Name SAM1  
Choose the format of your parameter string:  
 1 LE COBOL Default - Program Arguments / Run-time Options  
 2 Other LE Languages - Run-time Options / Program Arguments  
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments  
  
_ Enter / to modify parameters _____  
-----  
Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP  
-----  
***** Top of Data *****  
CEE0PTS 1 - SYSIN  
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR  
CUSRPT 1 - SYSOUT=*  
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR  
SYSOUT 1 - *  
SYSPRINT 1 - SYSOUT=*  
TRANFILE 1 - SYSIN  
***** Bottom of data *****
```

The program return code is displayed

F3

F3

The program ran to completion. When the program terminated, so did the debugger. The 'edit setup file' panel is displayed again, with the return code from the program.

F3 is pressed twice to return to the Debug Tool utilities menu.

Back to the Debug Tool Utilities menu



```
----- Debug Tool Utilities -----
Option ==> █
More: +
0 Job Card
  Create Job Card image.
1 Program Preparation
  Convert, compile, assemble or link edit program.
2 Debug Tool Setup File
  Manage setup files and start debug session in TSO foreground or batch.
3 Code Coverage
  Measure code coverage in programs.
4 IMS TM Setup
  Update Language Environment run-time options in IMS. Create message region.
5 Load Module Analyzer
  Analyze load modules and each CSECT in the load module.
6 Debug Tool User Exit Data Set
  Modify the data set used by user exit during program initialization.
```

The setup file was saved

The setup file was saved automatically. It can be re-used at any time to run and debug the program.

More about the setup file panels



- Next, you will see how to:
 - Display a list of available commands
 - Display a list of available prefix commands (line commands)
 - Specify detailed DD information
 - Create a concatenated DD statement (to reference multiple files)

Next, you will see how to display a list of available commands and line commands, specify detailed DD information, and how to specify concatenated DD statements.

Press F1 (help) to see a list of commands



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 7 of 7
Command ==> | Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSTRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - SYSOUT=*
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```



Pressing F1 from the 'edit setup file' panel will display help.

A list of commands available on the setup file panel



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 7 of 7
Command == E
Load Modul
Choose the
1 1 LE COB
2 Other
3 Non-LE
Enter /
Cmd DD Nam
CUSTFI
CUSTRP
STEPLI
SYSOUT
SYSPRI
TRANFI
*****

EDIT - Edit Setup File - Primary commands
Primary commands:
o Run - Allocate DD statements and run the program
o End - Save changes and exit this panel
o Cancel - Do not save changes and exit the edit session
o Copy - Copy setup information from JCL or Setup File
o Create - Create a new member from this Setup File
o Save - Save the current Setup File
o Submit - Create and submit JCL for batch execution
o ShowDD - Display only the DD statements.
Note: this command is not available when only
the DD statements are shown.
(Press HELP from this field-help panel to see help
for the full panel.)
**

F3
```

30

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

The help panel displays a list of available commands. You have already seen the RUN command used. It runs the setup file. COPY is used to copy information from another setup file that already exists or from JCL. Be aware that the SUBMIT command is available to automatically generate JCL from the setup file. The SUBMIT command lets you actually run JCL in batch. F3 returns from the help panel.

Use a slash (/) line command to see a list of line commands



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 8 of 8
Command ==> _____ Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
 / CEEOPTS 1 - SYSIN
 CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
 CUSRPT 1 - SYSOUT=*
 STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
 SYSOUT 1 - *
 SYSPRINT 1 - SYSOUT=*
 TRANFILE 1 - SYSIN
***** Bottom of data *****
```



A slash (/) line command displays a list of available line commands.

A list of available line commands is displayed



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'
Co                                     Row 1 to 8 of 8
                                     croll ==> PAGE
Modify Information for CUSTFILE
Option ==> █
Lo
Ch
1
1. SA Allocation      Specify Allocation information
2. SD DCB            Specify DCB      information
3. SS SMS            Specify SMS      information
4. SP Protection     Specify Protection information
5. SO Sysout         Specify Sysout   information
Cm 6. SX ALL (section) Specify all DD   information
7. SZ ALL (column)  Specify all DD   information
8. D Delete          Delete DD      information
/  9. I Insert        Insert DD      information
10.R Repeat          Repeat DD      information
11.E Edit            Edit Data set or Sysin contents
12.B Browse          Browse Data set or Sysin contents
**                                     *****
                                     *****
                                     F3
```

32

IBM Debug Tool for z/OS tutorial

© 2012 IBM Corporation

Here are the line commands you can use. There are several line commands that start with the letter S, such as SA and SD. These S line commands provide ways to specify more complex DD information than can be entered on the main panel.

Use a D line command to delete a DD statement, I to insert a new one, and R to repeat DD information to a new line to build a concatenation. If a DD refers to a sequential file, a PDS member, or in-stream data, you can use an E line command to edit the data in the file. A B line command is used to browse data in a file.

Press F3 to exit the line command help panel.

The SX line command displays all available DD information



```

EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'          Row 1 to 8 of 8
Command ==> |                                           Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy)  DISP
-----
***** Top of Data *****
  CEEOPTS  1  -  SYSIN
  CUSTFILE  1  -  'TSS16.ADLAB.CUST2'  SHR
  CUSRPT   1  -  SYSOUT=*
  STEPLIB  1  -  'TSS16.ADLAB.LOAD'  SHR
  SYSOUT   1  -  *
  SYSPRINT  1  -  SYSOUT=*
  TRANFILE  1  -  SYSIN
***** Bottom of data *****

```



On the 'edit setup file' panel, you can see the list of DDs. But only basic DD information, such as file name and disposition, can be entered. To specify more detailed DD information, use one of the line commands that begins with S. For example, here, an SX line command is entered on the CUSTFILE DD.

Result of SX (select all DD information) line command



```
Modify ALL Information for CUSTFILE in 'TSS16.ADLAB.DTSF(SAM1)'
Command ==>

DD Name . . CUSTFILE      Positional Parm . . .
DD Seq . . 1

Special Parameters
DSN . . 'TSS16.ADLAB.CUST2'
Status . . SHR           Normal Disp . . .      Abnormal Disp . . .
Space Unit. . .          Space Prim. . . .      Space Sec . . .
Space Dir . . .

Primary Parameters
ACCODE. . .             AVGREC. . .             BURST . . .
CCSID . . .             CHARS . . .            CHKPT . . .
CNTL . . .              COPIES. . .           DATACLAS. . .
DDNAME. . .            DEST. . .              DLM . . .
DSID. . .              DSNTYPE . . .          EXPDT . . .
FCB . . .              FILEDATA. . .          FLASH . . .
FREE . . .             HOLD. . .              KEYOFF. . .
LABEL . . .            LGSTREAM. . .          LIKE. . .
MGMTCLAS. . .          MODIFY. . .            OUTLIM. . .
OUTPUT. . .            PATH. . .              PATHDISP. . .
PATHMODE. . .          PATHOPTS. . .          PROTECT . . .
```

Similar to JCL, you can specify DDs that create, delete, and append to files

More: +

F8

The SX line command displays the 'All information' panel. From here, you can specify more complex DD information, similar to what can be done in JCL. For example, if you intend to have a new file created when the setup file runs, you can specify NEW in the 'Status' field, and specify allocation information including entries in the 'Space unit', 'Space prim'(ary), and other fields. There are so many options that they do not all fit on one panel. F8 is pressed to display more options.

Additional options available from the SX (select all DD information) line command



```

Modify ALL Information for CUSTFILE in 'TSS16.ADLAB.DTSF(SAM1)'
Command ==> █

DD Name . . CUSTFILE      Positional Parm . . . _____
DD Seq . . 1

Special Parameters
DSN . . 'TSS16.ADLAB.CUST2'
Status . . SHR           Normal Disp . . _____ Abnormal Disp . . _____
Space Unit. _____ Space Prim. . . _____ Space Sec . . . . _____
Space Dir . . _____

More: - +

PATHMODE. . . _____ PATHOPTS. . . _____ PROTECT . . . _____
QNAME . . . _____ REORG. . . _____ REFDD . . . _____
REF . . . _____ RETPD . . . _____ RLS . . . _____
SECMODEL. . . _____ SEGMENT . . . _____ SER . . . _____
SPIN. . . _____ STORCLAS. . . _____ SUBSYS. . . _____
SYSOUT. . . _____ TERM. . . _____ UCS . . . _____
UNIT. . . _____

DCB Parameters
BFALN . . . _____ BFTEK . . . _____ BLKSIZE . . . _____
BLKSZLIM. . . _____ BUFIN . . . _____ BUFL . . . _____
BUFMAX. . . _____ BUFNO . . . _____ BUFOFF. . . _____
    
```



The second 'All information' panel is displayed. Notice that DCB information can be specified here. F8 is pressed.

Additional options available from the SX (select all DD information) line command



```
Modify ALL Information for CUSTFILE in 'TSS16.ADLAB.DTSF(SAM1)'
Command ===>

DD Name . . CUSTFILE      Positional Parm . . .
DD Seq . . 1

Special Parameters
DSN . . 'TSS16.ADLAB.CUST2'
Status . . SHR           Normal Disp . . .      Abnormal Disp . . .
Space Unit. . .          Space Prim. . . .      Space Sec . . . .
Space Dir . . .

More: - +
BUFMAX. . .             BUFNO . . .             BUFOFF. . .
BUFOUT. . .             BUFSIZE . . .          CPRI . . .
CYLOFL. . .             DEN . . .              DIAGNS. . .
DSORG . . .             EROPT . . .           FUNC. . .
GNCP. . .              INTVL . . .           IPLTXID . . .
KEYLEN. . .            LIMCT . . .           LRECL . . .
MODE. . .              NCP . . .            NTM . . .
OPTCD . . .            PCI . . .            PRTSP . . .
RECFM . . .            RESERVE . . .         RKP . . .
STACK . . .            THRESH. . .          TRTCH . . .

AMP Parameters
```



The third 'All information' panel is displayed showing more options. F8 is pressed.

Additional options available from the SX (select all DD information) line command



```

Modify ALL Information for CUSTFILE in 'TSS16.ADLAB.DTSF(SAM1)'
Command ==> █

DD Name . . CUSTFILE      Positional Parm . . . _____
DD Seq . . 1

Special Parameters
DSN . . 'TSS16.ADLAB.CUST2'
Status . . SHR           Normal Disp . _____ Abnormal Disp . _____
Space Unit. _____ Space Prim. . _____ Space Sec . . . _____
Space Dir . _____

More: -
KEYLEN. . . _____ LIMCT . . . _____ LRECL . . . _____
MODE. . . _____ NCP . . . _____ NTM . . . _____
OPTCD . . . _____ PCI . . . _____ PRTSP . . . _____
RECFM . . . _____ RESERVE . . . _____ RKP . . . _____
STACK . . . _____ THRESH. . . _____ TRTCH . . . _____

AMP Parameters
ACCBIAS . . _____ BUFND . . . _____ BUFNI . . . _____
BUFSP . . . _____ CROPS . . . _____ RMODE31 . . . _____
SMBDFR. . . _____ SMBHWT. . . _____ SMBVSP. . . _____
STRNO . . . _____ SYNAD . . . _____ TRACE . . . _____
    
```



And the fourth and final 'All information' panel is displayed. Specify options as needed for your files. Press F3 to return.

Use an R (repeat) line command to concatenate a file in a DD

```

EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'          Row 1 to 8 of 8
Command ==> |                                           Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy)  DISP
-----
***** Top of Data *****
CEEOPTS 1 - SYSIN
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
R STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****

```

Enter

Next, you will see how to repeat a DD statement, and how to code a concatenation. To repeat a DD, type an R line command next to it, and press enter.

Result of the R (repeat) line command



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 9 of 9
Command ==> PAGE
Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments
_ Enter / to modify parameters

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CEEOPTS 1 - SYSIN
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
         2 - 'TSS16.ADLAB.LOAD' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
*****
```

The file information was repeated as a concatenation in the same DD. Notice the 'Seq' (sequence) number.

The DD was repeated. Notice that an R line command adds the repeated line as a concatenation of the DD. In this case, the R line command was specified on the STEPLIB DD. Now, STEPLIB is a concatenation of two files.

Overtyping DD information to change it



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)' Row 1 to 9 of 9
Command ==> | Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name Seq C DD Information (DSN/Sysin/Sysout/Dummy) DISP
-----
***** Top of Data *****
CEEOPTS 1 - SYSIN
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
          2 - 'TSS16.ADLAB.LOAD2' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```

The file name, DD name, and other data can be changed as needed

Now the name of the second file is changed. Notice that there is a '2' in the sequence field of the second file in the concatenation. You can create concatenations of multiple files in DDs, and you can change the number in the sequence field to change the order of the files in the concatenation.

If you do not want the second file to be a concatenation, that is, you want it to be its own DD, then type a DD name next to the file in the 'DD name field'.

Use a SAVE command to save the setup file



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'          Row 1 to 9 of 9
Command ==> SAVE                                     Scroll ==> PAGE

Load Module Name SAM1
Choose the format of your parameter string:
 1 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE COBOL - Run-time Options / Program Arguments

_ Enter / to modify parameters _____

Cmd DD Name  Seq C DD Information (DSN/Sysin/Sysout/Dummy)  DISP
-----
***** Top of Data *****
CEEOPTS  1  -  SYSIN
CUSTFILE 1  -  'TSS16.ADLAB.CUST2'  SHR
CUSRPT  1  -  SYSOUT=*
STEPLIB  1  -  'TSS16.ADLAB.LOAD'  SHR
          2  -  'TSS16.ADLAB.LOAD2' SHR
SYSOUT   1  -  *
SYSPRINT 1  -  SYSOUT=*
TRANFILE 1  -  SYSIN
***** Bottom of data *****
```



A setup is saved automatically when you press F3 to exit. But you can save a setup file without exiting. Type SAVE on the command line, and press Enter.

The setup file was saved



```
EDIT - Edit Setup File 'TSS16.ADLAB.DTSF(SAM1)'
Command ==> _____ Scroll ==> Saved PAGE
Load Module Name SAM1
Choose the format of your parameter string:
 1 1 LE COBOL Default - Program Arguments / Run-time Options
 2 Other LE Languages - Run-time Options / Program Arguments
 3 Non-LE Programs / Non-LE Languages / Program Arguments
_ Enter / to modify parameter string

Cmd DD Name Seq C DD Informa DISP
-----
***** Top of Data *****
CEEOPTS 1 - SYSIN
CUSTFILE 1 - 'TSS16.ADLAB.CUST2' SHR
CUSRPT 1 - SYSOUT=*
STEPLIB 1 - 'TSS16.ADLAB.LOAD' SHR
          2 - 'TSS16.ADLAB.LOAD2' SHR
SYSOUT 1 - *
SYSPRINT 1 - SYSOUT=*
TRANFILE 1 - SYSIN
***** Bottom of data *****
```

Note: the setup file is saved automatically when you exit with F3. Use a CANCEL command to exit without saving.

And the setup file is saved. Once you have created a setup file, you can use it again later when you need it.

That is the end of this section, using the 'Debug Tool setup file' utility to run and debug a batch application in your TSO session.

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_DTv12s10DebuggingBatchInTSO.ppt

This module is also available in PDF format at: [../DTv12s10DebuggingBatchInTSO.pdf](..../DTv12s10DebuggingBatchInTSO.pdf)

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



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.