

IBM IMS Batch Terminal Simulator for z/OS
4.1

User's Guide



Note:

Before using this information and the product it supports, read the information in [“Notices” on page 309.](#)

Fifth Edition (July 2022)

This edition applies to Version 4.1 of IBM IMS Batch Terminal Simulator for z/OS (program number 5655-BT4) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC19-3230-03.

© **Copyright International Business Machines Corporation 1982, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- About this information..... vii**

- Part 1. IMS Batch Terminal Simulator overview..... 1**

- Chapter 1. Introduction to IMS Batch Terminal Simulator..... 3
 - What's new in IMS Batch Terminal Simulator.....3
 - What is IMS Batch Terminal Simulator?.....5
 - Features and benefits.....7
 - Architecture and process flow..... 9
 - Supported applications and devices.....14
 - What's new in IMS Batch Terminal Simulator 4.1..... 15
 - Functional changes in IMS Batch Terminal Simulator 4.1.....15
 - Roadmap to IMS Batch Terminal Simulator information.....16
 - Service updates and support information..... 16
 - Product documentation and updates..... 16
 - Accessibility features..... 17

- Chapter 2. Key IMS Batch Terminal Simulator concepts.....19
 - IMS Batch Terminal Simulator terminology..... 19
 - Load modules and system flow.....20
 - Determining the types of IMS region..... 22
 - Runtime modes..... 24
 - Debugging capabilities..... 25
 - Support for IBM Debug Tool for z/OS and IBM z/OS Debugger.....26
 - Supported IMS calls..... 27
 - IMS call function codes..... 27
 - IMS call limitations..... 30
 - Formatting modes..... 33
 - Full screen image support (FSS)..... 35
 - FSS overview..... 35
 - FSS operation flow..... 36
 - FSS limitations..... 37

- Part 2. Installation and configuration..... 39**

- Chapter 3. Hardware and software prerequisites.....41

- Chapter 4. Configuring IMS Batch Terminal Simulator..... 43
 - Configuration checklist.....43
 - Customizing the BTS cataloged procedure..... 45
 - Customizing the BTS TSO command list (CLIST)..... 46
 - Customizing for IBM Debug Tool for z/OS or IBM z/OS Debugger..... 46
 - Customizing for specific applications..... 47
 - Customizing for IMS REXX applications.....47
 - Customizing for IMS Java applications..... 47
 - Customizing for IMS applications that access Db2 databases.....47
 - Customizing for IMS applications that access IBM MQ..... 48
 - Implementing IMS user-written routines..... 49
 - Configuring for IMS callable services..... 50
 - Setting defaults for environment parameters (./E commands)..... 51

Setting up the tutorial environment.....	52
Recording the resource information.....	52
Distributed Access Infrastructure configuration.....	53
Installing Eclipse interface.....	54
Installing the IMS Batch Terminal Simulator resource adapter.....	54
Adding and configuring the IMS Batch Terminal Simulator resource adapter J2C connection factory.....	55

Part 3. Testing applications.....57

Chapter 5. Restrictions and considerations.....	59
Restrictions.....	59
Considerations.....	60
Chapter 6. Testing applications in batch mode.....	65
Tutorial: Running the sample application in batch mode.....	65
Modifying the sample JCL and running a job.....	66
Running applications in batch mode.....	68
Step 1: Specifying the JOB and the EXEC statements.....	68
Step 2: Specifying the IMS Batch Terminal Simulator data sets.....	69
Step 3: Specifying the IMS data sets.....	70
Step 4: Defining the transactions and the simulation environment.....	72
Step 5: Specifying the simulator statements.....	74
Debugging applications in batch mode.....	75
Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger.....	76
Debugging with simulator commands.....	76
Debugging with VS COBOL II interactive debug.....	77
Chapter 7. Testing applications in interactive mode.....	79
Tutorial: Running the sample application in interactive mode.....	79
Step 1: Modifying the sample JCL.....	80
Step 2: Running the session.....	81
Running applications in interactive mode.....	86
Step 1: Specifying the TSO CLIST parameters.....	87
Step 2: Specifying the IMS Batch Terminal Simulator data sets.....	88
Step 3: Specifying the IMS data sets.....	90
Step 4: Defining the transactions and the simulation environment.....	91
Step 5: Starting the session.....	93
Step 6: Entering the simulation input.....	94
Debugging applications in interactive mode.....	98
Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger.....	98
Debugging with simulator commands.....	99
Debugging with the TSO TEST monitor.....	100
Debugging with VS COBOL II interactive debug.....	100
Tips for testing in interactive mode with FSS.....	102
Chapter 8. Tutorial: Testing an application from Eclipse.....	105
Step 1: Creating a server runtime environment.....	105
Step 2: Defining a launch configuration.....	107
Step 3: Testing an application in Eclipse.....	108
Step 4: Exporting a server runtime (optional).....	109
Step 5: Exporting a server runtime environment and launch configuration (optional).....	109
Chapter 9. Testing an application that uses the IMS Batch Terminal Simulator resource adapter.....	111
Transaction Manager resource adapter supported functions.....	111
Chapter 10. Customizing JCL for specific applications.....	113

JCL customization by application languages.....	113
Applications written in REXX.....	113
Applications written in Java.....	113
JCL customization by application types.....	113
Applications that access Db2 databases.....	114
Applications that access IBM MQ.....	116
Applications that access DEDBs.....	118
Applications that trigger secondary transactions.....	118
3270 applications.....	121
APPC applications.....	122
 Chapter 11. Including BTS optional user-written exit routines.....	125
Applying output formatting modules.....	125
Applying an attention exit routine.....	128
User exit interface for SAIL/ESA.....	128
User exit (BTSTSSE0) interface for TSS call.....	130
Application front-end routine.....	130
Extended user exit (BTSEEXIT) interface.....	131
 Chapter 12. Interpreting the simulation output.....	139
Analyzing the output listing.....	139
DL/I call trace listing.....	142
Db2 call trace listing.....	144
MQI call trace listing.....	148
Program analysis reports.....	149
 Part 4. Playback utility.....	153
 Chapter 13. Transaction screen formatting (Playback utility).....	155
Playback utility overview.....	155
Playback utility modules and process flow.....	156
Playback utility limitations and considerations.....	158
Creating screen images of transaction activities.....	161
Restarting the Playback utility after BTS0004W error.....	163
Tips for using the Playback utility.....	163
Input for the Playback utility.....	165
BTSARCN0 JCL requirements.....	165
BTSALOG0 JCL requirements.....	168
BTSAPRE0 JCL requirements.....	171
BTSAISR0 JCL requirements.....	174
B TSAEDT0 JCL requirements.....	176
Syntax rules for the control statements.....	177
Playback utility reports.....	178
BTSARCN0: RECON Query Summary report.....	178
BTSALOG0: Summary report.....	179
BTSAPRE0: Summary report.....	180
BTSAISR0: Summary report.....	181
B TSAEDT0: Screen Image Summary report.....	182
B TSAEDT0: Sample formatted screen.....	183
 Part 5. Reference.....	185
 Chapter 14. Command reference.....	187
Simulator commands.....	187
Simulator command summary.....	187
Syntax rules.....	191
./C command.....	192

./D command.....	193
./E command.....	198
./O command.....	205
./P command.....	212
./R command.....	217
./RETURN command.....	218
./S command.....	218
./SPA command.....	220
./T command.....	222
./* command.....	235
Debug commands.....	235
Simulator statements.....	239
Unformatted-mode simulator statements.....	240
Formatted-mode simulator statements.....	241
IMS commands.....	247
IMS /EXIT command.....	247
IMS /FORMAT command.....	247
IMS /RESET command.....	248
IMS /SET command.....	248
Chapter 15. Device feature indicator values.....	249
Chapter 16. BTS interface with a user-written BTSPIXT0 routine.....	251
Chapter 17. Playback utility record format.....	253
Extract file record and CROSSREF file record formats.....	253
MSGLIST file record format.....	255
Chapter 18. How to read syntax diagrams.....	257
Part 6. Troubleshooting.....	259
Chapter 19. Troubleshooting IMS Batch Terminal Simulator problems.....	261
Tips for troubleshooting IMS Batch Terminal Simulator problems.....	261
Gathering diagnostic information.....	261
Chapter 20. Messages and codes.....	263
IMS Batch Terminal Simulator return codes.....	263
IMS Batch Terminal Simulator abend codes.....	263
IMS Batch Terminal Simulator messages.....	268
IMS Batch Terminal Simulator resource adapter messages.....	291
Playback utility return codes.....	302
Playback utility abend codes.....	302
Playback utility messages.....	303
Notices.....	309
Index.....	313

About this information

IBM® IMS Batch Terminal Simulator for z/OS® (also referred to as IMS Batch Terminal Simulator or BTS) is a simulation tool that provides comprehensive services to test and debug IMS application and database activity.

These topics provide instructions for system administrators to install and configure IMS Batch Terminal Simulator, and application programmers to test applications by using IMS Batch Terminal Simulator.

These topics are designed to help technical support personnel, database administrators, system programmers, application programmers, and system operators who are involved in checking and debugging IMS application and database activities perform these tasks:

- Install and operate IMS Batch Terminal Simulator
- Customize your IMS Batch Terminal Simulator environment
- Check and debug IMS applications and database activities
- Simulate operations of IMS applications in a TSO environment and in a batch environment
- Run and test IMS applications from a development environment with the Eclipse interface
- Test applications before running them on production environments
- Diagnose and recover from IMS Batch Terminal Simulator problems

To use these topics, you should have a working knowledge of:

- The z/OS operating system
- ISPF
- SMP/E

Always check the IMS Tools Product Documentation page for complete product documentation resources:

<https://www.ibm.com/support/pages/node/712955>

The IMS Tools Product Documentation page includes:

- Links to [IBM Documentation](#) for the user guides ("HTML")
- Links to the PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guide, known as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, known as "Tech notes"
- White papers that describe product business scenarios and solutions

Part 1. IMS Batch Terminal Simulator overview

IBM IMS Batch Terminal Simulator for z/OS (also referred to as IMS Batch Terminal Simulator or BTS) is a simulation tool that provides comprehensive services to test and debug IMS applications.

Topics:

- [Chapter 1, “Introduction to IMS Batch Terminal Simulator,” on page 3](#)
- [Chapter 2, “Key IMS Batch Terminal Simulator concepts,” on page 19](#)

Chapter 1. Introduction to IMS Batch Terminal Simulator

IBM IMS Batch Terminal Simulator for z/OS (also referred to as IMS Batch Terminal Simulator or BTS) is a comprehensive testing and debugging tool that helps you to ensure that your IMS programs operate correctly.

Topics:

- [“What's new in IMS Batch Terminal Simulator” on page 3](#)
- [“What is IMS Batch Terminal Simulator?” on page 5](#)
- [“Features and benefits” on page 7](#)
- [“Architecture and process flow” on page 9](#)
- [“Supported applications and devices” on page 14](#)
- [“What’s new in IMS Batch Terminal Simulator 4.1” on page 15](#)
- [“Service updates and support information” on page 16](#)
- [“Product documentation and updates” on page 16](#)
- [“Accessibility features” on page 17](#)

What's new in IMS Batch Terminal Simulator

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

Revision markers do not necessarily indicate all the changes made to the information because deleted text and graphics cannot be marked with revision markers.

SC19-3230-04 (July 2022)

Description	Related APARs
Extended user exit (BTSEEXIT) interface enhancement. BTSEEXIT is dynamically loaded when it is not found in the BTS load module BTSRC000. The following topic is updated: “Extended user exit (BTSEEXIT) interface” on page 131	PH46987

SC19-3230-03 (January 2022)

Description	Related APARs
Support for the extended user exit (BTSEEXIT) interface and DLI RLSE calls. The following topics are added: <ul style="list-style-type: none">• “Extended user exit (BTSEEXIT) interface” on page 131• Abend code “4074” on page 264	PI98530

SC19-3230-02 (January 2018)

Description	Related APARs
BTSISRIO DD support: A new DD statement, BTSISRIO. This optional DD statement specifies the output data set for the data that the DLI ISRT call inserts to IOPCB. For more information, see the following topics: <ul style="list-style-type: none">• Batch mode: “Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 69• Interactive mode: “Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 88	PI88305
IMS 15 support and other enhancements: <ul style="list-style-type: none">• IMS Batch Terminal Simulator has been enhanced to support IMS 15.• A new keyword, LINE, for the ./D command. This keyword specifies how the input line is used. For more information see “./D command” on page 193.	PI72833
UTF8 to EBCDIC conversion support for IFI information A new keyword, IFIUTF, for the ./E command. This keyword converts the IFI information from UTF8 to EBCDIC when Db2 [®] subsystem parameter UIFCIDS=YES is specified. For more information, see “./E command” on page 198 .	PI63751
IMS-managed ACBs support and other enhancements: <ul style="list-style-type: none">• Batch mode: “Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 69 (BTSACB DD)• Interactive mode: “Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 88 (BTSACB DD)• “Applications that access DEDBs” on page 118• “DL/I call trace listing” on page 142• “./O command” on page 205 (ICALOLEN keyword)	PI50185
IMS 14.1 support	PI27382
New keywords: <ul style="list-style-type: none">• A new keyword, ENDROLB, for the ./E command. This keyword rolls back the database updates made by the application. For more information see “./E command” on page 198.• A new keyword, PERMLOAD, for the ./E command. This keyword keeps the application loaded. For more information see “./E command” on page 198.• A new keyword, TRXLINK, for the CNTLCRDS DD of the Playback module BTSAPRE0. This keyword processes the records associated with the specified transaction code. For more information, see “BTSAPRE0 JCL requirements” on page 171.	PM91061

Description	Related APARs
IMS 13.1 support	PM70739
IMS catalog database support and other enhancements: <ul style="list-style-type: none"> • Support for IMS catalog databases (GUR call) • A new operand, ALL, for the ELAPTIME keyword of the ./O command. This operand requests to include the elapsed time and the start timestamp of each call in the output. For more information, see “./O command” on page 205. 	PM61865

What is IMS Batch Terminal Simulator?

IMS Batch Terminal Simulator provides a comprehensive means of checking application program logic, IMS application interfaces, teleprocessing activity, 3270 format control blocks, and database activity. It simulates the behavior of applications and generates reports that contain detailed information about the processed transactions.

IMS Batch Terminal Simulator supports the running of IMS Data Base/Data Communications (DB/DC) applications through Time Sharing Option (TSO) CLIST or as MVS™ batch jobs. It also supports the running of IMS applications that access Db2 databases by using IMS Attachment Facility for BMP or JBP mode and Db2 DL/I Batch Support for DLI or DBB mode. IMS Batch Terminal Simulator can be used to run IMS Fast Path, BMP, MSG, and batch applications, and conversational or nonconversational teleprocessing applications. IMS applications that are written in Assembler, COBOL, PL/I, C, C++, REXX, and Java™ are supported. The user can specify the type of IMS region (DLI, DBB, BMP, or JBP) to run these IMS applications. Multiple IMS application programs can be processed in a single job step.

The operations of IMS Batch Terminal Simulator are transparent to the application. No changes to IMS code, control blocks, libraries, or application load modules are necessary. Directly or through IMS, IMS Batch Terminal Simulator utilizes the IMS ACB or PSB and DBD, and PGM libraries (ACBLIB, PSBLIB, DBDLIB, and PGMLIB).

IMS Batch Terminal Simulator intercepts each application program call to IMS and reports pertinent details related to the call, such as the segment search argument, scratchpad area, key feedback area, and I/O area. It provides statistics that are useful in uncovering system resource contention problems and in evaluating the impact of new applications on an IMS system. These statistics include the number of each type of DL/I call made against each program communication block (PCB).

By supporting IMS user-written transaction code (input) editing, providing for dynamic modification of the contents of main storage, and accepting ABEND and SNAP DUMP requests, IMS Batch Terminal Simulator facilitates the comprehensive testing and debugging of the application program logic.

Subsections:

- [“Terminal simulation”](#) on page 5
- [“Application activity summary”](#) on page 6
- [“3270 formatting support”](#) on page 6
- [“Full screen image support \(FSS\)”](#) on page 6
- [“Terminal action support for 3270 applications”](#) on page 6
- [“IMS command support”](#) on page 6
- [“Powerful debugging facilities”](#) on page 7

Terminal simulation

With IMS Batch Terminal Simulator, transaction message input from the terminal and output to the terminal are simulated in an IMS batch environment (DLIBATCH, DBBBATCH, BMP, or JBP), which means that system administrators do not need to configure terminals for each testing environment.

Application activity summary

IMS Batch Terminal Simulator produces a report that shows all activities during the run. This report is also available when the program is run under TSO. At the end of the job, IMS Batch Terminal Simulator can produce a sequential data set that contains all data input to IMS Batch Terminal Simulator during the TSO session. You can use this data set to create a regression test file for a subsequent run.

3270 formatting support

IMS Batch Terminal Simulator allows the formatting of 3270 input and output during the simulation. This feature provides a means for testing the format control blocks that are generated by the IMS Message Format Service utility. By either implicitly or explicitly specifying the position on the display screen, field-oriented input can be passed to the application program through the IMS Message Format Service. Screen images of the terminal input and output are printed including field attribute characters if requested.

When 3270 formatting is specified, IMS Batch Terminal Simulator output includes the following application input and output data:

- The contents of the 3270 screen upon receiving a /FORMAT command
- The input specifications made by the terminal operator to simulate keyboard entries, function keys, selector pen, or operator identification card reader
- The message segments passed to and from the application program to service GU, GN, ISRT, and PURG calls
- The first page of each output message and additional pages if requested

Full screen image support (FSS)

With IMS Batch Terminal Simulator full screen image support (also referred to as FSS), IMS Batch Terminal Simulator can be run in the TSO foreground. IMS Batch Terminal Simulator uses the TSO terminal as an IMS screen image for formatted output from the application program. It also allows you to enter new data from the IMS formatted screen image and, upon request, writes a trace of application program calls to the TSO terminal. During this trace, you can stop running the application program and enter the TSO TEST monitor.

Terminal action support for 3270 applications

In addition to the functional keys (CLEAR, ENTER, PA1 through PA3, PFK1 through PFK24, and ERASE (ERASIN)), the selector pen (PEN), operator identification card reader (IDCARD), and a request for all subsequent pages of an output message to be displayed (PAX) can also be specified in the simulator statements of IMS Batch Terminal Simulator. Tab, back tab, and new line commands can be simulated by implicit or explicit position references.

ERASE EOF is automatically assumed for short input fields, and RESET is simulated automatically after each keyboard lock situation. The INS MODE (Insert Mode), DUP (Duplicate), DEL (Delete Character), and TEST REQ (Test Request) functions are not supported.

IMS command support

The /EXIT, /FORMAT, /SET, and /RESET commands of IMS can be included in the input stream for an application. The /FORMAT command, which causes a specific format to be displayed on a physical terminal by using the IMS Message Format Service, can be especially useful during the early stages of application development when an error in the application program logic or format control block specification exists. These types of errors can cause misinterpretation of subsequent terminal activity for that application.

Powerful debugging facilities

TSO TEST is a powerful interactive test facility for application programs that are written in Assembler Language. Under TSO TEST, you can investigate the program status by displaying registers and storage, make temporary modifications, and resume the program run.

You can also use IBM Debug Tool for z/OS or IBM z/OS Debugger (Debug Tool) to debug your application program. Debug Tool helps you test programs and examine, monitor, and control the execution of application programs that are written in assembler, C, C++, COBOL, or PL/I on a z/OS system. You can use Debug Tool to debug your programs in batch mode, interactively in full-screen mode, or in remote debug mode by using a workstation user interface. When you use Debug Tool with IBM Rational® Developer for System z® (Rational Developer) or IBM Developer for z Systems®, you can debug your application through the graphical source-level debugger that the developer tools provide.

Features and benefits

IMS Batch Terminal Simulator helps system administrators and application programmers improve productivity and efficiency in a cost and time-saving manner.

IMS Batch Terminal Simulator offers the following features and benefits:

- Increased application programmer productivity through comprehensive interaction with an application during testing, producing information that is not available from an online execution.
- Reduced IMS resource requirements, significantly reducing system administrator tasks to set up test environments.
- Reduced cost and reduced MIPS consumption by providing a lightweight testing environment compared to testing with full IMS resources.
- Run and test IMS applications from a development environment with the Eclipse interface. You can use common Eclipse wizards to set up runtime parameters and application libraries to run IMS Batch Terminal Simulator.
- Test IMS applications that are started through the IMS Transaction Manager Resource Adapter (TMRA) without requiring an online IMS.
- Transparent testing of application programs.
 - No changes to IMS code, control blocks, libraries, or application load modules are required.
 - Flexible application test periods make application program testing easier to schedule and perform.
 - Offers a more stable online system in both test and production environments. After applications have been tested with IMS Batch Terminal Simulator, they are much less likely to cause serious problems to the system when the applications are put online.
 - Works as a regression test tool for testing modifications to applications as well as modifications to the system, including new releases.
 - Multiple transactions can be processed in one step.
 - Provides a means of accessing DL/I, Db2 databases, and IBM MQ from a TSO terminal.
- Generates additional information about the program because IMS Batch Terminal Simulator traces all IMS application interactions and prints 3270 input and output formats.
- Integrates with IBM Debug Tool for z/OS, IBM z/OS Debugger, IBM Rational Developer for System z, and IBM Developer for z Systems to provide an integrated application testing and debugging environment.
- Can be used as a tool for training personnel in the internal and external operation of an application.

You can use IMS Batch Terminal Simulator to reduce many complex configuration and testing tasks that are typically required to test IMS applications. IMS Batch Terminal Simulator especially helps you with the following tasks:

- [“Reducing IMS resource requirements for online application testing” on page 8](#)
- [“Testing multiple applications at the same time” on page 8](#)
- [“Recording the terminal output image” on page 8](#)

- [“Running applications in a development environment” on page 9](#)
- [“Testing applications that use the IMS TM resource adapter” on page 9](#)
- [“Providing training environments” on page 9](#)

Reducing IMS resource requirements for online application testing

By using IMS Batch Terminal Simulator, application programmers can run IMS online applications in TSO environments or as MVS batch jobs without starting an IMS control region. Transaction message input from the terminal and output to the terminal are simulated in an IMS batch environment (DLIBATCH or DBBBATCH), which means that system administrators do not need to define IMS control regions and configure terminals for each testing environment.

To simulate IMS applications that access MSDB or DEDB with IMS Batch Terminal Simulator, an IMS control region must be active. However, MPP and IFP applications can be run in the BMP region (online batch region), and the terminal input and output can be simulated under the BMP region. This capability allows application programmers to test the logic of MPP and IPF applications without defining terminals for each testing environment.

Application programs can function under IMS Batch Terminal Simulator without modification. Also, IMS definitions such as application names, transaction codes, and ACB libraries do not need to be modified for IMS Batch Terminal Simulator.

By enabling the batch execution of online applications and by using the IMS system definitions and terminal definitions that are used in the production environment as is, IMS Batch Terminal Simulator can concurrently run many test cases without preparing many IMS control regions and terminals that were traditionally required to establish test environments for application testing.

Testing multiple applications at the same time

By using IMS Batch Terminal Simulator, you can define the transaction messages and the runtime environment parameters for the application in the input stream BTSIN (SYSIN input.)

Without IMS Batch Terminal Simulator, changes to the definition of the transaction codes, application program identifiers (PSB names), and application names require you to redefine the IMS system definition. However, with IMS Batch Terminal Simulator, you can directly specify these changes in BTSIN to dynamically reflect the changes at run time. This capability is especially useful for testing temporary application fixes.

IMS Batch Terminal Simulator allows you to switch PSBs within an IMS Batch Terminal Simulator job to test the integration capabilities of applications. You can define transaction message input for multiple applications in a BTSIN data set and test the integration capabilities between the applications. By using these capabilities, application programmers can test their applications more efficiently by reducing the testing overhead.

Recording the terminal output image

For each transaction message that is processed, the terminal output image is written to BTSOUT (a sequential data set or SYSOUT stream) in text format. In BTSOUT, the input transaction messages and the response message to the terminal are written in chronological order. Optionally, trace information of the internal process is also written in BTSOUT. For applications that use IMS Message Format Service (MFS), application programmers can specify, in BTSIN, to print the screen images of the output terminal in BTSOUT.

You can keep the output that is recorded in BTSOUT as a test execution log during functional testing, and you can use that log later to compare the results in regression testing.

When you debug the application in unit testing or when you find a defect in functional testing or in regression testing, the trace information in BTSOUT can help you identify and analyze the error.

Running applications in a development environment

IMS Batch Terminal Simulator includes an Eclipse interface that helps application programmers work more independently. Traditionally, application programmers, such as a COBOL or PLI programmer, must rely on the expertise of a system programmer or database administrator to test their applications. The system programmer or database administrator would help customize and run the necessary JCL for the application developers.

Now, by using the Eclipse interface, application developers can use standard Eclipse wizards that help them specify runtime parameters, library definitions, and data definitions. These definitions can be exported and reused by other developers who use the same test environment. Also, developers can define custom BTSIN input and program libraries that are specific to their applications as part of the Eclipse launch configurations. The BTSIN specifies transaction messages and runtime environment parameters for the application.

Testing applications that use the IMS TM resource adapter

IMS Batch Terminal Simulator provides a resource adapter that you can use instead of the IMS Transaction Manager resource adapter (TMRA) to test TMRA client applications. By using the IMS Batch Terminal Simulator resource adapter (BTSRA), you can test and debug TMRA client applications without fully using IMS resources. Instead, the application is tested on IMS Batch Terminal Simulator by using the JCL DD statements and parameters that are specified in an XML file, which is exported from the Eclipse interface.

Providing training environments

IMS Batch Terminal Simulator can also be used as a training tool for new application programmers. Application programmers can use IMS Batch Terminal Simulator to learn the functions and the behavior of their application programs by studying the test scenarios that are described in BTSIN and the activity trace information that is recorded in BTSOUT.

Typical development activities include enhancement or modification to in-use applications rather than developing new applications. New programmers must learn and understand the behavior of their applications before they participate in development projects.

With the set of simulation functions that IMS Batch Terminal Simulator provides, the system administrator no longer needs to establish IMS environments for exclusive use in education or training activities. IMS libraries can be reused from the production environment, and each trainee can use independent batch jobs to learn the behavior of their application programs.

Architecture and process flow

IMS Batch Terminal Simulator simulates the functions of IMS and simulates the activities of IMS applications without starting an IMS control region.

Subsections:

- [“Architecture” on page 9](#)
- [“Process flow” on page 11](#)
- [“Distributed architecture” on page 12](#)

Architecture

The following figure shows the typical structure of an IMS online system environment.

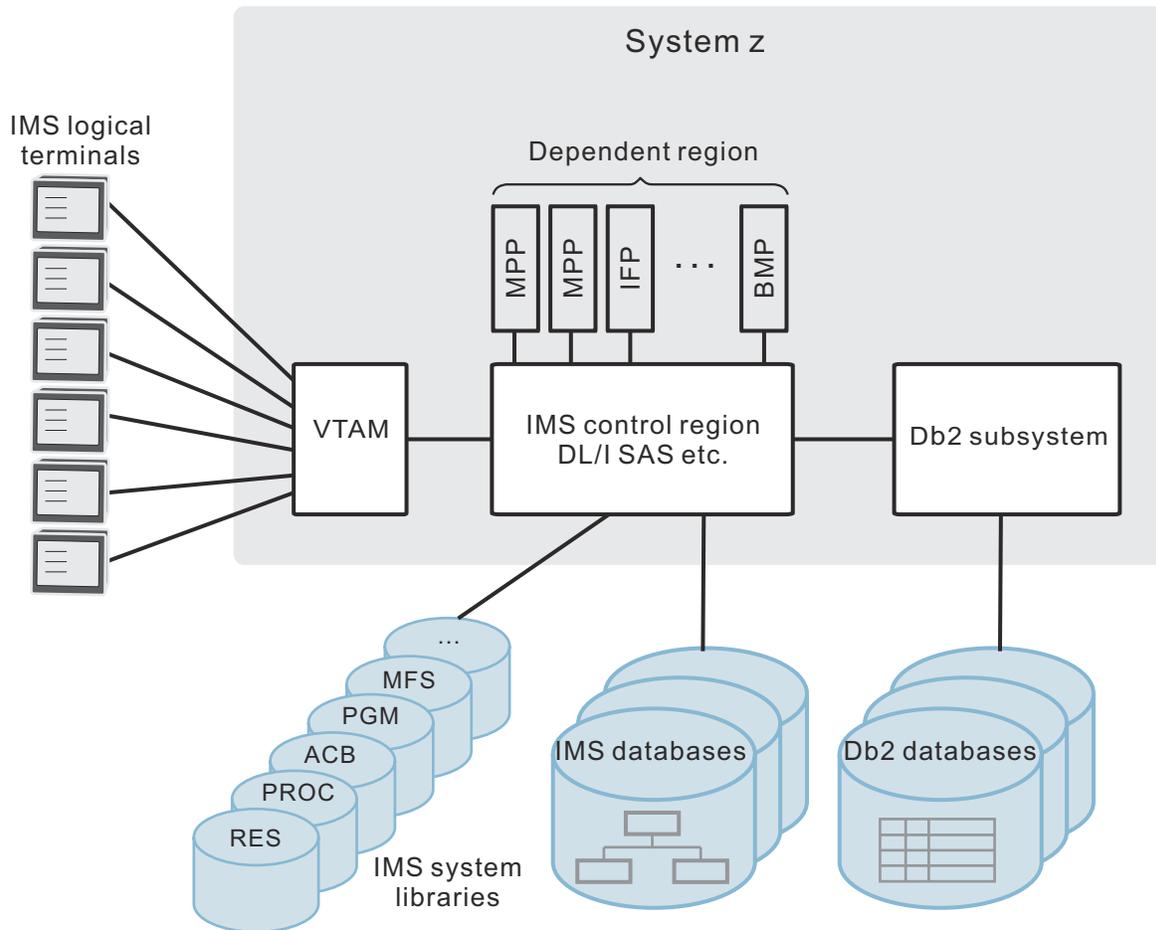


Figure 1. Typical structure of an IMS online system environment

Typically, when IMS Batch Terminal Simulator is not used, the first step to run IMS online applications is to start the IMS control region. For message-driven applications, you must also prepare and define one or more terminals to your IMS system to enter input for the transaction and to receive output from the transaction.

However, when IMS Batch Terminal Simulator is used, you can run IMS applications without starting an IMS control region or defining input and output terminals to your IMS system. The ability to test your applications without using full IMS resources significantly reduces the overhead that is typically required for preparing testing environments, and reduces cost and MIPS consumption.

IMS Batch Terminal Simulator simulates the following functions of IMS:

- Input and output terminals for IMS transactions
- A part of transaction management and database management capabilities of IMS
- Terminal session management of VTAM®

The following figure shows the structure of the system environment when IMS Batch Terminal Simulator is used.

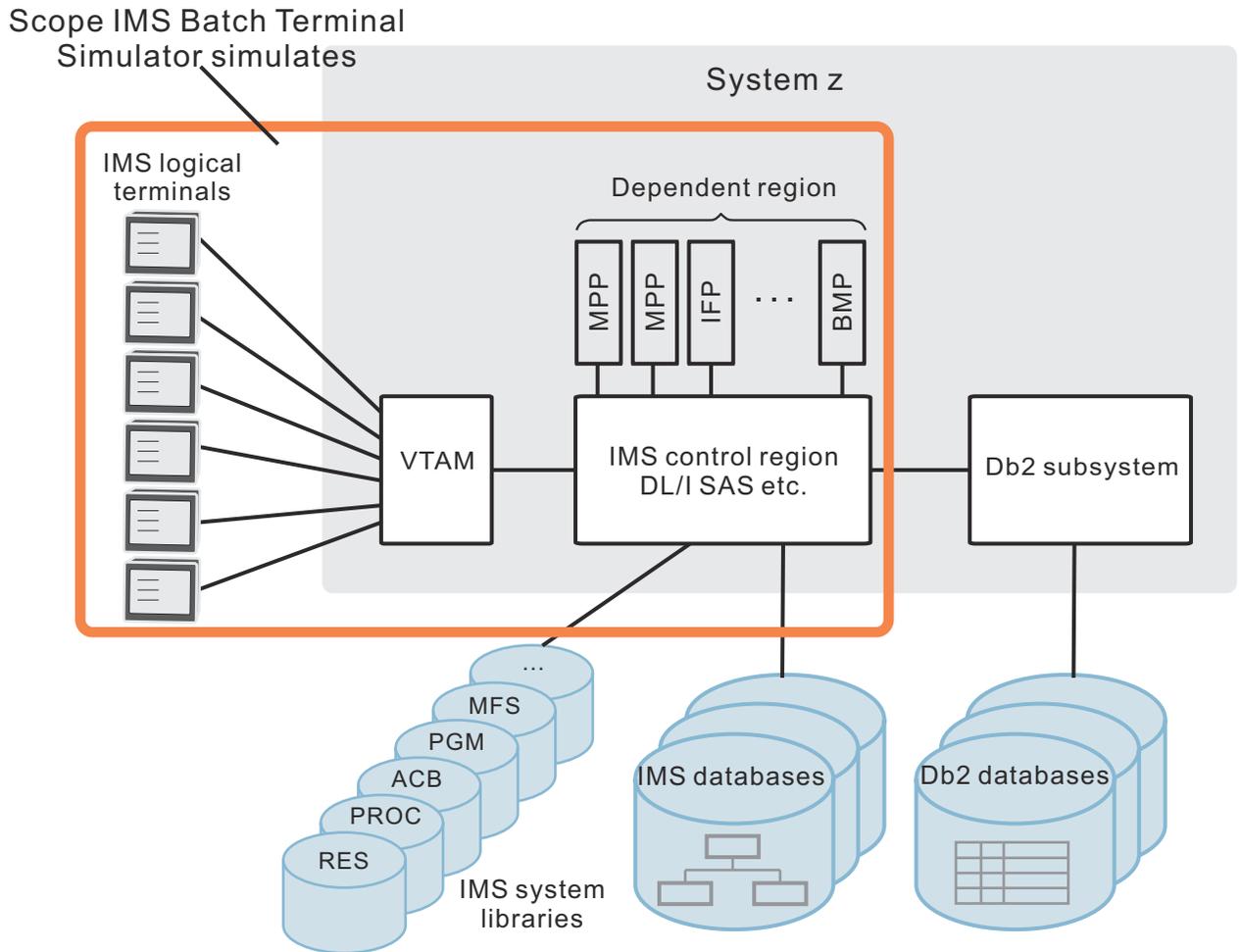


Figure 2. Structure of the system environment when IMS Batch Terminal Simulator is used

Process flow

The following figure shows basic process flow for IMS Batch Terminal Simulator.

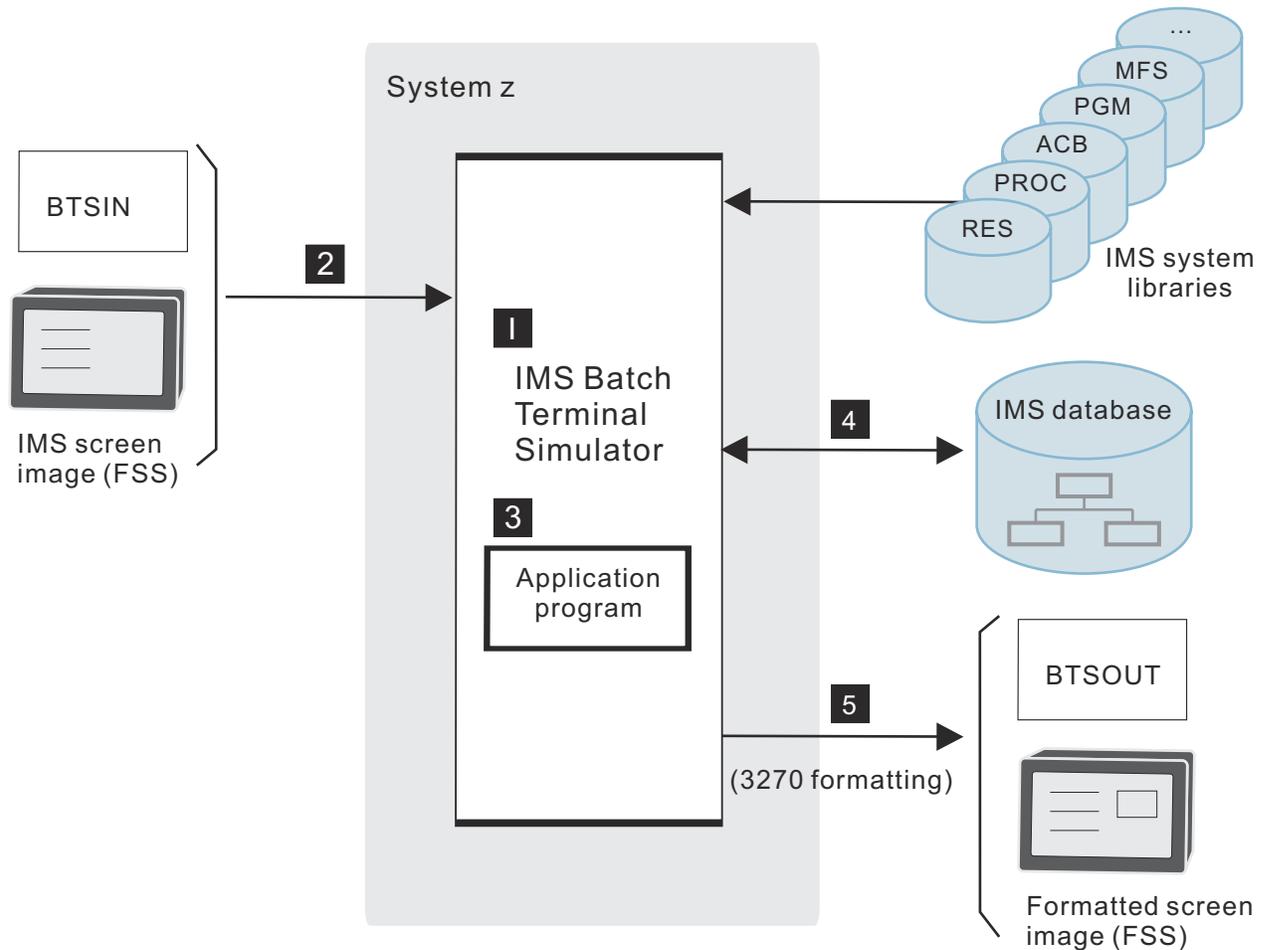


Figure 3. IMS Batch Terminal Simulator process flow

Input to IMS Batch Terminal Simulator consists of *simulator commands* and *simulator statements* in the BTSIN input stream. Simulator commands define the transactions to be processed and the format of simulator statements and output listings. Simulator statements define the transaction messages for simulating *terminal activities* through the input stream.

When IMS Batch Terminal Simulator receives a transaction code and a transaction message, IMS Batch Terminal Simulator accesses the IMS system libraries, obtains the resources from the libraries, simulates the application activities, and generates output.

Output from IMS Batch Terminal Simulator consists of a printed listing that shows all IMS Batch Terminal Simulator input and the requested information about each transaction that was processed during the IMS Batch Terminal Simulator run. These outputs are printed in the BTSOUT output stream.

The full screen image support (FSS) creates a 3270 data stream from the screen image (MFS library). This data stream, which contains all data and all field attribute characters that are present in the screen image, is sent to the TSO terminal. FSS issues a read from the TSO terminal. When the user enters data onto the screen image, the data is translated into data simulator statements for IMS Batch Terminal Simulator use. IMS Batch Terminal Simulator reads data simulator statements from FSS (or from BTSIN, if FSS is not active) and uses these data simulator statements to modify the current screen image of the simulated 3270 terminal.

Distributed architecture

The following figure shows how the Eclipse interface and the IMS Batch Terminal Simulator resource adapter (BTSRA) communicate with IMS Batch Terminal Simulator through Distributed Access Infrastructure.

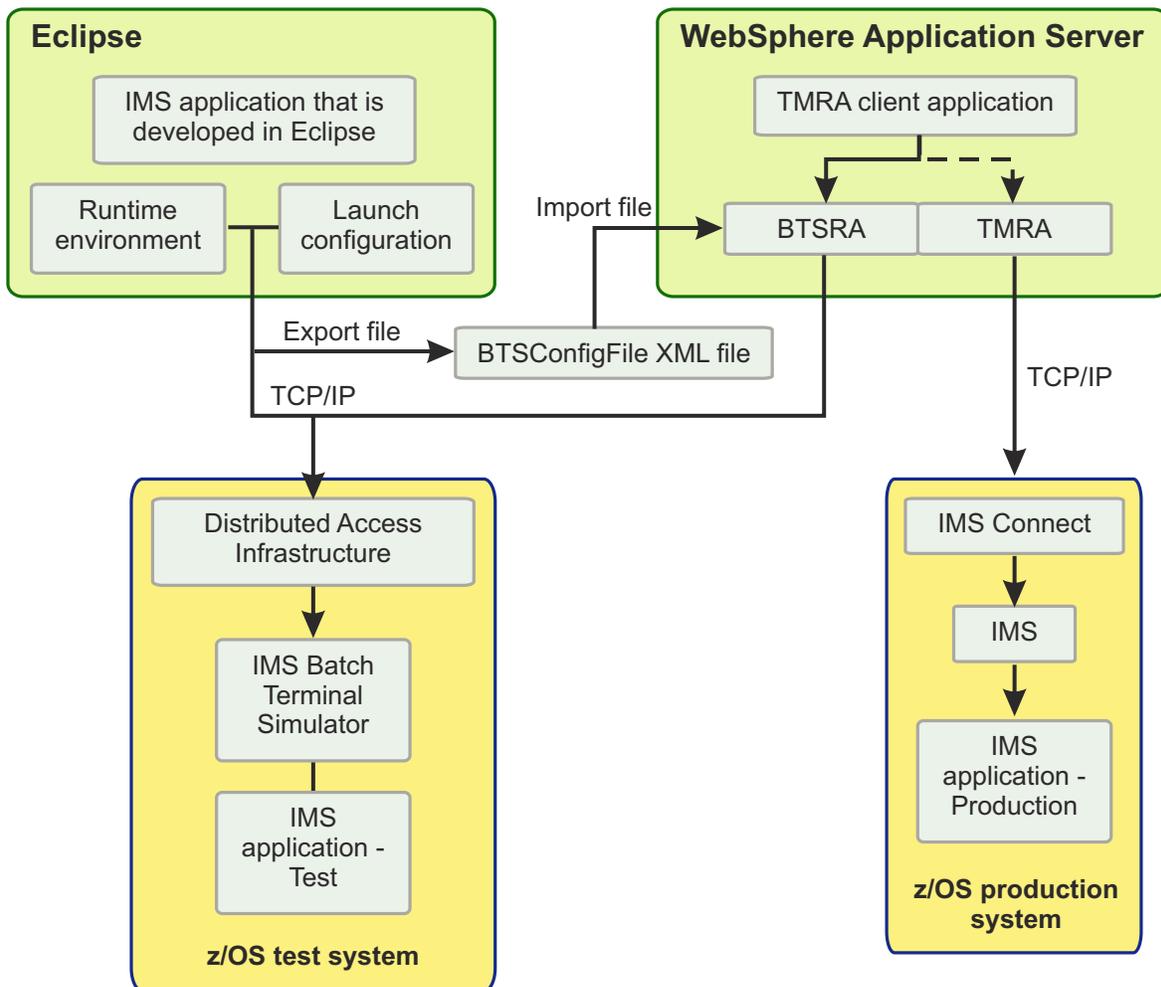


Figure 4. Distributed architecture for the IMS Batch Terminal Simulator Eclipse interface and resource adapter.

In the Eclipse interface, system programmers define an IMS Batch Terminal Simulator server runtime environment, which are the libraries that are used by IMS Batch Terminal Simulator. These server runtime environments can be exported and imported to other Eclipse development environments by an application developer.

After the server runtime environment is defined, application developers create *launch configurations* that are specific to the IMS application that they want to test from Eclipse. The launch configuration defines the server runtime environment that the application is tested in and the location of the application libraries. Then, application developers can run the launch configuration to test their applications on IMS Batch Terminal Simulator from Eclipse.

For client applications that are deployed in WebSphere® Application Server, developers can test the target IMS applications by using the BTSRA. The BTSRA acts similarly to the TMRA, but instead of communicating to IMS through IMS Connect, BTSRA runs IMS Batch Terminal Simulator through Distributed Access Infrastructure. BTSRA requires a runtime environment and a launch configuration XML file that developers can export from Eclipse. The XML file contains all the runtime information that is required to run IMS Batch Terminal Simulator for a target IMS application.

No changes are required to the WebSphere Application Server application if the application does not use TMRA specific functions that are not supported by BTSRA. Therefore, you can easily switch between testing and production environments by using the BTSRA or TMRA. Currently, any TMRA functions that are directly related to IMS Connect or IMS OTMA (such as resumeTpipe, asynchronous output, or reroute) are not supported by IMS Batch Terminal Simulator.

Related reference

Transaction Manager resource adapter supported functions

IMS Batch Terminal Simulator resource adapter (BTSRA) supports most of the Transaction Manager resource adapter (TMRA) functions that do not relate to IMS Connect or IMS OTMA.

Supported applications and devices

IMS Batch Terminal Simulator supports various types of IMS applications. It also provides device formatting support for various device types.

Application support

IMS Batch Terminal Simulator supports the following types of IMS applications:

- Applications that are written in Assembler Language, COBOL, PL/I, C, C++, REXX, and Java
- Applications that access IMS, Db2, and IBM MQ

Device formatting support

IMS Batch Terminal Simulator supports formatting for the following devices when used for the I/O PCB:

- 2740 Models 1 and 2
- 3275 Models 1 and 2
- 3276 Models 1, 2, 3, and 4
- 3277 Models 1 and 2
- 3278 Models 1, 2, 3, 4, and 5
- 3278 Model 52
- 3279 Models 2A, 2B, 3A, and 3B
- 3290
- PS/55

When IMS Batch Terminal Simulator simulates an alternate logical terminal, formatting for the following devices is also supported:

- 3275 Models 1 and 2
- 3276 Models 1, 2, 3, and 4
- 3277 Models 1 and 2
- 3278 Models 1, 2, 3, 4, and 5
- 3278 Model 52
- 3279 Models 2A, 2B, 3A, and 3B
- 3283 Model 52
- 3284 Models 1, 2, and 3
- 3286 Models 1 and 2
- 3287 Models 1, 1C, 2, and 2C
- 3289 Models 1 and 2
- 3290
- PS/55

IMS Batch Terminal Simulator supports formatting of data streams that contain programmed symbols (PS) and Extended Graphic Character Set (EGCS) characters. Also, for alternate logical terminal output for a printer device, IMS Batch Terminal Simulator supports formatting of data streams that contain SNA Character String (SCS) characters.

What's new in IMS Batch Terminal Simulator 4.1

IMS Batch Terminal Simulator 4.1 has been enhanced with features that simplify and modernize the testing of IMS applications. This section summarizes new functional changes and changes to the IMS Batch Terminal Simulator documentation.

Functional changes in IMS Batch Terminal Simulator 4.1

IMS Batch Terminal Simulator 4.1 offers several significant new features.

Support for distributed application development

IMS Batch Terminal Simulator can now accept and process input from distributed applications. Applications that run on WebSphere Application Server can test IMS applications on IMS Batch Terminal Simulator through an IMS Batch Terminal Simulator resource adapter rather than with the IMS Transaction Manager resource adapter. As a result, you can develop and test client applications with low-cost simulation.

Support for Eclipse application developers with an Eclipse plugin

IMS Batch Terminal Simulator can now be used with an Eclipse development environment, such as Rational Developer for System z and Debug Tool. By using Eclipse, application developers can run and test IMS applications from the Eclipse development environment, which helps reduce the amount of z/OS specific knowledge that an application developer requires to test IMS applications.

Improved productivity of application development activities

IMS Batch Terminal Simulator provides statistical reports about application program calls. These reports help application programmers analyze the results of their programs easily, which, in turn, improves their productivity.

These statistical reports provide the following information for each application program:

- A summary of the message and database activities of the application program, including the number of calls, elapsed time, function code, and status code
- Details about the message and database activities of the application program organized by the function codes and by the segment names for each function code
- A summary of the status codes for message and database DLI calls, which helps application programmers identify which status codes do not work, or how many instances of a specific status code do not work
- A summary of the ICAL call activities, the Db2 call activities, and the IBM MQ call activities of the application program, including the number of calls and the elapsed time by call types

Detailed information about the performance requirements for moving applications into the production environment

IMS Batch Terminal Simulator generates reports that provide the elapsed time of the database, Db2, and IBM MQ calls that were issued by the application program. It also generates a status summary report that helps database administrators define service level agreements (SLAs) that application programmers must meet before they can exit development and unit testing.

Improved verification productivity regression testing

Elapsed time, which is printed in the output, can be used to compare the results of simulation runs when IMS Batch Terminal Simulator is used as a regression test tool. When application programmers test modifications to an application or to the system, including testing of new releases, they can use the elapsed time information to improve the verification productivity in regression testing.

Support for IMS Call (ICAL) calls under BMP and JBP regions

When IMS Batch Terminal Simulator runs under a BMP or JBP region, the ICAL call enables the application program to send a synchronous request for data or services to a non-IMS application program or service that runs in a z/OS or distributed environment.

Support for variable-length BTSIN data sets

IMS Batch Terminal Simulator can process a variable-length BTSIN data set. The use of a variable-length BTSIN data set makes it easy for application programmers to simulate input messages from a distributed environment.

Roadmap to IMS Batch Terminal Simulator information

The IMS Batch Terminal Simulator User's Guide provides complete information for using IMS Batch Terminal Simulator.

The following list summarizes the new organization of the IMS Batch Terminal Simulator information:

- To learn about IMS Batch Terminal Simulator, see [Part 1, “IMS Batch Terminal Simulator overview,” on page 1](#).
- To find information about installing and configuring IMS Batch Terminal Simulator, see [Part 2, “Installation and configuration,” on page 39](#).
- To simulate applications with IMS Batch Terminal Simulator, follow the instructions in [Part 3, “Testing applications,” on page 57](#).
- To generate transaction screens from IMS logs, follow the instructions in [Part 4, “Playback utility,” on page 153](#).
- To find information about the simulator commands and statements, and other reference information, see [Part 5, “Reference,” on page 185](#).
- To troubleshoot IMS Batch Terminal Simulator problems, see explanations for IMS Batch Terminal Simulator abend codes and messages in [Part 6, “Troubleshooting,” on page 259](#).

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

[IBM Support: IMS Batch Terminal Simulator for z/OS](#)

Product documentation and updates

IMS Tools information is available at multiple places on the web. You can receive updates to IMS Tools information automatically by registering with the IBM My Notifications service.

Information on the web

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:

<https://www.ibm.com/support/pages/node/712955>

The IMS Tools Product Documentation web page includes:

- Links to [IBM Documentation](#) for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions

IBM Redbooks® publications that cover IMS Tools are available from the following web page:

<http://www.redbooks.ibm.com>

The IBM Information Management System website shows how IT organizations can maximize their investment in IMS databases while staying ahead of today's top data management challenges:

<https://www.ibm.com/software/data/ims/>

Receiving documentation updates automatically

To automatically receive automated emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The IMS Tools option is located under **Software > Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

How to send your comments

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS Tools information, you can take one of the following actions:

- Click the Feedback button at the top of the IBM Documentation topic that you are commenting on.
- Click the Contact Us tab at the bottom of any IBM Documentation topic.
- Send an email to ibmdocs@us.ibm.com. Be sure to include the book title, topic or section title, specific text, and your comment.

To help us respond quickly and accurately, please include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in IMS Batch Terminal Simulator enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *IBM z/OS ISPF User's Guide, Volume 1, SC34-4822*
 - *IBM z/OS TSO/E Primer, SA22-7787*
 - *IBM z/OS TSO/E User's Guide, SA22-7794*

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

The Eclipse interface uses the standard navigation keys for the operating system that it is installed on and supports keyboard-only operation.

Chapter 2. Key IMS Batch Terminal Simulator concepts

By interacting with IMS and with application programs, IMS Batch Terminal Simulator simulates the behavior of applications and generates reports that contain detailed information about processed transactions.

The following topics describe key concepts that you need to understand before using IMS Batch Terminal Simulator.

Topics:

- [“IMS Batch Terminal Simulator terminology” on page 19](#)
- [“Load modules and system flow” on page 20](#)
- [“Determining the types of IMS region” on page 22](#)
- [“Runtime modes” on page 24](#)
- [“Debugging capabilities” on page 25](#)
- [“Supported IMS calls” on page 27](#)
- [“Formatting modes” on page 33](#)
- [“Full screen image support \(FSS\)” on page 35](#)

IMS Batch Terminal Simulator terminology

IMS Batch Terminal Simulator information includes several unique terms that you need to understand before you begin to use IMS Batch Terminal Simulator.

Unique terms used in this information

Formatted mode

IMS Batch Terminal Simulator supports two formatting modes; formatted mode and unformatted mode. In formatted mode, IMS Batch Terminal Simulator enables the IMS Message Format Service and allows the formatting of 3270 input and output during the simulation.

Full screen image support or FSS

Full screen image support (FSS) is a function of IMS Batch Terminal Simulator. If you enable the FSS function, IMS Batch Terminal Simulator formats the screen image in the same way that the IMS terminal format the screen image. You can enter data into the screen image the same way that you can enter data with IMS.

Runtime modes

IMS Batch Terminal Simulator supports two runtime modes: batch mode and interactive mode. In batch mode, you can simulate your applications as a batch job, and in interactive mode, you can simulator your applications interactively.

Secondary transactions

A secondary transaction is a transaction that is scheduled by a (primary) transaction.

Simulator commands

Simulator commands define the transactions to be performed, the logical terminal to be simulated, the format of the simulator statements and output listing, and what debugging aids are requested.

Simulator statements

Simulator statements specify the transaction codes and transaction messages to request terminal activities.

Terminal activities

Terminal activities are the activities that are simulated by the terminal through IMS Batch Terminal Simulator.

Unformatted mode

IMS Batch Terminal Simulator supports two formatting modes; formatted mode and unformatted mode. In unformatted mode, IMS Batch Terminal Simulator does not use the IMS Message Format Service and, therefore, 3270 input and output are not formatted during the simulation.

Abbreviations

To make this information easier to read, the version and release levels of IMS are abbreviated, as follows:

- IMS 10 refers to IMS 10.1.
- IMS 11 refers to IMS 11.1.
- IMS 12 refers to IMS 12.1 and IMS Database Value Unit Edition 12.1.
- IMS 13 refers to IMS 13.1 and IMS Database Value Unit Edition 13.1.
- IMS 14 refers to IMS 14.1 and IMS Database Value Unit Edition 14.1.
- IMS 15 refers to IMS 15.1 and later, and IMS Database Value Unit Edition 15.1 and later.

The various versions of IMS are referred to simply as IMS, except where distinctions among them need to be made.

Load modules and system flow

IMS Batch Terminal Simulator is composed of four load modules. Each module interacts with one another to simulate IMS applications.

Subsections:

- [“Load modules” on page 20](#)
- [“System flow” on page 21](#)

Load modules

IMS Batch Terminal Simulator is composed of the following load modules.

BTS TSO controller (BTSTSOST)

The BTS TSO controller module is used only under TSO. This module attaches the BTS region controller.

BTS region controller (BTSRC000)

The BTS region controller module, linked by MVS or attached by BTS TSO controller, attaches the IMS region controller (DFSRR00). One of the CSECTs in this load module intercepts each Db2 call from the application program.

The major function of the BTS region controller is to set up the IMS environment to schedule the transactions. This scheduling is needed because all IMS message and batch message processing applications (that is, all teleprocessing applications under IMS) can trigger other secondary teleprocessing applications. These secondary transactions have their own PSB and DBD control blocks that must pass through IMS before their application programs can be run. The BTS region controller passes these blocks through IMS, thus uses IMS as often as necessary to run all secondary transactions.

BTS program controller (BTSPC000)

The BTS program controller module, linked by IMS, loads and calls the application program. One of the CSECTs in this load module intercepts each IMS call from the application program.

The BTS program controller has several functions, including initialization, application program load, IMS call interception, TP call handling, and termination.

The major function of the BTS program controller is to simulate the IMS DC calls. It is through its PCBs that IMS controls both communication to terminals and the passing of control to other teleprocessing applications. Under IMS Batch Terminal Simulator, the BTS program controller handles these calls so that the application program can function as if it were in the teleprocessing environment of online IMS.

BTS Db2 DL/I batch support program controller (BTSPCM00)

The BTS Db2 DL/I batch support program controller module is used only when IMS Batch Terminal Simulator runs with Db2 DL/I batch support. The module is linked by the BTS program controller through Db2 and calls one of the CSECTs in the BTS program controller.

System flow

The following figure shows a simplified picture of IMS Batch Terminal Simulator. The figure shows how IMS Batch Terminal Simulator components interact with IMS and the application program.

At the end of the IMS Batch Terminal Simulator run, the IMS Batch Terminal Simulator job step finishes with the maximum return code that is returned by the application program.

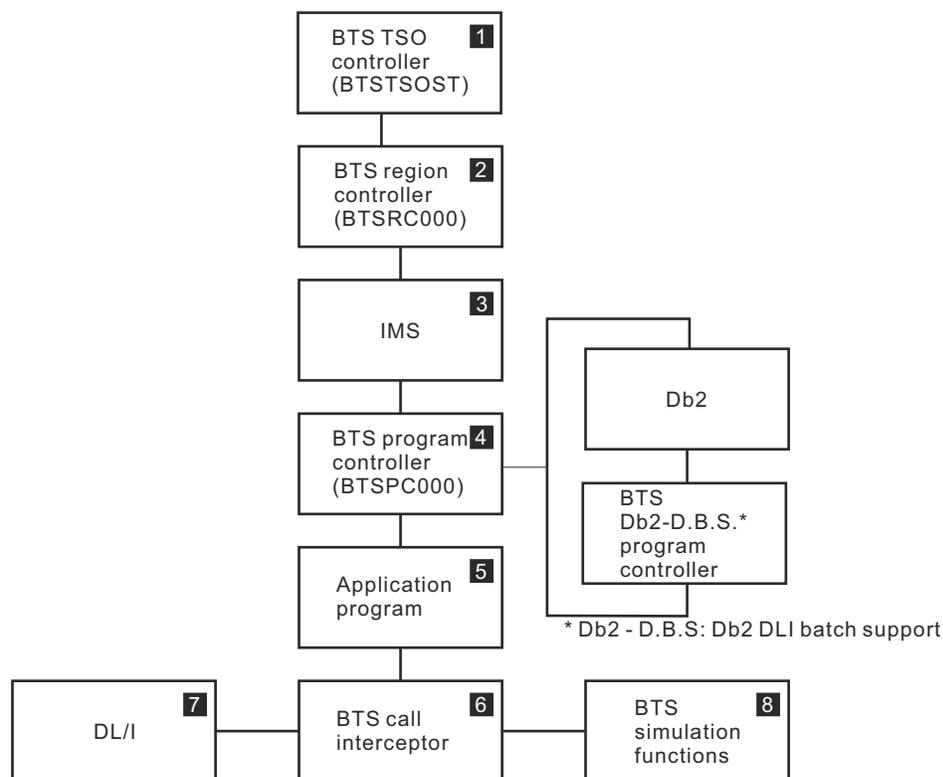


Figure 5. IMS Batch Terminal Simulator system flow

1. If TSO is being used, the BTS TSO controller (BTSTSOST) **1** receives control and attaches the BTS region controller (BTSRC000) **2**.
When control is regained, BTS TSO controller detaches BTS region controller and returns to the caller.
2. BTS region controller receives control from the operating system and performs the following functions:
 - a. Performs initialization
 - b. Receives a message or calls the IMS Message Format Service (MFS) functions
 - c. Finds a transaction to correspond with the message
 - d. Attaches the IMS region controller (DFSRR00) **3**

When the IMS region controller is attached, BTS region controller passes PSBNAME and IMS region controller EXEC parameters that are specified in the execution JCL or in the TSO CLIST.

After the IMS region controller returns control to BTS region controller, BTS region controller performs the following functions:

- a. Detaches the IMS subtask
- b. Loops back to get a message if the return code is zero, or terminates the run if the return code is not zero
- c. Returns to the operating system.

If TSO is being used, the operating system causes BTS TSO controller to regain control.

IMS Batch Terminal Simulator dynamically modifies IMS control blocks so that when the application program is subsequently run, all IMS calls from the application program go to an IMS Batch Terminal Simulator routine.

3. IMS loads the BTS program controller (BTSPC000) and transfers control to BTS program controller **4**.

Upon completion of BTS program controller, IMS returns to the operating system that causes BTS region controller to regain control **2**.

4. BTS program controller receives control from IMS and performs the following functions:

- a. Performs initialization
- b. Loads the application program
- c. Performs all memory patches as specified by the ./E or ./P commands
- d. Calls the application program **5**

Upon return from the application program, BTS program controller performs the following functions:

- a. Deletes the application program
- b. Prints any alternate terminal output
- c. Gets the next message or calls the IMS Message Format Service (MFS) functions
- d. Processes the transaction or returns to IMS **3**

5. Whenever the application program calls IMS, the BTS call interceptor intercepts the call **6**.

The application program can issue several IMS calls.

- For an IMS DB call, the IMS Batch Terminal Simulator routine calls the IMS Program Request Handler to process the request.
- For an IMS DC call, IMS Batch Terminal Simulator routines are called to process the request.

Upon completion of the application program, it returns to BTS program controller **4**.

6. The BTS call interceptor performs the following functions:

- a. Calls either DL/I **7** or IMS Batch Terminal Simulator simulation functions **8**
- b. Calls BTS PCB writer to print the call trace information
- c. Calls BTS statistics update routine to update the call statistics
- d. Calls BTS application snapper to perform any SNAP requests from the ./S command
- e. Returns to the application program **5**

Determining the types of IMS region

You specify the type of IMS region to be run (DLI, DBB, BMP, or JBP) in the IMS Batch Terminal Simulator execution JCL or in the TSO CLIST. Each region type has various advantages and disadvantages.

Tip: IMS Batch Terminal Simulator provides separate JCL procedures for running an IMS batch region (BTS) and an IMS online dependent region (BTSBMP and BTSJBP).

When evaluating the merits of each region, consider the following factors:

- [“Performance” on page 23](#)

- [“Resource utilization” on page 23](#)
- [“Access to Fast Path databases” on page 23](#)
- [“Handling of IMS calls” on page 23](#)
- [“Flexibility” on page 23](#)
- [“Messaging” on page 24](#)

Performance

Running DBB batch generally gives better performance than DLI batch. With DBB batch, the IMS region controller (DFSRR00) is attached only one time, except where GSAM is used. With DLI batch, the IMS region controller is attached each time a different program specification block (PSB) is scheduled by IMS Batch Terminal Simulator.

Resource utilization

When an online region is shared and used for multiple simulation environments, online region execution offers improved resource utilization. In an installation in which multiple users are running IMS applications concurrently, all users share the IMS facilities in the control region, which eliminates the need to duplicate these facilities in separate batch regions.

Access to Fast Path databases

The application can access Fast Path databases in an online region. IMS Batch Terminal Simulator does not support access to a terminal-related Main Storage Database (MSDB) but does support access to other Fast Path databases in an online environment.

Handling of IMS calls

IMS calls are handled differently depending on the type of IMS region in which the application is run.

- IMS Batch Terminal Simulator either passes IMS calls directly to DL/I or simulates the codes by itself. More IMS calls are passed directly to DL/I when applications are run in IMS online regions. Consider running your applications in IMS online region if you want to simulate IMS calls directly to DL/I.

Related reading: See [“IMS call function codes” on page 27](#) for more information about how the IMS calls are handled in each type of IMS region.

- If `./E SPLAPI=YES` is specified (specifies that Spool API functions are available) and if IMS Batch Terminal Simulator is run in an IMS online environment (`KW=BMP`), IMS Batch Terminal Simulator can write data to the IMS Spool API for IMS Spool API functions.

Flexibility

- Online execution is not as flexible as batch execution. The CTLBLKS (or NUCLEUS) generation for the control region must include specifications for each database description (DBD) and PSB to be used by the system. The DATABASE macro is used to define a DBD, and the APPLCTN macro is used to define a PSB.
 - A PSB must not be defined as `FPATH=YES` in an APPLCTN macro; otherwise, the region abends with a U1005 (The default value is `FPATH=NO`.)
 - A PSB can be defined in an APPLCTN macro as either `PGMTYPE=(TP)` or `PGMTYPE=(BATCH)`, except that `PGMTYPE=(TP)`, `SCHDTYP=(PARALLEL)` must not be specified; otherwise, the region abends with a U432 (The default is `SCHDTYP=SERIAL`.)

Note: A PSB can be defined as `PGMTYPE=(TP)` and `SCHDTYP=(PARALLEL)` for a non-message-driven BMP region (Abend U432 does not occur.)

- In an IMS online system, whenever an application program abends, the transaction is automatically stopped (`/PSTOP`). The MTO would normally `/START` the transaction to allow subsequent invocation

of the transaction. Consider having an AOI routine for your IMS online test system that automatically starts a transaction (with /START) following an abend. With this routine, the application programmer can modify the program and run again without delay.

Messaging

In an IMS online system, messages are sent to the master terminal operator (MTO) every time a BMP starts, stops, or invokes a checkpoint.

Tip: To reduce the number of messages that are displayed, consider having an automated operator interface (AOI) routine to intercept these messages.

Runtime modes

IMS Batch Terminal Simulator supports two runtime modes: batch mode and interactive mode.

In both modes, you can use IBM Debug Tool for z/OS or IBM z/OS Debugger on z/OS, and IBM Rational Developer for System z or IBM Developer for z Systems on the client system.

Subsections:

- [“Batch mode” on page 24](#)
- [“Interactive mode” on page 24](#)

Batch mode

In batch mode, an IMS Batch Terminal Simulator job is run as a batch job. IMS Batch Terminal Simulator reads the BTSIN data set as input and writes the output in the BTSOUT data set. This mode is especially beneficial in functional testing and regression testing where you need to run a test script (described in BTSIN) recursively.

IMS Batch Terminal Simulator provides three cataloged procedures for running applications in batch mode:

- The BTS procedure is used for running IMS Batch Terminal Simulator in an IMS batch processing environment.
- The BTSBMP procedure is used for running IMS Batch Terminal Simulator in an IMS BMP online environment.
- The BTSJBP procedure is used for running IMS Batch Terminal Simulator in an IMS JBP online environment.

By supplying the parameters that are required for these procedures, the DD statements that define the test database for the application, and the BTSIN DD statement, you can run an application as a batch job.

Interactive mode

In interactive mode, you establish an interactive IMS Batch Terminal Simulator session through TSO. IMS Batch Terminal Simulator provides BTS TSO command list, BTSCLIST, which you can customize and use to establish an interactive session.

In this mode, you can run the application interactively and debug it while it is running. This mode is beneficial in unit testing. You can enter input to IMS Batch Terminal Simulator directly through the TSO terminal while the application is running, or you can describe it in the BTSIN data set. The output from the transaction is immediately shown on the terminal and is also written to the BTSOUT data set.

For instructions to start IMS Batch Terminal Simulator in these modes, see the following topics:

- [Chapter 6, “Testing applications in batch mode,” on page 65](#)
- [Chapter 7, “Testing applications in interactive mode,” on page 79](#)

Debugging capabilities

IMS Batch Terminal Simulator integrates with IBM Debug Tool for z/OS, IBM z/OS Debugger, and COBOL interactive debug. It also provides debugging capabilities through IMS Batch Terminal Simulator simulator commands.

When you find application logic errors in unit testing, you can select from the following debugging methods.

- [“Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger” on page 25](#)
- [“Debugging with simulator commands” on page 25](#)
- [“Debugging with TSO TEST command” on page 25](#)
- [“Debugging with VS COBOL II interactive debug” on page 25](#)

Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger

By calling IBM Debug Tool for z/OS or IBM z/OS Debugger from IMS Batch Terminal Simulator, you can interactively test applications while using the various debugging functions that are provided by the debug tool.

Related reading: For more information about using these debug tools in IMS Batch Terminal Simulator jobs, see [“Support for IBM Debug Tool for z/OS and IBM z/OS Debugger” on page 26](#).

Debugging with simulator commands

By specifying the simulator commands that are provided by IMS Batch Terminal Simulator, you can run path tests as batch jobs without modifying your applications or compiling your applications with specific options.

Related reading: See the following topics for instructions to debug your applications by using simulator commands:

- To debug in batch mode, see [“Debugging with simulator commands” on page 76](#).
- To debug in interactive mode, see [“Debugging with simulator commands” on page 99](#).

Debugging with TSO TEST command

By using the TSO TEST command with IMS Batch Terminal Simulator, you can interactively run path tests without modifying your applications or compiling your applications with specific options.

In addition to the typical path tests, you can use the TSO TEST command following an abnormal termination in the program to determine where the termination occurred and the environment at the time of the termination (for example, storage and register contents).

Related reading: See [“Debugging with the TSO TEST monitor” on page 100](#) for more information about using the TSO TEST command.

Debugging with VS COBOL II interactive debug

The VS COBOL II interactive debug facility is a command processing program that operates under control of TSO. It provides a facility that allows COBOL programmers to debug an application program while it is being processed.

Related reading: See the following topics for instructions to debug your applications by using the VS COBOL II interactive debug facility:

- To debug in batch mode, see [“Debugging with VS COBOL II interactive debug” on page 77](#).
- To debug in interactive mode, see [“Debugging with VS COBOL II interactive debug” on page 100](#).

Support for IBM Debug Tool for z/OS and IBM z/OS Debugger

IBM Debug Tool for z/OS and IBM z/OS Debugger (Debug Tool) help you test, debug, examine, monitor, and control the execution of application programs that are written in Assembler, C, C++, COBOL, or PL/I on a z/OS system. You can use Debug Tool in batch mode, interactively in full-screen mode, or in remote debug mode by using a workstation user interface.

You can start Debug Tool through IMS Batch Terminal Simulator for application programs that start in Language Environment®. If the application program has been compiled with the TEST compiler option, and the TEST runtime option is specified, Debug Tool gains control of the program and begins a debug session.

You can use Debug Tool through IMS Batch Terminal Simulator in the following ways:

- To debug your batch IMS application programs as an MVS batch job, choose from the following methods:
 - If you want to specify the Debug Tool commands in an input data set and the progress of the debugging session to be recorded in an output data set, use batch mode.
 - If you want your Debug Tool session to be displayed on a terminal, use a VTAM terminal defined for Debug Tool or remote debug mode, such as in IBM Rational Developer for System z or IBM Developer for z Systems.

To start Debug Tool for a BTS batch job, see [“Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger”](#) on page 76.

- To debug your batch IMS application programs through a TSO CLIST, choose from the following methods:
 - If you want all interaction to be displayed on a single screen, use the full screen image support (FSS) function in TSO foreground.
 - If you want IMS Batch Terminal Simulator and FSS data displayed on your TSO terminal and your Debug Tool session to be displayed on another terminal, use a VTAM terminal defined for Debug Tool or remote debug mode, such as in IBM Rational Developer for System z or IBM Developer for z Systems.

FSS is the default option when IMS Batch Terminal Simulator is started in the TSO foreground, and is available only when you are running IMS Batch Terminal Simulator in the TSO foreground.

Tip: FSS can be turned off only by specifying TSO=NO on the ./O command. When you run in the TSO foreground, all call traces are displayed on your TSO terminal by default. The call traces can be turned off by specifying parameters on either the ./O or ./T command.

To start Debug Tool for an IMS Batch Terminal Simulator interactive session, see [“Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger”](#) on page 98.

Related reading: For details about Debug Tool, IBM Rational Developer for System z, and IBM Developer for z Systems, see the following information:

- *IBM Debug Tool for z/OS User's Guide*
- *IBM Debug Tool for z/OS References and Messages*
- *IBM Developer for z Systems Installation Guide*
- *IBM Developer for z Systems Host Configuration Guide*
- *IBM Rational Developer for System z Installation Guide*
- *IBM Rational Developer for System z Host Configuration Guide*
- *IBM z/OS Debugger User's Guide*
- *IBM z/OS Debugger Reference and Messages*
- *z/OS Language Environment Programming Reference*

Supported IMS calls

IMS Batch Terminal Simulator intercepts all IMS calls from the application program.

Subsections:

- [“EXEC DLI commands and limitations” on page 27](#)
- [“DL/I synchronization” on page 27](#)
- [“AIB interface” on page 27](#)
- [“CEE/CEL interface” on page 27](#)

EXEC DLI commands and limitations

IMS Batch Terminal Simulator supports the EXEC DLI application to use EXEC DLI commands, which read and update DL/I databases. However, it does not support a program that runs as an online CICS® program using DBCTL.

When multiple transactions are run by one JOB in a DBB region (that is, KW=DBB), you must reattach the IMS system, because IMS Batch Terminal Simulator must reattach the IMS region controller for each transaction, and thereby IMS Batch Terminal Simulator must get a fresh copy of the PSB.

Related reading: For the command specification to reattach the IMS region controller, see [“./E command” on page 198](#).

The following commands are not reported on BTSOUT or displayed on TSO screen, because they are not passed to IMS Batch Terminal Simulator.

- QUERY
- RETRIEVE

DL/I synchronization

An application program sync point results in committing the database changes made up to that point and resetting any database positioning. A sync point results from a CHKP call, a GU to the I/O PCB, or an application program return.

AIB interface

IMS Batch Terminal Simulator supports DL/I calls that use the Application Interface Block (AIB) interface.

For an ICAL DL/I call that uses the AIB interface, IMS Batch Terminal Simulator traces AIB information, the request area, and the response area. The I/O area and SSA information are not traced because they do not exist.

For the other DL/I calls that use the AIB interface, IMS Batch Terminal Simulator traces AIB information in addition to the I/O area and SSA information.

CEE/CEL interface

IMS Batch Terminal Simulator supports DL/I calls that use the Common Execution Environment (CEE) / Common Execution Library (CEL) interface. As a part of CEE/CEL interface support, IMS Batch Terminal Simulator allows the use of PSBs generated by the PSBGEN with LANG= (blank) option.

IMS call function codes

IMS Batch Terminal Simulator intercepts all IMS calls from the application program. The calls are handled differently depending on the type of application that is specified (./T TYPE= specification).

If TYPE=DLI is specified by the ./T command, all calls are passed directly to DL/I regardless of the PCB address specified in the call.

If TYPE=BMP, TYPE=MDB, TYPE=IFP, or TYPE=MSG is specified by the ./T command, then calls that refer to a PCB that is not a TPPCB are passed through to DL/I for processing. Calls that refer to the I/O PCB are handled as follows:

- When IMS Batch Terminal Simulator is run in an IMS batch region (that is, EXEC parameter KW=DLI or KW=DBB), IMS Batch Terminal Simulator handles the function codes listed in the following table, as well as GU, GN, ISRT, or PURG. Any other function code results in an 'AD' status code returned in the I/O PCB.
- When IMS Batch Terminal Simulator is run in an IMS batch region with ./T TYPE=IFP, IMS Batch Terminal Simulator handles the calls in the same way as when TYPE=MSG is specified.

The following table shows how the function codes are handled in an IMS batch region.

Table 1. IMS calls that refer to the I/O PCB and that are handled in an IMS batch region

Call	Explanation
AUTH	A 'bb' status code is returned in the I/O PCB, and the I/O area is filled up with X'00'.
CHKP	If ./T TYPE=MSG or TYPE=MDB, the call is passed directly to DL/I. The call is then handled like a GU if the status code 'bb' is returned.
CMD	If ./T TYPE=MSG or TYPE=MDB, a 'bb' status code is returned in the I/O PCB, and the I/O area is filled up with X'00'. If ./T TYPE=BMP, an 'AD' status code is returned in the I/O PCB.
DEQ	A 'bb' status code is returned in the I/O PCB. If IMS Batch Terminal Simulator is running in an IMS DCCTL batch environment, an 'AD' status code is returned in the I/O PCB.
GCMD	If ./T TYPE=MSG or TYPE=MDB, a 'bb' status code is returned in the I/O PCB, and the I/O area is filled up with X'00'. If ./T TYPE=BMP, a 'QE' status code is returned in the I/O PCB. If the I/O area length exceeds 132 bytes, the length is truncated to 132 bytes.
GSCD	The call is passed directly to DL/I.
INIT	The call is passed directly to DL/I.
INQY	The call is passed directly to DL/I, then the data is returned in the I/O area. The returned data depends on a subfunction. See “IMS call limitations” on page 30 .
LOG	If IEFRDER DD is valid and BKO=Y is specified by the EXEC parameter, this call is passed directly to the DL/I. For other cases, the call is checked for validity and the appropriate status code is returned. No logging occurs.
ROLB	If IEFRDER DD is valid and BKO=Y is specified by the EXEC parameter, this call is passed directly to the DL/I. For other cases, an 'AL' status code is returned in the I/O PCB.
ROLL	The call is passed directly to DL/I.
ROLS	If IMS Batch Terminal Simulator is running with Db2 DL/I Batch Support, an 'RC' status code is returned in the I/O PCB. Otherwise, the call is passed directly to DL/I.
SETO	This call is checked for validity and the appropriate status code is returned. You do not need to set the processing options.

Table 1. IMS calls that refer to the I/O PCB and that are handled in an IMS batch region (continued)

Call	Explanation
SETS	If IMS Batch Terminal Simulator is running with Db2 DL/I Batch Support, a 'bb' status code is returned in the I/O PCB. Otherwise, the call is passed directly to DL/I.
SETU	The call is passed directly to DL/I.
SYNC	If ./T TYPE=MSG or TYPE=MDB, an 'AD' status code is returned in the I/O PCB. If ./T TYPE=BMP, IMS Batch Terminal Simulator issues a CHKP call to DL/I on the behalf of the caller. A 'bb' status code is returned in the I/O PCB.
XRST	If ./T TYPE=MSG, an 'AD' status code is returned in the I/O PCB. If ./T TYPE=BMP or TYPE=MDB, the call is passed directly to DL/I.

When IMS Batch Terminal Simulator is run in an IMS online region (that is, EXEC parameter KW=BMP and JBP), IMS Batch Terminal Simulator handles the function codes listed in the following table, as well as GU or GN. Any other function code is passed directly to DL/I.

Table 2. IMS calls that refer to the I/O PCB and that are handled in an IMS online region

Call	Explanation
CHKP	If ./T TYPE=MSG or TYPE=MDB, the call is passed directly to DL/I. Then, if the status code 'bb' is returned, call is then handled like a GU. If ./T TYPE=BMP, the call is passed directly to DL/I.
CHNG	This call is simulated by IMS Batch Terminal Simulator. If ./E SPLAPI=YES and Advanced Print Function Options is specified, the call is passed directly to DL/I for Spool API Functions.
CMD	If ./T TYPE=MSG or TYPE=MDB, a 'bb' status code is returned in the I/O PCB, and the I/O area is filled up with X'00'. If ./T TYPE=BMP, an 'AD' status code is returned in the I/O PCB.
GCMD	If ./T TYPE=MSG or TYPE=MDB, a 'bb' status code is returned in the I/O PCB, and the I/O area is filled up with X'00'. If ./T TYPE=BMP, a 'QE' status code is returned in the I/O PCB. If the I/O area length exceeds 132 bytes, the length is truncated to 132 bytes.
INQY	The call is passed directly to DL/I, and then the data is returned in the I/O area. The returned data depends on a subfunction. See “IMS call limitations” on page 30 .
ISRT	If ISRT issues a call for an alternative PCB that has been set up for IAFP processing, the call is passed directly to DL/I.
PURG	If PURG issues a call for an alternative PCB that has been set up for IAFP processing, the call is passed directly to DL/I.
SETO	If ./E SPLAPI=YES and Advanced Print Function Options is specified, the call is passed directly to DL/I for Spool API Functions.
SYNC	If ./T TYPE=MSG or TYPE=MDB, an 'AD' status code is returned in the I/O PCB. If ./T TYPE=BMP or TYPE=JMP, IMS Batch Terminal Simulator issues a SYNC call to DL/I on the behalf of the caller. A 'bb' status code is returned in the I/O PCB.

Table 2. IMS calls that refer to the I/O PCB and that are handled in an IMS online region (continued)

Call	Explanation
XRST	If ./T TYPE=MSG, an 'AD' status code is returned in the I/O PCB. If ./T TYPE=BMP or TYPE=MDB, the call is passed directly to DL/I.

When IMS Batch Terminal Simulator is run in an IMS online region for a message-driven BMP (that is, EXEC parameter KW=BMP and IN=transaction code), IMS Batch Terminal Simulator handles the function codes listed in the following table for a message-driven BMP. Any other function code is handled as in a non-message-driven online IMS environment.

Table 3. IMS calls that refer to the I/O PCB and that are additionally handled in an IMS online region for message-driven BMP

Call	Explanation
CMD	If ./T TYPE=MSG or TYPE=MDB, the call is passed directly to DL/I.
GCMD	If ./T TYPE=MSG or TYPE=MDB, the call is passed directly to DL/I.

IMS Batch Terminal Simulator supports the ICAL function code by passing it directly to DL/I. When IMS Batch Terminal Simulator is run in an IMS online region (that is, EXEC parameter KW=BMP or JBP) where IMS supports the ICAL function code, an ICAL call enables the application program to send a synchronous request to a non-IMS program or a service that runs in a z/OS or distributed environment. When IMS Batch Terminal Simulator is run in an IMS batch region (that is, EXEC parameter KW=DLI or DBB) where IMS does not support the ICAL function code, an ICAL call fails with DL/I return and reason codes.

IMS call limitations

Be aware that certain limitations apply to handling of GSCD calls, INQY calls, and XRST calls.

Subsections:

- [“GSCD call” on page 30](#)
- [“INQY call with ENVIRON subfunction” on page 30](#)
- [“INQY call with NULL subfunction for APPC-type destinations” on page 31](#)
- [“INQY call with NULL subfunction for terminal-type destinations” on page 32](#)
- [“INQY call with NULL subfunction for transaction-type destinations” on page 32](#)
- [“INQY call with NULL subfunction for unknown destinations” on page 32](#)
- [“INQY call with PROGRAM subfunction” on page 32](#)
- [“INQY call with LERUNOPT subfunction” on page 32](#)
- [“XRST call” on page 33](#)

GSCD call

If an incorrect PCB is specified, its PCB number (that is, PCBN= operand on BTSOUT) is reported as zero ('000').

INQY call with ENVIRON subfunction

The following table summarizes the values that are returned in the I/O area when an INQY call is issued with the ENVIRON subfunction.

Table 4. INQY ENVIRON data output

Information returned	Value returned
Application Region Type	Data specified by the ./T TYPE= option 'BMP' if ./T TYPE=MDB
Transaction Name	Data specified by the ./T MBR= option
User Identifier	USER-ID in the BTS IOPCB If USER-ID in the BTS IOPCB is not specified by the ./E USERID= option: <ul style="list-style-type: none"> • Blank if ./E USERIND= option (Userid Indicator) is 'U', 'O', or not specified • LTERM name of BTS IOPCB if ./E USERIND= option is 'L' • PSB name of BTS IOPCB if ./E USERIND= option is 'P'
Group Name	GROUP name in the BTS IOPCB
Userid of the Address Space	The Userid of the address space in the BTSCOM00 contains a value specified by the ./E UIDDREG option
Userid Indicator	The Userid indicator in the BTS IOPCB

INQY call with NULL subfunction for APPC-type destinations

The following table summarizes the values that are returned in the I/O area when an INQY call is issued with the NULL subfunction for APPC-type destinations.

Table 5. INQY NULL data output for APPC-type destinations

Information returned	Value returned
Destination Type	' APPC '
Information Entry Name	Data specified by the ./T SIDE= option
Partner Logical Unit Name	Data specified by the ./T TPNAME= option
Partner Mode Table Entry Name	Data specified by the ./T MODE= option
User Identifier	USER-ID in the BTS IOPCB If USER-ID in the BTS IOPCB is not specified by the ./E USERID= option: <ul style="list-style-type: none"> • Blank if ./E USERIND= option (Userid Indicator) is 'U', 'O', or not specified • LTERM name of BTS IOPCB if ./E USERIND= option is 'L' • PSB name of BTS IOPCB if ./E USERIND= option is 'P'
Group Name	GROUP name in the BTS IOPCB
Synchronization Level	Data specified by the ./T SYNCLVL= option
Conversation Type	Data specified by the ./T CONVTYPE= option
Address of TPN	Address of the LL field of Transaction Program Name or 0
Userid Indicator	The Userid indicator in the BTS IOPCB

INQY call with NULL subfunction for terminal-type destinations

The following table summarizes the values that are returned in the I/O area when an INQY call is issued with the NULL subfunction for terminal-type destinations.

Table 6. INQY NULL data output for terminal-type destinations

Information returned	Value returned
Destination Type	' TERMINAL '
Terminal Location	' LOCAL '
Queue Status	' STARTED '
Session Status	' ACTIVE '

INQY call with NULL subfunction for transaction-type destinations

The following table summarizes the values that are returned in the I/O area when an INQY call is issued with the NULL subfunction for transaction-type destinations.

Table 7. INQY NULL data output for transaction-type destinations

Information returned	Value returned
Destination Type	' TRANSACT '
Terminal Location	' LOCAL '
Transaction Status	' STARTED '
Destination PSB Name	Data specified by the ./T PSB= option
Destination Program or Session Status	' STARTED '

INQY call with NULL subfunction for unknown destinations

When an INQY call is issued with the NULL subfunction and the destination type is unknown, ' UNKNOWN ' is returned in the I/O area.

Related reading: See [Example 18 for ./P command](#) to override the default location or status.

INQY call with PROGRAM subfunction

When an INQY call is issued with the PROGRAM subfunction, data that is specified by the ./T MBR= option is returned for the Application Program Name and stored in the I/O area.

INQY call with LERUNOPT subfunction

When an INQY call is issued with the LERUNOPT subfunction, the following values are returned in the AIB interface:

In the BTS batch (that is, KW=DLI or KW=DBB):

- AIBRETRN is zero
- AIBREASN is zero
- AIBRSA2 is filled up with X'00'

XRST call

CKPTID 'LAST' is not supported in the BTS batch (that is, KW=DLI or KW=DBB) environment. To use CKPTID 'LAST', you must run IMS Batch Terminal Simulator in an online IMS environment (that is, KW=BMP).

Formatting modes

IMS Batch Terminal Simulator supports two formatting modes; formatted mode and unformatted mode. Formatting modes influence the format of input to and the output from IMS Batch Terminal Simulator jobs.

Subsections:

- [“Formatted and unformatted modes” on page 33](#)
- [“How IMS Batch Terminal Simulator creates formatted images” on page 34](#)

Formatted and unformatted modes

Formatted mode can be used only for 3270 applications. The 3270 applications can be simulated in both formatted and unformatted modes. In formatted mode, IMS Batch Terminal Simulator enables the IMS Message Format Service and allows the formatting of 3270 input and output during the simulation.

You can request formatted mode by issuing one of the following commands:

- For an I/O PCB, issue a `./D` command that specifies the IBM 3270 Display Station.
- For an alternate PCB, issue the `./T` command with the MDL or TYPE keyword.
- Issue an IMS `/FORMAT` command.

Notes:

1. You can issue an IMS `/FORMAT` command only while the 3270 is in unformatted mode; otherwise, the results are unpredictable.
2. If you do not specify one of these commands, IMS Batch Terminal Simulator runs in unformatted mode.

A message output descriptor (MOD) controls the output formatting. The MOD name is specified in the Message Format Service Utility MSG macro statement. You can explicitly specify the name of the MOD as a fourth parameter in an ISRT or PURG call; otherwise, the current MOD name in the I/O PCB is used. IMS sets this field on a GU. The name is given by the MID control block with which the input was formatted. When no MOD name is available for output formatting (null or blank characters), the IMS default format DFSMO2 is used.

In batch mode, if the formatting is successful, the 3270 is thereafter in formatted mode. Subsequent input is assumed to consist of *formatted-mode simulator statements*. In interactive mode, if the formatting is successful, the session is thereafter in formatted mode (FSS). Subsequent input is assumed to be entered from the TSO terminal.

After a `./D` command is issued that specifies the 3270, a CLEAR or PAX simulator statement, or a severe error in formatting, the 3270 screen image is in unformatted mode. When a screen image is in unformatted mode, 3270 simulator statements have the same format that is used for other types of terminals (for example, the 2740 typewriter terminal). This format is the standard simulator statement format and is referred to as *unformatted-mode simulator statements*.

The following table summarizes the accepted format of simulator statements for each mode.

Table 8. Format of simulator statements

Runtime mode	When formatted mode is requested	When formatted mode is not requested
Batch mode	3270 formatting is started. Simulator statements must be specified with formatted-mode simulator statements.	Simulator statements must be specified with unformatted-mode simulator statements.
Interactive mode	Full screen image support (FSS) is started. Simulator statements must be entered directly through the TSO screen.	Simulator statements must be specified with unformatted-mode simulator statements.

Related reading:

- For more information about the formats of simulator statements, see [“Simulator statements” on page 239](#).
- For a description about how the MOD and the related control blocks (MID, DIF, and DOF) are specified, see *IMS Application Programming APIs*.

How IMS Batch Terminal Simulator creates formatted images

Terminal output is formatted by the BTS 2740 or 3270 formatting modules, or by a user-written formatting module. Each ISRT or PURG call of a message or message segment to the I/O PCB made by an application program is passed to a user exit routine for format purposes when a request through the .D device command statement has been supplied. The formatting module (user-written or the 2740 or 3270 modules supplied by IMS Batch Terminal Simulator) can reformat and print output by using the BTS common writer module. The user-written module can also reformat the message in place and request IMS Batch Terminal Simulator to print it.

When an IBM 3270 Display Station is being simulated in formatted mode, terminal input and output must be in a format that is compatible with the IMS Message Format Service. For this purpose IMS Batch Terminal Simulator maintains, in main storage, an up-to-date map of the IBM 3270 Display Station screen image or printer image (including field attribute characters.)

For terminal input, IMS Batch Terminal Simulator:

- Updates the internal 3270 screen image
- Sends teleprocessing line data to the IMS Message Format Service in the format it expects, with the data and terminal operator actions as supplied through the simulator statements
- Prints the screen image

Similarly, application program output to the terminal being simulated is intercepted by IMS Batch Terminal Simulator from the IMS Message Format Service, and IMS Batch Terminal Simulator:

- Updates the internal image of the display screen or printer
- Prints the screen image or printer image

When 3270 formatting is used, for each simulator statement or application output message to the terminal received by IMS Batch Terminal Simulator, a copy of the internal 3270 screen image is printed in the IMS Batch Terminal Simulator output listing. The output listing also contains the name of the message input descriptor or message output descriptor (MID or MOD) used, the location of the cursor, and the terminal action causing the message to be sent. In addition, the 3270 field attribute character for each field of the formatted screen can optionally be printed within the screen image.

When an MFS bypass for inbound D/T3270 data stream is used, the internal image of the display screen or printer is not updated.

Full screen image support (FSS)

With the full screen image support (FSS) of IMS Batch Terminal Simulator, you can take full advantage of a 3270 terminal that is interacting with TSO. The screen image is formatted in the same way as the IMS terminal would be formatted, and data is entered into the screen image in the same way that data would be entered with IMS.

FSS overview

Full screen image support (FSS) formats the screen image in the same way that the IMS terminal format the screen image. You can enter data into the screen image the same way that you can enter data with IMS.

The full screen image support can be used for 3270 applications that use IMS Message Format Service. IMS Batch Terminal Simulator, by referring to the message input descriptor (MID) and message output descriptor (MOD), displays the formatted screen image on TSO.

FSS can be used only in interactive mode and in a TSO foreground address space. When IMS Batch Terminal Simulator is started, it checks to see if IMS Batch Terminal Simulator is running in the TSO foreground.

Subsections:

- [“FSS functions” on page 35](#)
- [“Tracing actions and application program events” on page 36](#)

FSS functions

The BTSIN data set contains the input to IMS Batch Terminal Simulator. When FSS is active, you can also enter data directly through the TSO terminal. However, BTSIN is always used, even if FSS is started. In fact, FSS is only partially active until BTSIN has reached end of file. This permits the use of a predefined set of simulator commands to set up the IMS Batch Terminal Simulator environment. Typically, this set of commands consists of a `./D` command, some `./T` commands, and possibly a `./O` command. In addition, you can allocate BTSIN to a TSO terminal; in this case, `/*` must be entered at the terminal to activate the end of file for BTSIN (`/*` command must be entered only at the IMS Batch Terminal Simulator prompt for input.)

Until BTSIN has reached end of file, no full screen image writes are made to the TSO terminal. However, the read and write functions and the call trace function are active from the beginning, which means that if BTSIN contains data for a set of interactions, you can follow their progress on the terminal. The `BTS0002I INPUT RECORD...` message is copied to the terminal until BTSIN reaches the end of file.

If the 3270 input buffer contains data that is received from an IMS screen, it is converted to a series of data simulator statements and given to IMS Batch Terminal Simulator (A listing in the `BTSPUNCH` data set shows the format of these statements, which then later can be used as BTSIN.)

If the buffer contains data that is not received from an IMS screen image, it is assumed to be a simulator command or simulator statement. In that case, the buffer has a maximum length of 72 bytes and input is truncated if necessary.

If the buffer contains no data, the user is prompted to supply a command with the message `ENTER BTS COMMAND OR /FORMAT OR /*`.

When the multiple message application runs, the input data must be entered from an IMS screen.

The FSS full screen image function is not invoked until BTSIN has reached the end of file. Then it is invoked every time IMS Batch Terminal Simulator has a complete screen image of an IMS screen image for output.

The terminal is always left with the keyboard unlocked. All modified data tags on the screen image are reset to zero. (Any pre-modified fields are represented as such in the IMS Batch Terminal Simulator screen image.) The alarm sounds if the terminal has this capability and the audible alarm function was specified.

Once the IMS screen image is displayed, FSS issues a read against the terminal. The normal actions required by the application are then performed as though you were connected to IMS. You can, for example, enter data into the screen image or press some action key to transmit the data. Exceptions to this involve the use of the CLEAR, ERASE INPUT, and PFnn keys.

The PA1 key is normally used in TSO to bring attention to a higher-level task. This is also true with IMS Batch Terminal Simulator except when an IMS screen image is displayed on the terminal. During this time, an attention exit for PA1 is established to simulate the PA1 key function of paging forward in the IMS message.

Tracing actions and application program events

IMS Batch Terminal Simulator output messages indicate when a transaction is started, what program is used, the transactions performed, and statistics about the operation. When each message is written to BTSOUT, FSS is also invoked to send the identical message to the TSO terminal. In addition, call trace information is provided for any or all calls that were made by the application program.

Also with the FSS call trace function, you can receive control at the TSO terminal after the completion of every DL/I or SQL call. With this feature, you can selectively display the AIB, I/O area, SSAs, and so on. At these points, you can enter the TEST monitor of TSO for further investigation of what is happening in the program.

Related reading:

- See [“Displaying the call trace information” on page 96](#) for call trace information.
- See [“Debugging with the TSO TEST monitor” on page 100](#) for information about using the TSO TEST monitor.

FSS operation flow

The full screen image support (FSS) interacts with TSO and IMS Message Format Service (MFS) control blocks to create formatted screen images.

The following figure shows a simplified picture of how FSS interacts with IMS Batch Terminal Simulator and TSO.

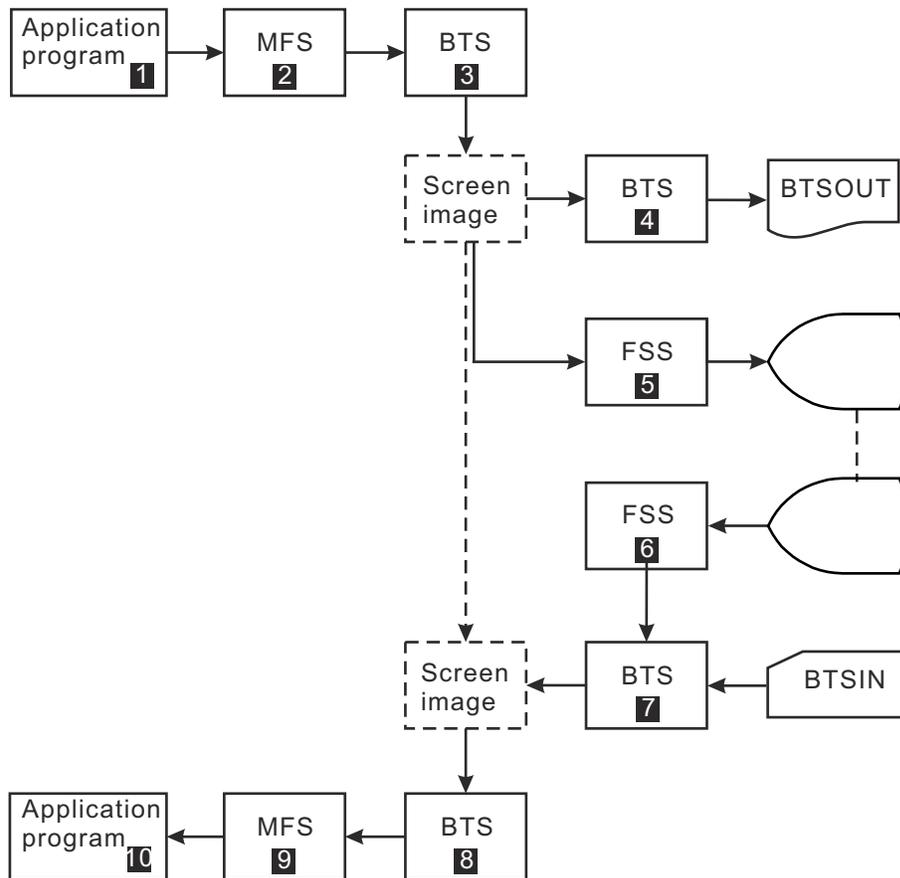


Figure 6. FSS operation flow

1. The application program issues an ISRT for a message **1**.
2. MFS receives the message segment and performs editing, which results in a 3270 data stream **2**.
3. IMS Batch Terminal Simulator uses the 3270 data stream to update the screen image for the simulated 3270 terminal. IMS Batch Terminal Simulator maintains this screen image in main storage **3**.
4. IMS Batch Terminal Simulator prints the 3270 screen image on the BTSOUT data set **4**.
5. FSS creates a 3270 data stream from the screen image. This data stream contains all data and all field attribute characters that are present in the screen image. This data stream is then sent to the TSO terminal **5**.
6. FSS issues a read from the TSO terminal **6**. When the user enters data onto the screen image and transmits it, the data is received and translated into appropriate simulator statements.
7. IMS Batch Terminal Simulator reads simulator statements either from BTSIN, if FSS is not active, or from FSS, if it is active. IMS Batch Terminal Simulator then uses these simulator statements to modify the current screen image of the simulated 3270 terminal **7**.
8. IMS Batch Terminal Simulator uses the modified screen image to create a 3270 data stream **8**.
9. MFS receives this data stream and performs some editing, which results in message segments **9**.
10. When the application program issues a GU or GN for a message, it receives the message segments created in step 9 **10**.

FSS limitations

Certain limitations apply to using the full screen image support (FSS).

Subsections:

- [“Device formatting” on page 38](#)

- [“TSO terminal equipment” on page 38](#)
- [“Deletion characters” on page 38](#)
- [“Immediate pen detectable fields” on page 38](#)
- [“PS/55 computer” on page 38](#)

Device formatting

A 3275 or 3277 Model 1 can be simulated only on a 3275 or 3277 Model 2. When a 3275 or 3277 Model 1 is simulated on a 3275 or 3277 Model 2 and a field extends from one line to the next, this condition is simulated on the TSO terminal by defining two fields with the same field attribute characters. The second field attribute character is located in column 80. When data is entered into such a field, the cursor automatically skips to the next line. If the TAB key is used, the cursor is positioned at the wrong location. Also, if this field contains data on both lines and only the data on the first line is modified, only that portion of the Model 1 field on the first line is presented to IMS Batch Terminal Simulator; as result, the data from the second line might be lost.

Any model of the 3276 or 3278 can be simulated on the same or larger size model of the 3276 or 3278.

TSO terminal equipment

The TSO terminal must be equipped with the same features as the simulated IMS terminal with the following exceptions:

- PFnn keys (because these keys can be simulated)
- Audible alarm (because the message BTS0055I is displayed in place of the alarm)

The following features are not simulated:

- Light pen
- Numeric lock

Deletion characters

The character-deletion and line-deletion characters in the TSO user profile might cause unpredictable problems. These deletion characters might cause certain characters, part of a line, or an entire line to be deleted before the data has been received by IMS Batch Terminal Simulator. To prevent this problem, consider specifying NOCHAR and NOLINE in the TSO user profile.

Immediate pen detectable fields

When immediate pen detectable fields are used, the data in modified fields are not passed to the application program.

PS/55 computer

A PS/55 computer can be simulated on any 3270 with the following restrictions:

- The screen size must be the same or larger (24,80).
- The mixed mode attribute is not transmitted to the 3270; Therefore, consider using only EBCDIC characters for mixed mode fields.
- The PS/55 computer allows an EGCS field to begin on an even or odd column. On a 3278 Model 52, MFS formats must specify EGCS fields to begin on an odd column number.

Part 2. Installation and configuration

After IMS Batch Terminal Simulator has been successfully installed, system administrators must configure the environment to meet the application testing needs of application programmers.

Topics:

- [Chapter 3, “Hardware and software prerequisites,” on page 41](#)
- [Chapter 4, “Configuring IMS Batch Terminal Simulator,” on page 43](#)

Chapter 3. Hardware and software prerequisites

IMS Batch Terminal Simulator is installed with the System Modification Program Extended (SMP/E) RECEIVE, APPLY, and ACCEPT commands.

IMS Batch Terminal Simulator is supplied with a Program Directory that contains specific instructions for installing the program. To install IMS Batch Terminal Simulator, follow the instructions in the *Program Directory for IBM IMS Batch Terminal Simulator for z/OS*.

This release of IMS Batch Terminal Simulator supports multiple IMS versions with one set of BTS libraries. After IMS Batch Terminal Simulator has been installed, BTS libraries can be used for any versions and releases of IMS that are supported by IMS Batch Terminal Simulator.

Hardware requirements

IMS Batch Terminal Simulator operates on any hardware configuration that supports the required versions of IMS.

Software requirements

IMS Batch Terminal Simulator requires a current version of IMS Database Manager. IMS Transaction Manager might be required instead of IMS Database Manager if no DL/I database is available.

The operating system requirements are the same as those required by the corresponding IMS release.

An IMS DB/DC system is required for running IMS Batch Terminal Simulator in an online IMS environment.

Conditional operational requirements include the following software:

- TSO/E is required for IMS Batch Terminal Simulator full screen image support (FSS) of the IBM 3270 display system simulation
- Distributed Access Infrastructure of IBM IMS Tools Base for z/OS 1.6 is required for the Eclipse interface and the resource adapter.
- For the IMS Batch Terminal Simulator plugin for Eclipse, IBM Rational Developer for System z 8 or 9, IBM Developer for z Systems 14, or Eclipse 3.6.2 environment or later with the Web Tools Platform (WTP) plugin (available from <http://www.eclipse.org/>) are required.
- For the IMS Batch Terminal Simulator resource adapter, WebSphere Application Server 7, 8, or 9, and the IMS Batch Terminal Simulator Eclipse interface are required. WebSphere Application Server Community Edition is not currently supported.
- IBM DFSORT (Data Facility Sort), which is part of z/OS, or a functionally equivalent sort and merge program is required for the Playback utility

IMS Batch Terminal Simulator can integrate with the following products:

- IBM Debug Tool for z/OS
- IBM z/OS Debugger
- IBM Developer for z Systems
- IBM Rational Developer for System z

Related reading: For details about Debug Tool, IBM Rational Developer for System z, and IBM Developer for z Systems, see the following information:

- *IBM Debug Tool for z/OS User's Guide*
- *IBM Debug Tool for z/OS References and Messages*
- *IBM Developer for z Systems Installation Guide*
- *IBM Developer for z Systems Host Configuration Guide*
- *IBM Rational Developer for System z Installation Guide*

- *IBM Rational Developer for System z Host Configuration Guide*
- *IBM z/OS Debugger User's Guide*
- *IBM z/OS Debugger Reference and Messages*
- *z/OS Language Environment Programming Reference*

Chapter 4. Configuring IMS Batch Terminal Simulator

You must configure the IMS Batch Terminal Simulator environment to meet your application testing needs.

Topics:

- [“Configuration checklist” on page 43](#)
- [“Customizing the BTS cataloged procedure” on page 45](#)
- [“Customizing the BTS TSO command list \(CLIST\)” on page 46](#)
- [“Customizing for IBM Debug Tool for z/OS or IBM z/OS Debugger” on page 46](#)
- [“Customizing for specific applications” on page 47](#)
- [“Implementing IMS user-written routines” on page 49](#)
- [“Configuring for IMS callable services” on page 50](#)
- [“Setting defaults for environment parameters \(./E commands\)” on page 51](#)
- [“Setting up the tutorial environment” on page 52](#)
- [“Recording the resource information” on page 52](#)
- [“Distributed Access Infrastructure configuration” on page 53](#)
- [“Installing Eclipse interface” on page 54](#)
- [“Installing the IMS Batch Terminal Simulator resource adapter” on page 54](#)
- [“Adding and configuring the IMS Batch Terminal Simulator resource adapter J2C connection factory” on page 55](#)

Configuration checklist

The configuration checklist is a summary of the tasks that you must complete after installation to ensure that the IMS Batch Terminal Simulator environment is ready for use.

Complete the tasks listed in the following checklist to configure the IMS Batch Terminal Simulator environment.

Table 9. Configuration checklist for IMS Batch Terminal Simulator

Task	Task description
Task 1	Customize the IMS Batch Terminal Simulator cataloged procedure that is supplied with the product. This task is required to use IMS Batch Terminal Simulator in batch mode. “Customizing the BTS cataloged procedure” on page 45
Task 2	Customize the BTS TSO command list (CLIST) that is supplied with the product. This task is required to use IMS Batch Terminal Simulator in interactive mode. “Customizing the BTS TSO command list (CLIST)” on page 46
Task 3	Configure the environment to enable the use of IBM Debug Tool for z/OS or IBM z/OS Debugger. This task is required to use Debug Tool with IMS Batch Terminal Simulator. “Customizing for IBM Debug Tool for z/OS or IBM z/OS Debugger” on page 46

Table 9. Configuration checklist for IMS Batch Terminal Simulator (continued)

Task	Task description
Task 4	Customize the IMS Batch Terminal Simulator JCL and configure the environment for specific applications. This task is required if the application to be simulated is written in Java or REXX, or if the application accesses Db2 or IBM MQ. “Customizing for specific applications” on page 47
Task 5	Implement IMS user-written routines. This task is required to implement IMS user-written routines. “Implementing IMS user-written routines” on page 49
Task 6	Configure for IMS callable services. This task is required to use IMS callable services. “Configuring for IMS callable services” on page 50
Task 7	Set default keywords and operands for environment parameters (./E simulator command). This task is required to set default environment parameters. “Setting defaults for environment parameters (./E commands)” on page 51
Task 8	IMS Batch Terminal Simulator information provides tutorial topics for application programmers who are new to IMS Batch Terminal Simulator. To perform these tutorials, the tutorial environment must be set up. “Setting up the tutorial environment” on page 52
Task 9	Summarize the resource information and provide it to the application programmers. This task is an easy way to provide application programmers with the resource and the environment information. “Recording the resource information” on page 52
Task 10	To use the Eclipse interface or the IMS Batch Terminal Simulator resource adapter, configure and run Distributed Access Infrastructure. “Distributed Access Infrastructure configuration” on page 53
Task 11	To use the Eclipse interface, install the IMS Batch Terminal Simulator plugins into your Eclipse installation. “Installing Eclipse interface” on page 54
Task 12	To use the resource adapter, install the IMS Batch Terminal Simulator resource adapter on a WebSphere Application Server. “Installing the IMS Batch Terminal Simulator resource adapter” on page 54
Task 13	To use the resource adapter, add and configure the IMS Batch Terminal Simulator resource adapter J2C connection factories. “Adding and configuring the IMS Batch Terminal Simulator resource adapter J2C connection factory” on page 55

Customizing the BTS cataloged procedure

Customize the IMS Batch Terminal Simulator cataloged procedure that is supplied with the product. This task is required to use IMS Batch Terminal Simulator in batch mode.

Procedure

1. By consulting with the application programmers, determine the type of IMS region to test the application.

Tip: See “Determining the types of IMS region” on page 22 to understand the differences between the region types.

IMS Batch Terminal Simulator provides the following sample cataloged procedures. These cataloged procedures are loaded into the BTS JCL library (SBTSJCL0) during installation. Use these cataloged procedures to create cataloged procedures for your IMS Batch Terminal Simulator environment.

Table 10. Sample cataloged procedures

Cataloged procedure	Description
BTS	Runs IMS Batch Terminal Simulator in an IMS batch (DLI and DBB) processing environment
BTSBMP	Runs IMS Batch Terminal Simulator in an IMS BMP online environment
BTSJBP	Runs IMS Batch Terminal Simulator in an IMS JBP online environment

2. Copy the sample procedure that you want to customize: BTS, BTSBMP, or BTSJBP, which is in the BTS JCL library (SBTSJCL0), to your JCL library and edit the copy of procedure.
3. Follow the instructions in the header description of the cataloged procedure and make the following changes:
 - a) Change the JOB card to meet your system requirement.
 - b) Change *\$prclib* to the name of the cataloged procedure library.
 - c) Change *\$sout* to the SYSOUT class for output.

4. If necessary, change the data set name that is defined for the BTSOUT DD statement.

The BTSOUT data set stores the output listing that is generated by IMS Batch Terminal Simulator jobs.

In the sample procedures, the BTSOUT data set is allocated as SYSOUT=A. As an alternative, you can allocate a DASD data set for BTSOUT. If you do so, do not specify RLSE on the SPACE parameter.

5. Modify the QIOPCB and the QALTPCB DD statements.

The QIOPCB and QALTPCB DD statements describe the data sets that contain the output message queue for a 3270 display screen or 3270 printer.

For these data sets, specify RECFM=VBS, LRECL=&IOSEG, and BLKSIZE=&IOBLK, where:

&IOSEG

Is an integer value in the range of 12 - 32760. This value must be greater than or equal to the maximum length of a message segment. The default value is IOSEG=512.

&IOBLK

Is an integer value in the range of 16 - 32760. The default value is IOBLK=3072.

6. Modify the QALTRAN DD statement.

The QALTRAN DD statement describes a work file for alternate PCB output for a 3270 display screen or 3270 printer.

For this DD statement, specify RECFM=U (with no blocking) and BLKSIZE=&ALTSEG, where &ALTSEG is an integer value from (MAXSEGL+26) to 32760, both inclusive. Here, MAXSEGL is the maximum length of a message segment. The default value is ALTSEG=536.

If the specified value is too small, IMS Batch Terminal Simulator issues message BTS0076A at run time.

7. Test the customized cataloged procedure.

To ensure that the cataloged procedure is customized correctly, use the installation verification program (BTSIVP) JCL.

BTSIVP is an IMS Batch Terminal Simulator data stream to run IMS Batch Terminal Simulator in IMS batch processing region without 3270 formatting. This data stream does not require the IMS sample application and database. For more information about BTSIVP, see the *Program Directory for IBM IMS Batch Terminal Simulator for z/OS*.

- a) In the copy of BTSIVP, replace the BTS IN-STREAM PROCEDURE section with the cataloged procedure that you have customized.
- b) Submit the modified IVP job and ensure that the job ends without errors.

Customizing the BTS TSO command list (CLIST)

Customize the BTS TSO command list (CLIST) that is supplied with the product. This task is required to use IMS Batch Terminal Simulator in interactive mode.

About this task

To start IMS Batch Terminal Simulator in interactive mode, IMS Batch Terminal Simulator must be started in a TSO environment by using the BTS TSO CLIST. This task provides instructions for creating this CLIST.

Procedure

A sample CLIST is provided as part of the IMS Batch Terminal Simulator installation material. IMS Batch Terminal Simulator provides BTSCLIST JCL in the BTS JCL library (SBTSJCL0). The sample does not impose any special requirements for the logon procedure except that enough DD cards must be available for dynamic allocation.

1. Copy BTSCLIST, which is in the BTS JCL library (SBTSJCL0), to your JCL library and edit the copy of BTSCLIST.
2. Follow the instructions in the header description of the CLIST procedure and make the following changes:
 - a) Change the JOB card to meet your system requirement.
 - b) Change *\$btshlq* to the appropriate data set prefix.
 - c) Change *\$cmdlib* to the data set name of the TSO CLIST library.
 - d) Change *\$sout* to the SYSOUT class for output.
3. Save and submit your customized BTSCLIST JCL.

Ensure that the job ends with return code 0.

Customizing for IBM Debug Tool for z/OS or IBM z/OS Debugger

To use IBM Debug Tool for z/OS or IBM z/OS Debugger (Debug Tool) to debug IMS applications, you must set certain system environment conditions in the cataloged procedure or CLIST.

Procedure

Complete the following procedures to specify the DD statements for the cataloged procedures and CLISTs that you have prepared in the previous steps.

- For cataloged procedures, ensure that the Debug Tool library is included in the JOBLIB, STEPLIB, or LINKLIST DD. If the library is not concatenated, make the Debug Tool load module available to IMS Batch Terminal Simulator by adding the SEQAMOD data set to the JOBLIB, STEPLIB, or LINKLIST DD.

- For CLISTs, ensure that the Debug Tool library is included in the TASKLIB. If the library is not concatenated, make the Debug Tool load module available to IMS Batch Terminal Simulator by adding the SEQAMOD data set to the TASKLIB DD.

Customizing for specific applications

Customize the IMS Batch Terminal Simulator JCL and configure the environment for specific types of applications. This task is required if the application to be simulated is written in Java or REXX, or if the application accesses Db2 or IBM MQ.

Topics:

- [“Customizing for IMS REXX applications” on page 47](#)
- [“Customizing for IMS Java applications” on page 47](#)
- [“Customizing for IMS applications that access Db2 databases” on page 47](#)
- [“Customizing for IMS applications that access IBM MQ” on page 48](#)

Customizing for IMS REXX applications

To run IMS REXX applications with IMS Batch Terminal Simulator, you must set certain system environment conditions in the cataloged procedure or CLIST.

Procedure

Complete the following steps to specify the DD statements of IMS.

For more information about these DD statements and IMS adapter for REXX, see *IMS Application Programming APIs*.

1. Ensure that the DFSREXX0 and the DFSREXX1 data sets are in a load library that is accessible to IMS Batch Terminal Simulator (for example, STEPLIB).
2. Ensure that SYSEXEC DD points to a list of data sets that contain the REXX EXECs that will be run in IMS. You must put this DD in the cataloged procedure or CLIST.
3. Ensure that SYSTSPRT DD is specified and that this DD is put in the cataloged procedure or CLIST. SYSTSPRT data set is used for REXX output (for example, tracings, errors, and SAY instructions). SYSTSPRT DD is generally allocated as SYSOUT=A or another class, depending on installation.

Customizing for IMS Java applications

To run IMS Java applications with IMS Batch Terminal Simulator, you must set certain system environment conditions in the cataloged procedure or CLIST.

Procedure

Complete the following steps to set up your environment for testing IMS Java applications.

To find information about IMS Java applications, see *IMS Application Programming*.

1. Ensure that the cataloged procedure for JBP region is in place. The IMS Java application can run under the JBP region by using stand-alone JVM.
2. Ensure that the data set for the IMS Java class libraries is included in the STEPLIB.

Customizing for IMS applications that access Db2 databases

To run IMS applications that access Db2 databases with IMS Batch Terminal Simulator, you must set certain system environment conditions in the cataloged procedure or CLIST.

Procedure

Complete the following steps to specify the DD statements of IMS and Db2.

1. Ensure that the Db2 libraries are included in the JOBLIB, STEPLIB, or LINKLIST.
2. Make the Db2 load modules available to IMS Batch Terminal Simulator by adding a DD statement for the SDSNLOAD data set to the JOBLIB, STEPLIB, or LINKLIST.
3. If application will be tested in an IMS online environment (KW=BMP), and if any of the libraries in the JOBLIB or STEPLIB statement for IMS Batch Terminal Simulator are not APF-authorized, add the DFSESL DD statement to specify the SDSNLOAD data set and the IMS load module library as shown in the following example.

```
//DFSESL DD DSN=imsvs.SDFSRESL,DISP=SHR
// DD DSN=DSNxxx.SDSNLOAD,DISP=SHR
```

4. Ensure that all the libraries that are specified on the DFSESL DD statement are APF-authorized.
5. To enable application programmers to obtain IFI SQL statement information during IMS Batch Terminal Simulator run, activate Db2 monitor trace class 1 for IFCID 124. In addition, set the Db2 authorization ID of the user to have MONITOR2 privilege or SYSADM authority.
6. To enable application programmers to use the Db2 SQL call trace option, install and configure the following functions:
 - In BMP mode, IMS Attachment Facility
 - In DLI and DBB modes, Db2 DL/I Batch Support

Customizing for IMS applications that access IBM MQ

To run IMS applications that access IBM MQ with IMS Batch Terminal Simulator, you must set certain system environment conditions in the cataloged procedure or CLIST.

Procedure

Complete the following steps to specify the DD statements of IMS.

- Complete these steps to run applications in BMP region:
 - a) Ensure that the IBM MQ libraries (SCSQAUTH, SCSQANLE, and SCSQLOAD data sets) are installed in the LINKLIST. If the libraries are not installed in the LINKLIST, concatenate the SCSQAUTH data set to the STEPLIB DD, and concatenate the SCSQAUTH data set, the SCSQANLE data set, and the IMS load module library to the DFSESL DD.

```
//DFSESL DD DSN=imsvs.SDFSRESL,DISP=SHR
// DD DSN=CSQxxx.SCSQAUTH,DISP=SHR
// DD DSN=CSQxxx.SCSQANLE,DISP=SHR
```

- b) Ensure that all the libraries that are specified on the DFSESL DD statement are APF-authorized.
- Complete these steps to run applications in DLI or DBB region:
 - a) Ensure that the IBM MQ libraries (SCSQAUTH, SCSQANLE, and SCSQLOAD data sets) are installed in the LINKLIST. If the libraries are not installed in the LINKLIST, concatenate the SCSQAUTH and SCSQLOAD data sets to STEPLIB DD.
 - b) To enable application programmers to obtain MQI call trace information, add the STEPLIB DD statement to specify the SCSQLOAD data set. If the DD statement is not specified, IMS Batch Terminal Simulator returns message BTS0114I and does not report the call trace information for MQI calls on BTSOUT when the MQI call is issued by the application program that is linked with the IBM MQ IMS stub CSQQSTUB.

Implementing IMS user-written routines

Implement IMS user-written routines. This task is required if you want to implement IMS user-written routines.

Procedure

You can include optional user-written routines in BTS load module BTSRC000. The coding sample to include the IMS user-written routines in the BTS load module is supplied as member BTSPIXT0. This member is loaded into the BTS JCL library (SBTSJCL0) during installation. Use this JCL to implement IMS user-written routines. You can also use this JCL to implement other user-written routines by modifying ASM.SYSIN DD.

The following table summarizes the IMS user-written routines that are supported by IMS Batch Terminal Simulator.

Table 11. IMS user-written routines

Routine	Explanation
MFS input edit routines (DFSMExxx)	These user-written routines can be included to provide common editing functions for message input fields and segments under IMS Message Format Service (MFS).
User Message table (DFSCMTU0)	This user-written CSECT can be included to support user-written MFS input edit routines. You can set MFS segment exit routines to issue return code 12 and have the routines pass the number of a message from DFSCMTU0 to be displayed if included. If DFSCMTU0 is not included, IMS Batch Terminal Simulator message BTS0093W is displayed.
BTS Physical Input exit routine (BTSPIXT0)	<p>This user-written routine can be included to support application programs that specify MFS bypass (for example, where the modname supplied is DFS.EDT or DFS.EDTN). No input edit is provided by IMS Batch Terminal Simulator when MFS bypass is used, but IMS Batch Terminal Simulator internally performs basic input-editing (see Note 1). When MFS bypass support without basic input-editing is active, IMS Batch Terminal Simulator does not perform basic input-editing and does not update the screen image (see “MFS bypass support without basic input-editing” on page 102). You cannot use this function with GDDM. BTSPIXT0 can perform input-editing functions when MFS bypass is used that is otherwise provided in an IMS DC environment. These functions can include:</p> <ul style="list-style-type: none">• Basic input-editing otherwise provided by IMS DC• User input-editing otherwise provided by user-written routine DFSPIXT0• Required setting of flags in IMS control blocks by user-written routine DFSPIXT0• Handling of return codes from user-written routine DFSPIXT0 <p>BTSPIXT0 can call DFSPIXT0 depending on your requirement.</p>
IMS Physical Terminal Input exit routine (DFSPIXT0)	This user-written routine can be included if required by user-written routine BTSPIXT0.

Table 11. IMS user-written routines (continued)

Routine	Explanation
---------	-------------

Notes:

- a. There is one exception where IMS Batch Terminal Simulator does provide input-editing. The Graphical Data Display Manager (GDDM) can be used by an application program to generate graphical displays under FSS. Such applications use MFS bypass. The same basic input-editing is performed by IMS Batch Terminal Simulator for input from a GDDM formatted screen image that is otherwise provided by IMS DC.
 - b. For information about the exit routines and the user message table, see *IMS Exit Routines*.
 - c. For a description of MFS bypass, see *IMS Operations and Automation*.
-

Related reference

[BTS interface with a user-written BTSPIXTO routine](#)

IMS Batch Terminal Simulator interfaces with a user-written BTSPIXTO routine.

Configuring for IMS callable services

Configure for IMS callable services. This task is required to use IMS callable services.

About this task

IMS provides IMS callable services to enable the use of IMS exit routines. IMS Batch Terminal Simulator simulates a part of this function.

IMS Batch Terminal Simulator enables users who code the IMS exit routines to use the following IMS callable services:

- Storage services
 - Get storage
 - Free storage

When other services are requested, IMS Batch Terminal Simulator ignores the request, issues a warning message, and continues the process.

Related reading: For a detailed description of the IMS callable services, see *IMS Exit Routines*.

Procedure

To use callable services, link your exit routine to the IMS Batch Terminal Simulator callable service interface module BTSCSI00 rather than to DFSCSI00.

For some exit routines, this module is linked automatically by IMS Batch Terminal Simulator. For others, you must manually link this module to your exit routine.

The following exit routines are automatically linked to the BTSCSI00 module by IMS Batch Terminal Simulator:

- DFSME000
- DFSME127
- DFSPIXTO
- BTSPIXTO

To use the IMS callable services, manually link the DFSCSMB0 exit routine to the BTSCSI00 module.

Limitation: The storage is obtained in the private storage under IMS Batch Terminal Simulator. Therefore, the released storage is also in the private storage. If CSA storage is specified by the function-specific

parameter list (CSSTRG), IMS Batch Terminal Simulator ignores the specification and simulates the storage in the private storage.

Setting defaults for environment parameters (./E commands)

Set default keywords and operands for environment parameters (./E simulator command). This task is required if you want to set default environment parameters.

About this task

You can create and use the environment specification table to set default parameters for the ./E simulator command. At run time, IMS Batch Terminal Simulator refers to the environment specification table, and applies the specified keywords and operands for the ./E command without modification to the BTSIN input.

You can create the environment specification table (BTSCHTBL) by using the BTSCHDEF macro to change the default value of ./E command. The environment specification table must be included in the STEPLIB concatenation to be in effect.

The keywords and their operands that are defined in BTSCHTBL can be overridden by the ./E command in BTSIN at run time.

Procedure

1. Copy the sample procedure for specifying the BTSCHTBL table, which is in the BTS JCL library (SBTSJCL0), to your JCL library and edit the copy of JCL.

2. Code the TYP=INITIAL statement.

This statement is a required statement that indicates the start of a parameter list. TYP= and KEY= are mutually exclusive.

One TYP=INITIAL statement is required for each CMD operand. The format of this statement is:

```
BTSCHDEF TYP=INITIAL ,CMD=command
```

where *command* specifies a simulator command. Only E can be specified.

3. Code the KEY= statement.

This statement is a required statement that defines the keyword parameter for the command that is specified on the CMD= operand. KEY= and TYP= are mutually exclusive.

One KEY= statement is required for each VALUE operand. The format of this statement is:

```
BTSCHDEF KEY=keyword ,VALUE=operand
```

where *keyword* specifies the keyword and *operand* specifies the operand for the keyword. An *operand* must be specified.

4. Code the TYP=FINAL statement.

No other parameters are valid on a TYP=FINAL statement. The format of this statement is:

```
BTSCHDEF TYP=FINAL
```

Example

The IMS Batch Terminal Simulator environment specification table can be coded as shown in the following figure.

```

BTSCHTBL  CSECT
          BTSCHDEF  TYP=INITIAL ,CMD=E
          BTSCHDEF  KEY=NOEDIT ,VALUE=YES
          BTSCHDEF  TYP=FINAL
          END

```

This specification is the same as writing the following statement in BTSIN.

```
./E NOEDIT=YES
```

Setting up the tutorial environment

IMS Batch Terminal Simulator information provides tutorial topics for application programmers who are new to IMS Batch Terminal Simulator. To access these tutorials, you must set up the tutorial environment.

Procedure

- For batch mode, install and set up the IMS sample database and the IMS sample application.
- For interactive mode, complete the following steps:
 - a) Install and set up the IMS sample database and the IMS sample application.
 - b) Install the IMS Batch Terminal Simulator sample MFS format by running the IMS MFS language utility by using BTSSAMFT as input.

The MFS format is installed as member BTSSAMFT, which is loaded into SBTSJCL0 during the installation of IMS Batch Terminal Simulator.

- c) Customize the TSO CLIST.
- d) Allocate a data set for BTSIN.

For the data set name, use your installation standard naming conventions (for example, DSN=USERID.BTSIN.DATA). The job that is used in the tutorial (BTSSAMP3) and the BTS CLIST assume that BTSIN is a sequential data set.

Related reading: For an instruction to install and set up the IMS sample database and the IMS sample application, see *IMS Installation*.

Recording the resource information

Summarize the resource information and provide it to the application programmers. This task is an easy way to provide application programmers with the resource and the environment information.

Procedure

Fill out the following tables and provide the tables to your application programmers who will be using the IMS Batch Terminal Simulator environment. Application programmers can refer to these tables to specify their job parameters.

Table 12. Data set names for IMS library

Name	Local DSN name
SDFSRESL	
DBDLIB	
PSBLIB	
ACBLIB	
PROCLIB	
REFERAL	

Table 12. Data set names for IMS library (continued)

Name	Local DSN name
FORMAT	

Table 13. Data set names for IMS Batch Terminal Simulator library

Name	Local DSN name
BTSIN	
BTSOUT	
BTSDEBUG	
BTSPUNCH	

Table 14. Other product libraries

Library	Name	Local DSN name
Db2 library	SDSNLOAD	
IBM MQ library	SCSQAUTH	
	SCSQANLE	
	SCSQLOAD	
IBM Debug Tool library	SEQAMOD	
LE runtime library	SCEERUN	

Table 15. Data set names for user libraries

Name	Local DSN name
Application program library	
CLIST library	

Distributed Access Infrastructure configuration

For the Eclipse interface or the IMS Batch Terminal Simulator resource adapter, Distributed Access Infrastructure is required on the z/OS system that you want to test on. Distributed Access Infrastructure acts as a gateway for communication between distributed platforms and z/OS.

Configure and start the Distributed Access Infrastructure servers before using the Eclipse interface or resource adapter.

After configuring Distributed Access Infrastructure, you must have the following information to configure the Eclipse interface and resource adapter:

- The Distributed Access Infrastructure server address and port number
- An authorized user ID and password for the z/OS system that Distributed Access Infrastructure is installed on
- The location of the keystore and truststore certificates and the keystore and truststore passwords if SSL enabled for the port that Distributed Access Infrastructure is using

For more information about Distributed Access Infrastructure configuration or SSL security, see the *IMS Tools Base Distributed Access Infrastructure User's Guide and Reference*.

Installing Eclipse interface

To use the IMS Batch Terminal Simulator Eclipse interface, you must install IMS Batch Terminal Simulator plugins into the Eclipse installation that you want to test IMS applications in.

Procedure

1. After the IMS Tools Base for z/OS SMP/E installation is complete, use FTP to transfer the AIIBTSUI member from the *hlq.SAIIGENU* partitioned data set on z/OS to the computer with the Eclipse installation that you want to use.
2. Rename the AIIBTSUI file to *file_name.zip*.
You can use any name for the file, but it must have the .zip extension.
3. If you are migrating from a previous version of the Eclipse interface, remove any plugins that start with *com.ibm.imstools* from the *Eclipse_installation\dropins* directory.
4. Extract the contents of the .zip file into the *Eclipse_installation\dropins* directory.
5. Start Eclipse.

Your Situation	Action to Take
If you are migrating from a previous version	Start Eclipse by using the following command: <pre>eclipse.exe -clean</pre>
If you did not have a previous version installed	Start Eclipse as you normally would.

6. Verify that IMS Batch Terminal Simulator is a server type.
 - a) From the **File** menu, click **Window > Show View > Servers** to open the Servers window.
 - a) Right-click in the Servers window, and then click **New > Server** to start the New Server wizard.
 - b) From the server type list, select **IBM > IMS Batch Terminal Simulator**.
If you see IMS Batch Terminal Simulator as a server type, the plugins are installed correctly. If you do not see IMS Batch Terminal Simulator, ensure that you have the correct Eclipse version and that the plugins are in the correct directory.

Related concepts

[Architecture and process flow](#)

IMS Batch Terminal Simulator simulates the functions of IMS and simulates the activities of IMS applications without starting an IMS control region.

Installing the IMS Batch Terminal Simulator resource adapter

Install and configure the IMS Batch Terminal Simulator resource adapter (BTSRA) by using the WebSphere Application Server administrative console.

About this task

BTSRA provides connectivity between IMS Batch Terminal Simulator, the application server, and the client application. For more information about resource adapters, see the WebSphere Application Server documentation.

Procedure

1. After the IMS Tools Base for z/OS SMP/E installation is complete, use FTP to transfer the AIIBTSRA member from the *hlq.SAIIGENU* partitioned data set on z/OS to a location where WebSphere Application Server can access the member.
2. Rename the AIIBTSRA member to *file_name.rar*.

You can use any name for the file, but it must have the .rar extension.

3. Log in to the WebSphere Application Server administrative console.
4. From the navigation menu, click **Resources > Resources adapters**.
5. On the Resource adapters page, click **Install RAR**.
6. Specify a node and cell.

The values for the node and cell are specified by the WebSphere Application Server system administration. Contact the system administrator for the correct node and cell values.

7. Specify the location of the .rar file.
8. Click **Next** and accept the default settings.

After you save the configuration, BTSRA is installed.

Results

Verify the installation by viewing the Resource adapters page. If BTSRA is listed as one of the resource adapters that you can administer, the installation was successful.

Related concepts

[Architecture and process flow](#)

IMS Batch Terminal Simulator simulates the functions of IMS and simulates the activities of IMS applications without starting an IMS control region.

Adding and configuring the IMS Batch Terminal Simulator resource adapter J2C connection factory

After installing the IMS Batch Terminal Simulator resource adapter (BTSRA), use the WebSphere Application Server administrative console to add and configure the BTSRA J2C connection factory.

Before you begin

From the IMS Batch Terminal Simulator Eclipse interface, export the server runtime environment and launch configuration settings for the application that you want to test. See [“Step 5: Exporting a server runtime environment and launch configuration \(optional\)”](#) on page 109 for more information about exporting an XML file.

Important: You must remove Transaction Manager resource adapter (TMRA) and its connection factories to successfully deploy BTSRA in WebSphere Application Server. Also, the BTSRA connection factory property values must be the same as the TMRA property values except for values that are unique to IMS Batch Terminal Simulator. With similar property values, applications can behave consistently when you switch between resource adapters.

About this task

A J2C connection factory is required for each application that accesses an enterprise information system (EIS), such as IMS Batch Terminal Simulator when testing IMS TMRA client applications. The BTSRA J2C connection factory contains information to connect to Distributed Access Infrastructure and to run IMS Batch Terminal Simulator.

Procedure

1. Log in to the WebSphere Application Server administrative console.
2. From the navigation menu, click **Resources > Resource adapters**.
3. On the Resource adapters page, click **BTSRA**.

If BTSRA is not listed, you must install the BTSRA on WebSphere Application Server. For more information, see [“Installing the IMS Batch Terminal Simulator resource adapter”](#) on page 54.

4. In the Additional properties section, click **J2C connection factories**.
5. Click **New** and specify the Name and JNDI Name for this connection factory.

Specify the same name and JNDI name that are used in the TMRA client application that you are testing. By using the same names, you do not have to modify the client application to communicate with BTSRA.

6. Click **OK** to save the J2C connection factory.

The connection factory is listed as part of the J2C connection factories for BTSRA.

7. On the J2C connection factories page, click the name of the connection factory that you created.

8. In the Additional properties section, click **Custom properties**.

9. In the Custom properties page, specify the following properties:

- For the **BTSConfigFile** field, specify the location of the IMS Batch Terminal Simulator configuration XML file, which specifies the IMS application and IMS Batch Terminal Simulator runtime settings.
- For the **BTSLoggingConfigFile** field, specify the location of the `log4j.properties` file, which specifies the log settings for the client application.
- For the **BTSOutputDirectory** field, create and specify the location where all the IMS Batch Terminal Simulator output is saved.
- For the **HostName** and **Port number** fields, specify the Distributed Access Infrastructure address and port number.
- For the **User ID** and **Password** fields, specify an authorized z/OS user ID and password that can access Distributed Access Infrastructure.

Remember: To test applications that already use a defined TMRA, specify the same J2C connection factory custom properties that are defined in the TMRA J2C connection factory custom properties. However, you are not required to specify property values that are not supported by BTSRA. For example, the Local Option function in TMRA is not supported in BTSRA, so you do not have to specify a value for `IMSCoordinateName`. For more information about supported TMRA functions, see [“Transaction Manager resource adapter supported functions”](#) on page 111.

10. Click **Save**.

BTSRA is deployed and you can start testing applications with IMS Batch Terminal Simulator. If the client application invokes more than one IMS application, create additional connection factories for each IMS application.

What to do next

After you install BTSRA and configure the required connection factories, you can run the client application to test applications in IMS Batch Terminal Simulator instead of IMS.

Related tasks

[Testing an application that uses the IMS Batch Terminal Simulator resource adapter](#)

For TMRA client applications that run on WebSphere Application Server, you can use the IMS Batch Terminal Simulator resource adapter (BTSRA) to run IMS applications on IMS Batch Terminal Simulator instead of on IMS.

Part 3. Testing applications

With IMS Batch Terminal Simulator, you can run your application programs in either of the following two runtime modes; batch mode or interactive mode.

Topics:

- [Chapter 5, “Restrictions and considerations,” on page 59](#)
- [Chapter 6, “Testing applications in batch mode,” on page 65](#)
- [Chapter 7, “Testing applications in interactive mode,” on page 79](#)
- [Chapter 8, “Tutorial: Testing an application from Eclipse,” on page 105](#)
- [Chapter 9, “Testing an application that uses the IMS Batch Terminal Simulator resource adapter,” on page 111](#)
- [Chapter 10, “Customizing JCL for specific applications,” on page 113](#)
- [Chapter 11, “Including BTS optional user-written exit routines,” on page 125](#)
- [Chapter 12, “Interpreting the simulation output,” on page 139](#)

Chapter 5. Restrictions and considerations

Certain restrictions and considerations apply to using IMS Batch Terminal Simulator.

Topics:

- [“Restrictions” on page 59](#)
- [“Considerations” on page 60](#)

Restrictions

Be aware that certain restrictions apply to using IMS Batch Terminal Simulator.

Subsections:

- [“IMS data communication functions” on page 59](#)
- [“Logging capability” on page 59](#)
- [“Extended operator control functions” on page 59](#)
- [“Unsupported IMS parameters” on page 59](#)
- [“Abend in a PL/I application” on page 60](#)
- [“Printing of segment search argument \(SSA\)” on page 60](#)
- [“Access to data entry databases \(DEDBs\)” on page 60](#)
- [“2740 Typewriter output formatting in 24 bit mode” on page 60](#)
- [“Waiting-for-input \(WFI\) BMPs” on page 60](#)
- [“Scheduling multiple PSBs” on page 60](#)
- [“SQL calls in REXX application” on page 60](#)

IMS data communication functions

While IMS Batch Terminal Simulator approximates IMS Data Communications (DC) functions in most instances, it does not duplicate all the complexities of the IMS DC environment in precisely the same way.

IMS Batch Terminal Simulator does not support the IMS PASSWORD function or the PAGDEL=NO function of the IMS DC generation TERMINAL macro.

Logging capability

If a log data set is provided for an IMS Batch Terminal Simulator run, IMS can log database operations as usual. However, log records that are associated with teleprocessing activity are not produced, and certain consideration applies when using the log as input to the IMS Batch Backout utility. For more information, see [“Using the log data set for database recovery” on page 63](#).

Extended operator control functions

IMS Batch Terminal Simulator does not support the following four extended operator control functions: NEXTMSGP, NEXTLP, NEXTTP, and NEXTMSG.

Unsupported IMS parameters

The MFS SYSMMSG field is not supported.

The PSBGEN MAXQ operand is not supported.

Abend in a PL/I application

If an abend occurs while running a PL/I application program, the results might be unpredictable due to an incompatibility between IMS Batch Terminal Simulator and the PL/I SPIE routine.

Printing of segment search argument (SSA)

When IMS Batch Terminal Simulator prints the segment search argument (SSA) following a D/B PCB call to IMS, the print line is terminated when IMS Batch Terminal Simulator encounters a right parenthesis or SSA length of 512 bytes, whichever occurs first. If the SSA contains a binary value that includes the EBCDIC representation of a right parenthesis, output of the SSA stops at that location.

Access to data entry databases (DEDBs)

IMS Batch Terminal Simulator allows accessing of Fast Path databases when IMS Batch Terminal Simulator is run in an online IMS BMP type region. The application can access a data entry database (DEDB) and a non-terminal-related MSDB. However, it cannot access a terminal-related MSDB. In this environment, IMS is not aware that a terminal is associated with the dependent region.

2740 Typewriter output formatting in 24 bit mode

You can run application programs in 31 bit mode (AMODE=31 and RMODE=ANY). Parameters passed in DL/I or Db2 can refer to code or storage areas above the 16 megabyte line. However, all parameter addresses passed in DL/I or Db2 calls must refer to code or storage areas below the 16 megabyte line when using 2740 Typewriter Output Formatting in 24 bit mode (AMODE=24 and RMODE=24).

Waiting-for-input (WFI) BMPs

IMS Batch Terminal Simulator has no special facilities for the simulation of waiting-for-input (WFI) BMPs. IMS Batch Terminal Simulator does not interface to the IMS message queue. A BTS BMP can look like a WFI BMP by routing the BTSIN DD to a terminal. Delaying between inputs from the terminal causes the BMP to wait for input. There is no way for IMS Batch Terminal Simulator to simulate the execution of multiple transactions concurrently as could happen in the case of a real IMS WFI environment. Examples of this would be a transaction that initiated a WFI transaction through an express TP PCB, or a transaction that initiated a WFI BMP transaction that had multiple transactions on the message queue and a PLC >= 2. Either of these cases could cause the initiating transaction and a WFI BMP transaction to be executed concurrently.

Scheduling multiple PSBs

When multiple PSBs are scheduled by a BTS batch job (KW=DLI or KW=DBB), the log data set cannot be used to backout database changes made by a PSB that is different from the first PSB that made update (type X'50') log records on the log data set. The IMS batch backout utility fails with the abend code U0071 and the message DFS0428I.

When multiple PSBs are scheduled with DBRC by the IMS Batch Terminal Simulator batch job (KW=DBB), you must specify REATTCH=YES on the ./E command or the ./T command.

SQL calls in REXX application

When the SQL calls are run in REXX application, IMS Batch Terminal Simulator does not handle the SQL calls written in REXX. Therefore the SQL call trace is not printed on the BTSOUT or displayed on the TSO terminal.

Considerations

Be aware that certain considerations apply to using IMS Batch Terminal Simulator.

Subsections:

- [“Execution time” on page 61](#)
- [“Virtual storage requirements” on page 61](#)
- [“Applications that update databases” on page 62](#)
- [“LRECL value for BTSOUT data set for 3270 formatting” on page 62](#)
- [“Termination prior to end of job” on page 62](#)
- [“Using the log data set for database recovery” on page 63](#)

Execution time

IMS Batch Terminal Simulator does not simulate the execution times that could be experienced in an IMS production environment. Application program execution times under IMS Batch Terminal Simulator can be affected by the following factors:

- Machine speed
- Application program processing
- Mix of other jobs within the machine

Virtual storage requirements

The IMS Batch Terminal Simulator program modules require 2 MB of virtual storage. This virtual storage is in addition to the storage that is required by IMS, and the amount of virtual storage IMS Batch Terminal Simulator must acquire dynamically by issuing GETMAINS. More virtual storage might be required depending on other products in your environment, such as Db2, IBM MQ, and Language Environment, and on the storage requirements of individual application programs.

Insufficient storage problems might occur when a large number of transactions are run. These problems are due to storage fragmentation after many attaches of IMS. To avoid these problems, consider taking the following actions:

- Increase the region size.
- Increase the value for PLC on the `./T` command.
- Use multiple job steps.

The requirements summarized in the following table can help you estimate the amount of virtual storage required for running a particular IMS application with IMS Batch Terminal Simulator.

Table 16. Application-dependent main storage requirements

BTS queue	Storage requirements
<code>./T</code> Transaction Data Queue	162 bytes per <code>./T</code> command. This space remains for the duration of the BTS run.
<code>./P</code> Patch Data Queue or <code>./E</code>	Maximum of 160 bytes per <code>./P</code> or <code>./E</code> command. After a Patch Data Queue element has been used, the element is dequeued and the space is freed. This queue is also used to manage the SPAs that are passed to application programs. There is no maximum length for an SPA on this queue. After the SPA is passed to the application program, the element is dequeued and the space is freed.
<code>./S</code> SNAP Data Queue	48 bytes per <code>./S</code> command. When the use count is reduced to zero, the associated SNAP queue element is dequeued and its space freed. The SNAP queue is also used to queue requests for a particular status code on a DL/I call.
Message Queue	For secondary transactions only; (segment length + 28) per segment. (Multi-segment messages are processed one segment at a time.) The message queue is managed dynamically. When an element causes an application program to be scheduled, the element is dequeued and its space is freed. The scheduled program, however, might add elements to the queue.

Table 16. Application-dependent main storage requirements (continued)

BTS queue	Storage requirements
Statistics Queue	288 bytes per PCB against which any IMS calls are made by an application program.
3270 Teleprocessing Buffer	Enough space to contain the largest I/O message of the application program.
Format Block Pool	Enough space for an input area for the FORMAT data set.
Internal 3270 Screen Image	3 x lines x columns of the I/O PCB device type. The device type is specified by the ./D command.
Input Message Work Area	Two times maximum input message length.
Output Message Work Area	Sum of LRECL + BLKSIZE for the queue data sets.

Applications that update databases

If you are debugging an application program that updates a database, it is generally easier to have exclusive use of that database. That way you have better control of the test and the test results (exclusive use is not a requirement.) Dynamic Backout as well as Database Data Set Recovery considerations apply as they would for online updates by any other dependent region.

LRECL value for BTSOUT data set for 3270 formatting

The LRECL value specified for the BTSOUT data set can affect the output IMS Batch Terminal Simulator produces. IMS Batch Terminal Simulator checks the BTSOUT LRECL when it writes an output record. If the output is greater than the LRECL value, the output is broken up into two or more records. Normally this is not a matter of concern because the recommended value of 133 gives the best results. However, if the IBM 3278 Model 5 with the 132 character line is being simulated, some problems might occur. In supporting the Model 5, IMS Batch Terminal Simulator attempts to produce only one line of output for each line on the screen image. If the LRECL is 133, the framing characters indicating the line numbers are not shown. Also the first half byte of any field attribute character in column 1 is not shown. If the LRECL is 132 or less, IMS Batch Terminal Simulator shows all characters and framing, and two lines are produced for every line of the screen. If your printer has a large enough print line, consider setting the LRECL to 141. This setting allows all Model 5 screen image information on one print line.

Whenever you change terminal type, the screen image might appear different depending on the LRECL of the BTSOUT data set and terminal screen size. For example, after a switch from a 3278-2 (24 x 80) to a 3278-5 (27 x 132), the screen images might appear to be inconsistent with each other. This condition is caused by IMS Batch Terminal Simulator using the default screen rows and screen length from TSO for a particular terminal. In this example (3278-2 to 3278-5), to keep the screen image the same, specify the 'TERMINAL SCRSIZE(24,80)' in the TSO command language used for the application. Also, specify the ./D command similar to the following example:

```
./D TYPE=3270-A2 SIZE=(24,80)
```

To make full use of the 3278-5 screen size, the MOD and the DOF must be generated again with the DEV statement of the MOD indicating TYPE=(3270,7). A ./D TYPE=3270-A7 SIZE=(27,132) command is also necessary.

Termination prior to end of job

Termination of an IMS Batch Terminal Simulator simulation run prior to normal end of job (EOJ) imposes considerations identical to the considerations in a production IMS application run. There are no logging or security functions provided by IMS Batch Terminal Simulator. However, all recovery or restart procedures,

inherent in the installation's mode of operation with respect to IMS applications, are applicable to an IMS Batch Terminal Simulator run. Databases are modified with or without logging.

Note: If the application program ends abnormally and output has been inserted to alternate logical terminals, this output is lost. Correct the application program and rerun to see the output.

Using the log data set for database recovery

The following considerations apply when using the log data set for database recovery:

- The log data set can be used for forward recovery of an IMS database data set.

The log data set is suitable for input to the IMS Database Data Set Recovery Utility, the IMS Change Accumulation Utility, and other log utilities (including Database Recovery Control DFSLOG) whose function is to provide for forward recovery.

- The log data set *can be used* to back out database changes made by a transaction that *did not* complete successfully.

The log data set can be input to the IMS Batch Backout Utility where the *CHECKPOINT ID* is *not specified*. In the Batch Backout Utility JCL, either do not supply SYSIN DD statement or specify SYSIN DD DUMMY. This returns the database to its state at the last successful checkpoint written to the log.

Related reading: See [“DL/I synchronization” on page 27](#) for information about application sync point.

- In general, the log data set *cannot be used* to back out database changes made by transactions that completed successfully.

In general, the log data set *cannot be* input to the IMS Batch Backout Utility where a *CHECKPOINT ID* is *specified*.

To enable the IMS logging facility, you must specify the IEFORDER DD statement.

Chapter 6. Testing applications in batch mode

You can use IMS Batch Terminal Simulator to test your applications in batch mode or in interactive mode. In batch mode, you can specify the input for IMS Batch Terminal Simulator in the BTSIN data set and run your application without intervention. Batch mode is especially helpful in functional verification testing and regression testing.

Topics:

- [“Tutorial: Running the sample application in batch mode” on page 65](#)
- [“Running applications in batch mode” on page 68](#)
- [“Debugging applications in batch mode” on page 75](#)

Tutorial: Running the sample application in batch mode

Use this tutorial to learn how to run applications in batch mode.

For application developers who want to test their applications as a batch job, IMS Batch Terminal Simulator provides a method for testing applications in batch.

Learning objectives

This tutorial uses BTSSAMP1 JCL that is supplied with the product and is copied to the BTS JCL library (SBTSJCL0) during the installation. This JCL stream runs the IMS sample application in the following ways:

- In batch mode
- In an IMS batch processing region (DLI)
- Without 3270-formatting

By completing the tutorial, you will learn how to complete the following tasks:

- Test IMS sample application in batch mode
- Code JCL statements for running IMS Batch Terminal Simulator in batch mode
- Understand the input and the output data sets that are used in batch mode

Time required

This tutorial takes approximately 30 minutes to complete.

Audience

This tutorial is targeted for application developers with some z/OS and IMS knowledge.

Prerequisites

Make sure that your environment is properly set up for this tutorial by checking with your system administrator.

- Verify with your system administrator that the IMS sample application has been installed. This tutorial requires that the IMS sample application that uses the DI21PART database has been installed.
- Obtain the data set name template from your system administrator. The data set name template contains the information about the required system resources. For more information about the data set name template, see [“Recording the resource information” on page 52](#).

Modifying the sample JCL and running a job

In this lesson, you will modify the sample JCL to run sample applications in batch mode.

About this task

This tutorial runs the following transactions of the IMS sample application:

PART, DSPALLI, DSPINV, ADDPART, ADDINV, DLETINV, DLETPART, CLOSE, DISBURSE

To learn more about the IMS sample application, see *IMS Installation*.

Procedure

1. Copy BTSSAMP1 JCL to your JCL library and edit the copy of BTSSAMP1 JCL.
2. Follow the instructions in the header description of BTSSAMP1 JCL and make the following changes:
 - a) Change the JOB card to meet your system requirements.
 - b) Specify the IMS Batch Terminal Simulator libraries.

Locate *\$btshlq*. Change the value to the data set prefix name that is described in the data set name template.
 - c) Specify the IMS libraries.

Locate *\$imshlq*. Change the value to the data set prefix name that is described in the data set name template.
 - d) Specify the DI21PART database.

DI21PART database is the IMS Sample Database provided by IMS.
Locate *\$btsuhlq*. Change the value to the data set prefix name that is described in the data set name template.
3. Locate the STEPLIB DD statement and ensure that the PGMLIB data set is specified for the DD statement.

The PGMLIB data set contains the user-written application programs. In this tutorial, the PGMLIB data set contains the IMS sample application.
4. Locate the BTSOUT DD statement.

The BTSOUT data set contains the output listing from the IMS Batch Terminal Simulator job. If you want to change the output data set, specify the data set name for BTSOUT DD.
5. Locate the BTS SAMPLE JCL (BTSSAMP1) section.

This section contains the JCL for running IMS Batch Terminal Simulator. The following figure shows the BTS SAMPLE JCL (BTSSAMP1) section.

```

//*------*
//* BTS SAMPLE JCL (BTSSAMP1) *
//*------*
//BTSSAMP1 EXEC BTS, 1
//      BTSIDX=$btshlq,
//      IMSIDX=$imshlq
//G.BTSIN DD *
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02 2
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./T TC=ADDPART LANG=CBL MBR=DFSSAM04
./T TC=ADDINV LANG=CBL MBR=DFSSAM04
./T TC=DLETINV LANG=CBL MBR=DFSSAM04
./T TC=DLETPART LANG=CBL MBR=DFSSAM04
./T TC=CLOSE LANG=CBL MBR=DFSSAM05
./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
PART AN960C10$ 3
DSPALLI AN960C10$
DSPINV AN960C10,28009126$
ADDPART AB960C10,RIVET,74$
ADDINV AB960C10,8009126A$
DSPINV AB960C10,8009126A$
DLETINV AB960C10,8009126A$
DLETPART AB960C10$
//G.DI21PART DD DSN=$btsuhlq.DI21PART,DISP=OLD
//G.DI21PARO DD DSN=$btsuhlq.DI21PARO,DISP=OLD
//G.DFSVSAMP DD *
4096,4
8192,4
/*

```

Figure 7. BTSSAMP1 JCL

- a) The first line (1) shows the EXEC statement for IMS Batch Terminal Simulator. The BTSIDX and the IMSIDX that are used in the cataloged procedure (BTS) are the high level qualifiers of the IMS Batch Terminal Simulator and the IMS system libraries.
- b) The BTSIN DD statement contains the input to IMS Batch Terminal Simulator and your application.
 - The lines that start with ./T (2) specify the IMS teleprocessing system generation information about the transactions. The TC operand specifies the transaction code, and the MBR operand specifies the module name. These specifications are called the *simulator commands*. IMS Batch Terminal Simulator provides various simulator commands to support testing of various types of applications.
 - The lines that follow the ./T commands (3) and that end with a dollar sign (\$) specify the transaction codes and transaction message input to the application. These specifications are called the *simulator statements*. For example, PART AN960C10\$ requests the PART transaction to function with input of AN960C10. A dollar sign (\$) indicates the end of a simulator statement.
6. Submit the BTSSAMP1 job. IMS Batch Terminal Simulator processes all the input that is specified on the BTSIN DD. When the job ends, ensure that the return code is 0.
7. Check the output in the BTSOUT data set.

The BTSOUT data set contains the output from the IMS Batch Terminal Simulator run. The output consists of BTSIN echo back, DL/I call information, and trace information of the internal processes.

Lesson checkpoint

You have run an IMS sample application in batch mode in the IMS Batch Terminal Simulator environment.

You learned how to code the IMS Batch Terminal Simulator JCL statements and how to run an IMS application in batch mode.

Running applications in batch mode

To run an IMS Batch Terminal Simulator job in batch mode, submit the IMS Batch Terminal Simulator JCL. You can use IBM Rational Developer for System z or IBM Developer for z Systems to perform these steps.

Before you begin

If you want to implement exit routines, follow the instructions in the following topics:

- [Chapter 11, “Including BTS optional user-written exit routines,” on page 125](#)
- [“Implementing IMS user-written routines” on page 49](#)

About this task

Complete these steps to run applications in batch mode:

- [“Step 1: Specifying the JOB and the EXEC statements” on page 68](#)
- [“Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 69](#)
- [“Step 3: Specifying the IMS data sets” on page 70](#)
- [“Step 4: Defining the transactions and the simulation environment” on page 72](#)
- [“Step 5: Specifying the simulator statements” on page 74](#)

Step 1: Specifying the JOB and the EXEC statements

Define the JOB and the EXEC statements in the IMS Batch Terminal Simulator JCL.

Procedure

1. You can run your application in a DLI region, DBB region, BMP region, or JBP region. Determine the region type by evaluating the characteristics of each region in [“Determining the types of IMS region” on page 22](#).

To run your application in BMP or JBP region, you must start the IMS control region. You do not need to start the IMS control region if you are running your application in a DLI or DBB region.

2. Prepare your IMS Batch Terminal Simulator JCL and code the JOB and the EXEC statements.

Select the BTS cataloged procedure to use. The BTS cataloged procedures define the default values for the EXEC statement. To override these default values, provide the override values in the EXEC statement.

For the EXEC statement, you can specify the following parameters:

KW=region_type

Specifies the type of IMS region.

SOUT=x

Specifies the SYSOUT class.

The following figure shows an example. In this example:

- Cataloged procedure BTS is selected to run IMS Batch Terminal Simulator in an IMS batch processing environment.
- The KW parameter is set to DBB to run IMS Batch Terminal Simulator in a DBB region.
- The SOUT class specifies B.

```
//BTSJOB  JOB  ACCNT,NAME,MSGLEVEL=1
//          EXEC  BTS,KW=DBB,SOUT=B
```

For the definitions of other execution parameters that can be specified, see *IMS System Definition*.

Important: For KW=DBB, if you do not supply a log data set in the JCL, you must specify DBRC=N as an EXEC parameter; otherwise, the region abends with a U0073. IMS always assumes update intent for BTS DBBATCH execution. For an explanation of the abend, see *IMS Messages and Codes*.

Step 2: Specifying the IMS Batch Terminal Simulator data sets

Define the input and output data sets for IMS Batch Terminal Simulator in the IMS Batch Terminal Simulator JCL.

Procedure

Specify the IMS Batch Terminal Simulator DD statements.

The BTS cataloged procedures define the default parameters for the DD statements. To override these default parameters, provide the override parameters in the JCL.

The following IMS Batch Terminal Simulator DD statements are supported.

Table 17. IMS Batch Terminal Simulator DD statements for batch mode

DD name	Description
BTSIN	<p>The BTSIN DD is a required DD statement that defines the input data set for IMS Batch Terminal Simulator. This data set contains the simulator commands and simulator statements.</p> <p>Important:</p> <ul style="list-style-type: none"> • The BTSIN data set must not be a concatenated SMS-managed data set. If BTSIN is a concatenated SMS-managed data set, unpredictable results might occur. • The BTSIN data set must be in fixed, fixed-block, variable, or variable-block format. • When the BTSIN data set is in fixed or fixed-block format, BTSIN LRECL that is longer than 80 bytes is truncated to 80 bytes. • When the BTSIN data set is in variable or variable-block format, BTSIN LRECL can be in the range of 5 - 32756 without any truncation.
BTSOUT	<p>The BTSOUT DD is a required DD statement that defines the output data set for IMS Batch Terminal Simulator. After the job is run, this data set contains the output listing from the job.</p> <p>If you are allocating BTSOUT for IBM 3278 Model 5 display, the LRECL value that you specify can affect the output that IMS Batch Terminal Simulator produces. See “LRECL value for BTSOUT data set for 3270 formatting” on page 62 for more information.</p>
BTSACB	<p>The BTSACB DD is an optional DD statement that specifies the ACB library. Consider specifying this DD when simulating the application programs that access data entry databases (DEDBs) in BMP or JBP region with IMS 13 or earlier.</p> <p>See “Applications that access DEDBs” on page 118 for more information about specifying BTSACB DD.</p>
BTSISRIO	<p>The BTSISRIO DD is an optional DD statement that defines the output data set for the data that the DLI ISRT (Insert) call inserts to IOPCB.</p> <p>The BTSISRIO data set must be in variable or variable-block format. The LRECL of BTSISRIO must be 32756 to avoid truncation.</p>

Table 17. IMS Batch Terminal Simulator DD statements for batch mode (continued)

DD name	Description
BTSSNAP	The BTSSNAP DD is an optional DD statement that defines the output data set for snapshot dumps. This data set is used only when you request a snapshot dump by issuing the ./S command.
BTSDEBUG	<p>The BTSDEBUG DD is an optional DD statement that defines the output data set for snapshot dumps of the trace table and various control blocks. These snapshots are taken at critical points during an IMS Batch Terminal Simulator run. You can specify this DD in the following formats:</p> <ul style="list-style-type: none"> • //BTSDEBUG DD DUMMY • //BTSDEBUG DD SYSOUT=A <p>Allocation of a data set with the DD name of BTSDEBUG starts the BTS DEBUG function.</p>

The following figure shows an example.

```
//BTSJOB      JOB      ACCNT,NAME,MSGLEVEL=1
//PROCLIB    JCLLIB  ORDER=(BTS.PROCLIB)
//BTS        EXEC    BTS
//G.BTSOUT    DD      SYSOUT=B,
//           DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//G.BTSDEBUG  DD      DUMMY
//G.BTSSNAP   DD      SYSOUT=B,SPACE=(TRK,(10,15))
//G.BTSIN     DD      *
BTS commands and simulator statements
(optional '/FORMAT' statements)
(optional '/EXIT' statements)
/*
```

Step 3: Specifying the IMS data sets

Define the input and output data sets for IMS in the IMS Batch Terminal Simulator JCL.

Procedure

Specify the IMS and z/OS DD statements.

The BTS cataloged procedures define the default parameters for the DD statements. To override these default parameters, provide the override parameters in the JCL.

The following table summarizes the IMS and the z/OS DD statements that are required to override the values in the BTS cataloged procedures.

Table 18. IMS and z/OS DD statements for batch mode

DD name	Description
STEPLIB or JOBLIB	<p>The STEPLIB or the JOBLIB DD statement is a required DD statement that defines the following input data sets:</p> <p>IMS library The library that contains the IMS load modules.</p> <p>Program library (PGMLIB) The program library that contains the user-written application programs.</p> <p>Db2 library (optional) The library that contains the Db2 load modules.</p> <p>IBM MQ library (optional) The library that contains the IBM MQ load modules.</p> <p>Language Environment library (optional) The library that contains the Language Environment modules.</p> <p>Debug Tool library (optional) The library that contains the IBM Debug Tool modules.</p> <p>Other libraries Other libraries such as language libraries.</p>
IMS	The IMS DD statement is an optional DD statement. This DD statement defines the library that contains the DBD and the PSB that describe the database to be processed.
IMSACB	The IMSACB DD statement is an optional DD statement. This DD statement defines the library that contains the DMB for the database. You can use the IMSACB DD statements to identify the active ACB library.
FORMAT	The FORMAT DD is an optional DD statement that defines the IMS MFS control block library. This DD statement is required for testing IMS MFS applications.
IEFRDER	<p>The IEFRDER DD is an optional DD statement that enables the IMS logging facility. For this DD statement, define the primary system log data set. If IEFRDER DD DUMMY is specified, the log data sets are not used.</p> <p>Important: For KW=DBB, if you do not supply a log data set in the JCL, you must specify DBRC=N as an EXEC parameter; otherwise, the region abends with a U0073.</p>
DFSVSAMP	<p>The DFSVSAMP DD is an optional DD statement that defines the buffer pool configuration of an IMS subsystem. Specify this DD statement if VSAM databases are used. This statement is sometimes required for ISAM and OSAM databases. When KW=DBB is specified, the IOBF keyword must be used to define the length and number of buffers for the ISAM and OSAM buffer pool.</p> <p>Because IMS Batch Terminal Simulator might attach to IMS more than once, the DFSVSAMP DD statement must be defined as a DASD data set so that it can be reread.</p> <p>Related reading: For more information about DFSVSAMP DD, see <i>IMS System Definition</i>.</p>
<p>Note: The DFSRESLB, IEFRDER, IMS, and IMSACB data sets are allocated only when KW=DLI or KW=DBB is specified. For more information about these data sets, see <i>IMS System Definition</i>.</p>	

The following example shows how to specify IMS data sets.

```

//BTSJOB      JOB      ACCNT ,NAME ,MSGLEVEL=1
//PROCLIB    JCLLIB  ORDER=(BTS.PROCLIB)
//BTS        EXEC     BTS
//G.STEPLIB  DD DISP=SHR,DSN=IMS.PGMLIB
//           DD DISP=SHR,DSN=BTS.SBTSLMDO
//           DD DISP=SHR,DSN=IMS.SDFSRESL
//G.FORMAT   DD DISP=SHR,DSN=IMS.FORMAT
//G.BTSOUT   DD SYSOUT=B,
//           DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//G.BTSDEBUG DD DUMMY
//G.BTSSNAP  DD SYSOUT=B,SPACE=(TRK,(10,15))
//G.BTSIN    DD *
BTS commands and simulator statements
(optional '/FORMAT' statements)
(optional '/EXIT' statements)
/*
//G.IEFRDER  DD DSN=IMSLOG,DISP=(MOD,KEEP)
//DFSVSAMP  DD DISP=SHR,DSN=IMS.VSAM.PARM(OPTIONS)

```

Step 4: Defining the transactions and the simulation environment

Add the appropriate simulator commands in the BTSIN data set to define the transactions to be performed, the logical terminal to be simulated, the format of the simulator statements and output listing, and the debugging aids that you want to use.

About this task

Before IMS Batch Terminal Simulator can process a transaction code, the IMS teleprocessing system generation information about that transaction must be made available to IMS Batch Terminal Simulator. You supply this information by using the `./T` command.

Note: `./T` command is the only required simulator command of IMS Batch Terminal Simulator. Other simulator commands are optional.

Procedure

1. In the BTSIN data set, specify the terminal and transaction names by using `./T` commands.

The only destination name that is supplied by the BTS cataloged procedure is the logical terminal PCB name for the source of the input message. You must explicitly define each alternate logical terminal name and primary and secondary transaction name with the `./T` command.

The following example shows a format to define a transaction. In this example, the `TC=` operand defines the transaction code, and the `MBR=` operand defines the load module name.

```

./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03

```

The following example shows a format to define an alternate logical terminal name. In this example, the `TC=` operand specifies the logical terminal name.

```

./T TC=BTSTERM MDL=1

```

If the messages that are inserted at the alternate logical terminal are to be presented in an output format by using the Message Format Service control blocks, you must also specify the device and feature code (if `FEAT=7F` is not acceptable) in the `./T` command. Spool data sets are defined as alternate logical terminals.

Related reading: See [“./T command” on page 222](#) for the keywords, operands, and examples for specifying the `./T` commands.

Tips:

- Unlike IMS, IMS Batch Terminal Simulator allows the use of a load module member name other than the PSB name for the transaction. This facility can be used for testing multiple versions of a single

application with IMS Batch Terminal Simulator, but the modules must be correctly named for IMS operation.

- Except for the required operands, if the operands are not specified, IMS Batch Terminal Simulator default values are used. If you do not want to use the default values, specify additional `./T` commands to define the PSB name, SPA size, input message edit module name, programming language, program type, and process limit count (PLC).

Related reading: See [“Simulator command summary” on page 187](#) for the list of default values.

2. If necessary, define the I/O logical terminal name by coding the `./D` command with the `LTERM=` operand.

The default I/O logical terminal is set to `LTERM=IOPCB`. Specify `LTERM=name` to override the default.

Tips: You can also use the `./D` command to specify the following attributes:

- You can specify different values for the end-of-segment (EOS) and end-of-message (EOM) indicators if the default values (`*` and `$` respectively) are not acceptable.
- If you want to use a device-dependent output formatter to format input and output messages, specify additional `./D` command keywords to complete the specification.

3. If necessary, specify other simulator commands.

The `./T` command is the only required simulator command; however, IMS Batch Terminal Simulator provides additional simulator commands. After the `./T` commands, place additional simulator commands that are applicable.

Examples:

- You might want to specify the `./R` command, which indicates to IMS Batch Terminal Simulator a change in the data format of all simulator statements that follow during the run.
- To substitute a given PCB status code on a designated DL/I call, specify the `./S` command with the `STCD=` keyword.
- You can set default keywords and operands for `./E` commands by using the environment specification table (BTSCHTBL). When the table is used, you do not need to specify the `./E` commands in your BTSIN data set.

Related reading:

- See [“Simulator command summary” on page 187](#) for the list of simulator commands.
- See [“Simulator commands” on page 187](#) for the format of the simulator commands and specification examples.
- See [“Setting defaults for environment parameters \(./E commands\)” on page 51](#) for the instructions to set the default values.

4. Organize the order of the simulator commands.

Consider putting all `./T` commands first to ensure that the teleprocessing generation information is made available to IMS Batch Terminal Simulator for the duration of the run.

If a primary transaction generates secondary transactions that are enqueued by IMS Batch Terminal Simulator, they are scheduled and processed by IMS Batch Terminal Simulator immediately upon completion of the primary transaction. When all the secondary transactions have been processed, IMS Batch Terminal Simulator continues to process primary transactions. Therefore, simulator commands that relate to secondary transactions must be included prior to the completion of the primary transaction because IMS Batch Terminal Simulator does not read the next statement from the input stream until all secondary transactions have already been processed.

5. If necessary, customize the JCL or the BTSIN to accommodate the characteristics of your application. The following characteristics might require additional customization:

- Applications that are written in REXX or Java
- Applications that access Db2, IBM MQ, or data entry databases (DEDBs)
- Applications that trigger secondary transactions

- APPC applications or 3270 applications

For the customization procedures, see [Chapter 10, “Customizing JCL for specific applications,”](#) on page 113.

Related reference

Command reference

IMS Batch Terminal Simulator supports simulator commands to define the transactions, simulator statements to supply transaction messages, and IMS commands to facilitate the testing activities.

Step 5: Specifying the simulator statements

You define the transaction messages by coding the simulator statements in the BTSIN data set.

Procedure

IMS Batch Terminal Simulator supports two format types for simulator statements: *unformatted-mode simulator statements* and *formatted-mode simulator statements*. The procedure differs by whether you use 3270 formatting or not.

- If 3270 formatting is not requested (/FORMAT command is not specified), specify the unformatted-mode simulator statements.

The first unformatted-mode simulator statement must begin with a transaction code to which it is related, followed by a transaction message (input for the transaction), and finally an end-of-segment indicator or an end-of-message indicator. This indicator can also be coded as either a single-EBCDIC character or as two hexadecimal digits. In the following example, dollar signs (\$) are used for the end-of-message indicator.

```
PART AN960C10$
DSPALLI AN960C10$
```

Related reading: See [“Unformatted-mode simulator statements”](#) on page 240 for information about specifying the unformatted-mode simulator statements.

- If 3270 formatting is requested (/FORMAT command is specified), specify the formatted-mode simulator statements.

The formatted-mode simulator statements conform to the following format:

```
{[LxCy|CyLx] ['data'] [terminal-action]} ... {terminal-action|eom}
```

where:

- LxCy and CyLx specify the location (line and column) in the screen where data is entered
- *data* specifies the transaction message
- *terminal-action* specifies the terminal actions such as PF keys
- *eom* specifies the end-of-message indicator

For example, you can code formatted-mode simulator statements as follows:

```
L03C30 'AN960C10' PFK2 $
L04C30 '28009126' PFK3 $
```

Tip: Specifying the location of the lines and columns can be complicated in some cases. You can generate the formatted-mode simulator statements automatically by running IMS Batch Terminal Simulator in interactive mode with full screen image support (FSS). In interactive mode with FSS, IMS Batch Terminal Simulator converts all the inputs that were entered through the TSO terminal into the formatted-mode simulator statements, and stores them in the BTSPUNCH data set in the order that they were entered. Consider using auto-generated simulator statements in the BTSPUNCH data set as BTSIN input.

Related reading: See [“Formatted-mode simulator statements” on page 241](#) for information about specifying the formatted-mode simulator statements.

What to do next

If necessary, you can use the following IMS commands while simulating your application:

- IMS /EXIT command to end the IMS Batch Terminal Simulator job at the particular conversation point.
- IMS /FORMAT command to enable the 3270 formatting option. This command is valid only for 3270 applications.
- IMS /SET command to establish the destination of all messages that are entered into this terminal. The destination can be another terminal or a particular transaction code.
- IMS /RESET command to eliminate the preset mode that is established by the /SET command.

Related reading: See [“IMS commands” on page 247](#) for more information about the IMS commands that are supported by IMS Batch Terminal Simulator.

You can change your simulation environment even after transactions are processed. To change the environment, specify the simulator commands (simulator commands can be specified after the simulator statements.) For example, to change the simulation environment in unformatted mode after processing transaction PART, the BTSIN data set might include the following lines:

```
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./E USERID=USER1
PART AN960C10$
./E USERID=USER2
DSPINV AB960C10,8009126A$
```

The following example is for formatted mode:

```
./D LTERM=BTS3270 DD0F=327021
/FOR SAMOUX
PA1 $ PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
PA1 $ PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 3).
L03C30 'AN960C10' PFK2 $ ENTER 'DSPALLI' TRANS CODE BY PF KEY 2
L04C30 '28009126' PFK3 $ USE PREMODIFIED FIELD TO ENTER PART NO
./D LTERM=BTS3270 DD0F=327016
./O ATR=NO
/FORMAT SAMOUX
```

When you have specified all the simulator commands and statements, submit the IMS Batch Terminal Simulator job.

When the job ends, see [Chapter 12, “Interpreting the simulation output,” on page 139](#) to confirm the application behavior.

Related reference

[Simulator statements](#)

Simulator statements define the transaction messages. There are two types of simulator statements: unformatted-mode simulator statements and formatted-mode simulator statements.

Debugging applications in batch mode

In batch mode, you can use IBM Debug Tools for z/OS, simulator commands that are provided by IMS Batch Terminal Simulator, or the VS COBOL II interactive debug facility to debug applications.

About this task

To debug applications in batch mode, see the following topics:

- [“Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger” on page 76](#)
- [“Debugging with simulator commands” on page 76](#)

- [“Debugging with VS COBOL II interactive debug” on page 77](#)

Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger

To use IBM Debug Tool for z/OS or IBM z/OS Debugger (Debug Tool), you must modify the IMS Batch Terminal Simulator JCL.

Before you begin

To enable an IMS application for use with Debug Tool, it must be compiled with the TEST option. For example:

```
//          PARM.COBOL=' MAP , OFFSET , VBREF , XREF , OBJ , TEST ' ,
```

Tip: Several other methods are available to enable your application for use with Debug Tool. For more information, see the *IBM Debug Tool for z/OS User's Guide* or the *IBM z/OS Debugger User's Guide*.

Procedure

1. Ensure that the SEQAMOD data set is included in the STEPLIB concatenation.
2. Specify the following DD statements for Debug Tool:

CEEOPTS

Defines additional invocation-level runtime options. This DD statement is required to debug applications through Debug Tool (TEST option).

INSPIN

Defines the input data set that contains the Debug Tool commands.

INSPLOG

Defines the output data set for recording the progress of the debugging session.

3. Specify the TEST runtime option in CEEOPTS DD. For example, TEST (, INSPIN , ,)
4. Specify the Debug Tool commands in INSPIN DD, or specify the Debug Tool commands in a data set and allocate the data set to INSPIN DD.

Debugging with simulator commands

By using the simulator commands that are provided by IMS Batch Terminal Simulator, you can run path tests as batch jobs without modifying your applications or compiling your applications with specific options.

Procedure

You can use the following simulator commands to debug your applications:

./P command

You can use the ./P command to apply a patch at a particular point in application program processing. You can apply a patch to the application program by specifying the module offset and the values in character or hexadecimal.

For example, you can copy (zap) the constant DC C'January' that is defined at offset X'0DAC' in your single CSECT application program (MYPGM) and test the following conditions:

- Replace to DC C'February' in the first transaction
- Use the original value in the second transaction
- Replace to DC C'March' in the third transaction

To achieve these results, your input stream might contain the following lines:

```

./T TC=ZAPTEST MBR=MYPGM
./P MBR=MYPGM PA=000DAC PC=February
ZAPTEST SECOND
./P MBR=MYPGM PA=000DAC PC=March
ZAPTEST THIRD

```

With this specification, you can run three path tests in one IMS Batch Terminal Simulator.

Related reading: See [“./P command”](#) on page 212 for more information about the ./P command and its keywords.

./S command

You can use the ./S command to set specific action to be taken at a particular point during the processing of the application program. The action can be either to issue a SNAP macro or to substitute a given PCB status code on a designated DL/I call. You can set only one specific action per call.

The following example shows how to use the ./S command to take a snapshot dump of a 62-byte work area at offset '1F20' in the PARTRAN application program when a GNP call is issued in the PART database.

```

./T TC=PARTRAN MBR=PARTRAN
./S MBR=PARTRAN SA=001F20 BYTES=62 PCB=PART X
FUNC=GNP TIMES=5

```

The following example shows how to use the ./S command to generate a status code of A5 on the fifth ISRT to the I/O PCB while the XYZ transaction is being processed.

```

./T TC=XYZ MBR=XYZPGM...
./D LTERM=IOPCB...
./S MBR=XYZPGM PCB=IOPCB X
FUNC=ISRT TIMES=4 STCD=00
./S MBR=XYZPGM PCB=IOPCB X
FUNC=ISRT TIMES=1 STCD=A5

```

Related reading: See [“./S command”](#) on page 218 for more information about the ./S command and its keywords.

MSGABENDxxx command (debug command)

You can use this command to cause a user 4091 abend after IMS Batch Terminal Simulator issues a specific message.

For example, when Db2 changed-data-capture exit routine (DB2CDCEX) is called, IMS Batch Terminal Simulator issues BTS0107W DB2 CHANGED DATA CAPTURE EXIT STARTED. If you specify MSGABEND107 in BTSIN, IMS Batch Terminal Simulator causes a user 4091 abend when Db2 changed-data-capture exit routine (DB2CDCEX) is called.

Related reading: See [“Debug commands”](#) on page 235 for more information about the MSGABEND command.

Debugging with VS COBOL II interactive debug

To use VS COBOL II interactive debug (COBOL interactive debug) under a batch job with IMS Batch Terminal Simulator, you must modify the JCL.

Procedure

To run COBOL interactive debug under a batch job, complete these steps:

1. Define the BTSCMD library (SBTSCMD0) on the STEPLIB DD statement.
2. Specify a SYSTSIN DD statement to call the BTS command processor BTSCP.
3. In the SYSTSIN DD, specify all COBTEST commands (instead of SYSDBIN.)
4. Specify a SYSTSPRT DD statement for COBTEST output.

Example

The following example calls COBTEST under a batch job:

```

//COBTESTB  JOB ACCNT,NAME,MSGLEVEL=1
//COBGO     EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=BTS.SBTSCMD0
//TASKLIB  DD DISP=SHR,DSN=BTS.SBTSLMD0
//          DD DISP=SHR,DSN=BTS.PGMLIB
//          DD DISP=SHR,DSN=IMSVS.SDFSRESL
//          DD DISP=SHR,DSN=COB2.COB2LIB
//BTSOUT    DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//QIOPCB   DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(LRECL=4096,BLKSIZE=4096)
//QALTPCB  DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(LRECL=4096,BLKSIZE=4096)
//DFSRESLB DD DISP=SHR,DSN=IMSVS.SDFSRESL
//IEFRDER  DD DUMMY
//IMS      DD DISP=SHR,DSN=IMSVS.PSBLIB
//          DD DISP=SHR,DSN=IMSVS.DBDLIB
//          *
//SYSDBOUT DD SYSOUT=A,OUTLIM=100
//SYSOUT   DD SYSOUT=A,OUTLIM=100
//SYSUDUMP DD DUMMY
//          *
//SYSTSIN DD *
BTSCP 'DLI,12,0100,,0,,N,0,T,SY1,,N,N,,N'
QUALIFY PGMCOBOL
LIST(SSA-MAE) PRINT
GO
QUIT
//          *
//BTSIN    DD *
./T TC=TCCCOBOL LANG=CBL MBR=PGMCOBOL TYPE=MSG
./C COBTEST PGMCOBOL
./O DB=YES MSG=YES TSODB=ALL TSOMSG=ALL
TCCCOBOL $
//          *
//DFSVSAMP DD *
4096,4
8192,4
//          *
//SYSTSPRT DD SYSOUT=A,OUTLIM=100
//

```

Figure 8. JCL example for calling COBTEST

Chapter 7. Testing applications in interactive mode

You can use IMS Batch Terminal Simulator to test your applications in batch mode or in interactive mode. In interactive mode, you can enter the input for IMS Batch Terminal Simulator interactively and check the results of each input. Interactive mode is especially helpful in unit testing.

With the full screen image support (FSS), you can take full advantage of a 3270 terminal that is interacting with TSO. The screen image is formatted in the same way as the IMS terminal would be formatted, and data is entered into the screen image in the same way that data would be entered with IMS.

Topics:

- [“Tutorial: Running the sample application in interactive mode” on page 79](#)
- [“Running applications in interactive mode” on page 86](#)
- [“Debugging applications in interactive mode” on page 98](#)
- [“Tips for testing in interactive mode with FSS” on page 102](#)

Tutorial: Running the sample application in interactive mode

Use this tutorial to learn how to run applications in interactive mode.

For application developers who want to test their applications interactively, IMS Batch Terminal Simulator provides a method for testing applications in interactive mode.

Learning objectives

This tutorial uses BTSSAMP3 JCL that is supplied with the product and is copied to the BTS JCL library (SBTSJCL0) during the installation. This tutorial runs the IMS sample application in the following ways:

- In interactive mode
- In an IMS batch processing region (DLI)
- With full screen image support (FSS)

By completing the tutorial, you will learn how to complete the following tasks:

- Test IMS sample application in interactive mode
- Code JCL statements for running IMS Batch Terminal Simulator in interactive mode
- Understand the input and the output data sets that are used in interactive mode

Time required

This tutorial takes approximately 60 minutes to complete.

Audience

This tutorial is targeted for application developers with some z/OS and IMS knowledge.

Prerequisites

Make sure that your environment is properly set up for this tutorial by checking with your system administrator.

- Verify with your system administrator that the IMS sample application has been installed. This tutorial requires that the IMS sample application that uses the DI21PART database has been installed.
- Obtain the data set name template from your system administrator. The data set name template contains the information about the required system resources. For more information about the data set name template, see [“Recording the resource information” on page 52](#).

Step 1: Modifying the sample JCL

Understand how IMS Batch Terminal Simulator JCL is structured and how to modify the JCL.

Procedure

1. Copy BTSSAMP3 JCL to your JCL library and edit the copy of BTSSAMP3 JCL.
2. Follow the instructions in header description of BTSSAMP3 JCL and make the following changes:
 - a) Change the JOB card to meet your system requirements.
 - b) Specify the IMS libraries.

Locate *\$imshlq*. Change the value to the data set prefix name that is described in the data set name template.
 - c) Specify the IMS Message Format Services (MFS) libraries.

Locate *\$referral*. Change the value to the data set prefix name of your MFS REFERAL library.

Locate *\$tformat*. Change the value to the data set prefix name of your MFS FORMAT library.
 - d) Specify the IMS Batch Terminal Simulator libraries.

Locate *\$btshlq*. Change the value to the data set prefix name that is described in the data set name template.
 - e) Specify the BTSIN data set.

Locate *\$btsin*. Change the value to the data set name of BTSIN that will be created in this job.
 - f) Specify the command procedure library.

Locate *\$cmdlib*. Change the value to the data set name for CMDLIB that will be created in this job.
 - g) Specify the volume serial for BTSIN and CMDLIB.

Locate *\$volser*. Change the value to the volume serial for the BTSIN data set and the CMDLIB data set that will be created in this job.
 - h) Specify the sysout class for output.

Locate *\$sout*. Change the value to the sysout class for output.
3. Locate the BTS SAMPLE JCL (BTSSAMP3) section.

This section contains the JCL for running IMS Batch Terminal Simulator. The following figure shows the BTS SAMPLE JCL (BTSSAMP3) section.

```
//*-----*
//* BTS SAMPLE JCL (BTSSAMP3)                      *
//*-----*
...
...
//BTSSAMP3 EXEC BTSTSO, 1
//          SOUT=$sout
//BTSIN.SYSUT1 DD DATA,DLM=$$
./D DDOF=327029 2
./T TC=PART      MBR=DFSSAM02 LANG=CBL
./T TC=DSPINV   MBR=DFSSAM03 LANG=CBL
./T TC=ADDPART  MBR=DFSSAM04 LANG=CBL
./T TC=ADDINV   MBR=DFSSAM04 LANG=CBL
./T TC=DLETINV  MBR=DFSSAM04 LANG=CBL
./T TC=DLETPART MBR=DFSSAM04 LANG=CBL
./T TC=CLOSE    MBR=DFSSAM05 LANG=CBL
./T TC=DISBURSE MBR=DFSSAM06 LANG=CBL
./T TC=DSPALLI  MBR=DFSSAM07 LANG=CBL
$$
//*
//BTSCLIST.SYSIN DD DATA,DLM=$$
./          ADD LIST=ALL,NAME=BTSSAMP3
```

Figure 9. BTSSAMP3 JCL

- a) The first line (1) shows the EXEC statement for BTS TSO command list (BTS CLIST).

b) The BTSIN DD statement contains the input for IMS Batch Terminal Simulator.

The lines that start with `./D` and `./T` (2) specify the IMS teleprocessing system generation information about the transactions. On the `./T` commands, the TC operands specify the transaction codes, and the MBR operands specify the module names. These specifications are called the *simulator commands*. IMS Batch Terminal Simulator provides various simulator commands to support testing of various types of applications.

4. Submit the BTSSAMP3 job and ensure that the return code is 0 except for the MFSGEN MFSBLD1 step. Return code 4 from the MFSGEN MFSBLD1 step is an expected return. You can ignore this return code.

Lesson checkpoint

You have prepared a JCL to run IMS Batch Terminal Simulator in interactive mode.

You learned about the input and the output data sets and how to code the IMS Batch Terminal Simulator JCL statements.

Step 2: Running the session

Simulate the sample application interactively and understand how an IMS Batch Terminal Simulator TSO session operates.

About this task

This tutorial runs the following transactions of the IMS sample application:

DSPALLI, DSPINV

To learn more about the IMS sample application, see *IMS Installation*.

Procedure

1. Run the TSO CLIST procedure by entering the following command.

```
EXEC 'cmdlib(BTSSAMP3)'
```

where *cmdlib* is the name of the CLIST library.

2. When you are prompted to enter ALLOC, ALLOCDS, FREE, DELETE, or ENTER, enter ALLOCDS and press Enter to start IMS Batch Terminal Simulator. The CLIST procedure allocates the required data sets.

The meaning of the options are as follows:

ALLOC

Dynamically allocates the existing data sets that are required to run the IMS Batch Terminal Simulator job by issuing the TSO ALLOCATE command, and runs the IMS Batch Terminal Simulator job.

ALLOCDS

Dynamically allocates the new data sets that are required to run the IMS Batch Terminal Simulator job by issuing the TSO ALLOCATE command, and runs the IMS Batch Terminal Simulator job.

FREE

Deallocates the data sets that were previously allocated for IMS Batch Terminal Simulator jobs by issuing the TSO FREE command.

DELETE

Deletes the data sets that were previously allocated for IMS Batch Terminal Simulator jobs by issuing the TSO DELETE command.

ENTER

Runs the IMS Batch Terminal Simulator job.

Tip: If this is not the first run of this tutorial, enter DELETE and press Enter. IMS Batch Terminal Simulator deletes the data sets that were used in the previous run. Then, issue the EXEC 'cmdlib(BTSSAMP3)' command again, and enter ALLOCDS and press Enter.

When IMS Batch Terminal Simulator starts, a BTS0007I message and a list of simulator commands are displayed.

```
BTS0007I BTS V4R1 SIMULATION STARTED. TIME=20:24:40, DATE=2021.087, IMS=14.1.
./D DD0F=327029
./T TC=PART MBR=DFSSAM02 LANG=CBL
./T TC=DSPINV MBR=DFSSAM03 LANG=CBL
./T TC=ADDPART MBR=DFSSAM04 LANG=CBL
./T TC=ADDINV MBR=DFSSAM04 LANG=CBL
./T TC=DLETINV MBR=DFSSAM04 LANG=CBL
./T TC=DLETPART MBR=DFSSAM04 LANG=CBL
./T TC=CLOSE MBR=DFSSAM05 LANG=CBL
./T TC=DISBURSE MBR=DFSSAM06 LANG=CBL
./T TC=DSPALLI MBR=DFSSAM07 LANG=CBL
ENTER BTS COMMAND OR /FORMAT OR /*
```

These simulator commands (./D and ./T) are read from the BTSIN data set that was created by the CLIST procedure. At the end of the list, when IMS Batch Terminal Simulator completes reading all the simulator commands, IMS Batch Terminal Simulator prompts for input.

3. For this tutorial, we will use full screen image support (FSS) to display the formatted screen image. To start FSS, type the following command and press Enter.

```
/FORMAT SAMOUX
```

The /FORMAT command is the IMS command that starts FSS, and SAMOUX specifies the name of the sample MFS control block that is provided by IMS Batch Terminal Simulator.

The first screen image is displayed.

```
                BTS 3270 SAMPLE PROGRAM

THE SAMPLE PROGRAM CONTAINS MESSAGE-PROCESSING AND ASSOCIATED
TRANSACTIONS TO EXECUTE IN AN IMS ONLINE CONTROL REGION TO:

    1. INQUIRE ABOUT A PART AND ITS DESCRIPTION (PART)           ...PFK1
    2. INQUIRE ABOUT A PART BY SPECIFIC LOCATION (DSPALLI)       ...PFK2
       OR A PART'S TOTAL INVENTORY IN ALL LOCATIONS (DPSINV)     ...PFK3
    3. ADD A NEW PART AND ITS DESCRIPTION (ADDPART)               ...PFK4
    4. ADD PART INVENTORY INFORMATION BY LOCATION TO AN EXISTING
       PART DESCRIPTION (ADDINV)                                   ...PFK5
    5. DELETE PART INVENTORY INFORMATION BY LOCATION (DLETNV)    ...PFK6
    6. DELETE A PART AFTER DELETION OF ALL ITS SUBORDINATE PART
       INVENTORY INFORMATION (DLETPART)                           ...PFK7

                                PRESS PA1 TO CONTINUE
```

4. Press PA1 (as indicated at the right bottom of screen) to display the next screen.

7. CLOSE A PART ORDER TO INCREASE THE PART INVENTORY AT A SPECIFIC LOCATION (CLOSE) ...PFK8
8. DISBURSE A SPECIFIC QUANTITY OF A PARTICULAR PART ON A PLANNED OR UNPLANNED BASIS AT A PARTICULAR PART ON A LOCATION THEREBY REDUCING INVENTORY (DISBURSE) ...PFK9

CHOOSE THE PROGRAM DESIRED AND DEPRESS THE APPROPRIATE PFK1 THROUGH 9

PART	PFK1	DLETINV...	PFK6
DSPALLI...	PFK2	DLETPART..	PFK7
ADDPART....	PFK4	CLOSE.....	PFK8
ADDINV.....	PFK5	DISBURSE..	PFK9
DSPINV.....	PFK3		

5. Press PA1 to display the next screen.

```

B T S 3270 F O R M A T T E R S A M P L E P R O G R A M .
* * * * * I N P U T S E C T I O N * * * * *
PART NUMBER : TRANSACTION :
INVENTORY NO: DESCRIPTION:
PROC CODE/DISBURSE(U/P)/QUANT:

* * * * * O U T P U T S E C T I O N * * * * *

```

6. Run the DSPALLI transaction.

Enter AN960C10 for PART NUMBER field and press PF2. The PF2 key is assigned for transaction DSPALLI as indicated in the previous screens.

```

B T S 3270 F O R M A T T E R S A M P L E P R O G R A M .
* * * * * I N P U T S E C T I O N * * * * *
PART NUMBER : AN960C10 TRANSACTION :
INVENTORY NO: DESCRIPTION:
PROC CODE/DISBURSE(U/P)/QUANT:

* * * * * O U T P U T S E C T I O N * * * * *

```

IMS Batch Terminal Simulator starts the simulation.

```

BTS0100I ATTACHING DFSRRC00, PARM=DLI,BTSPC000,DFSSAM07,12,0100,,0,,N,0,T,,N,
N,,N,,,,,,,,,,,,,
BTS0006I TRANSACTION STARTED: DSPALLI .
MBR=DFSSAM07 PSB=DFSSAM07 EDIT=          SPA=0      PLC=1      LANG=CBL TYPE=MSG
**** MSG CALL- FUNC=GU , PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000023, PCBN=001 -MSG-      GU
***** DB CALL- FUNC=GU , PCB=DI21PART, STATUS= , LEVEL=01, SEGMENT=PARTROOT
      IOLENGTH=000050, PCBN=002 -DB-      GU
***** DB CALL- FUNC=GN , PCB=DI21PART, STATUS= , LEVEL=02, SEGMENT=STANINFO
      IOLENGTH=000085, PCBN=002 -DB-      GN
***** DB CALL- FUNC=GN , PCB=DI21PART, STATUS=GK, LEVEL=02, SEGMENT=STOKSTAT
      IOLENGTH=000160, PCBN=002 -DB-      GN GK
BTS0031I MODNAME: SAMOUX
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000063, PCBN=001 -MSG-      ISRT
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000086, PCBN=001 -MSG-      ISRT
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000086, PCBN=001 -MSG-      ISRT
***** DB CALL- FUNC=GN , PCB=DI21PART, STATUS= , LEVEL=02, SEGMENT=STOKSTAT
      IOLENGTH=000160, PCBN=002 -DB-      GN
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000084, PCBN=001 -MSG-      ISRT
***** DB CALL- FUNC=GN , PCB=DI21PART, STATUS= , LEVEL=02, SEGMENT=STOKSTAT
      IOLENGTH=000160, PCBN=002 -DB-      GN
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000084, PCBN=001 -MSG-      ISRT
***** DB CALL- FUNC=GN , PCB=DI21PART, STATUS=GA, LEVEL=01, SEGMENT=PARTROOT
      IOLENGTH=000050, PCBN=002 -DB-      GN GA
**** MSG CALL- FUNC=ISRT, PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001
      LENGTH=000084, PCBN=001 -MSG-      ISRT
***
BTS0020I STATISTICS REPORT FOR TRANSACTION: DSPALLI .
PCBNAME  GU  GN  GNP  GHU  GHN  GHP  ISRT  PURG  REPL  DLET  DEQ  CHPK  LOG  STAT
XRST  CHNG  ROLB  OPEN  CLSE  OTHR
IOPCB          1                               6
DI21PART      1   5
BTS0101A ENTER NULL LINE TO OBTAIN IMS-SCREEN FOR PCB(IOPCB )

```

A BTS0006I message is issued to indicate that a transaction has started. Trace output of the actions that were taken by the application program follows the message. At the end of the trace output, IMS Batch Terminal Simulator provides a statistics report for the transaction (message BTS0020I).

When the transaction ends, a BTS0101A message prompts you to press Enter.

7. Press Enter.

The screen shows the reply from the application program.

```

      B T S   3270   F O R M A T T E R   S A M P L E   P R O G R A M .
      * * * * * INPUT SECTION * * * * *
      PART NUMBER : AN960C10      TRANSACTION :
      INVENTORY NO:      DESCRIPTION: WASHER
      PROC CODE/DISBURSE(U/P)/QUANT:      74

      * * * * * OUTPUT SECTION * * * * *

      AREA  INV  PROJ  DIV   UNIT  CURRENT  ON      IN      TOTAL  COUNT  BA
      DEPT  CD           PRICE  REQMTS  ORDER  STOCK  DISBURSE  TAKEN  ORD
      1.    AA  165  11           146    20    126    104    N    0

```

8. Run the DISPINV transaction.

Enter 28009126 in the INVENTORY NO field and press PF3. The PF3 key is assigned for transaction DISPINV.

```

B T S 3270 F O R M A T T E R S A M P L E P R O G R A M .
* * * * * INPUT SECTION * * * * *
PART NUMBER : AN960C10 TRANSACTION :
INVENTORY NO: 28009126 DESCRIPTION: WASHER
PROC CODE/DISBURSE(U/P)/QUANT: 74

```

```

* * * * * OUTPUT SECTION * * * * *
AREA INV PROJ DIV UNIT CURRENT ON IN TOTAL COUNT BA
DEPT CD PRICE REQMS ORDER STOCK DISBURSE TAKEN ORD
1. AA 165 11 146 20 126 104 N 0

```

9. Press Enter until the following screen is displayed.

This screen shows the reply from the application program.

```

B T S 3270 F O R M A T T E R S A M P L E P R O G R A M .
* * * * * INPUT SECTION * * * * *
PART NUMBER : AN960C10 TRANSACTION :
INVENTORY NO: 28009126 DESCRIPTION: WASHER
PROC CODE/DISBURSE(U/P)/QUANT: 74

```

```

* * * * * OUTPUT SECTION * * * * *
AREA=2; INV DEPT=80; PRJ=091; DIV=26; PRICE= .000; STK CT DATE=513; UNIT=
CURR REQMS= 630 ; ON ORDER= 15 ; TOTAL STOCK= 680
DISB PLANNED= 1053 ; DISB UNPLANNED= 104 ; STK CT VARIANCE= 0

```

10. To end the session, press the CLEAR key, and then press Enter.

The following screen is displayed.

```

BTS0004W NO TRANSACTION INFORMATION SUPPLIED. UNABLE TO SCHEDULE TRANSACTION:
BTS0011I CONTINUING WITH NEXT TRANSACTION.
ENTER BTS COMMAND OR /FORMAT OR /*

```

11. To end the session, enter /* and press Enter.

The following screen is displayed and the session ends.

```

BTS0008I END OF INPUT DATA SET ENCOUNTERED.
DFS627I IMS RTM CLEANUP ( EOT ) COMPLETE FOR ST TEAM93 .DBAUSER . ,RC=0
0
BTS0005I END OF BTS RUN.
***

```

12. Check the output in the BTSOUT data set.

The BTSOUT data set contains the output from the IMS Batch Terminal Simulator run. The output consists of BTSIN echo back, DL/I call information, and trace information of the internal processes.

13. Check the output in the BTSPUNCH data set.

Check the BTSPUNCH data set. The input data that was entered through the screen is converted to a series of formatted-mode simulator statements and are stored in the BTSPUNCH data set. The contents of BTSPUNCH data set can be reused as input for IMS Batch Terminal Simulator.

Tip: If you specify this BTSPUNCH data set for the BTSIN DD in batch mode, IMS Batch Terminal Simulator uses the contents of the BTSPUNCH data set as input and runs the same job in batch mode.

Based on the data that was entered in this tutorial, the following simulator statements are written to the BTSPUNCH data set.

```
./D DDOF=327029
./T TC=PART      MBR=DFSSAM02 LANG=CBL
./T TC=DSPINV   MBR=DFSSAM03 LANG=CBL
./T TC=ADDPART  MBR=DFSSAM04 LANG=CBL
./T TC=ADDINV   MBR=DFSSAM04 LANG=CBL
./T TC=DLETINV  MBR=DFSSAM04 LANG=CBL
./T TC=DLETPART MBR=DFSSAM04 LANG=CBL
./T TC=CLOSE    MBR=DFSSAM05 LANG=CBL
./T TC=DISBURSE MBR=DFSSAM06 LANG=CBL
./T TC=DSPALLI  MBR=DFSSAM07 LANG=CBL
/FORMAT SAMOXX
PA1 $
PA1 $
L3C28 'AN960C10' L3C36 CURSOR PFK2 $
L4C28 '28009126' L4C36 CURSOR PFK3 $
L15C48 CURSOR ENTER $
```

Figure 10. Contents of the BTSPUNCH data set

Related reading: See [“Formatted-mode simulator statements” on page 241](#) for the format of formatted-mode simulator statements.

Lesson checkpoint

You have run an IMS sample application in interactive mode in the IMS Batch Terminal Simulator environment.

You learned how to run an IMS application in interactive mode and the IMS Batch Terminal Simulator outputs in interactive mode.

Running applications in interactive mode

You start IMS Batch Terminal Simulator in interactive mode from a TSO command list (CLIST). You can use IBM Rational Developer for System z or IBM Developer for z Systems to perform these steps.

Before you begin

If you want to implement exit routines, follow the instructions in the following topics:

- [Chapter 11, “Including BTS optional user-written exit routines,” on page 125](#)
- [“Implementing IMS user-written routines” on page 49](#)

About this task

Complete these steps to run applications in interactive mode:

- [“Step 1: Specifying the TSO CLIST parameters” on page 87](#)
- [“Step 2: Specifying the IMS Batch Terminal Simulator data sets” on page 88](#)
- [“Step 3: Specifying the IMS data sets” on page 90](#)
- [“Step 4: Defining the transactions and the simulation environment” on page 91](#)
- [“Step 5: Starting the session” on page 93](#)
- [“Step 6: Entering the simulation input” on page 94](#)

Step 1: Specifying the TSO CLIST parameters

You start IMS Batch Terminal Simulator in interactive mode by using a TSO command list (CLIST). You must define the execution parameters for IMS Batch Terminal Simulator in the TSO CLIST procedure.

About this task

You can create a TSO CLIST to allocate data sets and run IMS Batch Terminal Simulator in the TSO foreground. TSO CLISTs are TSO capability and are not part of IMS Batch Terminal Simulator. The content of this CLIST depends on your installation's standards.

IMS Batch Terminal Simulator provides a sample CLIST, BTSCLIST. You can copy and use the BTSCLIST, or you can use the CLIST provided by your system administrator to create your TSO CLIST procedure.

Procedure

1. You can run your application in a DLI region, DBB region, BMP region, or JBP region. Determine the region type by evaluating the characteristics of each region in [“Determining the types of IMS region” on page 22](#).

To run your application in BMP or JBP region, you must start the IMS control region. You do not need to start the IMS control region if you are running in a DLI region or DBB region.

2. Define the type of IMS region by specifying the KW(*type*) parameter.

The BTS CLIST has a number of optional parameters. One use of the optional parameters is to specify the EXEC PARMs that are passed to the IMS region controller. The CLIST generates the EXEC PARMs based on the KW parameter specification.

KW(*type*) indicates the type of IMS region to be used. KW(DLI) and KW(DBB) are used for batch region execution, and KW(BMP) and KW(JBP) are used for online dependent region execution. The following table describes each KW(*type*) parameter.

Table 19. KW(*type*) parameters

Type	Description
KW(DLI)	Requests a DLI region. The parameter string that is generated for this option is the same as that for IMS Procedure DLIBATCH except that the second and third parameters, &MBR and &PSB, are not included.
KW(DBB)	Requests a DBB region. The parameter string that is generated for this option is the same as that for IMS procedure DBBBATCH except that the second and third parameters, &MBR and &PSB, are not included. Important: For KW(DBB), if you do not supply a log data set in the CLIST, you must specify DBRC=N as an EXEC parameter; otherwise, the region abends with a U0073. IMS always assumes update intent for BTS DBBBATCH execution. For an explanation of the abend, see <i>IMS Messages and Codes</i> .
KW(BMP)	Requests a BMP region. The parameter string that is generated for this option is the same as that for IMS procedure IMSBATCH except that the second and third parameters, &MBR and &PSB, are not included.
KW(JBP)	Requests a JBP region. The parameter string that is generated for this option is the same as that for IMS procedure IMSBATCH except that the second and third parameters, &MBR and &PSB, are not included.

3. Specify other CLIST parameters.

Table 20. CLIST parameters

Mode	Description
CALLTYPE(<i>mode</i>)	<p>Specifies the mode of execution for IMS Batch Terminal Simulator. The default mode is CALLTYPE(CALL).</p> <ul style="list-style-type: none">• CALLTYPE(CALL): Indicates that a TSO CALL is generated.• CALLTYPE(TEST): Indicates that IMS Batch Terminal Simulator and the application program are run under control of the TSO TEST monitor.• CALLTYPE(BTSCP): Indicates that IMS Batch Terminal Simulator is run under control of TSO command processor BTSCP. Under BTSCP, COBOL application programs can be run under control of COBOL interactive debug (COBTEST). <p>Restriction: IMS Batch Terminal Simulator does not support testing of dynamically linked COBOL subroutines. The ./C information is ignored or not processed.</p> <p>Related reading: For more information about using COBOL interactive debug, see the following topics:</p> <ul style="list-style-type: none">– “Debugging with VS COBOL II interactive debug” on page 100– “./C command” on page 192
ALLOCD	Specifies the creation of the IMS Batch Terminal Simulator data sets and the allocation of all data sets that are necessary for the IMS Batch Terminal Simulator run. After the data sets are created and allocated, IMS Batch Terminal Simulator is started.
ALLOC	Specifies that the IMS Batch Terminal Simulator data sets already exist and that only the allocation is to be performed. After the data sets are allocated, IMS Batch Terminal Simulator is started.
DELETE	Specifies that the IMS Batch Terminal Simulator data sets are to be deleted. IMS Batch Terminal Simulator is not started.
FREE	Specifies that the IMS Batch Terminal Simulator data sets are to be deallocated. IMS Batch Terminal Simulator is not started.

For the definitions of the execution parameters that can be specified, see *IMS System Definition*.

Step 2: Specifying the IMS Batch Terminal Simulator data sets

Define the input and output data sets for IMS Batch Terminal Simulator in the TSO CLIST.

Procedure

Specify the IMS Batch Terminal Simulator data sets.

The following IMS Batch Terminal Simulator data sets are supported.

Table 21. IMS Batch Terminal Simulator data sets for interactive mode

DD name	Description
BTSIN	<p>The BTSIN data set is a required input data set. This data set can contain simulator commands and simulator statements. You can enter simulator commands and statements directly to the TSO terminal; however, you can also define a complete test run in the BTSIN data set.</p> <p>Important:</p> <ul style="list-style-type: none">• The BTSIN data set must not be a concatenated SMS-managed data set. If BTSIN is a concatenated SMS-managed data set, unpredictable results might occur.• The BTSIN data set must be in fixed, fixed-block, variable, or variable-block format.• When the BTSIN data set is in fixed or fixed-block format, BTSIN LRECL that is longer than 80 bytes is truncated to 80 bytes.• When the BTSIN data set is in variable or variable-block format, BTSIN LRECL can be in the range of 5 - 32756 without any truncation.
BTSOUT	<p>The BTSOUT data set is a required output data set. After the IMS Batch Terminal Simulator job is run, this data set contains the output listing from the job. To obtain a hard copy of a TSO session, BTSOUT must be allocated to a sequential data set and then routed to a printer.</p> <p>If you are allocating BTSOUT for IBM 3278 Model 5 display, the LRECL value that you specify can affect the output that IMS Batch Terminal Simulator produces. See “LRECL value for BTSOUT data set for 3270 formatting” on page 62 for more information.</p>
BTSACB	<p>The BTSACB data set is an optional data set that specifies the ACB library.</p> <p>Consider specifying this data set when simulating the application programs that access data entry databases (DEDBs) in BMP or JBP region with IMS 13 or earlier.</p> <p>See “Applications that access DEDBs” on page 118 for more information about specifying BTSACB DD.</p>
BTSISRIO	<p>The BTSISRIO DD is an optional DD statement that defines the output data set for the data that the DLI ISRT (Insert) call inserts to IOPCB.</p> <p>The BTSISRIO data set must be in variable or variable-block format. The LRECL of BTSISRIO must be 32756 to avoid truncation.</p>
BTSSNAP	<p>The BTSSNAP data set is an optional output data set that contains snapshot dumps. This data set is used only when you request a snapshot dump by issuing the <code>./S</code> command.</p>
BTSPUNCH	<p>The BTSPUNCH data set is an optional output data set that contains regression test input data.</p> <p>IMS Batch Terminal Simulator attempts to open a sequential output data set that is named BTSPUNCH. In this data set, IMS Batch Terminal Simulator records the input data that was entered though TSO as formatted-mode simulator statements. It contains everything that IMS Batch Terminal Simulator receives as input. The data is truncated to 80 bytes when the input data is greater than 80 bytes.</p>

Table 21. IMS Batch Terminal Simulator data sets for interactive mode (continued)

DD name	Description
BTSDEBUG	The BTSDEBUG data set is an optional output data set that contains snapshot dumps of the trace table and various control blocks. These snapshots are taken at critical points during an IMS Batch Terminal Simulator run. Allocation of the BTSDEBUG data set starts the BTS DEBUG function.

Step 3: Specifying the IMS data sets

Define the input and output data sets for IMS in the TSO CLIST.

Procedure

Specify the IMS and z/OS data sets.

The following table summarizes the IMS and the z/OS data sets that are required when you want to customize the values in the TSO CLIST procedure.

Table 22. IMS and z/OS data sets for interactive mode

DD name	Description
TASKLIB	The TASKLIB DD statement is a required DD statement that defines the following input data sets: IMS library The library that contains the IMS load modules. Program library (PGMLIB) The program library that contains the user-written application programs. Db2 library (optional) The library that contains the Db2 load modules. IBM MQ library (optional) The library that contains the IBM MQ load modules. Language Environment library (optional) The library that contains the Language Environment modules. Debug Tool library (optional) The library that contains the IBM Debug Tool modules. Other libraries Other libraries such as language libraries.
IMS	The IMS data set is an optional data set. This data set defines the library that contains the DBD and the PSB that describe the database to be processed.
IMSACB	The IMSACB data set is an optional data set. This data set defines the library that contains the DMB for the database. You can use the IMSACB data sets to identify the active ACB library.
FORMAT	The FORMAT data set is an optional data set that defines the IMS MFS control block library. This data set is required for testing IMS MFS applications.
IEFRDER	The IEFRDER data set is an optional data set that enables the IMS logging facility. For IEFRDER data set, define the primary system log data set. Important: For KW(DBB), if you do not supply a log data set in the JCL, you must specify DBRC=N as an EXEC parameter; otherwise, the region abends with a U0073.

Table 22. IMS and z/OS data sets for interactive mode (continued)

DD name	Description
DFSVSAMP	<p>The DFSVSAMP data set is an optional data set that defines the buffer pool configuration of an IMS subsystem. Specify this data set if VSAM databases are used. The DFSVSAMP data set is sometimes required for ISAM and OSAM databases. When KW(DBB) is specified, the IOBF keyword must be used to define the length and number of buffers for the ISAM and OSAM buffer pool.</p> <p>Because IMS Batch Terminal Simulator might attach to IMS more than once, the DFSVSAMP data set must be defined as a DASD data set so that it can be reread.</p> <p>Related reading: For more information about DFSVSAMP data set, see <i>IMS System Definition</i>.</p>

Note: The DFSRESLB, IEFRDER, IMS, and IMSACB data sets are allocated only when KW(DLI) or KW(DBB) is specified. For more information about these data sets, see *IMS System Definition*.

Step 4: Defining the transactions and the simulation environment

Add the appropriate simulator commands in the BTSIN data set to define the transactions to be performed, the logical terminal to be simulated, the format of the simulator statements and output listing, and what debugging aids that you want to use.

About this task

Before IMS Batch Terminal Simulator can process a transaction code, the IMS teleprocessing system generation information about that transaction must be available to IMS Batch Terminal Simulator. You supply this information by using the `./T` command.

Note: `./T` command is the only required simulator command of IMS Batch Terminal Simulator. Other simulator commands are optional.

In interactive mode, you can specify the simulator commands in the BTSIN data set or enter them directly through the TSO screen while the TSO session is active.

Procedure

1. In the BTSIN data set, specify the terminal and transaction names by using `./T` commands.

The only destination name that is supplied by the BTS procedure is the logical terminal PCB name for the source of the input message. You must explicitly define each alternate logical terminal name and primary and secondary transaction name with the `./T` command.

The following example shows a format to define a transaction. In this example, the `TC=` operand defines the transaction code, and the `MBR=` operand defines the load module name.

```
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
```

The following example shows a format to define an alternate logical terminal name. In this example, the `TC=` operand specifies the logical terminal name.

```
./T TC=BTSTERM MDL=1
```

If the messages that are inserted at the alternate logical terminal are to be presented in an output format by using the Message Format Service control blocks, you must also specify the device and feature code (if `FEAT=7F` is not acceptable) in the `./T` command. Spool data sets are defined as alternate logical terminals.

Related reading: See [“./T command”](#) on page 222 for the keywords, operands, and examples for specifying the ./T commands.

Tips:

- Unlike IMS, IMS Batch Terminal Simulator allows the use of a load module member name other than the PSB name for the transaction. This facility can be used for testing multiple versions of a single application with IMS Batch Terminal Simulator, but the modules must be correctly named for IMS operation.
- Except for the required operands, if the operands are not specified, IMS Batch Terminal Simulator default values are used. If you do not want to use the default values, specify additional ./T commands to define the PSB name, SPA size, input message edit module name, programming language, program type, and process limit count (PLC).

Related reading: See [“Simulator command summary”](#) on page 187 for the list of default values.

2. If necessary, define the I/O logical terminal name by coding the ./D command with the LTERM= operand.

The default I/O logical terminal is set to LTERM=IOPCB. Specify LTERM=*name* to override the default.

Tips: You can also use the ./D command specify the following attributes:

- You can specify different values for the end-of-segment (EOS) and end-of-message (EOM) indicators if the default values (* and \$ respectively) are not acceptable.
 - If you want to use a device-dependent output formatter to format input and output messages, specify additional ./D command keywords to complete the specification.
3. If necessary, specify other simulator commands.

The ./T command is the only required simulator command; however, IMS Batch Terminal Simulator provides additional simulator commands. After the ./T commands, place additional simulator commands that are applicable.

Examples:

- You might want to specify the ./R command, which indicates to IMS Batch Terminal Simulator a change in the data format of all simulator statements that follow during the run.
- To substitute a given PCB status code on a designated DL/I call, specify the ./S command with the STCD= keyword.
- Messages that are suffixed with the letter I are informational messages. To suppress IMS Batch Terminal Simulator informational messages and call trace information, specify ./O TSOMLVL=0. With this option, the information is not displayed on the TSO terminal, and the TSO terminal normally shows only IMS screens.
- You can set default keywords and operands for ./E commands by using the environment specification table (BTSCHTBL). When the table is used, you do not need to specify the ./E commands in your BTSIN data set.

Related reading:

- See [“Simulator command summary”](#) on page 187 for the list of simulator commands.
 - See [“Simulator commands”](#) on page 187 for the format of the simulator commands and specification examples.
 - See [“Setting defaults for environment parameters \(./E commands\)”](#) on page 51 for the instructions to set the default values.
4. Organize the order of the simulator commands.

If you are specifying the simulator commands in the BTSIN, consider putting all ./T commands first in the input stream to ensure that the teleprocessing generation information is available to IMS Batch Terminal Simulator for the duration of the run.

If a primary transaction generates secondary transactions that are enqueued by IMS Batch Terminal Simulator, they are scheduled and processed by IMS Batch Terminal Simulator immediately upon

completion of the primary transaction. When all the secondary transactions have been processed, IMS Batch Terminal Simulator continues to process primary transactions. Therefore, simulator commands that relate to secondary transactions must be included prior to the completion of the primary transaction because IMS Batch Terminal Simulator does not read the next statement from the input stream until all secondary transactions have already been processed.

5. If necessary, customize the TSO CLIST or the BTSIN to accommodate the characteristics of your application. The following characteristics might require additional customization:
 - Applications that are written in REXX or Java
 - Applications that access Db2, IBM MQ, or data entry databases (DEDBs)
 - Applications that trigger secondary transactions
 - APPC applications or 3270 applications

For the customization procedures, see [Chapter 10, “Customizing JCL for specific applications,” on page 113.](#)

Related reference

[Command reference](#)

IMS Batch Terminal Simulator supports simulator commands to define the transactions, simulator statements to supply transaction messages, and IMS commands to facilitate the testing activities.

Step 5: Starting the session

You start an interactive session by running the TSO CLIST procedure.

Procedure

1. Run the TSO CLIST procedure by entering the following command.

```
EXEC 'cmdlib(clist_name)'
```

2. When you are prompted with the following message, enter ALLOCDS and press Enter.

```
CHOOSE ONE OF THE FOLLOWING OPTIONS: ALLOC ALLOCDS FREE DELETE OR JUST HIT ENTER
```

The meaning of the options are as follows:

ALLOC

Dynamically allocates the existing data sets that are required to run the IMS Batch Terminal Simulator job by issuing the TSO ALLOCATE command, and runs the IMS Batch Terminal Simulator job.

ALLOCDS

Dynamically allocates the new data sets that are required to run the IMS Batch Terminal Simulator job by issuing the TSO ALLOCATE command, and runs the IMS Batch Terminal Simulator job.

FREE

Deallocates the data sets that were previously allocated for IMS Batch Terminal Simulator jobs by issuing the TSO FREE command.

DELETE

Deletes the data sets that were previously allocated for IMS Batch Terminal Simulator jobs by issuing the TSO DELETE command.

ENTER

Runs the IMS Batch Terminal Simulator job.

Tips:

- The following tips can help you use these options:
 - To delete and re-create new output data sets:
 - a. Enter DELETE and press Enter.

- b. Issue the EXEC 'cmdlib(clist_name)'command again.
 - c. Enter ALLOCDS and press Enter.
- To run a different test case and you want to change the data set names:
 - a. Enter FREE and press Enter.
 - b. Issue the EXEC 'cmdlib(clist_name)'command again.
 - c. Enter ALLOC and press Enter.
- To repeat the same test case and you want to use the same data set names, enter ENTER and press Enter.
- You do not need to reallocate data sets between IMS Batch Terminal Simulator executions during the same TSO session. You can restart IMS Batch Terminal Simulator by entering the simulation commands.

IMS Batch Terminal Simulator starts. When it is completely initialized, it reads the specifications in the BTSIN data set. When it reaches the end of file, you are prompted with the following message:

```
ENTER BTS COMMAND OR /FORMAT OR /*.
```

3. If you want to enable the full screen image support (FSS), enter the /FORMAT command followed by the MFS control block name. Press Enter.

The TSO screen displays the formatted screen. After that, FSS is invoked every time IMS Batch Terminal Simulator has a complete screen image of an IMS screen image for output.

Tip: FSS can be used only in a TSO foreground address space. When IMS Batch Terminal Simulator is started, it checks to see if it is running in the TSO foreground. If you want to run IMS Batch Terminal Simulator in the TSO foreground but without FSS, you must include an ./O TSO=NO command in the BTSIN data set. ./O TSO=NO, if coded, must be presented to IMS Batch Terminal Simulator before end of file is reached on BTSIN.

Step 6: Entering the simulation input

To start the simulation, provide a transaction message by entering a simulator statement or directly entering input in the formatted screen image.

Procedure

IMS Batch Terminal Simulator supports two format types for simulator statements: *unformatted-mode simulator statements* and *formatted-mode simulator statements*. When full screen image support (FSS) is active (that is, the /FORMAT command is specified), the simulator statements cannot be used, and the transaction messages can be entered only through the TSO terminal. If FSS is not used, the unformatted-mode simulator statements are accepted. The procedure differs by whether FSS is active or not.

- If FSS is not active (/FORMAT command is not specified), specify the unformatted-mode simulator statements.

The first unformatted-mode simulator statement must begin with a transaction code to which it is related, followed by a transaction message (input for the transaction), and finally an end-of-segment indicator or an end-of-message indicator. This indicator can be coded as either a single-EBCDIC character or as two hexadecimal digits. In the following example, a dollar sign (\$) is used for the end-of-message indicator.

```
PART AN960C10$
```

Related reading: See [“Unformatted-mode simulator statements” on page 240](#) for information about specifying the unformatted-mode simulator statements.

- If FSS is active (the /FORMAT command is specified) and your screen displays the formatted screen image, specify only the transaction message in the location where the transaction message is to be entered.

Simulator statements define the transaction messages. There are two types of simulator statements: unformatted-mode simulator statements and formatted-mode simulator statements.

Simulating the terminal actions with FSS

You can operate a TSO terminal that runs under IMS Batch Terminal Simulator with FSS almost in the same way a user operates an IMS terminal for the same application program. Only the differences are related to the use of the CLEAR, ERASE INPUT, and PF n keys.

Procedure

The following steps describe how to simulate the behavior of the CLEAR, ERASE INPUT, and PF n keys.

Simulating the use of the CLEAR key

The CLEAR key is intercepted by TCAM, and no indication of this action is passed to FSS. Therefore, you must use another method to inform FSS that the screen image has been cleared.

- a. Press the CLEAR key to eliminate the IMS format of the screen image.
- b. Enter CLEAR or &&CLEAR.
- c. Press Enter.

FSS receives the text &&CLEAR and passes it to IMS Batch Terminal Simulator as a simulator command. IMS Batch Terminal Simulator then clears the screen image. To obtain more input, IMS Batch Terminal Simulator invokes the FSS input function. Because FSS finds the buffer empty, it prompts you with the "ENTER BTS COMMAND" message. Now you can enter whatever would normally have followed the clear screen image action.

Simulating the use of the ERASE INPUT key

The ERASE INPUT key does not transmit any data, so the function must be simulated.

- a. Press the CLEAR key.
- b. Enter ERASE or &&ERASE.
- c. Press Enter.

FSS recognizes this command and rewrites the full IMS screen image to the terminal. However, this time the screen image is subjected to an EUA (Erase Unprotected All) order after the original data is displayed. Now the screen image is identical to what would have been displayed after the ERASE INPUT key is pressed. After you have entered data and transmitted it, FSS includes the keyword ERASIN in the simulator statement.

Simulating the use of the PF n keys

If the IMS session uses PF n keys but the TSO terminal that you are using is not equipped with them, simulate their use as follows:

- a. Specify a ./O TSOAID=YES command.
- b. When the prompting message is displayed in response to pressing Enter, you then must:
 - i) Enter a value for PFK1 through PFK24.
 - ii) Press Enter.

Note: If you actually want an Enter action, press Enter again.

Displaying the call trace information

If you enable the call trace option, you can view the call trace information for each DL/I, SQL, MQI call on the TSO terminal.

Procedure

To view the call trace information, specify ./O TSOMLVL=1. FSS displays the call trace information on the TSO terminal. To suppress FSS call trace information, specify ./O TSOMLVL=0.

By specifying the following options, you can receive control at the completion of each call:

- `./O TSOMSG=PROMPT` for DL/I MSG calls
- `./O TSODB=PROMPT` for DL/I DB calls
- `./O TSOSQL=PROMPT` for SQL calls
- `./O TSOMQI=PROMPT` for MQI calls

DL/I call trace option `DLITRACE=8`, SQL call trace option `SQLTRACE=9`, and MQI call trace option `MQITRACE=9` indicate that you want to receive control at the terminal at the completion of each call.

You can use the call trace options of the `./T` command to override the `./O` call trace options for a particular transaction.

When you are prompted with the following message:

```
ENTER 'L CALL', 'L IOAR', 'END' OR NULL LINE
```

Your options and their effect are as follows:

L CALL

Causes a display of the first line of the trace and, if a database call, a trace of the key feedback area as well as the SSAs.

L IOAR

Causes a display of the AIB and I/O area.

END

Turns off `./T` DL/I call trace option 8 and turns on option 2 for this PCB. FSS call trace for subsequent calls for this PCB is displayed as a single-line message to identify each call.

If `./O TSOMSG=PROMPT`, `TSODB=PROMPT`, or `TSOSQL=PROMPT` was specified, they are reset to `./O TSOMSG=YES`, `TSODB=YES`, or `TSOSQL=YES`, respectively. Otherwise, the `./O` FSS call trace options are left unchanged.

Application program processing is resumed.

Null Line (ENTER)

Resumes processing.

If you are using IMS Batch Terminal Simulator with the TSO TEST monitor, you can use the PA1 key at this point to activate TEST. You can then examine your application program by using the TSO TEST facilities.

Reformatting the screen image

You can reformat the current IMS screen image on the TSO terminal by using the `./RETURN` command.

About this task

The `./RETURN` command can be used only with FSS. This command requests reformat of the current IMS screen image on the TSO terminal. If no screen image is available, you receive an error message at the terminal. The command is intended to be used when the TSO terminal is formatted with an IMS screen image, and you want to enter a simulator command before entering data into the screen image.

Procedure

The following steps show how to use the `./RETURN` command.

1. While the IMS formatted screen image is displayed, press the CLEAR key.
2. Enter any simulator command. For example, type `&&./O`, then press Enter.
3. Enter `./RETURN` and press Enter.

The screen image is reformatted. You can now enter data.

Related reference

[./RETURN command](#)

Use the `./RETURN` command to reformat the current IMS screen image on the TSO terminal. This command is intended to be used when the TSO terminal is formatted with an IMS screen image, and you want to enter a simulator command before entering data into the screen image.

Debugging applications in interactive mode

In interactive mode, you can use IBM Debug Tools for z/OS, simulator commands that are provided by IMS Batch Terminal Simulator, TSO TEST monitor, or the VS COBOL II interactive debug facility to debug applications.

About this task

To debug applications in interactive mode, see the following topics:

- [“Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger” on page 98](#)
- [“Debugging with simulator commands” on page 99](#)
- [“Debugging with the TSO TEST monitor” on page 100](#)
- [“Debugging with VS COBOL II interactive debug” on page 100](#)

Debugging with IBM Debug Tool for z/OS or IBM z/OS Debugger

To use IBM Debug Tool for z/OS or IBM z/OS Debugger (Debug Tool), you must modify the TSO CLIST.

Before you begin

To enable an IMS application for use with Debug Tool, it must be compiled with the TEST option. For example:

```
//          PARM.COBOL=' MAP ,OFFSET ,VBREF ,XREF ,OBJ ,TEST ' ,
```

Tip: Several other methods are available to enable your application for use with Debug Tool. For more information, see the *IBM Debug Tool for z/OS User's Guide* or the *IBM z/OS Debugger User's Guide*.

Procedure

Define the data set for the CEEOPTS DD in the TSO CLIST. The data set contains additional invocation-level runtime options. This data set is required to debug applications through Debug Tool (TEST option).

Also ensure that the SEQAMOD data set is included in the TASKLIB concatenation.

IMS Batch Terminal Simulator provides three methods to start Debug Tool in interactive mode. You can display the Debug Tool session on a single screen, you can display the Debug Tool session on another VTAM terminal, or you can display the Debug Tool session with remote debug mode.

- To display the Debug Tool session on a single screen, complete these steps:
 - a) Specify the TEST runtime option in a data set, and allocate the data set to CEEOPTS DD in the BTS CLIST. For example:

```
TEST( , , ,*)
```

- b) Specify the parameter CALLTYPE(CALL) in the BTS CLIST.
- To display the Debug Tool session on another VTAM terminal, specify the MFI suboption of the TEST runtime option with the LU name of the VTAM terminal or the VTAM suboption of the TEST runtime option with your TSO user ID. For example:

```
TEST( , , ,MFI%TRMLU001:*)
```

or

```
TEST( , , ,VTAM%USERABCD:*)
```

In this way, the job starts Debug Tool in full-screen mode with a VTAM terminal. The VTAM terminal controls the Debug Tool session. For a BTS CLIST, you can specify the TEST runtime option in a data set and allocate the data set to CEEOPTS DD.

- To display the Debug Tool session with remote debug mode, specify the TCPIP suboption of the TEST runtime option with the IP address and port number that the remote debugger is listening to. For example:

```
TEST( , , ,TCPIP&9.99.999.99%8001:*)
```

In this way, the job starts Debug Tool in remote debug mode with a remote debugger. For a BTS CLIST, you can specify the TEST runtime option in a data set and allocate the data set to CEEOPTS data set.

Debugging with simulator commands

By using the simulator commands that are provided by IMS Batch Terminal Simulator, you can run path tests as batch jobs without modifying your applications or compiling your applications with specific options.

Procedure

You can use the following simulator commands to debug your applications:

./P command

You can use the ./P command to apply a patch at a particular point in application program processing. You can apply a patch to the application program by specifying the module offset and the values in character or hexadecimal.

For example, you can copy (zap) the constant DC C'January' that is defined at offset X'0DAC' in your single CSECT application program (MYPGM) and test the following conditions:

- Replace to DC C'February' in the first transaction
- Use the original value in the second transaction
- Replace to DC C'March' in the third transaction

To achieve these results, your input stream might contain the following lines:

```
./T TC=ZAPTEST MBR=MYPGM  
./P MBR=MYPGM PA=000DAC PC=February  
ZAPTEST SECOND  
./P MBR=MYPGM PA=000DAC PC=March  
ZAPTEST THIRD
```

With this specification, you can run three path tests in one IMS Batch Terminal Simulator.

Related reading: See [“./P command”](#) on page 212 for more information about the ./P command and its keywords.

./S command

You can use the ./S command to set specific action to be taken at a particular point during the processing of the application program. The action can be either to issue a SNAP macro or to substitute a given PCB status code on a designated DL/I call. You can set only one specific action per call.

The following example shows how to use the ./S command to take a snapshot dump of a 62-byte work area at offset '1F20' in the PARTRAN application program when a GNP call is issued in the PART database.

```
./T TC=PARTRAN MBR=PARTRAN  
./S MBR=PARTRAN SA=001F20 BYTES=62 PCB=PART X  
FUNC=GNP TIMES=5
```

The following example shows how to use the `./S` command to generate a status code of A5 on the fifth ISRT to the I/O PCB while the XYZ transaction is being processed.

```
./T TC=XYZ MBR=XYZPGM...
./D LTERM=IOPCB...
./S MBR=XYZPGM PCB=IOPCB X
    FUNC=ISRT TIMES=4 STCD=00
./S MBR=XYZPGM PCB=IOPCB X
    FUNC=ISRT TIMES=1 STCD=A5
```

Related reading: See [“./S command” on page 218](#) for more information about the `./S` command and its keywords.

MSGABENDxxx command (debug command)

You can use this command to cause a user 4091 abend after IMS Batch Terminal Simulator issues a specific message.

For example, when Db2 changed-data-capture exit routine (DB2CDCEX) is called, IMS Batch Terminal Simulator issues BTS0107W DB2 CHANGED DATA CAPTURE EXIT STARTED. If you specify MSGABEND107 in BTSIN, IMS Batch Terminal Simulator causes a user 4091 abend when Db2 changed-data-capture exit routine (DB2CDCEX) is called.

Related reading: See [“Debug commands” on page 235](#) for more information about the MSGABEND command.

Debugging with the TSO TEST monitor

If you enable the TSO TEST monitor, you can use the TSO TEST command while testing application programs under IMS Batch Terminal Simulator.

About this task

You can use the TSO TEST command following an abnormal termination in the program to determine where the termination occurred and the environment at the time of the termination (for example, storage and register contents). You can also set breakpoints in a program before running it; then, during the run, you can study the program at these points or make temporary changes to the program or data.

A known restriction is the use of the GO subcommand after abnormal termination in the program. This restriction is due to an IMS restriction that does not allow the application program to continue after terminating abnormally. If the GO subcommand is used in this situation, the subcommand continues to completion, but IMS issues an U0032 ABEND upon return from the application program.

Procedure

To enable the TSO TEST command, change the CALLTYPE parameter in CLIST to CALLTYPE(TEST).

Related reading: For more information about the TSO TEST command, see the *TSO/E Command Reference*.

Debugging with VS COBOL II interactive debug

To use VS COBOL II interactive debug (COBOL interactive debug), you must modify the TSO CLIST.

Procedure

To activate COBOL interactive debug, complete these steps:

1. Make the command processor BTSCP available under TSO.

Create the LOGON procedure that defines the BTSCMD library (SBTSCMD0) on the STEPLIB DD statement. Alternatively, enter BTSCP as a member of the partitioned data set SYS1.CMDLIB.

2. To run COBOL interactive debug command processor (COBTEST) in full-screen mode, include ISPF libraries in the JOBLIB or STEPLIB DD statement to make them available to IMS Batch Terminal Simulator.

Note: The COBTEST runs in line mode or, if ISPF (version 2.3 or later) is active, in full-screen mode.

3. Specify the CALLTYPE(BTSCP) parameter in the BTS CLIST.
4. Start BTS CLIST by entering the following command in ISPF:

```
ISPEXEC SELECT CMD(BTSCLIST CALLTYPE(BTSCP)) NEWAPPL(IGZ)
```

Note: With COBTEST in full-screen mode, the ISPEXEC SELECT PGM service cannot be used to start IMS Batch Terminal Simulator under ISPF.

5. Enter one or more ./C commands.

Notes:

- a. All ./C commands are ignored when IMS Batch Terminal Simulator is not run under BTSCP.
 - b. You cannot test dynamically linked COBOL subroutines under IMS Batch Terminal Simulator. The ./C information is ignored or not processed.
6. When the same COBOL program is iteratively invoked (via LINK, XCTL, LOAD, or scheduled by IMS Batch Terminal Simulator) under control of an interactive debug command processor, make either of the following specifications:
 - ./E REATTCH=YES
 - ./P MBR=BTSCOM00 PA=004DE PX=01

See [Example 2](#) for ./E command.

7. After the prompting message is displayed, enter any COBOL interactive debug subcommands (for example, enter the AT subcommand to set breakpoints.) To begin execution, key GO or RUN and press Enter.

When 'PROGRAM ENDED' is displayed, key END for TESTCOB or QUIT for COBTEST, and then press Enter.

Related reading: See *VS COBOL II Application Programming: Debugging* for more information.

JCL example: Using COBINDD with COBTEST

To use the COBINDD with COBTEST under IMS Batch Terminal Simulator, add the following CLIST statements immediately before calling BTSCP in the BTS CLIST:

```
CONTROL MAIN
SET &RCODE = 0
ERROR +
DO
  SET &RCODE = &LASTCC
  RETURN
END
SET &INFILE = NO
OPENFILE COBINDD
IF &RCODE = 0 THEN +
DO
  CLOSFIL COBINDD
  SET &INFILE = YES
END
ISPEXEC VPUT (INFILE) SHARED
```

What to do next

When a transaction is subsequently started for one of the COBOL programs specified, IMS Batch Terminal Simulator links to the COBTEST. The COBOL program is then loaded and called under control of COBOL interactive debug.

Related reference

[./C command](#)

Use the `./C` command to define the usage of interactive debug of some languages.

Tips for testing in interactive mode with FSS

You can use certain tips when testing applications in interactive mode with full screen image support (FSS).

Subsections:

- [“MFS bypass support without basic input-editing” on page 102](#)
- [“Entering simulator commands” on page 102](#)
- [“Using alternate terminals” on page 102](#)
- [“Generating formatted-mode simulator statements” on page 103](#)

MFS bypass support without basic input-editing

No input edit is provided by IMS Batch Terminal Simulator when MFS bypass is used, but IMS Batch Terminal Simulator internally performs basic input-editing. If IMS Batch Terminal Simulator users do not want to perform basic input-editing when MFS bypass (mod name is DFS.EDTN) is used, use MFS bypass special support.

This special support performs MFS bypass without basic input-editing. When this support is used, IMS Batch Terminal Simulator internally does not perform basic input-editing, which IMS Batch Terminal Simulator normally performs, for inbound data and does not insert inbound data into the BTSPUNCH data set.

In addition, the image of the display screen or printer is not updated. This function is useful when IMS Batch Terminal Simulator users use an inbound D/T3270 data stream that cannot perform basic input-editing. GDDM cannot be used at the same time this special support is used. See [Example 16 for `./P` command](#) for the details.

Entering simulator commands

Enter a simulator command with no command keywords, and IMS Batch Terminal Simulator displays a list of the commonly used keywords for that command.

If you want to enter a simulator command, you must do so from an unformatted screen image. With some releases of VTAM or TCAM, it is not possible to tell if the data received by IMS Batch Terminal Simulator came from an unformatted screen image. To bypass this problem, an escape sequence has been defined to allow IMS Batch Terminal Simulator to recognize its commands. The escape sequence consists of the two characters `&&`, which must precede the simulator command. If simulator commands are entered without the proper escape sequence, they might be treated as data and passed to MFS. Try entering commands with and without the escape sequence to determine if the escape sequence is needed. To enter a simulator command from a TSO terminal, formatted with an IMS screen image, proceed as follows:

1. Press the CLEAR key.
2. Enter a simulator command (for example, PAX or `&&PAX`, CLEAR or `&&CLEAR`).
3. Press Enter.

Using alternate terminals

If alternate terminals are used with FSS, the following rules apply:

- IMS Batch Terminal Simulator attempts to display the output of an alternate terminal for a screen device on the TSO terminal. If the screen size of the TSO terminal is smaller than the simulated alternate terminal, the results are unpredictable.

- Data cannot be entered into an IMS screen image that is an alternate terminal. FSS ignores any input that comes from such a screen image, even simulator commands, and continues by displaying the next physical page. To continue, press Enter.
- If you press the PA1 key when multi-page output is being displayed at the alternate terminal, FSS does not receive control. To resume the processing, press the CLEAR key and then press Enter.
- Before a physical page is transmitted to an alternate terminal, you receive a message giving the name of the alternate destination.

Generating formatted-mode simulator statements

If you specify the BTSPUNCH data set and simulate your application in interactive mode with FSS, IMS Batch Terminal Simulator converts all the inputs that were entered through the TSO terminal into the formatted-mode simulator statements and records them in the data set.

In some cases, coding the formatted-mode simulator statements from scratch can be complicated because you must specify the location of the lines and columns for input. The formatted-mode simulator statements that are generated in the BTSPUNCH data set can be used as BTSIN input in batch mode.

Chapter 8. Tutorial: Testing an application from Eclipse

Learn how to specify the runtime parameters, program libraries, and input to start testing applications from the Eclipse interface.

For application developers who use the Eclipse development environment, the IMS Batch Terminal Simulator Eclipse interface provides developers with a familiar interface for testing applications.

Learning objectives

By completing the tutorial, you will learn how to complete the following tasks:

- Create an IMS Batch Terminal Simulator server runtime environment
- Define a launch configuration for testing specific IMS application
- Test an application by using the Eclipse interface
- Export a server runtime environment configuration to an XML file

Time required

This tutorial takes approximately 45 minutes to complete.

Audience

This tutorial is targeted for application developers with some z/OS knowledge.

Prerequisites

IMS Batch Terminal Simulator must be configured, and Distributed Access Infrastructure must be configured and started on the z/OS system that you want to test on. The IMS Batch Terminal Simulator plugins must also be installed in the Eclipse environment that you test from.

For this tutorial, you need the following information:

- The Distributed Access Infrastructure host name and port number
- z/OS system login information

Related tasks

Installing Eclipse interface

To use the IMS Batch Terminal Simulator Eclipse interface, you must install IMS Batch Terminal Simulator plugins into the Eclipse installation that you want to test IMS applications in.

Step 1: Creating a server runtime environment

When you create an IMS Batch Terminal Simulator server runtime environment, you define the IMS system libraries, database libraries, operational parameters, and other data definitions that you require to run your applications.

About this task

An IMS Batch Terminal Simulator server defines a generic environment in which most of your applications can run so that you do not have to create a server for each application. You define application-specific libraries and data definitions, such as STEPLIB information, in the launch configuration.

Procedure

1. In Eclipse, click **Window** > **Show View** > **Servers** from the **File** menu to display the Servers window.
2. Right-click in the Servers window, and then click **New** > **Server** to start the New Server wizard.

3. From the server type list, select **IBM > IMS Batch Terminal Simulator**.
4. Specify the Distributed Access Infrastructure server name as the host name, any name for the server name.

The host name is location of the Distributed Access Infrastructure server, and the server name is the name that is displayed in the server window. Specify a server name that helps you identify what parameters and libraries are specified for this server.

5. To create a server runtime environment, click **Next**.

After you have created an IMS Batch Terminal Simulator server, you create new server runtime environments by clicking **Add** next to the Server runtime environment drop-down.

6. Select **Create a new configuration**, and then click **Next**.

Tip: If you have an existing configuration that was exported to an XML file, you can specify that file by selecting **Use an existing configuration**. By using an existing configuration, you do not have to manually specify the parameter values and library definitions.

To learn how to export a configuration, see [“Step 4: Exporting a server runtime \(optional\)” on page 109](#).

7. In the Runtime Parameters tab, specify the runtime parameter values.

- a) Specify a region type.

Depending on your application, you can select one of four IMS region types. See [“Determining the types of IMS region” on page 22](#) for more information about the different region types.

- b) Specify additional runtime parameters , such as IMSID or DIRCA size, by clicking **Show Advanced**.

8. In the Library Definitions tab, specify the libraries for the server runtime environment.

- a) Specify DFSRESLB data set name and volume by clicking a line in the DFSRESLIB table.

- b) Optional: Specify the STEPLIB, IMS, and IMSACB libraries by clicking a line in their respective tables.

You can move definitions up or down by selecting the definition and clicking **Move Up** or **Move down**. Data sets that are higher in the table are concatenated first.

9. In the More Data Definitions tab, click **Add** to specify any additional data definitions, such as FORMAT.

The More Data Definitions tab contains a table with required data definition names, which cannot be removed. You are not required to change these data definitions, but you can specify data set names to store any output.

10. Optional: If you have Debug Tool for z/OS or IBM z/OS Debugger and have installed the Debug Tool Eclipse plugin, you can define Debug Tool load libraries in the Debug Tool Data Definitions tab.

- a) In the first table, specify the STEPLIB data sets that are specific to Debug Tool.

The following data sets are required for Debug Tool: SEQAMOD, SCEERUN, EQAOPTS, and CEEBINIT.

If you add additional data sets, they are concatenated into the STEPLIB DD.

- b) In the second table, specify the CEEOPTS DD.

You can add multiple data definitions by typing the DD names in different rows. If you have more than one DD, they are concatenated. The CEEOPTS DD name is specified by default.

For more information about Debug Tool or the Debug Tool data sets, see the *IBM Debug Tool for z/OS User's Guide* or the *IBM z/OS Debugger User's Guide*.

11. Click **Next**.

12. Specify the Distributed Access Infrastructure connection settings, which are required to communicate with IMS Batch Terminal Simulator.

- a) Specify the Distributed Access Infrastructure hostname and port number.

- b) Select the EBCDIC code page that your z/OS system is using.

The code page is required to correctly translate messages from the local encoding where Eclipse is running to the remote encoding where IMS Batch Terminal Simulator is running. The default code page is IBM-1047.

- c) If your z/OS environment is using AT-TLS to enable SSL on the port that Distributed Access Infrastructure is using, specify the keystore, truststore, and passwords. The keystore, truststore, and passwords are created from the certificate that AT-TLS generated.
- d) Specify the user name and password for the z/OS system that you are connecting to.

13. Click **Finish**.

Results

The server is listed in the Servers window. You can specify this server when you create a launch configuration.

You can also modify the server configuration by selecting it and clicking the Runtime Environment link in the Overview window.

Lesson checkpoint

You learned to define a server runtime environment.

You used the IMS Batch Terminal Simulator Eclipse interface to create an IMS Batch Terminal Simulator region to test your applications.

Step 2: Defining a launch configuration

After creating an IMS Batch Terminal Simulator server, you define application-specific libraries, data definitions, and inputs by creating a launch configuration.

Procedure

1. From the **File** menu, click **Run > Run configurations**.

2. From the list of run configurations, double-click **IMS Batch Terminal Simulator**.

In the list of run configuration, a new configuration is created under IMS Batch Terminal Simulator.

3. Select the configuration that you created to customize it.

4. In the **Name** field that is on top of the tabs, specify a name for this launch configuration.

Use a name that reminds you of the launch configuration settings so that you can easily identify it.

5. In the Server tab, select the IMS Batch Terminal Simulator server that you want to use each time you run this launch configuration.

The server defines the runtime environment in which the application runs. To create a server, see [“Step 1: Creating a server runtime environment” on page 105](#).

6. In the BTSIN tab, specify command statements and IMS messages.

- a) If you have a BTSPUNCH or BTSIN file that contains all the input data that you require, click **Import** to specify it.

A BTSPUNCH file records all the input that was received by IMS Batch Terminal Simulator and is generated each time that you run an application.

- b) In the BTS Commands section, specify IMS Batch Terminal Simulator command statements.

For example, to define a translation, add a transaction command (./T), a transaction code, and member name in this section. For more information about command descriptions, default values, and syntax specifications, see [Chapter 14, “Command reference,” on page 187](#).

- c) In the IMS messages section, specify IMS messages that IMS Batch Terminal Simulator processes.

IMS Batch Terminal Simulator reads a transaction from each line, passes the transaction to the application for processing, and then reads the next line.

You can also interactively pass messages to IMS Batch Terminal Simulator by selecting **Prompt for additional IMS messages**. By interacting with IMS Batch Terminal Simulator, you can view the output of your current run before entering the next message that IMS Batch Terminal Simulator processes.

7. In the Program Libraries and Additional Data Sets tab, specify the program libraries and the data definitions that are required to run your application.

You can move definitions up or down, which changes the order in which they are concatenated. Data sets that are higher in the list are concatenated first.

8. Optional: If you have Debug Tool for z/OS or IBM z/OS Debugger, you can add application-specific debug data sets in the Debug Tool Data Sets tab.

The debug data sets are not required for Debug Tool. These data sets, such as EQADEBUG, EQADBG, EQAMDBG, and EQAUEDAT, define data definitions that are required by specific applications for debugging.

For more information about Debug Tool or the Debug Tool data sets, see the *IBM Debug Tool for z/OS User's Guide* or the *IBM z/OS Debugger User's Guide*.

9. Click **OK** to complete the launch configuration wizard.

Lesson checkpoint

You learned how to create a launch configuration to test an application.

You specified a server runtime environment, libraries, and input to test a specific application.

Step 3: Testing an application in Eclipse

After you create a server and launch configuration, you can start testing applications from Eclipse by running the launch configuration.

Procedure

1. From the **File** menu, click **Run > Run configurations**.
2. In the navigation tree, expand IMS Batch Terminal Simulator.
3. Select the launch configuration for the application that you want to test.
4. Click **Run**.

Results

The following output files are saved in the `Servers\server_name\Reports\` directory:

OUTPUT.BTSOUT

This file contains output from IMS Batch Terminal Simulator, such as BTSIN echo back, DL/I call information, and trace information of the internal processes.

OUTPUT.BTSDEBUG

This file contains snapshot dumps of the trace table and various control blocks.

OUTPUT.BTSSNAP

This file contains snapshot dumps when you specify the `./S` command.

OUTPUT.BTSWTO

This file contains all the write-to-operator (WTO) calls.

OUTPUT.BTSPUNCH

This file contains all the input that was received by IMS Batch Terminal Simulator. You can reuse the input in BTSPUNCH by importing the BTSPUNCH file when you create a launch configuration. With BTSPUNCH, you can replay the same transaction sequence or build a regression suite.

Tip: If you do not see the latest output files, refresh the server directory by pressing F5.

Step 4: Exporting a server runtime (optional)

Export the configurations for a server runtime environment to an XML file to create similar servers on different Eclipse clients.

Procedure

1. In the Servers window, right-click test server and select **Export Runtime**.
2. Specify the location and name for the configuration XML file.
3. Click **Save**.

You can import the XML file when you create a server in the New Server wizard.

Step 5: Exporting a server runtime environment and launch configuration (optional)

Export a server runtime environment and launch configuration to an XML file to specify an IMS Batch Terminal Simulator configuration for the resource adapter.

Procedure

1. From the **File** menu, click **Run > Run configurations**.
2. Select the IMS Batch Terminal Simulator launch configuration that you want to export.
3. Click the **Export Configurations** tab.
4. Click **Export**, and then specify the location and name of the XML file.
5. Click **Save**.

When you configure an IMS Batch Terminal Simulator resource adapter (BTSRA), use the XML file that you exported to specify the `BTSTConfig` file.

Chapter 9. Testing an application that uses the IMS Batch Terminal Simulator resource adapter

For TMRA client applications that run on WebSphere Application Server, you can use the IMS Batch Terminal Simulator resource adapter (BTSRA) to run IMS applications on IMS Batch Terminal Simulator instead of on IMS.

Before you begin

Install and configure BTSRA on the WebSphere Application Server that has the TMRA client applications that you want to test. Also, when you test a TMRA client application, remove the TMRA resource adapter that the client application uses so that the application communicates only with the BTSRA.

Procedure

After you install BTSRA and configure the required connection factories, run the client application.

Because the BTSRA is configured with the same properties as the TMRA, you do not need to modify the client application. When you run the client application, it uses BTSRA and runs IMS applications in IMS Batch Terminal Simulator.

What to do next

After you complete testing applications with the BTSRA, you can remove BTSRA and redeploy TMRA to run the applications in a production environment.

Related tasks

[Installing the IMS Batch Terminal Simulator resource adapter](#)

Install and configure the IMS Batch Terminal Simulator resource adapter (BTSRA) by using the WebSphere Application Server administrative console.

[Adding and configuring the IMS Batch Terminal Simulator resource adapter J2C connection factory](#)

After installing the IMS Batch Terminal Simulator resource adapter (BTSRA), use the WebSphere Application Server administrative console to add and configure the BTSRA J2C connection factory.

Related reference

[Transaction Manager resource adapter supported functions](#)

IMS Batch Terminal Simulator resource adapter (BTSRA) supports most of the Transaction Manager resource adapter (TMRA) functions that do not relate to IMS Connect or IMS OTMA.

Transaction Manager resource adapter supported functions

IMS Batch Terminal Simulator resource adapter (BTSRA) supports most of the Transaction Manager resource adapter (TMRA) functions that do not relate to IMS Connect or IMS OTMA.

Supported TMRA functions

The following TMRA functions are supported by BTSRA:

- Commit Mode 1
- Execution Timeout/Socket Timeout (BTSRA supports only socket timeout. Set the timeout value to the lower value between the Execution Timeout and the Socket Timeout.)
- Old-style conversations (same socket)
- SendOnly
- SendRecv
- Sync Level NONE
- SSL

For more information about each of these functions, see the IMS TM Resource Adapter information in the IMS documentation.

Tolerated TMRA functions

You can use the following functions without error, but they are not passed to IMS Batch Terminal Simulator.

Important: You can use these function without error, but they are not passed to IMS Batch Terminal Simulator.

- Commit Mode 0
- SendOnly with ACK
- Sync Level CONFIRM/ACK/NAK (For NAK, the changes are not backed out.)

For more information about each of these functions, see the IMS TM Resource Adapter information in the IMS documentation.

Chapter 10. Customizing JCL for specific applications

Depending on the types of IMS applications that you are testing, you must make certain changes to the JCL stream or in the BTS CLIST.

Topics:

- [“JCL customization by application languages” on page 113](#)
- [“JCL customization by application types” on page 113](#)

JCL customization by application languages

To test applications that are written in REXX or Java, you must make certain modifications to your JCL.

Applications written in REXX

If your application is written in REXX, the following requirements and considerations apply.

To test IMS REXX applications with IMS Batch Terminal Simulator, you must specify ASM or CBL on the `./T LANG=` keyword.

When the conversational application on REXX has SPA with data mapping defined by MAPDEF, SPA data might need to be specified by the `./SPA` command. If SPA data defined by MAPDEF is of the variable-data type or the packed-decimal-data type, you must specify the `./SPA` command.

IMS issues an INQY call with the 'ENVIRON' subfunction code internally for REXX application programs. This INQY call is reported on the BTSOUT as an application program call.

Related reading:

- See [“./T command” on page 222](#) for `./T` command specification.
- See [Example 3 for ./SPA command](#) for the `./SPA` command specification.

Applications written in Java

If your application is written in Java, the following requirements and considerations apply.

To test Java application programs with IMS Batch Terminal Simulator, make the following specifications:

- The IMS Java application can run only in the JBP region by using stand-alone JVM. To set the JBP region, specify `KW=JBP` in IMS Batch Terminal Simulator JCL or in CLIST.
- Specify the application parameter (APARM parameter) on the EXEC statement as `APARM=BTS`.
- Specify ASM or CBL on the `./T LANG=` keyword.

Related reading:

- See [“./T command” on page 222](#) for `./T` command specification.
- For information about IMS Java applications, see *IMS Application Programming*.

JCL customization by application types

To test certain types of applications, you must modify your JCL appropriately.

The following types of applications require special modifications:

- Applications that access Db2 databases
- Applications that access IBM MQ
- Applications that access DEDBs
- Applications that trigger secondary transactions

- 3270 applications
- APPC applications

Applications that access Db2 databases

If your application accesses Db2 databases, the following requirements and considerations apply.

Subsections:

- [“Considerations in a BTS batch \(KW=DLI or KW=DBB\) environment” on page 114](#)
- [“Considerations in a BTS online \(KW=BMP\) environment” on page 114](#)
- [“Simulator commands for Db2 SQL call trace” on page 114](#)
- [“DDITV02 DD specifications by IMS Batch Terminal Simulator” on page 115](#)

Considerations in a BTS batch (KW=DLI or KW=DBB) environment

To run Db2 applications with IMS Batch Terminal Simulator in a BTS batch (that is, KW=DLI or KW=DBB) environment, the SSID must be specified with the `./E` command `SSID= keyword` or the `./P` command. If it is not specified, IMS Batch Terminal Simulator ends with an IMS ABEND U3057 when the SQL call is issued.

If Db2 libraries are included in the JOBLIB, STEPLIB, or LINKLIST with KW=DLI or KW=DBB, and the SSID is specified with the `./E` command `SSID= keyword` or the `./P` command, IMS Batch Terminal Simulator always runs with Db2 DL/I Batch Support even if the application program does not issue the SQL call. In this environment, consider the following factors:

SSID

To establish the connection with Db2 when running in IMS Batch environment, you do not need to specify a value for SSM in an EXEC parameter. If the value for SSM is specified as an EXEC parameter, IMS Batch Terminal Simulator translates it into a null value and then passes the EXEC parameter to IMS.

ROLS

When IMS Batch Terminal Simulator is running with Db2 DL/I Batch Support, the ROLS call does not backout database changes to an intermediate back out point established by a prior SETU call or to the prior commit point. The ROLS call is rejected with an 'RC' status code.

Related reading: For SSID= specifications, see [Example 3 for ./E command](#).

Considerations in a BTS online (KW=BMP) environment

To run Db2 applications in a BTS online environment (KW=BMP), you must connect Db2 to IMS before running IMS Batch Terminal Simulator.

If the SSM is specified on the EXEC parameter, you must specify the IMS PROCLIB data set on the PROCLIB DD statement.

Related reading: For more information, see the *Db2 for z/OS Administration Guide*.

Simulator commands for Db2 SQL call trace

IMS Batch Terminal Simulator optionally provides call trace information for Db2 SQL calls that are issued by application programs.

Db2 SQL call trace information consists of the following information:

- Statement type
- Program or DBRM name
- PLAN name (only in a BTS batch environment)
- Statement number

- Section number
- Error status in the SQL communication area (SQLCA)
- IFI SQL statement information (see notes)
- Input and output variables

Notes:

1. To obtain IFI SQL statement information, Db2 monitor trace class 1 must be active for IFCID 124, and your Db2 authorization ID must have MONITOR2 privilege or SYSADM authority.
2. If the thread is not currently executing an SQL statement in Db2, IMS Batch Terminal Simulator might not be able to obtain SQL statement data from Db2 IFI. In this case, IMS Batch Terminal Simulator issues the following information as IFI SQL statement information:

```
IFI INFORMATION:
SQL STATEMENT INFORMATION UNAVAILABLE
```

3. If `./O SQL=ALL` or `TSOSQL=ALL` is specified, host variable values are printed on BTSOUT or displayed on the TSO terminal in the following cases:
 - SQL STATEMENT VARS (values from the applications) and VARS FROM DB2 (values from Db2) when the SQL call ends normally.
 - SQL STATEMENT VARS when the SQL call ends abnormally.
 - If you use the LOB (large objects) data type, and you specify `./O SQLOBLN=`, IMS Batch Terminal Simulator puts the LOB data type and the length on the BTSOUT or the TSO terminal display as follows:

LEN= shows the actual data length
DATA= shows the LOB data

Use the following simulator commands for Db2 SQL call trace:

- To print SQL call trace information in BTSOUT, specify the SQL keyword for the `./O` command.
- To display SQL call trace information on the TSO terminal, specify the `TSOSQL` keyword for the `./O` command.
- To specify the SQL call trace options for each SQL call for each transaction, specify the `SQLTRACE` keyword for the `./T` command. The `SQLTRACE` options override the call trace options of the `./O` command.
- In BTS batch environments:
 - Specify the Db2 subsystem ID with the `./E` command.
 - You can specify the Db2 plan name by specifying the `PLAN` keyword for the `./T` command. This function is supported only in a BTS batch environment.

Related reading:

- See [“./O command” on page 205](#) for `./O` command specification.
- See [“./T command” on page 222](#) for `./T` command specification.
- See [Example 3 for ./E command](#) for an example to specify the Db2 subsystem ID with the `./E` command.
- See [“Db2 call trace listing” on page 144](#) for call trace information.

DDITV02 DD specifications by IMS Batch Terminal Simulator

IMS Batch Terminal Simulator uses Db2 DL/I batch support when running in IMS batch environment (KW=DLI or KW=DBB). An input data set, specified by the DDITV02 DD statement, is required by Db2. However, you can omit the DDITV02 in your JCL or CLIST. When omitted, IMS Batch Terminal Simulator builds and allocates the DDITV02.

IMS Batch Terminal Simulator allocates the DDITV02 to the unit group-name SYSDA. If this unit group-name is not available for your MVS system, contact the system administrator in charge of the installation

and override the unit group-name in the BTS common area with the UGN= keyword of the ./E command or of the ./P command. An OUTPUT data set, specified by the DDOTV02 DD statement, is optional. You can get it by specifying the DDOTV02 DD statement in your JCL or CLIST.

The following table summarizes the specifications used when DDITV02 is built and allocated by IMS Batch Terminal Simulator.

Table 23. Content of the DDITV02 data set

Field	Action
SSN	IMS Batch Terminal Simulator sets the SSID in the BTS common area. See the SSID= keyword example in Example 3 for ./E command .
LIT	IMS Batch Terminal Simulator processes as if LIT were not specified.
ESMT	'DSNMIN10'
RTT	IMS Batch Terminal Simulator processes as if RTT were not specified
REO	IMS Batch Terminal Simulator sets the region error option specified on the ./E command. If this option is not specified, IMS Batch Terminal Simulator processes as if REO were not specified. See the ERR= keyword in “./E command” on page 198 .
CRC	IMS Batch Terminal Simulator processes as if CRC were not specified.
CONNECTION_NAME	IMS Batch Terminal Simulator sets the CONNECTION_NAME specified on the ./E command. If it is not specified, IMS Batch Terminal Simulator sets the TSO authorization ID or job name. See the CONNECT= keyword in “./E command” on page 198 .
PLAN	IMS Batch Terminal Simulator sets the PLAN name specified with the ./T command. See the PLAN= keyword in “./T command” on page 222 .
PROG	Used internally

Applications that access IBM MQ

If your application accesses IBM MQ, the following requirements and considerations apply.

Subsections:

- [“Considerations in a BTS batch \(KW=DLI or KW=DBB\) environment” on page 116](#)
- [“Considerations in a BTS online \(KW=BMP\) environment” on page 117](#)
- [“Simulator commands for MQI call trace” on page 117](#)
- [“IBM MQ trigger message for IMS application \(MQTMC2\)” on page 117](#)

Considerations in a BTS batch (KW=DLI or KW=DBB) environment

If you are running IMS Batch Terminal Simulator in a BTS batch environment (KW=DLI or KW=DBB), and if you need the call trace information for MQI calls, add the STEPLIB DD statement to specify the SCSQLOAD data set. If you do not specify the DD statement, IMS Batch Terminal Simulator returns message BTS0114I and does not report the call trace information for MQI call on BTSOUT when the MQI call is issued by the application program that is linked with the IBM MQ IMS stub CSQQSTUB.

Considerations in a BTS online (KW=BMP) environment

To run IBM MQ applications in a BTS online environment (KW=BMP), you must connect IBM MQ to IMS before IMS Batch Terminal Simulator runs.

If the SSM is specified on the EXEC parameter, you must specify the IMS PROCLIB data set for the PROCLIB DD statement.

Related reading: For information about IBM MQ, see *IBM MQ Installing*.

Simulator commands for MQI call trace

IMS Batch Terminal Simulator provides the call trace information for MQI calls that are issued by application programs.

The MQI call trace consists of the following information:

- MQI call function
- Compression code
- Reason code
- Putting or getting messages
- Putting or getting message lengths

IMS Batch Terminal Simulator does not provide the call trace information for the following message property APIs that were introduced in WebSphere MQ 7.0:

MQCRTMH, MQSETMP, MQINQMP, MQMHBUF, MQBUFMH, MQDLTMP, MQDLTMH

The message property APIs are taken into consideration when building or using data-structures that contain message properties, and they are passed on to the queue manager on MQPUT or retrieved on MQGET. No transactional work exists in these APIs. These APIs execute the entire code within the application region, and the APIs are not passed on to IMS.

Use the following simulator commands for MQI call trace:

- To print MQI call trace information in BTSOUT, specify the MQI keyword for the `./O` command.
- To display MQI call trace information on the TSO terminal, specify the TSOMQI keyword for the `./O` command.
- To specify the MQI call trace options for each MQI call for each transaction, specify the MQITRACE keyword for the `./T` command. The MQITRACE options override the call trace options of the `./O` command.

Related reading:

- See [“./O command”](#) on page 205 for `./O` command specification.
- See [“./T command”](#) on page 222 for `./T` command specification.
- See [“MQI call trace listing”](#) on page 148 for call trace information.

IBM MQ trigger message for IMS application (MQTMC2)

When `./T MSGTYPE=MQTMC2` is specified, IMS Batch Terminal Simulator can pass the trigger message (MQTMC2) to the application without running the CSQQTRMN transaction supplied by IBM MQ. To start the application, you must specify only the transaction code in the BTS input stream.

If the QNAME= keyword is specified for the `./T` command, IMS Batch Terminal Simulator provides the following information in the trigger message by using the IBM MQ batch adapter.

Table 24. Data in the trigger message

Parameter	Value
StrucId	'TMC'

Table 24. Data in the trigger message (continued)

Parameter	Value
Version	'2'
QName	IMS Batch Terminal Simulator sets the name of the triggered queue specified in the ./T command. See the QNAME= keyword in “./T command” on page 222 .
ProcessName	IMS Batch Terminal Simulator sets the name of process object.
TriggerData	IMS Batch Terminal Simulator sets the trigger data.
ApplType	IMS Batch Terminal Simulator sets blanks.
ApplId	IMS Batch Terminal Simulator sets the transaction code specified on the ./T command. See the TC= keyword in “./T command” on page 222 .
EnvData	IMS Batch Terminal Simulator sets the environment data.
UserData	IMS Batch Terminal Simulator sets the user data.
QMgrName	IMS Batch Terminal Simulator sets the default queue manager name. Use the ./T QMGRNAME= keyword to override the default queue manager name. See the QMGRNAME= keyword in “./T command” on page 222 .

Note: To obtain more information about trigger messages, IMS Batch Terminal Simulator issues MQI calls by using the IBM MQ batch adapter. If the MQI calls fail, message BTS0115I is returned.

If the QNAME= keyword of the ./T command is not specified, the following items contain blanks:

- QName
- ProcessName
- TriggerData
- EnvData
- UserData
- QMgrName

Applications that access DEDBs

If your application accesses data entry databases (DEDBs) with IMS 13 or earlier, the following requirement applies.

To obtain the correct length of the DEDB compressed segment or the DEDB fixed-length segment, add a BTSACB DD statement to the JCL or the ALLOC statement in the CLIST. BTSACB DD statement specifies ACBLIB including the member with compressed segment or fixed-length segment. If you do not specify a BTSACB DD statement in the JCL or the ALLOC statement in the CLIST, the length after compression or the length "fixed-length + 2 bytes" is put on the BTSOUT or the TSO screen when the application program issues a DEDB ISRT call.

Applications that trigger secondary transactions

A primary transaction is a transaction that is scheduled by reading BTSIN or input from the MFS screen. A secondary transaction is a transaction that is scheduled by the primary transaction.

If your application triggers secondary transactions, the following considerations apply.

A secondary transaction is defined in precisely the same manner as a primary transaction. The process limit count (PLC) for a secondary transaction is treated as follows:

- If PLC=0, the secondary transaction is not processed. The message insert is traced, but the message is not queued.
- If PLC>0, the secondary transaction is to be processed on completion of the primary transaction. However, all messages inserted as secondary transactions are processed (regardless of the specified PLC) before the next primary transaction is considered.

The SCNDTRX= parameter of the ./E command specifies the order in which the secondary transaction is scheduled. The following subsections explain the processing order of secondary transactions.

Subsections:

- [“When SCNDTRX=CONT is specified” on page 119](#)
- [“When SCNDTRX=NOCONT is specified” on page 120](#)
- [“Secondary transactions in conversational processing” on page 120](#)

When SCNDTRX=CONT is specified

Assume that ./E SCNDTRX=CONT is specified (specifies to schedule secondary transaction immediately after a primary transaction) and that the input stream includes the following specifications:

```
./T TC=PRIMARY  MBR=PROGA  PLC=5
./T TC=SECOND1  MBR=PROGB  PLC=0
./T TC=SECOND2  MBR=PROGC  PLC=2
./T TC=SECOND3  MBR=PROGD  PLC=5
PRIMARY  MESSAGE1
PRIMARY  MESSAGE2
```

The first message to be processed is PRIMARY MESSAGE1.

Further assume that in processing the PRIMARY transaction, PROGA inserts message segments to an alternate PCB in the following sequence:

1. SECOND2 MESSAGE1
2. SECOND2 MESSAGE2
3. SECOND3 MESSAGE1
4. SECOND1 MESSAGE1
5. SECOND2 MESSAGE3
6. Response to I/O PCB.

At the next GU call by PROGA, IMS Batch Terminal Simulator:

1. Ends PROGA, even though PLC is not attained, because there are messages to be processed on the secondary transaction queue.
2. Schedules PROGC to process:
SECOND2 MESSAGE1
SECOND2 MESSAGE2
SECOND2 MESSAGE3
3. Ends PROGC.
4. Schedules PROGD to process:
SECOND3 MESSAGE1
5. Ends PROGD.
6. Schedules PROGA to process:
PRIMARY MESSAGE2

Note: PROGB is never scheduled, because PLC=0.

When SCNDTRX=NOCONT is specified

Assume that ./E SCNDTRX=NOCONT is specified (specifies that a secondary transaction is not to be scheduled immediately after a primary transaction) and that the input stream includes the following specifications:

```
./T TC=PRIMARY  MBR=PROGA  PLC=5
./T TC=SECOND1  MBR=PROGB  PLC=0
./T TC=SECOND2  MBR=PROGC  PLC=2
./T TC=SECOND3  MBR=PROGD  PLC=5
./E SCNDTRX=NOCONT
PRIMARY  MESSAGE1
PRIMARY  MESSAGE2
```

The secondary transaction is scheduled as follows:

The first message to be processed is PRIMARY MESSAGE1. Further assume that in processing the PRIMARY transaction, PROGA inserts message segments into an alternate PCB in the following sequence:

1. SECOND2 MESSAGE1
2. SECOND2 MESSAGE2
3. SECOND3 MESSAGE1
4. SECOND1 MESSAGE1
5. SECOND2 MESSAGE3
6. Response to I/O PCB

At the next GU call by PROGA, IMS Batch Terminal Simulator:

1. Restarts PROGA to get the second message (PRIMARY MESSAGE2), because SCNDTRX=NOCONT is specified on the ./E command. The second message to be processed is PRIMARY MESSAGE2.
 - a. SECOND2 MESSAGE1
 - b. SECOND2 MESSAGE2
 - c. SECOND3 MESSAGE1
 - d. SECOND1 MESSAGE1
 - e. SECOND2 MESSAGE3
 - f. Response to I/O PCB
2. Ends PROGA, because there are no input messages for PROGA.
3. Schedules PROGC to process:
 - SECOND2 MESSAGE1 (from PRIMARY MESSAGE1)
 - SECOND2 MESSAGE2 (from PRIMARY MESSAGE1)
 - SECOND2 MESSAGE3 (from PRIMARY MESSAGE1)
 - SECOND2 MESSAGE1 (from PRIMARY MESSAGE2)
 - SECOND2 MESSAGE2 (from PRIMARY MESSAGE2)
 - SECOND2 MESSAGE3 (from PRIMARY MESSAGE2)
4. Ends PROGC.
5. Schedules PROGD to process:
 - SECOND3 MESSAGE1 (from PRIMARY MESSAGE1)
 - SECOND3 MESSAGE1 (from PRIMARY MESSAGE2)
6. Ends PROGD.

Note: PROGB is never scheduled, because PLC=0.

Secondary transactions in conversational processing

Conversational processing operates differently. An application program in conversational mode continues until one of the following conditions is met:

- An IOPCB ISRT call of the SPA is issued, where the transaction code in the SPA is not the current transaction code.
- An alternate PCB ISRT call of the SPA is issued.
- The conversation is stopped by an /EXIT command.

When an /EXIT command is entered, IMS Batch Terminal Simulator ends the current transaction but also discards any secondary transactions that might have been previously inserted (ISRT) to the message queue.

Related reference

[./E command](#)

Use the ./E command to define the environment that the transaction is executed in.

3270 applications

If the application to be tested is a 3270 application, the following considerations apply.

The IMS Message Format Service offers a field-oriented interface with the terminals. A utility program generates the control blocks needed for converting input fields to message segments and output segments to terminal fields. All the specifications, connections, and mapping operations in this interface should, for practical reasons, be tested offline. This is where IMS Batch Terminal Simulator is especially useful.

It is recommended that 3270 applications be tested in parallel as much as possible without losing control over the testing environment. This parallelism is attained with IMS Batch Terminal Simulator primarily in two ways:

- One approach is to construct input segments (that is, simulator statements) in the layout expected by the application program and use the unformatted-mode simulator statements (as though the terminal being simulated were a 2740, for example). At the same time, 3270 formatted-mode simulator statements can be used in other portions of the simulation to test the validity of the format control block specifications.
- A second approach is to divide 3270 format control block testing into subunits, each one beginning with a /FORMAT command. With this approach, the body of each subunit is a sequence of statements that are error-free, and lead to the display of one specific page. Thus, one function (such as data entry, function key operation, or selector pen detection) is tested per subunit. Finally, the PAX keyword can display all output.

Consider the application in the following figure. The terminal operator is supposed to start the terminal session by requesting the SELECT format. The operator then selects the part, customer, or sales information application by entering data in the appropriate fields on the screen. In response, the application program selects the correct output format (PART, CUSTOM, or SALES) to guide the operator in subsequent data entries. When the customer application is chosen, output can be shown in (and input accepted from) either of two different formats (CUSTOM1 or CUSTOM2), selected by data contents or control block chaining.

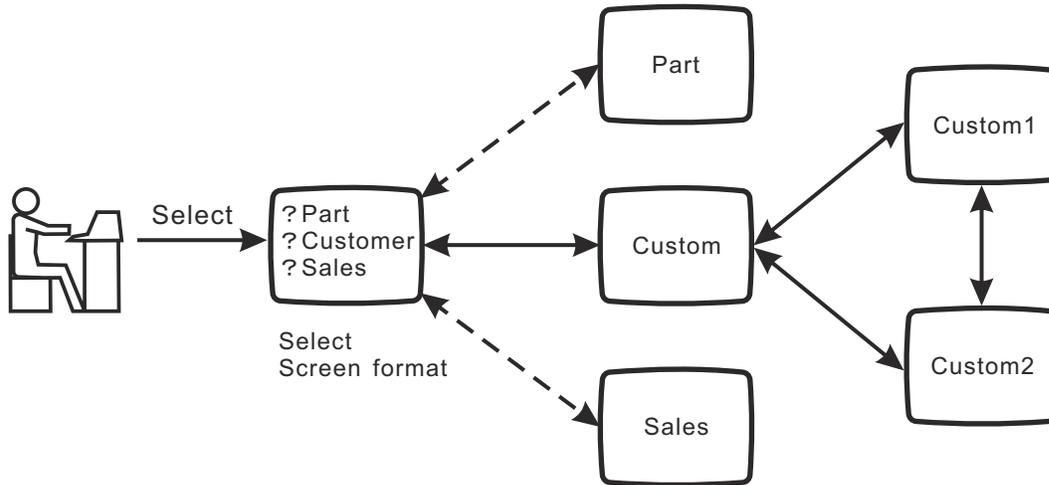


Figure 11. Application for testing example

Two input streams in this application environment might be:

```
./D LTERM=MODEL1 TYPE=3270-A1 SIZE=(12,40)
/FORMAT SELECT
PAX $
./D LTERM=MODEL2 TYPE=3270-A2 SIZE=(24,80)
/FOR SELECT
PAX $
```

This example might be the first run to display the initial format for both models of the 3270 Display Station. At a later stage, the input stream might be more like the following:

```
./D LTERM=UNFORM
PART AB9012C3F BRASS SCREW $
CUSTOM JOHN DOE & CO. $
.
.
.
(input segments to application programs)
.
.
./D LTERM=MODEL2 TYPE=3270-A2 SIZE=(24,80)
/FORMAT SELECT
L03C27 'AB9012C3F' L05C30 'BRASS SCREW' PFK1 $
PFK2 $
L04C37 '=+4' ENTER $ (page to a particular application page)
L23C01 PEN $
PAX $
./D LTERM=MODEL1 TYPE=3270-A1 SIZE=(12,40)
/FORMAT SELECT
.
.
.
```

(Repeat the 3270 formatted-mode simulator sequence to test the formats for a 3270 Model 1.)

As the application testing progresses, additional sections are introduced as needed, using an established format as the basis.

APPC applications

If the application to be tested is an APPC (Advanced Program-to Program Communications) application, the following requirements and considerations apply.

Subsections:

- [“Requirements for simulator command” on page 123](#)
- [“Supported APIs for APPC conversations” on page 123](#)

- [“Limitations for the APPC support function” on page 123](#)

Requirements for simulator command

To test APPC applications with IMS Batch Terminal Simulator, specify the DDOF=LU62 option of the ./D command.

To initialize the LU 6.2 descriptors on IMS Batch Terminal Simulator, specify the TYPE=LU62 option of the ./T command.

Related reading: See [“Simulator commands” on page 187](#) to specify the ./D command and the ./T command.

Supported APIs for APPC conversations

IMS Batch Terminal Simulator supports the following implicit APIs for APPC conversations:

- The CHNG call with LU 6.2 options in the options list.

IMS Batch Terminal Simulator checks the options list format and traces it. If the options list is available, IMS Batch Terminal Simulator sets the destination name in the alternate PCB to 'DFSLU62 '.

- The CHNG call to the LU 6.2 descriptor.

IMS Batch Terminal Simulator checks the destination name.

- The ISRT call to the LU 6.2 destination.

IMS Batch Terminal Simulator checks that a SETO call with the DEALLOCATE_ABEND option has been previously issued for the destination. Any new APPC conversation is not processed. The message insert is traced, but the message is not queued.

- The SETO call with LU 6.2 options in the options list.

IMS Batch Terminal Simulator checks the options list format and traces it. When the DEALLOCATE_ABEND option is specified, IMS Batch Terminal Simulator checks the following occurrences:

- SETO call with the DEALLOCATE_ABEND option was previously issued for the destination
- There were previous inserts for the destination

- The INQY call with null subfunction for LU 6.2 destinations.

IMS Batch Terminal Simulator returns a character string data in the I/O area.

Limitations for the APPC support function

IMS Batch Terminal Simulator has the following limitations for the APPC conversations:

- IMS Batch Terminal Simulator does not really talk to the LU 6.2 device.
- IMS Batch Terminal Simulator does not support the modified DL/I application program and the CPI communication application program.
- IMS Batch Terminal Simulator does not use DFS62DTx member in IMS.PROCLIB.
- IMS Batch Terminal Simulator does not support the IMS system service DFSAPPC.

Chapter 11. Including BTS optional user-written exit routines

IMS Batch Terminal Simulator provides several user-written exit routines.

Saving registers and the save area chain

IMS Batch Terminal Simulator exit routines need to save registers in the save area that is pointed to by Register 13. This save area is provided at entry. Before processing any data, the exit routine must:

1. Get the forward save area
2. Set the save area address at offset 4 of the forward save area
3. Set the forward save area address at offset 8 of the save area
4. If necessary, make Register 13 point to the forward save area

Before returning control to IMS Batch Terminal Simulator, the routine must step back to the original save area and restore IMS Batch Terminal Simulator registers.

The following topics describe how to include IMS Batch Terminal Simulator user-written exit routines.

Topics:

- [“Applying output formatting modules” on page 125](#)
- [“Applying an attention exit routine” on page 128](#)
- [“User exit interface for SAIL/ESA” on page 128](#)
- [“User exit \(BTSTSSE0\) interface for TSS call” on page 130](#)
- [“Application front-end routine” on page 130](#)
- [“Extended user exit \(BTSEEXIT\) interface” on page 131](#)

Applying output formatting modules

IMS Batch Terminal Simulator is structured such that user-written output formatting modules can be appended to it. Each ISRT or PURG call of a message or message segment to the I/O PCB that is made by an application program can be passed to a user-written routine for formatting purposes.

About this task

This topic contains Product-sensitive Programming Interface information.

These modules can be used to:

- Reformat the message or message segment and print it themselves by using the BTS common writer module
- Reformat the message in place and let IMS Batch Terminal Simulator print it (IMS Batch Terminal Simulator interprets and prints non-printable characters)

Output formatting can be done only for the I/O PCB. IMS Batch Terminal Simulator uses standard OS/390® and z/OS linkage conventions in its own modules, and any appended modules are expected to do the same. Therefore, when a user-written module receives control, the register contents are as shown in the following table.

Table 25. Register outputs of IMS Batch Terminal Simulator

Register	Content
0	The user control switch in the last-order byte

Table 25. Register outputs of IMS Batch Terminal Simulator (continued)

Register	Content
1	The address of a doubleword parameter list
13	The save area address of the caller
14	The return point address of the caller
15	The entry point address of the user-written module (The high-order byte of Register 15 contains a user switch when running in 24 bit mode.)

The relationship between the address (@) in Register 1 and the output message or message segment is illustrated in the following figure.

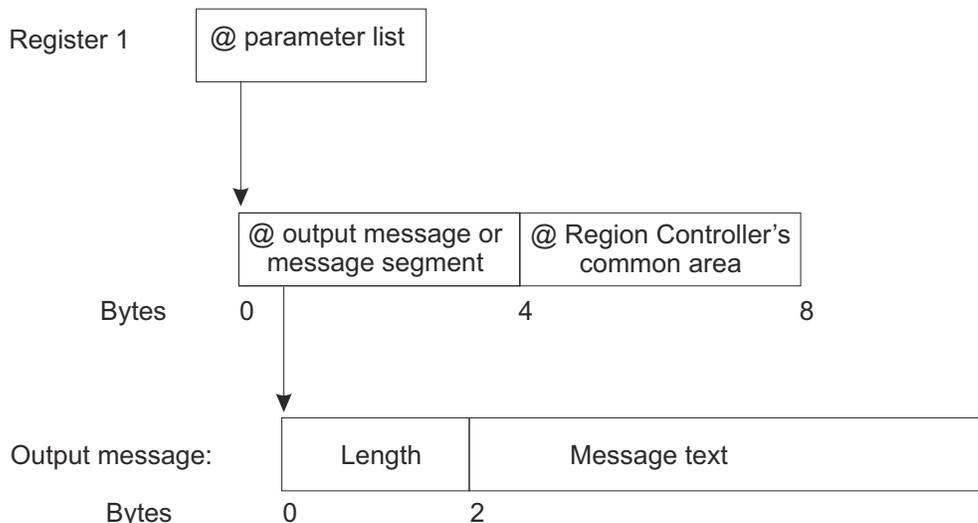


Figure 12. Relationship between Register 1 address and message segment

If no additional printing is requested, set Register 15 to zero when returning to the caller. When the caller is to print the message, set Register 15 to 4 before returning to the caller.

When the BTS common writer is used, the first 2 bytes of the output must be the binary length of the output text (not including the 2-byte length of the length field itself), and all output is blocked according to the logical record length specified in the BTSOUT DD statement. The actual assembler coding necessary to use the BTS common writer module is:

```

BTSD    DSECTS=COMMON
MVC     CAPRINTA,=A(output record) (address of output)
L       15,CAWRT (contains the address of the
                BTS common writer module)
BALR    14,15
  
```

A user-written output formatting module must reside in the BTS region controller (BTSRC000) load module. It must also be known to the BTSDFTBL table, which has information of output formatting modules. BTSDFTBL table is created by using IMS Batch Terminal Simulator macro BTSDEFDF.

Note: The sample JCL for including the BTSDFTBL table in IMS Batch Terminal Simulator is supplied as the member BTSDFTBL and is loaded into the BTS JCL library (SBTSJCL0) during the installation.

Procedure

To append a user-written output formatting module, follow these steps:

1. Create the BTSDFTBL table module by coding the BTSDEFDF macro statements.

The BTSDEFDF macro has three statement types (as indicated by the TYPE= parameter). Code one TYPE=INITIAL statement to start the parameter list build, and then code many TYPE=DEF statements as necessary. Finally, code a TYPE=FINAL statement to end the list. The TYPE=DEF statement has additional parameters.

a) Code the TYP=INITIAL statement.

This statement is a required statement that indicates the start of a parameter list build. No other parameters are valid on a TYPE=INITIAL statement. The format of this statement is:

```
BTSDEFDF TYPE=INITIAL
```

b) Code the TYPE=DEF statement.

This statement defines information of an output formatting module. One TYPE=DEF statement is required for each DDOF operand. The format of this statement is:

```
BTSDEFDF TYPE=DEF,DDOF=ddofop,CTLSW=ctlsw,MODULE=modname,MODE=mode
```

where:

DDOF=

Specifies the DDOF operand. Specify all the valid DDOF keyword operands that can be specified to indicate that the particular user-written module is to be used. Specify one operand per TYPE=DEF statement.

CTLSW=

Specifies the user control switch that is to be passed to the user-written output formatting module for the DDOF operand.

MODULE=

Specifies the entry name of the user-written output formatting module for the DDOF operand.

MODE=

Specifies the running mode (31 for 31 bit mode or 24 for 24 bit mode) of the user-written output formatting module for the DDOF operand. The default value is 24.

c) Code the TYPE=FINAL statement.

This statement indicates the end of a parameter list build and is required. No other parameters are valid on a TYPE=FINAL statement. The format of this statement is:

```
BTSDEFDF TYPE=FINAL
```

The following example shows the entry of the BTS-supplied 2740 output formatting module in the table:

```
BTSDFDFTBL CSECT
BTSDEFDF TYPE=INITIAL
BTSDEFDF TYPE=DEF,DDOF=274011,CTLSW=4,MODULE=BTS2740F,MODE=24
BTSDEFDF TYPE=DEF,DDOF=274012,CTLSW=8,MODULE=BTS2740F,MODE=24
BTSDEFDF TYPE=DEF,DDOF=274010,CTLSW=0,MODULE=BTS2740F,MODE=24
BTSDEFDF TYPE=FINAL
END
```

The user control switch is passed to the formatting module in the high-order byte of Register 15 and in the last-order byte of Register 0 when running in 24 bit mode (AMODE=24 and RMODE=24). You can use whichever register you like to get the user control switch.

The user control switch is passed to the formatting module in the last-order byte of Register 0 when running in 31 bit mode (AMODE=31 and RMODE=ANY).

2. Code the output formatting module by using standard OS linkage conventions and placing a return code of 0 or 4 in register 15 upon return to IMS Batch Terminal Simulator.
3. Assemble the BTS CSECT BTSDFTBL and the user-written module, and link-edit them with BTSRC000.

Applying an attention exit routine

If you want to stop processing, or if you see a message requesting information you do not have, press the attention key. Control is passed to the ATTENTION EXIT routine for interrupting or ending a process.

About this task

The BTS ATTENTION EXIT routine takes any of the following actions:

- Receive the control in the 31 bit addressing mode
- Put the messages BTS0102W and BTS0103A on TSO screen
- Wait for a response from the TSO terminal operator, who must press Enter or the PA1 key

A user-written ATTENTION EXIT routine must have the CSECT name of BTSTSOAT and reside in the BTS TSO controller (BTSTSOST) load module.

IMS Batch Terminal Simulator allows you to replace its ATTENTION EXIT routine with one of your own. Your ATTENTION EXIT routine receives control in the 31 bit addressing mode.

Note: Sample JCL for replacing the BTS ATTENTION EXIT routine with your own is supplied as member BTSTSOAT in the BTS JCL library.

Procedure

To replace the BTS ATTENTION EXIT routine with your own, follow these steps:

1. Code your ATTENTION EXIT routine.
2. Assemble your routine and link-edit it with BTSTSOST.
3. Set the ATTNEXIT=Y parameter into the CLIST.

User exit interface for SAIL/ESA

IMS Batch Terminal Simulator provides the user exit interface for SAIL/ESA.

If you use the interface for SAIL/ESA, you must include the user exits in BTS load module BTSSRC000. IMS Batch Terminal Simulator provides the user work area (4 K bytes) for the user exit.

Subsections:

- [“BTSSAILO” on page 128](#)
- [“BTSSAIL1” on page 129](#)
- [“BTSSAIL2” on page 129](#)

BTSSAILO

This user exit is called from IMS Batch Terminal Simulator before the trace information of ISRT call is put on the BTSOUT.

Registers at entry to BTSSAILO

R1

The address of A(parameter list)

Parameter list includes:

- A(I/O area)
- A(work area)
- A(length of work area)

R13

The save area address of the caller

R14

The return point address of the caller

R15

The entry point address of BTSSAILO

Registers on return from BTSSAILO

None

BTSSAIL1

This user exit is called from IMS Batch Terminal Simulator before the input card (BTSIN) is read by IMS Batch Terminal Simulator.

Registers at entry to BTSSAIL1**R1**

The address of A(parameter list)

Parameter list includes:

A(IMS Batch Terminal Simulator input card)

A(work area)

A(length of work area)

R13

The save area address of the caller

R14

The return point address of the caller

R15

The entry point address of BTSSAIL1

Registers on return from BTSSAIL1**R15**

A return code in which:

0

Indicates that IMS Batch Terminal Simulator must get a card input after returning to BTSSAIL1.

4

Indicates that IMS Batch Terminal Simulator does not need to get a card input after returning to BTSSAIL1. Instead of IMS Batch Terminal Simulator, BTSSAIL1 has already read the input data.

BTSSAIL2

This user exit is called from IMS Batch Terminal Simulator when an ISRT call for ALT PCB is issued. IMS Batch Terminal Simulator provides BTSEXIT0 output data set for this user exit. BTSSAIL2 can put the data into BTSEXIT0.

Registers at Entry to BTSSAIL2**R1**

The address of A(parameter list)

Parameter list includes:

A(I/O area)

A(work area)

A(length of work area)

A(BTSEXIT0 data set)

R13

The save area address of the caller

R14

The return point address of the caller

R15

The entry point address of BTSSAIL2

Registers on Return from BTSSAIL2

None

User exit (BTSTSSE0) interface for TSS call

When a TSS call is issued, IMS Batch Terminal Simulator calls the user exit BTSTSSE0 instead of BTSCINT0.

If BTSTSSE0 is used for TSS call, you must include its user exit in the load module BTSPC000.

BTSTSSE0**Registers at Entry to BTSTSSE0****R1**

The address of A(PCB)

R3

The address of (TSS call parameter list)

Parameter list includes:

A(C'TSS ')

A(PCB)

:

:

R9

The address of the BTS COMMON DSECT base

R13

The save area address of the caller

R14

The return point address of the caller

R15

The entry point address of BTSTSSE0

Registers on return from BTSTSSE0

None

Application front-end routine

IMS Batch Terminal Simulator supports application front-end routines. Use of an application front-end routine is designated by the ./E simulator command.

If you specify the ./E APPLFE command, IMS Batch Terminal Simulator performs the following functions:

1. When the first transaction starts (that is, when a BTS0006I message is issued), IMS Batch Terminal Simulator loads the application front-end routine, calls the application front-end routine for initialization processing, then calls the application front-end routine.
2. When the next transaction starts, IMS Batch Terminal Simulator calls the application front-end routine.
3. Before the IMS Batch Terminal Simulator job ends, IMS Batch Terminal Simulator calls the application front-end routine for shutdown processing.

For details about the application front-end routine, see *IMS System Definition*.

Extended user exit (BTSEEXIT) interface

IMS Batch Terminal Simulator provides the extended user exit (BTSEEXIT) interface.

To use the BTSEEXIT interface, you must prepare a user-written exit routine. Two methods are supported:

- Prepare a user-written exit routine with the CSECT name of BTSEEXIT and include the exit routine in the BTS load module BTSRC000.
- Prepare a user-written exit routine with any CSECT name and create a load module member named BTSEEXIT with the exit routine in a STEPLIB library.

The BTSEEXIT interface is dynamically loaded and used from the STEPLIB library if it is not included in the BTS load module BTSRC000. If it is included in BTSRC000, the BTSEEXIT interface included in BTSRC000 is used without dynamically loading a member named BTSEEXIT from the STEPLIB library.

The AMODE and RMODE of the load module member named BTSEEXIT must be 24 and 24 when one of the following exit numbers is used; 5, 6, 8, or 9. Otherwise, AMODE and RMODE can be 31 and ANY.

Exit type

When the BTSEEXIT interface is used, IMS Batch Terminal Simulator calls the following initialization and termination exits. Other exits are called only when the corresponding bits are on at offset 1 and 2 of the parameter area.

Exit number	Exit name	Description
0	Initialization	Always called once, at the beginning, for initialization
1	Termination	Always called once, at the end, for termination
2	BTSIN	Called for BTSIN processing
3	BTSOUT	Called for BTSOUT processing
4	TPUT	Called for TPUT processing
5	Pre-DLI call	Called for pre-DLI call processing
6	Post-DLI call	Called for post-DLI call processing
7	PSB post-load	Called for PSB post-load processing
8	Pre-other call	Called for pre-other call processing
9	Post-other call	Called for post-other call processing
10	Pre-transaction call	Called for pre-transaction call processing
11	Post-transaction call	Called for post-transaction call processing
12	Pre-application call	Called for pre-application call processing

Register convention

Registers at entry to BTSEEXIT

R0

Exit number

R1

The address of the exit parameter area

R13

The save area address of the caller

R14

The return point address of the caller

R15

The entry point address of BTSEEXIT

Registers on return from BTSEEXIT**R15**

Return code

Other registers must be restored on return.

IMS Batch Terminal Simulator ends abnormally with abend code 4074 when the exit program returns nonzero return code in register 15.

Addressing mode

When IMS Batch Terminal Simulator intercepts a DLI call or any other call from an application program, IMS Batch Terminal Simulator calls exits in 24-bit or 31-bit addressing mode for exit numbers 5, 6, 8, and 9. For exits with other exit numbers, IMS Batch Terminal Simulator calls them in 31-bit addressing mode.

The exit program must return control in the same addressing mode in which it was invoked.

Exit parameter area

IMS Batch Terminal Simulator prepares the following 32-byte parameter area below the 16 MB line and passes it to the initialization exit. This area contains zeros except for the first two bits.

IMS Batch Terminal Simulator passes the same area to subsequent exits.

Offset	Byte length	Description
0	1	<p>IMS Batch Terminal Simulator sets the following values to the first two bits ('XX.....'B) according to the running mode of IMS Batch Terminal Simulator.</p> <ul style="list-style-type: none"> • '00'B: DLI • '01'B: DBB • '10'B: BMP • '11'B: JBP <p>Other six bits (third bit to eighth bit) are reserved.</p>
1	1	<ul style="list-style-type: none"> • '10000000'B: BTSIN exit request • '01000000'B: BTSOUT exit request • '00100000'B: TPUT exit request • '00010000'B: Pre-DLI call exit request • '00001000'B: Post-DLI call exit request • '00000100'B: PSB post-load exit request • '00000010'B: Pre-other call exit request • '00000001'B: Post-other call exit request <p>All request bits can be switched on or off by any exit.</p>
2	1	<ul style="list-style-type: none"> • '10000000'B: Pre-transaction call exit request • '01000000'B: Post-transaction call exit request • '00100000'B: Pre-application call exit request <p>All request bits can be switched on or off by any exit.</p> <p>Other five bits (fourth bit to eighth bit) are reserved.</p>

Offset	Byte length	Description
3	1	'00000001'B: BTSOUT exit call repeat request When both this bit and the BTSOUT request bit (at offset 1) are on, IMS Batch Terminal Simulator repeatedly calls the BTSOUT exit until the BTSOUT exit call repeat request bit is turned off by an exit. This request bit can be switched on or off by any exit other than the TPUT exit. Other seven bits (first bit to seventh bit) are reserved.
4	4	Used only for BTSEEXIT. This field is never used by IMS Batch Terminal Simulator.
8	12	Used by each exit.
20	12	Reserved.

Usage notes:

- Unused areas and bits are reserved for future use and must not be updated.
- The data pointed to by an address field in the parameter list at each exit entry must not be updated.

Initialization exit (exit number 0)

IMS Batch Terminal Simulator calls this exit once at the beginning immediately after issuing message BTS0007I.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.

The exit program must set the corresponding bits of the required exits, which are at offset 1 and 2 of the parameter area, to on.

Termination exit (exit number 1)

IMS Batch Terminal Simulator calls this exit once at the end immediately before issuing message BTS0005I.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.

BTSIN exit (exit number 2)

IMS Batch Terminal Simulator calls this exit immediately before processing a concatenated command or a statement from BTSIN DD.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the area containing the leading 2-byte length field and the original data of the concatenated command or the statement from BTSIN DD.

Offset	Byte length	Description
12	4	<p>IMS Batch Terminal Simulator clears this field upon calling the exit. If this field is zero on return of the exit, IMS Batch Terminal Simulator processes the original data without change.</p> <p>If the exit program changes the original data, the program must prepare an area that contains the leading 2-byte length field and the changed data, and set the address to this field. In this case, IMS Batch Terminal Simulator processes the changed data instead of the original data.</p>

BTSOUT exit (exit number 3)

IMS Batch Terminal Simulator calls this exit immediately before generating outputs to BTSOUT DD.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the area containing the leading 2-byte length field and the original data for the output for BTSOUT DD.
12	4	<p>IMS Batch Terminal Simulator clears this field upon calling the exit. If this field is zero on return of the exit, IMS Batch Terminal Simulator processes the original data without change.</p> <p>If the exit program changes the original data, the program must prepare an area that contains the leading 2-byte length field and the changed data, and set the address to this field. In this case, IMS Batch Terminal Simulator processes the changed data instead of the original data.</p>

When the BTSOUT exit call repeat request ('00000001'B at offset 3 in the parameter area) is on, IMS Batch Terminal Simulator repeatedly calls this BTSOUT exit with the original data address that points to the leading 2-byte length field ('0002'X) and 2 blank characters (blank carriage return and blank). An exit program can change the original data as described in the table. If the exit program turns on the bit, it must set the bit off at some point to avoid loops.

TPUT exit (exit number 4)

IMS Batch Terminal Simulator calls this exit immediately before it transmits outputs to a TSO terminal by TPUT.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the original data that the TPUT exit transmits.

Offset	Byte length	Description
12	4	IMS Batch Terminal Simulator clears this field upon calling the exit. If this field is zero on return of the exit, IMS Batch Terminal Simulator processes the original data without change. If the exit program changes the original data, the program must prepare an area for the changed data and set the address to this field. In this case, IMS Batch Terminal Simulator processes the changed data instead of the original data.
16	2	Length of the original data.
18	2	Length of the changed data.

Pre-DLI call exit (exit number 5)

IMS Batch Terminal Simulator calls this exit immediately before processing a DLI call that IMS Batch Terminal Simulator intercepted.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the parameter list of the original DLI call.
12	4	IMS Batch Terminal Simulator clears this field upon calling the exit. If this field is zero on return of the exit, IMS Batch Terminal Simulator processes the original DLI call without change. If the exit program changes the parameter list of the original DLI call, the program must prepare an area for the changed DLI call parameter list in the same language format (LILAN) as the original language format and set the address to this field. In this case, IMS Batch Terminal Simulator processes the changed DLI call instead of the original DLI call. The area for the changed parameter list must be below the 16 MB line when this exit is called in 24-bit addressing mode.
16	1	LIFLG1 field in PARMBLK macro of IMS.
17	1	LILAN field in PARMBLK macro of IMS.
18	1	'10000000'B: EXEC DLI call indicator If this bit is on, the call indicates an EXEC DLI call. Other seven bits (second bit to eighth bit) are reserved.

Pre-DLI call exits and post-DLI call exits are called in pairs. Generally the post-DLI call exit is called after the pre-DLI call exit as follows:

1. Pre-DLI call exit with LIDEXIT bit off
2. Post-DLI call exit with LIDEXIT bit off
3. Pre-DLI call exit with LIDEXIT bit off
4. Post-DLI call exit with LIDEXIT bit off

IMS Batch Terminal Simulator supports segment data user exit that corresponds to the LIDEXIT bit in the LIFLG1 field of the PARMBLK macro. When a segment data user exit is enabled, additional DLI calls can be made by the segment data user exit program that is triggered by a DLI call. Pairs of pre-DLI call and post-DLI call exits with the LIDEXIT bit on can be nested inside a pair of pre-DLI call and post-DLI call exits with the LIDEXIT bit off as follows:

1. Pre-DLI call exit with LIDEXIT bit off
2. Pre-DLI call exit with LIDEXIT bit on
3. Post-DLI call exit with LIDEXIT bit on
4. Pre-DLI call exit with LIDEXIT bit on
5. Post-DLI call exit with LIDEXIT bit on
6. Post-DLI call exit with LIDEXIT bit off

Post-DLI call exit (exit number 6)

IMS Batch Terminal Simulator calls this exit immediately before returning to the application program that issued the DLI call.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.

PSB post-load exit (exit number 7)

IMS Batch Terminal Simulator calls this exit immediately after loading a PSB.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the original PCBLIST returned from IMS.
12	4	IMS Batch Terminal Simulator clears this field upon calling the exit. If this field is zero on return of the exit, IMS Batch Terminal Simulator uses the original PCBLIST without change. If the exit program changes the original PCBLIST, the program must prepare an area for the changed PCBLIST and set the address to this field. In this case, IMS Batch Terminal Simulator uses the changed PCBLIST instead of the original PCBLIST.
16	4	Address that points to the 8-byte PSB name.

Pre-other call exit (exit number 8)

IMS Batch Terminal Simulator calls this exit immediately before processing a non-DLI call that IMS Batch Terminal Simulator intercepted.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the 4-byte function code. <ul style="list-style-type: none"> • 'SQL ' or 'IFC ': Db2 call • 'ELI ': MQI call • Other: Any other call
12	1	LIFLG1 field in PARMBLK macro of IMS.

Pre-other call exits and post-other call exits are called in pairs. Generally the post-other call exit is called after the pre-other call exit as follows:

1. Pre-other call exit with LIPCC50 bit off
2. Post-other call exit with LIPCC50 bit off
3. Pre-other call exit with LIPCC50 bit off
4. Post-other call exit with LIPCC50 bit off

IMS Batch Terminal Simulator supports Db2 change data capture user exit that corresponds to the LIPCC50 bit in the LIFLG1 field of the PARMBLK macro. When a Db2 change data capture user exit is enabled, additional Db2 calls can be made by the Db2 change capture user exit program that is triggered by a Db2 call. Pairs of pre-other Db2 call and post-other Db2 call exits with the LIPCC50 bit on can be nested inside a pair of pre-other Db2 call and post-other Db2 call exits with the LIPCC50 bit off as follows:

1. Pre-other Db2 call exit with LIPCC50 bit off
2. Pre-other Db2 call exit with LIPCC50 bit on
3. Post-other Db2 call exit with LIPCC50 bit on
4. Pre-other Db2 call exit with LIPCC50 bit on
5. Post-other Db2 call exit with LIPCC50 bit on
6. Post-other Db2 call exit with LIPCC50 bit off

Post-other call exit (exit number 9)

IMS Batch Terminal Simulator calls this exit immediately before returning control to the application program that issued the non-DLI call.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.

Pre-transaction call exit (exit number 10)

IMS Batch Terminal Simulator calls this exit immediately before issuing message BTS0006I.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the 8-byte transaction code name.
12	4	Address that points to the 8-byte load module name of the application program.

Post-transaction call exit (exit number 11)

IMS Batch Terminal Simulator calls this exit immediately before issuing message BTS0020I.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.

Pre-application call exit (exit number 12)

IMS Batch Terminal Simulator calls this exit immediately before calling an application program.

Offset	Byte length	Description
0	8	Refers to the exit parameter area.
8	4	Address that points to the PCBLIST to be passed to the application program.

Chapter 12. Interpreting the simulation output

The BTSOUT data set contains the output listing from IMS Batch Terminal Simulator run. The output listing includes information about the processed transactions.

Topics:

- [“Analyzing the output listing” on page 139](#)
- [“DL/I call trace listing” on page 142](#)
- [“Db2 call trace listing” on page 144](#)
- [“MQI call trace listing” on page 148](#)
- [“Program analysis reports” on page 149](#)

Analyzing the output listing

The BTSOUT data set contains precise information about the transactions that were processed. By analyzing the information in the BTSOUT data set, you can determine if the application ran as expected.

Procedure

1. Browse the BTSOUT data set and locate the BTS0007I message.

The BTS0007I message indicates the time and date when the IMS Batch Terminal Simulator job started and the version of IMS that was used. After the BTS0007I message, the simulator commands that are specified in the BTSIN data set or entered through TSO are listed with BTS0002I messages.

```
BTS0007I BTS V4R1 SIMULATION STARTED. TIME=00:30:24, DATE=2021.013, IMS=14.1.
BTS0002I INPUT RECORD: ./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
BTS0002I INPUT RECORD: ./T TC=DSPINV LANG=CBL MBR=DFSSAM03
BTS0002I INPUT RECORD: ./T TC=ADDPART LANG=CBL MBR=DFSSAM04
BTS0002I INPUT RECORD: ./T TC=ADDINV LANG=CBL MBR=DFSSAM04
BTS0002I INPUT RECORD: ./T TC=DLETINV LANG=CBL MBR=DFSSAM04
BTS0002I INPUT RECORD: ./T TC=DLETPART LANG=CBL MBR=DFSSAM04
BTS0002I INPUT RECORD: ./T TC=CLOSE LANG=CBL MBR=DFSSAM05
BTS0002I INPUT RECORD: ./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
BTS0002I INPUT RECORD: ./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
```

Figure 13. BTSOUT: Echo of the simulator commands

2. If you have specified a /FORMAT command (requests to format the screen image) in BTSIN or through the TSO session, locate the BTS0002I message that contains the /FORMAT command specification.

```
BTS0002I INPUT RECORD: /FORMAT SAMOUX
```

Figure 14. BTSOUT: /FORMAT command

The formatted screen image is recorded after the BTS0002I message.

Note: If no /FORMAT command is specified during the job, the formatted screen images are not printed in BTSOUT. Continue with Step [“3” on page 141](#).

Table 26. 3270 extended attribute display

Color	Highlighting	Programmed symbols
Blue: X'80'	blink: X'40'	user defined
Red: X'10'	reverse video: X'80'	
Pink: X'18'	underscore: X'C0'	
Green: X'20'		
Turquoise: X'28'		
Yellow: X'30'		
White: X'38'		

3. Locate the BTS0002I message that contains the first simulator statement.

When IMS Batch Terminal Simulator encounters the first simulator statement that specifies the transaction code and transaction message, or an input that was made through TSO terminal (when FSS is used), IMS Batch Terminal Simulator prints the simulator statement in a BTS0002I message.

```
BTS0002I INPUT RECORD: PART AN960C10$
```

Figure 16. BTSOUT: Simulator statement

When an input is made though the TSO terminal while FSS is active, IMS Batch Terminal Simulator converts the input to formatted-mode simulator statement and prints the formatted-mode simulator statement as follows:

```
BTS0002I INPUT RECORD: L01C09 'AN960C10' L02C05 'PART ' ENTER $
```

Figure 17. BTSOUT: Simulator statement (converted simulator statement)

4. Locate the BTS0006I message.

IMS Batch Terminal Simulator prints message BTS0006I to indicate the start of a transaction. Detailed information about the input data is included after the message.

The following figure shows the output listing when the input simulator statement is PART AN960C10\$.

```
BTS0006I TRANSACTION STARTED: PART
MBR=DFSSAM02 PSB=DFSSAM02 EDIT= SPA=0 PLC=1 LANG=CBL TYPE=MSG
**** MSG CALL- FUNC=GU , PCB=IOPCB , STATUS= , MESSAGE NUMBER=000001 LENGTH=000017, PCBN=001 -MSG- GU
-----1-----2-----3-----4-----5-----6-----7-----8-----9-----10
IOAREA= PART AN960C10
0100DCDE4CDFFCFF
01007193015960310
```

Figure 18. BTSOUT: Transaction message read

By analyzing this section, you can determine that the following events occurred:

- The PART transaction started.
- The DFSSAM02 application program was called.
- The DFSSAM02 application program issued an IOPCB GU call. Status code blank was simulated by IMS Batch Terminal Simulator.
- The input message that was specified in BTSIN and that was simulated was stored in the I/O area.

5. Locate the DL/I call information.

When the transaction starts, IMS Batch Terminal Simulator prints the DL/I call information as shown in the following figure:

```

***** DB CALL- FUNC=GU , PCB=DI21PART, STATUS= , LEVEL=01, SEGMENT=PARTR00T IOLENGTH=000050, PCBN=002 -DB- GU
KFB=02AN960C10
FFCDDFFFCFF4444444
02159603100000000
SSA1=PARTR00T(PARTKEY =02AN960C10 )
DCDEDDDE4DCDEDC447FFCDDFFFCFF44444445
71939663D719325800E02159603100000000D

```

Figure 19. *BTSOUT: DL/I call information*

Application program DFSSAM02 issued a DB GU call to DI21PART. Status code blank is returned from IMS.

The subsequent output listing contains the information and related fields of the subsequent IMS calls that were issued by the applications.

6. Locate the BTS0020I message.

When the application ends, IMS Batch Terminal Simulator summarizes the statistics for the transaction.

Message BTS0020I indicates the start of a statistics report.

```

BTS0020I STATISTICS REPORT FOR TRANSACTION:PART .
PCBNAME GU GN GNP GHU GHN GHNP ISRT PURG REPL DLET DEQ CHKP LOG STAT XRST CHNG ROLB OPEN CLSE OTHR
IOPCB 1 2
DI21PART 1 1

```

Figure 20. *BTSOUT: Statistics report*

7. Locate the BTS0008I and the BTS0005I messages, which indicate the end of IMS Batch Terminal Simulator run.

IMS Batch Terminal Simulator checks for additional simulator statements. If additional simulator statements are found, the same set of output listing for the next transaction is recorded in the BTSOUT. Otherwise, IMS Batch Terminal Simulator issues BTS0008I and BTS0005I and ends the run.

```

BTS0008I END OF INPUT DATA SET ENCOUNTERED.
BTS0005I END OF BTS RUN.

```

Figure 21. *BTSOUT: End of job*

DL/I call trace listing

When the DL/I call trace option is used, IMS Batch Terminal Simulator prints the call trace information for DL/I calls that were issued by the application.

Subsections:

- [“DL/I call trace options for printing DL/I call trace listing” on page 142](#)
- [“Examples of the DL/I call trace listing” on page 142](#)
- [“Description of the fields” on page 143](#)

DL/I call trace options for printing DL/I call trace listing

To manage the DL/I call trace listing, specify the following options:

- `./O DB=` to print DL/I database call traces
- `./O MSG=` to print DL/I message call traces
- `./T DLITRACE=` to print a DL/I call trace for each PCB for the specified PSB

Examples of the DL/I call trace listing

The following example shows the DL/I database call trace and DL/I message call trace in the default format.

Table 27. DL/I call trace listing fields (continued)

Field	Description
IOAREA=	The content of the I/O area. This information is available only for calls with the I/O area.
WORK=	The content of the I/O area. This information is available only for XRST calls or Symbolic CHKP calls.
I/O LENGTH _x =	The length of the area, where x is a serial number for the area length name. This information is available only for XRST calls or Symbolic CHKP calls.
I/O AREA _x =	The content of the area, where x is a serial number for the area name. This information is available only for XRST calls or Symbolic CHKP calls.
KFB=	The key feedback area. This information is available only for successful database calls.
SSA=	The SSA. This information is available only for database calls.
AIB=	The AIB information. This information is available only when the AIB interface is used.
OPTLIST=	The option list for the DL/I call.
FEEDBACK=	The feedback area for the DL/I call.
ELAPSED SEC=	The elapsed seconds of the DL/I call. This information is available only when ./O ELAPTIME=Y is specified.
REQUEST=	The content of the request area for the ICAL DL/I call. This information is available only when the AIB interface is used with a function code of ICAL.
RESPONSE=	The content of the response area for the ICAL DL/I call. This information is available only when the AIB interface is used with a function code of ICAL.
CONTROL DATA=	The content of the control data area for the ICAL DL/I call. This information is available only when the AIB interface is used with a function code of ICAL and control data.

Db2 call trace listing

When the Db2 call trace option is used, IMS Batch Terminal Simulator prints the call trace information for Db2 SQL and IFI calls that were issued by the application.

Db2 SQL call trace information consists of the following information:

- Statement type
- Program or DBRM name
- PLAN name (only in a BTS batch environment)
- Statement number
- Section number
- Error status in the SQL communication area (SQLCA)
- IFI SQL statement information (see notes)
- Input and output variables

Notes:

1. When the IFI call trace option is used, only the IFI call trace is available; the statistics report is not printed.

2. To obtain IFI SQL statement information, Db2 monitor trace class 1 must be active for IFCID 124, and your Db2 authorization ID must have MONITOR2 privilege or SYSADM authority.
3. If the thread is not currently executing an SQL statement in Db2, IMS Batch Terminal Simulator might not be able to obtain SQL statement data from Db2 IFI. In this case, IMS Batch Terminal Simulator issues the following information as IFI SQL statement information:

```
IFI INFORMATION:  
SQL STATEMENT INFORMATION UNAVAILABLE
```

4. If `./O SQL=ALL` or `TSOSQL=ALL` is specified, host variable values are printed on BTSOUT or displayed on the TSO terminal in the following cases:
 - SQL STATEMENT VARS (values from the applications) and VARS FROM DB2 (values from Db2) when the SQL call ends normally.
 - SQL STATEMENT VARS when the SQL call ends abnormally.
 - If you use the LOB (large objects) data type, and you specify `./O SQLOBLN=`, IMS Batch Terminal Simulator puts the LOB data type and the length on the BTSOUT or the TSO terminal display as follows:
 - LEN= shows the actual data length
 - DATA= shows the LOB data

Subsections:

- [“Db2 call trace options for printing Db2 call trace listing” on page 145](#)
- [“Examples of the Db2 SQL call trace listing” on page 145](#)
- [“Description of the Db2 SQL call trace” on page 146](#)
- [“Example of the IFI call trace listing” on page 147](#)
- [“Description of the IFI call trace fields” on page 147](#)

Db2 call trace options for printing Db2 call trace listing

To manage the Db2 call trace listing, specify the following options:

- `./O SQL=` to print SQL call traces
- `./T SQLTRACE=` to print a SQL call trace for each SQL call that is issued by the application program

Examples of the Db2 SQL call trace listing

The following example shows the SQL call trace in the default format.

Table 29. Db2 SQL call trace listing information for each SQL statement type

SQL statement	Information provided
OPEN	<ul style="list-style-type: none"> • Host variable or parameter marker values (if used) • IFI SQL statement information (DECLARE CURSOR or SELECT statement string)
FETCH	<ul style="list-style-type: none"> • Column values for the fetched row (with data types) • All column names (if DESCRIBEd)
SELECT	<ul style="list-style-type: none"> • Host variable values in search condition (if used) • Column values for the selected row (with data types) • IFI SQL statement information (SELECT statement string)
INSERT	<ul style="list-style-type: none"> • Host variable values (if used) • Number of rows INSERTed • IFI SQL statement information (INSERT statement string)
DELETE	<ul style="list-style-type: none"> • Host variable values (if used) • Number of rows DELETED • IFI SQL statement information (DELETE statement string)
UPDATE	<ul style="list-style-type: none"> • Host variable values (if used) • Number of rows UPDATED • IFI SQL statement information (UPDATE statement string)
PREPARE	<ul style="list-style-type: none"> • String expression • SQLDA field values (if INTO used)
EXECUTE	<ul style="list-style-type: none"> • Parameter marker values (if used) • IFI SQL statement information
EXECUTE IMMEDIATE	<ul style="list-style-type: none"> • String expression
DESCRIBE	<ul style="list-style-type: none"> • SQLDA field values • IFI SQL statement information

Related reading: For information about the SQL statements, see the *Db2 for z/OS SQL Reference*.

Example of the IFI call trace listing

```
**** IFI CALL- FUNC=COMMAND , IFI RETURN CODE= 00, IFI REASON CODE= 00000000
**** IFI CALL- FUNC=WRITE , IFI RETURN CODE= 00, IFI REASON CODE= 00000000
**** IFI CALL- FUNC=READA , IFI RETURN CODE= 00, IFI REASON CODE= 00000000
**** IFI CALL- FUNC=COMMAND , IFI RETURN CODE= 00, IFI REASON CODE= 00000000
```

Figure 26. Example of the IFI call trace listing

Description of the IFI call trace fields

The following table summarizes the information that is available for each IFI statement.

Table 30. Db2 IFI call trace listing fields

Field	Description
FUNC=	The function of the IFI statement.
IFI RETURN CODE=	The return code is returned from Db2 in the IFI communication area (IFCA), and is displayed in decimal notation.
IFI REASON CODE=	The reason code is returned from Db2 in the IFI communication area (IFCA), and is displayed in hexadecimal notation.
ELAPSED SEC=	The elapsed seconds of the IFI statement. This information is available only when ./O ELAPTIME=Y is specified.

MQI call trace listing

When the MQI call trace option is used, IMS Batch Terminal Simulator prints the call trace information for MQI calls that were issued by the application.

The MQI call trace consists of the following information:

- MQI call function
- Compression code
- Reason code
- Putting or getting messages
- Putting or getting message lengths

Notes:

1. If you run in a DLI or DBB region without specifying the SCSQLOAD data set, message BTS0114I is issued to BTSOUT instead of the trace information of MQI calls.
2. Only the MQI call trace is available; the statistics report is not printed.

Subsections:

- [“MQI call trace options for printing MQI call trace listing” on page 148](#)
- [“Examples of the MQI call trace listing” on page 148](#)
- [“Description of the fields” on page 149](#)

MQI call trace options for printing MQI call trace listing

To manage the MQI call trace listing, specify the following options:

- ./O MQI= to print MQI call traces
- ./T MQITRACE= to print an MQI call trace for each MQI call that is issued by the application program

Examples of the MQI call trace listing

The following example shows the MQI call trace in the default format.

- [“Example of the program analysis statistics by function code and segment name” on page 150](#)
- [“Example of the program analysis statistics by status code” on page 151](#)
- [“Example of the program analysis statistics by ICAL OTMA descriptor” on page 151](#)
- [“Example of the program analysis statistics by SQL statement” on page 151](#)
- [“Example of the program analysis statistics by MQI function” on page 151](#)
- [“Description of the header fields” on page 152](#)

Example of the program analysis statistics by PCB total

The following figure shows an example of the program analysis statistics by PCB total report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=T`.

```
PROGRAM ANALYSIS STATISTICS BY PCB TOTAL FOR TRANSACTION: DSPALLI
MBR=DFSSAM07 PSB=DFSSAM07 FROM 2021/07/06 16:43:40 FOR 0.354110SEC
PCBNAME  FUNC  LVL  SEGNAME  SC  CALLS  MAX-ELAPS-SEC  AVG-ELAPS-SEC
IOPCB    **  TOTAL                7      0.000044      0.000006
DI21PART **  TOTAL                6      0.275356      0.045952
-----
** TOTAL                13      0.275356      0.021212
```

Figure 29. Example of the program analysis statistics by PCB total report

Example of the program analysis statistics by function code

The following figure shows an example of the program analysis statistics by function code report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=F`.

```
PROGRAM ANALYSIS STATISTICS BY FUNCTION CODE FOR TRANSACTION: DSPALLI
MBR=DFSSAM07 PSB=DFSSAM07 FROM 2021/07/06 16:43:40 FOR 0.354110SEC
PCBNAME  FUNC  LVL  SEGNAME  SC  CALLS  MAX-ELAPS-SEC  AVG-ELAPS-SEC
IOPCB    GU      1      0.000044      0.000044
          ISRT    6      0.000001      0.000000
IOPCB    **  TOTAL                7      0.000044      0.000006
DI21PART GU      1      0.275356      0.275356
          GN      5      0.000346      0.000072
DI21PART **  TOTAL                6      0.275356      0.045952
-----
** TOTAL                13      0.275356      0.021212
```

Figure 30. Example of the program analysis statistics by function code report

Example of the program analysis statistics by function code and segment name

The following figure shows an example of the program analysis statistics by function code and segment name report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=G`.

```
PROGRAM ANALYSIS STATISTICS BY FUNCTION-SEGNAME FOR TRANSACTION: DSPALLI
MBR=DFSSAM07 PSB=DFSSAM07 FROM 2021/07/06 16:43:40 FOR 0.354110SEC
PCBNAME  FUNC  LVL  SEGNAME  SC  CALLS  MAX-ELAPS-SEC  AVG-ELAPS-SEC
IOPCB    GU      1      0.000044      0.000044
          ISRT    6      0.000001      0.000000
IOPCB    **  TOTAL                7      0.000044      0.000006
DI21PART GU      01  PARTROOT    1      0.275356      0.275356
          GN      02  STANINFO    1      0.000007      0.000007
                   02  STOKSTAT GK  1      0.000002      0.000002
                   01  PARTROOT GA  1      0.000346      0.000346
DI21PART **  TOTAL                6      0.275356      0.045952
-----
** TOTAL                13      0.275356      0.021212
```

Figure 31. Example of the program analysis statistics by function code and segment name report

Example of the program analysis statistics by status code

The following figure shows an example of the program analysis statistics by status code report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=C`.

```
PROGRAM ANALYSIS STATISTICS BY STATUS CODE FOR TRANSACTION: DSPALLI
MBR=DFSSAM07 PSB=DFSSAM07 FROM 2021/07/06 16:43:40 FOR 0.354110SEC
SC PCBNAME CALLS MAX-ELAPS-SEC AVG-ELAPS-SEC
IOPCB 7 0.000044 0.000006
DI21PART 4 0.275356 0.068842
GK DI21PART 1 0.000002 0.000002
GA DI21PART 1 0.000346 0.000346
-----
** TOTAL 13 0.275356 0.021212
```

Figure 32. Example of the program analysis statistics by status code report

Example of the program analysis statistics by ICAL OTMA descriptor

The following figure shows an example of the program analysis statistics by ICAL OTMA descriptor report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=I`.

```
PROGRAM ANALYSIS STATISTICS BY ICAL OTMA DESCRIPTOR FOR TRANSACTION: ICALCBL
MBR=SSBICAL PSB=DFSSAM02 FROM 2021/06/27 14:15:51 FOR 0.208677SEC
ICAL-OTMA-DESCRIPTOR CALLS WARNES ERRORS MAX-ELAPS-SEC AVG-ELAPS-SEC
IVPDT0B4 7 1 4 0.047512 0.019893
IVPDT0B9 1 0 1 0.000054 0.000054
-----
** TOTAL 8 1 5 0.047512 0.017413
```

Figure 33. Example of the program analysis statistics by ICAL OTMA descriptor report

Example of the program analysis statistics by SQL statement

The following figure shows an example of the program analysis statistics by SQL statement report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=S`.

```
PROGRAM ANALYSIS STATISTICS BY SQL STATEMENT FOR TRANSACTION: REPOSQ
MBR=SSASQ SQL PSB=DFSSAM02 FROM 2021/06/29 12:21:13 FOR 1.715030SEC
SQL-STATEMENT CALLS WARNES ERRORS MAX-ELAPS-SEC AVG-ELAPS-SEC
OPEN 1 0 0 0.000970 0.000970
CLOSE 1 0 0 0.000012 0.000012
SELECT 3 0 0 0.001980 0.001570
FETCH 1 0 0 0.000026 0.000026
INSERT 2 0 0 0.001238 0.001133
DELETE 0 0 0 0.000000 0.000000
UPDATE 1 0 0 0.002094 0.002094
PREPARE 0 0 0 0.000000 0.000000
DESCRIBE 0 0 0 0.000000 0.000000
EXECUTE 0 0 0 0.000000 0.000000
EXECUTE IMMEDIATE 0 0 0 0.000000 0.000000
OTHER 7 0 0 0.754645 0.151591
-----
** TOTAL 16 0 0 0.754645 0.066951
```

Figure 34. Example of the program analysis statistics by SQL statement report

Example of the program analysis statistics by MQI function

The following figure shows an example of the program analysis statistics by MQI function report. You can generate this analysis report by specifying `./O REPORT=YES` or `REPORT=M`.

```

PROGRAM ANALYSIS STATISTICS BY MQI FUNCTION FOR TRANSACTION: MQREPORT
MBR=SSAMQI PSB=DFSSAM02 FROM 2021/06/29 15:08:14 FOR 0.275878SEC
MQI-FUNCTION CALLS WARNS ERRORS MAX-ELAPS-SEC AVG-ELAPS-SEC
MQCONN 1 0 0 0.050573 0.050573
MQDISC 1 0 0 0.014761 0.014761
MQOPEN 1 0 0 0.008358 0.008358
MQCLOSE 1 0 0 0.007690 0.007690
MQGET 5 0 1 0.018390 0.012261
MQPUT 2 0 0 0.022744 0.022310
MQPUT1 2 0 0 0.007415 0.007391
MQINQ 1 0 0 0.007277 0.007277
MQSET 1 0 0 0.008225 0.008225
MQCMIT 1 0 0 0.009272 0.009272
MQBACK 1 0 0 0.007621 0.007621
MQCONNX 0 0 0 0.000000 0.000000
MQSUB 0 0 0 0.000000 0.000000
MQSUBRQ 0 0 0 0.000000 0.000000
MQSTAT 0 0 0 0.000000 0.000000
OTHER 0 0 0 0.000000 0.000000
-----
** TOTAL 17 0 1 0.050573 0.013793

```

Figure 35. Example of the program analysis statistics by MQI function report

Description of the header fields

The following table summarizes the information about the header fields of the reports.

Table 32. Header columns of reports

Header column	Meaning
MBR=	Load module name
PSB=	PSB name
FROM	Start time of the transaction
FOR	Elapsed time of the transaction
PCBNAME	PCB name
FUNC	Function code
LVL	Segment level
SEGNAME	Segment name
SC	Status code
CALLS	Call counts
MAX-ELAPS-SEC	Maximum elapsed seconds
AVG-ELAPS-SEC	Average elapsed seconds
ICAL-OTMA-DESCRIPTOR	ICAL OTMA descriptor name
WARNS	Warning counts
ERRORS	Error counts
SQL-STATEMENT	SQL statement
MQI-FUNCTION	MQI function

Part 4. Playback utility

You can use the Playback utility to generate the transaction screens from the IMS log data set.

Topics:

- [Chapter 13, “Transaction screen formatting \(Playback utility\),” on page 155](#)

Chapter 13. Transaction screen formatting (Playback utility)

The Playback utility is a transaction screen formatting utility that creates a file of screen images that correspond to the transaction activities that are captured in the IMS log data set.

Topics:

- [“Playback utility overview” on page 155](#)
- [“Playback utility modules and process flow” on page 156](#)
- [“Playback utility limitations and considerations” on page 158](#)
- [“Creating screen images of transaction activities” on page 161](#)
- [“Restarting the Playback utility after BTS0004W error” on page 163](#)
- [“Tips for using the Playback utility” on page 163](#)
- [“Input for the Playback utility” on page 165](#)
- [“Playback utility reports” on page 178](#)

Playback utility overview

By using the IMS online log data sets (OLDS) and system log data sets (SLDS) as input, the Playback utility formats the transaction log into transaction screen images.

The IMS OLDS and SLDS traditionally are used to maintain IMS system integrity, but users have often found many other uses for the data on the logs.

Transaction queue-related data is contained on the IMS log as '01'x records (INPUT MESSAGE QUEUED) and '03'x records (OUTPUT MESSAGE QUEUED). These records contain, among other things, information that identifies the user who initiated the transaction, the terminal from which it originated, and the target to where the data is being sent. The data that will be processed by the application program or displayed on a terminal is also contained on these records.

One of the functions of IMS Batch Terminal Simulator is to simulate transaction processing in batch mode. To accomplish this, IMS Batch Terminal Simulator uses the MFS format library to format the transaction data into a screen display. As part of its function, IMS Batch Terminal Simulator processes the ISRT calls that are issued by an application program to the message queue.

The Playback utility uses these functions of IMS Batch Terminal Simulator to create its report.

Functions

The function of the Playback utility is to create a file of screen images that correspond to the transaction activities captured in an IMS OLDS or SLDS.

The Playback utility provides the following functions:

- Queries RECON data sets to find log data set names.
- Finds and extracts the data part of an IMS log record.
- Finds the MOD name in the IMS log record, and extracts it.
- Finds and extracts the data that identifies the user ID, the logical terminal (LTERM) name, the transaction code, and the logged time in the IMS log record.
- Issues DLI ISRT calls to drive IMS Batch Terminal Simulator through its normal functions, which include formatting a screen based on the information in the data.
- Edits the screen that results from IMS Batch Terminal Simulator and appends the previously extracted data related to user identification.

Because the application programs that initially generated the messages are never called, and the associated databases are never accessed, the application programs or the databases are not needed by Playback processing.

Features and benefits

The Playback utility is designed to provide users with an easy-to-use, flexible tool to translate the queue log records into a readable format. Although the default processing mode is set to select and process all records, the utility can selectively process a subset of the records based upon the selection criteria that is driven by the control statement.

The Playback utility uses standard IMS DL/I calls to ensure that it functions correctly with all levels of IMS.

The reports and files that are generated by the Playback utility can be used to:

- Monitor IMS transaction activity
- Create an audit trail of transactions
- Track activity of specific users or transactions
- Create a history of IMS transaction activity
- Determine which MODs are active in a system in order to discover unused format blocks
- Provide a tool for technical support personnel and application administrators

You can use the Playback utility to create the following output:

- A list of the transaction codes that have executed within an IMS log
- A list of the user IDs that have executed transactions within an IMS log
- A list of the MFS MOD names that have been involved with transactions within an IMS log
- A list of the transaction codes that have been executed, cross referenced by the user IDs that scheduled them, within an IMS log
- Screen images that are associated with the transactions within an IMS log

With the Playback utility, you can:

- Specify which user IDs you are interested in tracking
- Specify which transactions you are interested in tracking
- Specify which transactions you are not interested in tracking
- Specify which terminals you are interested in tracking
- Specify which terminals you are not interested in tracking
- Specify which MFS MODs you are interested in tracking
- Specify which MFS MODs you are not interested in tracking
- Specify which log type you are not interested in tracking
- Narrow your search to specific time intervals in the log
- Skip records in the IMS log or extraction files before beginning your search
- Process a subset of the records in the IMS log or extraction files
- Restart the process after errors have prematurely terminated the process
- Run IMS Batch Terminal Simulator as a batch program (DLI/DBB)
- Process data from any source that has been converted to match formats used by the Playback utility

Playback utility modules and process flow

The Playback utility consists of five modules.

Subsections:

- [“Modules” on page 157](#)
- [“Process flow” on page 157](#)

Modules

BTSARCNO

This module extracts log data set names from the RECON data sets.

The module analyzes the RECON data sets to find appropriate log data set names. This module is optional because you can specify the log data set names directly in the JCL of the BTSALOG0 module.

BTSALOG0

This module extracts log records ('01'x and '03'x records) from the IMS log data sets that were identified in the BTSARCNO step or that are specified in the JCL.

The log records can be selected by one or more of the following criteria through the CNTLCRDS control statement specification:

- User ID
- Transaction code
- Logical terminal (LTERM) name
- Logged time

BTSAPRE0

This module prepares the following data, which is used by IMS Batch Terminal Simulator in the BTSAISRO step:

- A cross reference file relating messages to the corresponding sequence numbers in the extract file and containing user-related information
- A file of input statements that are required by IMS Batch Terminal Simulator (BTSIN file)

Note: BTSIN input statements include enough 'PA2 \$' statements to run through all the generated screens.

- A report that lists the search criteria and statistics related to the output creation

The log records can be selected by one or more of the following criteria through the CNTLCRDS control statement specification

- MOD name
- Transaction code
- User ID

You can ignore log records by one or more of the following criteria:

- MOD name
- Terminal code
- Log type

Related reading: See [“Sort dependency for BTSAPRE0” on page 158](#) for considerations about the sort dependency.

BTSAISRO

This module issues DLI ISRT calls through IMS Batch Terminal Simulator. These calls include calls to format the screen based on information in the data. This module then obtains the extracted data. This job runs as an IMS job in the IMS Batch Terminal Simulator environment.

BTSAEDTO

This module edits the BTSOUT report that is created by the IMS Batch Terminal Simulator run.

Process flow

The following figure outlines the process flow of the Playback utility.

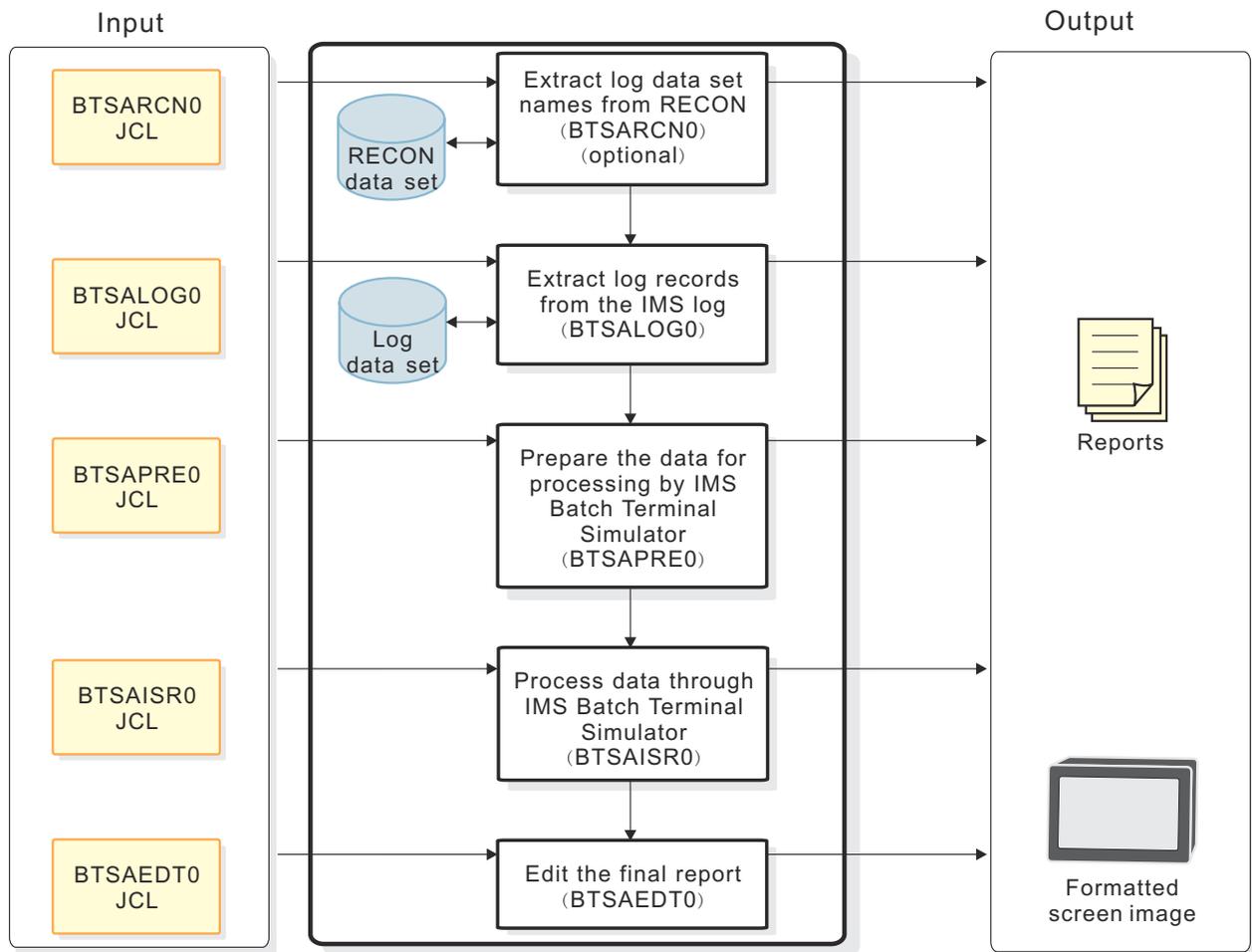


Figure 36. Playback utility process flow

Playback utility limitations and considerations

Certain limitations and considerations apply to using the Playback utility.

Subsections:

- [“Sort dependency for BTSAPREO” on page 158](#)
- [“Separation of BTSARCNO and BTSALOGO in separate jobs” on page 159](#)
- [“Use of MOD MFS block with '01'x records” on page 159](#)
- [“Single logical page and multiple physical pages” on page 159](#)
- [“PSB requirement” on page 159](#)
- [“Restarting Playback after BTS0004W error” on page 159](#)

Sort dependency for BTSAPREO

An input or output queue message might span multiple log records. These multiple records might not be contiguous in the log data set, and might even be mixed with records for other queue messages.

BTSAPREO requires that all of the records for each message are contiguous. A preliminary sort step is used to ensure that the records are processed in the appropriate order.

The TRXOUT file, which is generated by BTSALOGO, is sorted by the character representations of the following fields from the queue message log records:

- IMS System ID and store clock time

- Device Relative Record Number (DRRN)
- Log record code ('01' or '03')

Note: These fields were converted from a binary to a displayable format and written in the TRXOUT file.

The SORT SYSIN control statement that corresponds to this file record layout and that must be used by SORT is `SORT FIELDS=(39,50,CH,A,89,4,CH,A,5,1,CH,A)`.

Related reading: See [“Extract file record and CROSSREF file record formats”](#) on page 253 for the layout of the TRXOUT file record.

Separation of BTSARCNO and BTSALOG0 in separate jobs

The BTSARCNO and BTSALOG0 cannot be executed in the same job. Not only must they be in different jobs, but the job containing BTSALOG0 must not be submitted until the job that contains BTSARCNO has completed.

The reason is related to the following two factors:

- BTSARCNO creates JCL in the JCLPDS for the DD statements that identify the logs that BTSALOG0 accesses through the INCLUDE statement.
- JES (Job Entry System) expands JCL from INCLUDEs or PROCs as soon as the job has been submitted.

If both programs are executed in the same job, at the time this job is received by JES, the JCL referenced by the INCLUDE statement is expanded. However, at this time, obviously BTSARCNO has not yet run, so it could not have populated the JCLPDS member yet. The contents of the member at this time will not be what is expected.

If the two steps are not separated, JES will read either:

- An empty PDS member if this member has never been used before
- The contents of the prior run if this PDS member is reused

Use of MOD MFS block with '01'x records

The LOG records contain only the MOD name of the records. IMS does not support the use of a MID name in a DLI ISRT call to the message queue. The Playback utility always uses the MOD name in the ISRT calls, whether a '01'x record or '03'x record was the source of the data in the log. The MOD matches the layout for the '03'x (OUTPUT MESSAGE QUEUED), but might not accurately match the layout for the '01'x records (INPUT MESSAGE QUEUED). As a result, the screen display generated for the '01'x record might not match the actual screen, if the MID and MOD are different.

Single logical page and multiple physical pages

The Playback utility cannot detect when a transaction has single logical pages that are associated with multiple physical pages.

An optional facility is provided for you to specify the number of physical pages that are associated with a MOD name. In these cases, BTSIN statements are generated to drive IMS Batch Terminal Simulator through the multiple pages.

If this facility is not used, only the first physical page of a multi-page MOD is presented.

PSB requirement

The Playback utility issues ISRT and PURG calls to the I/O PCB. To ensure that the I/O PCB is present, the `CMPAT=YES` keyword must be coded for the PSB that is being used by the Playback utility. Any PSB on the execution system that meets this criterion is suitable for use by the Playback utility.

Restarting Playback after BTS0004W error

When you process the BTSAISRO module, you might receive a BTS0004W message.

During the processing of the records by IMS Batch Terminal Simulator, IMS Batch Terminal Simulator and the BTSAISRO operates asynchronously: BTSAISRO processes all of its messages (ISRT and PURG) and receive a zero completion code before IMS Batch Terminal Simulator formats all of the screens.

Occasionally, IMS Batch Terminal Simulator cannot successfully format some of the messages. The most common reasons for this problem are:

- The MOD name specified in a record is missing from the MFS format library.
- The DIFs or DOFs that are associated with the format do not match the terminal type that is specified in the ./D BTSIN control statement.

When these errors occur, the screen generated by IMS Batch Terminal Simulator in the BTSOUT contains the message DFS057I REQUESTED BLOCK NOT AVAILABLE : . After this screen, IMS Batch Terminal Simulator no longer continues processing the messages that are queued by BTSAISRO. All subsequent queued messages generate the message BTS0004W NO TRANSACTION INFORMATION SUPPLIED.

An excessive number of pages indicated for a multi-page MOD also triggers this message. When this condition occurs, the DFS0571 message does not occur and IMS Batch Terminal Simulator generates the message BTS0004W NO TRANSACTION INFORMATION SUPPLIED as it tries to run through non-existent panels.

Another occurrence that triggers this message is an excessive number of pages indicated for a multi-page MOD. In this case, while there will be no occurrence of the DFS0571 message, IMS Batch Terminal Simulator will generate the message BTS0004W NO TRANSACTION INFORMATION SUPPLIED as it tries to run through non-existent panels.

In either situation, IMS Batch Terminal Simulator terminates with condition code 0, even though it has not successfully produced the desired result. However, BSAEDT0 detects that the IMS Batch Terminal Simulator step was not completed. BSAEDT0 formats all of the screens up until the BTS0004W message is detected, at which point it terminates with condition code 8, and generates the following messages:

- BSA1003E, which indicates a processing error has occurred.
- BSA1004E, which indicates the location in the CROSSREF file at which the problem with the MOD originated.
- BSA1009E, which indicates the location in the extract file at which the problem with the MOD originated.

The number of records and starting position in the CROSSREF must exactly match the number of screens that are processed from the BTSOUT. The user-related information that is extracted from the CROSSREF and applied to the BTSOUT records is matched by record count. If the location within these two files gets out of sync, the user-related information will be applied incorrectly.

To continue the process, there are two possible techniques that ensures that the files stay in sync:

IGNORE technique

Use the IGNOREMOD= CNTLCRDS control statement keyword to bypass the problem log records from TRXIN and restart at the BTSAPRE0 process from the beginning.

The *IGNORE technique* ensures that, regardless of how many occurrences there are of the MOD in question, all are bypassed. However, this process does begin at the top of the TRXIN file and does reprocess all of the transactions. The Playback process can sometimes be long, and it might not be desirable to process the records again.

Skip records and IGNORE technique

Use the IGNOREMOD= CNTLCRDS control statement keyword and the SKIP= CNTLCRDS control statement keyword to skip past the point of failure in BTSAPRE0. This method bypasses the problem log records from TRXIN, and restarts at the BTSAPRE0 process from the beginning.

The *skip records and IGNORE technique* eliminates reprocessing of records and ensures that all occurrences of the MOD in question are bypassed.

This technique is the recommended approach. However, one drawback is that the results of the original run must be retained and combined with the results of the subsequent run in order to get a complete report.

For instructions for restarting the Playback utility after you receive a BTS0004W error, see [“Restarting the Playback utility after BTS0004W error”](#) on page 163.

Creating screen images of transaction activities

To create screen images of transaction activities, code the JCL statements and the control statements for each of the Playback utility modules and run the modules.

Before you begin

The following sample JCL streams are in the BTS JCL library (SBTSJCL0). You can modify and use these JCL streams.

BTSASMP1 (runs BTSALOG0 and BTSAPRE0)

This JCL analyzes the log data. In this sample JCL, the log data set is defined in the JCL (LOG DD in RUNSUM step).

BTSASMP2 (runs BTSARCNO)

This JCL is an alternative method for analyzing the log data. This JCL is separated into two jobs. In this method, BTSASMP2 is used to determine the log data set names from RECON.

BTSASMP3 (runs BTSALOG0 and BTSAPRE0)

This JCL analyzes the log data set names that were discovered by BTSASMP2.

BTSASMP4 (runs BTSAISRO and BTSAEDT0)

This JCL is used to create the Playback screens.

About this task

The following two methods are available for log analysis:

- Run BTSASMP1 and BTSASMP4. Use this method if you want to define the log data set.
- Run BTSASMP2, BTSASMP3, and BTSASMP4. Use this method if you want to determine the log data set names from RECON.

Procedure

1. Optional: To extract the log data set names from the RECON data sets, run the BTSARCNO module.

Note: You can omit this step if you want to specify the log data set names manually in the JCL of the BTSALOG0.

If you run this step, ensure that you run it as a separate job from the job that runs BTSALOG0. BTSARCNO creates a part of JCL that is used by BTSALOG0, and the job must be completed before the Job Entry System (JES) analyzes the JCL for BTSALOG0.

After you run this module, ensure that the return code is 0 and check how the job ran by viewing the RECON Query Summary report in the REPORT data set.

Related reading:

- See [“BTSARCNO JCL requirements”](#) on page 165 for a list of DD statements and control statements.
- See [“Separation of BTSARCNO and BTSALOG0 in separate jobs”](#) on page 159 for more information about separating the BTSARCNO job and the BTSALOG0 module.
- See [“BTSARCNO: RECON Query Summary report”](#) on page 178 for an example of the summary report.

2. Extract log records from the IMS log by running the BTSALOG0 module.

After you run this module, ensure that the return code is 0 and check how the job ran by viewing the summary report in the REPORT data set.

Related reading:

- See [“BTSALOG0 JCL requirements”](#) on page 168 for a list of DD statements and control statements.
 - See [“BTSALOG0: Summary report”](#) on page 179 for an example of the summary report.
3. Prepare the data for processing by IMS Batch Terminal Simulator by running the BTSAPRE0 module.
- After you run this module, ensure that the return code is 0 and check how the job ran by viewing the summary report in the SYSPRINT data set.

Related reading:

- See [“BTSAPRE0 JCL requirements”](#) on page 171 for a list of DD statements and control statements.
 - See [“BTSAPRE0: Summary report”](#) on page 180 for an example of the summary report.
4. Process the data through IMS Batch Terminal Simulator by running the BTSAISR0 module.

This module runs as an IMS job in the IMS Batch Terminal Simulator environment. Before you run this step, make the following modifications to your IMS Batch Terminal Simulator JCL:

- a) Specify either of the following IMS region types in the supplied BTS procedure.

Table 33. Specifying the IMS region type in BTS procedure

IMS region type	Task
DLI	i) Set the KW parameter to a value of DLI. ii) Ensure that the PSB that is used by BTSAISR0 is contained in the PSBLIB that is referenced by the //IMS DD statement.
DBB	i) Set the KW parameter to a value of DBB. ii) Perform an ACBGEN on the PSB; the ACB must be contained in the ACBLIB that is referenced by the //IMSACB DD statement.

- b) Specify the DD statements that define input and output.
- c) After you run this module, ensure that the return code is 0 and check how the job ran by viewing the summary report in the SYSPRINT data set.

Related reading:

- See [“Customizing the BTS cataloged procedure”](#) on page 45 for information about the BTS procedure.
 - See [“BTSAISR0 JCL requirements”](#) on page 174 for a list of DD statements and control statements.
 - See [“Default device type for BTSAISR0”](#) on page 164 for information about how IMS Batch Terminal Simulator determines the default device type.
 - If you receive a BTS0004W message while processing the BTSAISR0 module, follow the instructions in [“Restarting the Playback utility after BTS0004W error”](#) on page 163.
 - See [“BTSAISR0: Summary report”](#) on page 181 for an example of the summary report.
5. Edit the final report by running the BTSAEDT0 module.

After you run this module, ensure that the return code is 0 and check how the job ran by viewing the summary report in the SYSPRINT data set and the formatted screen images of the transaction activities in the OUTFILE data set.

Related reading:

- See [“BTSAEDT0 JCL requirements”](#) on page 176 for a list of DD statements and control statements.
- See [“BTSAEDT0: Screen Image Summary report”](#) on page 182 for an example of the Screen Image Summary report and [“BTSAEDT0: Sample formatted screen”](#) on page 183 for an example of the formatted screen image.

Restarting the Playback utility after BTS0004W error

When you process the BTSAISRO module, you might receive a BTS0004W message.

Before you begin

Two techniques are provided to restart the Playback utility job after BTS0004W error:

- IGNORE technique: Restart by using the IGNOREMOD= keyword in BTSAPRE0
- Skip records and IGNORE technique: Restart by using the IGNORE=keyword and the SKIP=keyword

Before you restart the Playback utility job, determine which techniques to use by referring to [“Restarting Playback after BTS0004W error”](#) on page 159. The BTSALOGO summary report can help determine which technique is most appropriate. This report determines if there are multiple occurrences of this problem MOD. Also, other MODs with similar names due to naming conventions might also share the same trait that caused the failure with the identified MOD; a decision on how to avoid processing these other MODs can be made without waiting until an actual failure has occurred.

Procedure

- Restart by using the IGNOREMOD= keyword in BTSAPRE0 (IGNORE technique)

Message BTS1004E identifies the MOD where the problem originated.

To restart Playback by using this technique, follow these steps:

- a) (Optional) Back up the OUTFILE and SYSPRINT data sets that were produced by BTS1004E.

Note: This technique starts at the beginning of the log extract file and reprocesses records, so the backup is not essential.

- b) Code the IGNOREMOD= keyword for the BTSAPRE0 where the MOD name is the value identified in message BTS1004E.
- c) Repeat Step b of the process beginning in the BTSAPRE0.

In this case, all of the TRXIN records, except for the records that are identified by the IGNOREMOD= keyword, are processed.

- Restart by using the IGNORE=keyword and the SKIP=keyword (Skip records and IGNORE technique)

To restart Playback by using this technique, follow these steps:

- a) Back up the OUTFILE and SYSPRINT data sets that were produced by BTS1004E.
- b) Code the SKIP= keyword for the BTSAISRO where the value for SKIP is the value identified in message BTS1009E associated within the TRXIN file.
- c) Code the IGNOREMOD= keyword for the BTSAPRE0 where the MOD name is the value identified in message BTS1004E.
- d) Repeat Step b of the process beginning in the BTSAPRE0.

In this case, all the TRXIN records after the skipped records, except for the records that are identified by the IGNOREMOD= keyword, are processed.

Tips for using the Playback utility

The following tips can simplify the use of the Playback utility.

Subsections:

- [“Tips for data set space allocation”](#) on page 164
- [“DASD and elapsed time saving tips”](#) on page 164
- [“Default device type for BTSAISRO”](#) on page 164

Tips for data set space allocation

Because IMS logs can be large, extracts of records from logs can be large also.

The following table lists the data sets that can become large and indicates an example of space allocations for them. Although the sizes are accurate, they are presented for illustrative purposes only.

Table 34. Example for data set space allocation

Data set	Allocation size
IMS log that contains 1076904 records. Of these records, 50682 were selected for processing approximately 5% by record count, approximately 20% by DASD usage	760 Cyls (Block size 32760)
BTSALOG0.TRXOUT that contains 51000 records	142 Cyls of 3,390 (Block size 33000)
SORT.SORTOUT that contains 51000 records	142 Cyls of 3,390 (Block size 33000)
BTSAPRE0.TRXOUT	130 Cyls of 3,390 (Block size 33000)
BTSAPRE0.BTSIN that contains 44000 records	5 Cyls of 3,390 (Block size 3120)
BTSAPRE0.CROSSREF that contains 44000 records	125 Cyls of 3,390 (Block size 33000)
BTSAISR0.BTSOUT	400 Cyls of 3390 (Block size 33000)
BTSAEDT0.OUTFILE	280 Cyls of 3390 (Block size 33000)

DASD and elapsed time saving tips

The following usage-related tips might help to reduce DASD usage and improve overall performance:

- Avoid processing every '01'x and '03'x record if possible. The BTSAISR0 process might run for a significantly long time when it must process every record. You can use the statistics from the BTSALOG0 Summary Report to narrow your selection of records.
- Use the IGNORETERM= control statement in BTSALOG0 to eliminate many unwanted records early in the process. Records that are selected in this step but eliminated later will unnecessarily increase the size of the BTSALOG0.TRXOUT and SORT.SORTOUT data sets prior to the next opportunity to use IGNORETERM= (in BTSAPRE0).

Potential candidates to be ignored include:

- Messages destined to or originating from the Master and Secondary consoles
- Messages that might have originated from an MSC link
- Use the IGNOREMOD= control statement in BTSAPRE0 to eliminate records with invalid MOD names. Potential candidates to be ignored include messages with blank or null values for MOD name.

Default device type for BTSAISR0

If the ./D DDOF or the ./TYPE statement is not specified on the BTSIN DD card when running BTSAISR0, IMS Batch Terminal Simulator cannot determine the device type attribute. But when the ./E IMSSUF keyword, is specified, IMS Batch Terminal Simulator determines the device type attribute automatically from the LTERM name of DFSCLV0x, which is determined by the keyword, by using the LTERM name of the first record in TRXIN DD.

This function is available under the following conditions:

- TC= (Transaction code) in BTSIN DD is BTSAISR0
- Both ./D DDOF= and ./D TYPE= are not specified in BTSIN DD
- The ./E IMSSUF= keyword is specified in BTSIN DD

If `./D DDOF` or `./D TYPE=` is specified, it is used as the device (`./D`) type even though `./E IMSSUF=` is specified. Message `BTS0124W` is issued and `./E IMSSUF=` is ignored.

IMS Batch Terminal Simulator provides a default device type for `BTSAISR0`. The default is `DEV TYPE=(3270,2),FEAT=IGNORE`. If all of `./E IMSSUF=`, `./D DDOF=`, and `./D TYPE=` are not specified, `BTSAISR0` runs with a default device type after issuing message `BTS0123W`.

Restriction: IMS Batch Terminal Simulator does not change the device type to the type that is specified by each `LTERM` specification. Whatever is specified by `LTERM`, IMS Batch Terminal Simulator uses the device type that is determined first.

Input for the Playback utility

Each of the Playback modules requires you to code JCL statements.

BTSARCNO JCL requirements

The JCL for the `BTSARCNO` module, which extracts the log data set names from the `RECON` data sets, must satisfy certain JCL requirements.

Subsections:

- [“JCL statements” on page 165](#)
- [“CNTLCRDS control statement keywords” on page 166](#)

JCL statements

The following list summarizes the JCL statements for `BTSARCNO` JCL.

CNTLCRDS (Input)

Contains the `CNTLCRDS` control statements.

The DCB for this data set is `RECFM=FB,LRECL=80`.

JCLOUT (Output)

A `DD` statement that specifies where the selected IMS log data sets is written. This `DD` must be specified as a member of the partitioned data set (`PDS`) for use in the `INCLUDE` statement on `BTSALOG0`.

The DCB for this data set is `RECFM=FB,LRECL=80`.

RECON1, RECON2, RECON3 (Input)

The `RECON` data sets that are used by the `DBRC` function of IMS.

These data sets can also be identified through dynamic allocation. In this case, the dynamic allocation libraries must be included in the `STEPLIB` concatenation, or an equivalent method of identifying the data sets.

REPORT (Output)

Diagnostic messages and summary reports are written.

The DCB for this data set is `RECFM=FB,LRECL=133`.

STEPLIB (Input)

Describes the library that contains Playback.

The `STEPLIB` `DD` statement is a required `DD` statement that defines the following input data sets:

- The Playback load module library
- The IMS Load module library

SYSIN (Input)

Contains the `LIST.LOG` statement that is used by `DBRC` for processing.

The DCB for this data set is `RECFM=FB,LRECL=80`.

SYSPRINT (Input and Output)

SYSPRINT is a data set that contains the results of a DBRC LIST.LOG command that can be used in two ways. If you do not use the NORECON control statement, BTSARCNO invokes DBRC and the results are written to SYSPRINT. If you use the NORECON control statement, you must prepare the SYSPRINT file with your own LIST.LOG results, and you can specify RECFM=VB on the DCB for this data set.

BTSARCNO can get read and write access to the SYSPRINT file. A data set on a DASD must be specified.

The DCB for this data set is RECFM=FB,LRECL=133.

CNTLCRDS control statement keywords

Playback processing can be influenced by the CNTLCRDS control statements that are read from the CNTLCRDS file.

Most control statements are optional. The default result of statement omission is discussed for each keyword.

Related reading: For syntax rules, see [“Syntax rules for the control statements” on page 177](#).

The following keywords apply to BTSARCNO.

LOGCOUNT= keyword

This keyword specifies the number of log data sets that BTSARCNO processes.

The format of the LOGCOUNT= keyword is LOGCOUNT=*value*, where *value* is a 1- to 8-digit numeral.

The LOGCOUNT= keyword is optional. If you omit this keyword, all logs that are within the time interval specification are processed.

Only one LOGCOUNT= control statement can be specified.

LOGTYPE= keyword

This keyword identifies the type of IMS log.

The format of the LOGTYPE= keyword is LOGTYPE=*value*, where *value* is one of the following values:

OLDS

The primary online log data sets are used.

SLDS

The primary system log data sets are used.

PRIOLD

The primary online log data sets are used.

PRISLD

The primary system log data sets are used.

SECOLD

The secondary online log data sets are used.

SECSLD

The secondary system log data sets are used.

LOGS

The primary log data sets are used.

PRILOG

The primary log data sets are used.

SECLOG

The secondary log data sets are used

The LOGTYPE= keyword is optional. If omitted, the default value is SLDS.

MAXLOGS= keyword

This keyword specifies the maximum number of log data sets that can be listed by BTSARCNO.

The format of the MAXLOGS= keyword is MAXLOGS=*value*, where *value* is a 1- to 8-digit numeral.

The MAXLOGS= keyword is optional. If omitted, the default value is 100.

Specify this keyword only if error message BTS1011E indicates that the maximum number of log data sets has been exceeded.

Only one MAXLOGS= control statement can be specified.

NORECON keyword

This keyword identifies that the results of a prior LIST.LOG are supplied to BTSARCNO in the SYSPRINT DD statement, and that the program evaluates this data instead of issuing the LIST.LOG command.

The format of the NORECON keyword is: NORECON

The NORECON keyword is optional. If this keyword is omitted, the LIST.LOG command is issued to DBRC.

SSID= keyword

This keyword specifies the subsystem identifier associated with the IMS subsystem.

The format of the SSID= keyword is SSID=*value*, where *value* is a 4-character value.

The SSID= keyword is required.

STOPTIME= keyword

This keyword specifies the time of the last log data set to be processed. An exact match of time is not required; processing stops with the first record equal to or greater than the indicated time.

The format of the STOPTIME= keyword is STOPTIME=*value*, where *value* is the 19-digit time that is used by DBRC. The format of this value is *yyydddhhmmssthmiju*. The Playback utility uses the time precision that is being used by DBRC.

The STOPTIME= keyword is optional. If omitted, the concatenation of selected logs terminates with the last log identified in RECON.

STRTTIME= keyword

This keyword specifies the time of the first log data set to be processed. An exact match of time is not required; processing begins with the first record equal to or greater than the indicated time.

There are two formats of the STRTTIME= keyword:

- STRTTIME=*value*

where *value* is the 19-digit time that is used by DBRC. The format of this value is *yyydddhhmmssthmiju*. The Playback utility uses the time precision that is being used by DBRC.

- STRTTIME=LAST

in which case, only the latest log is used

Recommendation: STRTTIME=LAST can be used in conjunction with the LOGCOUNT= keyword to get the latest logs, up to the LOGCOUNT value.

The STRTTIME= keyword is optional. If omitted, the concatenation of selected logs begins with the first log identified in RECON.

UNIT= keyword

This keyword specifies a unit type that will be used in the DD statements generated for the log data sets instead of the unit type associated with these data sets in RECON.

The format of the UNIT= keyword is UNIT=*value*, where *value* is an allowable unit type for this installation.

The UNIT= keyword is optional. If this keyword is omitted, and a unit is required by the generated DD statement, the unit indicated in RECON is used.

The UNIT= keyword must be specified with the VOLSER keyword.

VOLSER keyword

This keyword specifies that the volume serial number and unit type must be included on the DD statements that are generated for the log data sets, instead of assuming that the log data sets are cataloged.

The format of the VOLSER keyword is: VOLSER

The VOLSER keyword is optional. If this keyword is omitted, the log data sets are assumed to be cataloged, and the UNIT and the VOL=SER keywords are not included in the generated JCL.

When you want to change UNIT=, you must specify UNIT= and VOLSER together.

BTSALOGO JCL requirements

The JCL for the BTSALOGO module, which extracts the log records from the IMS log data sets, must satisfy certain JCL requirements.

Subsections:

- [“JCL statements” on page 168](#)
- [“CNTLCRDS control statement keywords” on page 169](#)

JCL statements

The following list summarizes the JCL statements for BTSALOGO JCL.

JCLLIB ORDER=*jclpds* (Input)

This statement is a JCL statement. It must directly follow the JOB statement in the JCL stream.

This statement references the partitioned data set with a member that contains the LOG DD statements that identify the log data set names.

This statement is required only if the log data set names are to be extracted from RECON.

The DCB for this data set is RECFM=FB,LRECL=80.

CNTLCRDS (Input)

Contains the CNTLCRDS control statements.

The DCB for this data set is RECFM=FB,LRECL=80.

INCLUDE (Input)

This statement is a JCL statement. This statement specifies the member name from the partitioned data set identified in JCLLIB that identifies the LOG DD statements when BTSARCN0 was used.

LOG (Input)

Refers to the IMS OLDS or SLDS.

This DD statement can be supplied in one of the following ways:

- When BTSARCN0 was not used to generate LOG DD statement, specify it in the JCL.
- When BTSARCN0 was used to generate LOG DD statement, specify it as a member of the PDS—specified by the JCLLIB statement—of the INCLUDE statement.

The DCB for this data set is RECFM=VB,LRECL=32756, or equivalent DCB, if the log data sets were created with a different LRECL.

MSGLIST (Output)

Records that are related to extracted messages are written.

You can process these records to create additional reports, if you include the MSGLIST keyword in the CNTLCRDS statements.

The DCB for this data set is RECFM=FB,LRECL=80.

Related reading: See [“MSGLIST file record format” on page 255](#) for information about the MSGLIST file records.

REPORT (Output)

Diagnostic messages and summary reports are written.

The DCB for this data set is RECFM=FB,LRECL=133.

STEPLIB (Input)

Describes the library that contains Playback.

SYSPRINT (Output)

Diagnostic messages and summary reports are written.

The DCB for this data set is RECFM=FB,LRECL=133.

TRXOUT (Output)

Formatted extracts of the '01'x and '03'x records are written.

The DCB for this data set is RECFM=VB,LRECL=32756.

Related reading: See [Chapter 17, “Playback utility record format,”](#) on page 253 for information about the records.

CNTLCRDS control statement keywords

Playback processing can be influenced by the CNTLCRDS control statements that are read from the CNTLCRDS file.

Most control statements are optional. The default result of statement omission is discussed for each keyword.

If none of the IGNORE-related keywords are used, no records will be specifically bypassed for processing. If none of the selection-related keywords are used, all records will be candidates for processing.

If a record matches some criteria for processing and also some criteria for being ignored, it will be ignored.

Related reading: For syntax rules, see [“Syntax rules for the control statements”](#) on page 177.

The following keywords apply to BTSALOG0.

IGNORETERM= keyword

This keyword specifies an LTERM name for which associated records will not be processed.

The format of the IGNORETERM= keyword is IGNORETERM=*lterm_name*, where *lterm_name* is an 8-character field.

Additional allowable *lterm_name* value:

- HIGH#VAL - eight characters of 'FF'X
- NULL#VAL - eight characters of '00'X

The IGNORETERM= keyword is optional. If omitted, no records will be ignored based upon the LTERM name.

If IGNORETERM= and TERMID= have the same value, TERMID is ignored.

You can specify up to 500 IGNORETERM= specifications.

IMSLEVEL= keyword

This keyword identifies the level of IMS that was used to create the IMS log.

The format of the IMSLEVEL= keyword is IMSLEVEL=*value* where *value* is the version of IMS. For example, 15 indicates IMS 15.

The IMSLEVEL= keyword is required.

Requirement: IMSLEVEL= control statement must specify one of the currently supported versions of IMS.

LCLTIME= keyword

This keyword identifies the difference between the local time and Coordinated Universal Time (UTC), the format in which log time stamps are stored. The use of this keyword adjusts the STRTTIME and STOPTIME values, which can be entered as local time values to be consistent with the STCK values on the log, which are in UTC. The value represents the hours and minutes that need to be added to or subtracted from local time to make it match UTC.

The format of the LCLTIME= keyword is LCLTIME=*shhmm* where:

- *s* is a sign value (+ or -).

s is required.

- *hh* is the hours difference between UTC and local time.

hh is required.

- *mm* is the additional minutes difference between UTC and local time.

mm is optional. Most time zones do not have additional minutes in the difference between UTC and local time.

The LCLTIME= keyword is optional. If omitted, the STRTTIME and STOPTIME values are considered to be UTC.

MSGLIST keyword

This keyword advises that records be written to the MSGLIST data set.

The format of the MSGLIST keyword is: MSGLIST

The MSGLIST keyword is optional. If omitted, no records are written to the MSGLIST data set.

PROCESS= keyword

This keyword specifies the number of records to be processed.

The format of the PROCESS= keyword is PROCESS=*value* where *value* is a 1- to 8-digit numeral.

The PROCESS= keyword is optional. If omitted, all of the records from where processing begins will be processed.

You can specify only one PROCESS= control statement.

SKIP= keyword

This keyword specifies the number of records to be skipped in the LOG file prior to processing records.

The format of the SKIP= keyword is SKIP=*value* where *value* is a 1- to 8-digit numeral.

The SKIP= keyword is optional. If omitted, no records are skipped prior to record processing.

You can specify only one SKIP= control statement.

STOPTIME= keyword

This keyword specifies the time of the first record to be processed. An exact match of time is not required; processing will begin with the first record equal to or greater than the indicated time.

The format of the STOPTIME= keyword is STOPTIME=*value*, where *value* is the 19-digit time that is used in the logs. The format of this value is *yyyymmddhhmmsssthmiju*. The last five digits are ignored.

The STOPTIME= keyword is optional. If omitted, the log will process until the end is reached, unless the PROCESS= keyword has been used.

STRTTIME= keyword

This keyword specifies the time of the first record to be processed. An exact match of time is not required; processing begins with the first record equal to or greater than the indicated time.

The format of the STRTTIME= keyword is STRTTIME=*value*, where *value* is the 19-digit time that is used in the logs. The format of this value is *yyyymmddhhmmsssthmiju*. The last five digits are ignored.

The STRTTIME= keyword is optional. If omitted, the log will be processed from the beginning of the file, unless the SKIP= keyword has been used.

TERMID= keyword

This keyword specifies an LTERM name for which you want to process associated records.

The format of the TERMID= keyword is TERMID=*lterm_name* where *lterm_name* is an 8-character field.

The TERMID= keyword is optional. If omitted, no records will be selected specifically based upon the TERMID= value.

If IGNORETERM= and TERMID= have the same value, TERMID is ignored.

You can specify up to 500 TERMID= specifications.

TRX= keyword

This keyword specifies a transaction code for which associated records are to be processed.

The format of the TRX= keyword is TRX=*transaction_code* where *transaction_code* is an 8-character field.

The TRX= keyword is optional. If omitted, no records will be selected specifically based upon the TRX= value.

You can specify up to 500 TRX= specifications.

USERID= keyword

This keyword specifies a user ID for which associated records are to be processed.

The format of the USERID= keyword is USERID=*userid* where *userid* is an 8-character field.

The USERID= keyword is optional. If omitted, no records are selected specifically based upon the USERID= value.

You can specify up to 500 USERID= specifications.

BTSAPRE0 JCL requirements

The JCL for the BTSAPRE0 module, which prepares the data to be processed through IMS Batch Terminal Simulator, must satisfy certain JCL requirements.

Subsections:

- [“JCL statements” on page 171](#)
- [“CNTLCRDS control statement keywords” on page 172](#)
- [“PAGECNTL control statement keywords” on page 174](#)

JCL statements

The following list summarizes the JCL statements for BTSAPRE0 JCL.

BTSIN (Input)

Additional statements required by in the BTSIN of IMS Batch Terminal Simulator will be written.

The DCB for this data set is RECFM=FB,LRECL=80.

CNTLCRDS (Input)

Contains the CNTLCRDS control statements.

The DCB for this data set is RECFM=FB,LRECL=80.

CROSSREF (Output)

Formatted extracts of the '01'x and '03'x records from BTSALOG0 are written. An additional sequence number is prefixed to the first record for each message.

The DCB for this data set is RECFM=VB,LRECL=32756.

PAGECNTL (Input)

Input file used to specify the number of physical pages to be associated with a MOD name, in the case of a multi-page MOD.

The DCB for this data set is RECFM=FB,LRECL=80.

STEPLIB (Input)

Describes the library that contains Playback.

SYSPRINT (Output)

Diagnostic messages and summary reports will be written.

The DCB for this data set is RECFM=FB,LRECL=133.

TRXIN (Input)

Contains the formatted extracts of the '01'x and '03'x records created by BTSALOG0.

The DCB for this data set is RECFM=VB,LRECL=32756.

TRXOUT (Output)

Formatted extracts of the '01'x and '03'x records from BTSALOG0 are written, each with an additional sequence number prefixed.

The DCB for this data set is RECFM=VB,LRECL=32756.

CNTLCRDS control statement keywords

Playback processing can be influenced by the CNTLCRDS control statements that are read from the CNTLCRDS file.

Most control statements are optional. The default result of statement omission is discussed for each keyword.

All of the keywords are optional. If none of the IGNORE-related keywords are used, no records will be specifically bypassed for processing. If none of the selection-related keywords are used, all records will be candidates for processing.

If a record matches some criteria for processing and also matches some criteria for being ignored, it will not be processed.

Related reading: For syntax rules, see [“Syntax rules for the control statements” on page 177](#).

The following keywords apply to BTSAPRE0.

IGNOREMOD= keyword

This keyword specifies a MOD name for which associated records will not be processed.

The format of the IGNOREMOD= keyword is IGNOREMOD=*modname* where *modname* is an 8-character field.

Additional allowable *modname* value:

- BLNK#VAL - eight characters of blank

The IGNOREMOD= keyword is optional. If omitted, no records will be ignored based upon the MOD name.

If IGNOREMOD= and MODNAME= have the same value, MODNAME is ignored.

You can specify up to 1000 IGNOREMOD= specifications.

IGNORETERM= keyword

This keyword specifies an LTERM name for which associated records will not be processed.

The format of the IGNORETERM= keyword is IGNORETERM=*lterm_name* where *lterm_name* is an 8-character field.

Additional allowable *lterm_name* values are:

- HIGH#VAL - eight characters of 'FF'X
- NULL#VAL - eight characters of '00'X

The IGNORETERM= keyword is optional. If omitted, no records will be ignored based upon the LTERM name.

You can specify up to 1000 IGNORETERM= specifications.

IGNORETYPE= keyword

This keyword specifies a log type, for which associated records will not be processed.

The format of the IGNORETYPE= keyword is IGNORETYPE=*log_type* where *log_type* is '01' or '03'.

The IGNORETYPE= keyword is optional. If omitted, no records are ignored based upon the log type.

MODNAME= keyword

This keyword specifies a MOD name for which associated records will be processed.

The format of the MODNAME= keyword is MODNAME=*modname* where *modname* is an 8-character field.

The MODNAME= keyword is optional. If omitted, no records will be selected specifically based upon the MOD name.

If IGNOREMOD= and MODNAME= have the same value, MODNAME is ignored.

You can specify up to 1000 MODNAME= specifications.

PROCESS= keyword

This keyword specifies the number of records to be processed from the CROSSREF file.

The format of the PROCESS= keyword is PROCESS=*value* where *value* is a 1- to 8-digit numeral.

The PROCESS= keyword is optional. If omitted, all of the records from where processing begins are processed.

You can specify only one PROCESS= control statement.

SKIP= keyword

This keyword specifies the number of records to be skipped in the TRXIN file prior to processing records.

The format of the SKIP= keyword is SKIP=*value* where *value* is a 1- to 8-digit numeral.

The SKIP= keyword is optional. If omitted, no records are skipped prior to record processing.

You can specify only one SKIP= control statement.

TRX= keyword

This keyword specifies a transaction code for which associated records will be processed.

The format of the TRX= keyword is TRX=*transaction_code* where *transaction_code* is an 8-character field.

The TRX= keyword is optional. If omitted, no records will be selected specifically based upon the TRX= value.

You can specify up to 1000 TRX= specifications.

TRXLINK keyword

This keyword specifies that the records associated with the specified transaction are also processed.

Use this keyword together with the TRX= keyword. When both the TRX= keyword and the TRXLINK keyword are specified, in addition to the records identified by the TRX= keyword, BTSAPRE0 also processes the records that have the same originating timestamp as the records identified by the TRX= keyword.

To specify the TRXLINK keyword, do not code the TRX= keyword in CNTLCRDS DD of the BTSALOG0 module. If coded, necessary records will not be extracted in the BTSALOG0 step.

The TRXLINK keyword is optional. If this keyword is omitted and the TRX= keyword is specified, only the records that contain the specified transaction code are processed.

USERID= keyword

This keyword specifies a user ID for which associated records will be processed.

The format of the USERID= keyword is USERID=*userid* where *userid* is an 8-character field.

The USERID= keyword is optional. If omitted, no records are selected specifically based upon the USERID= value.

You can specify up to 1000 USERID= specifications.

PAGECNTL control statement keywords

Playback supports screen processing when there are multiple physical pages associated with a MOD. However, Playback cannot automatically detect which MODS have multiple pages. You can use the PAGECNTL file, which is used by BTSAPRE0, to specify the number of physical pages that are associated with a MOD. If there is only one page associated with a MOD, you do not need to use the PAGECNTL file.

Records that contain the control statements that are described in this topic are read from the PAGECNTL file.

Related reading: For syntax rules, see [“Syntax rules for the control statements” on page 177](#).

The following keywords apply to the PAGECNTL DD statement.

MODNAME= keyword

This keyword specifies a MOD name for which associated records will be processed.

The format of the MODNAME= keyword is MODNAME=*modname* where *modname* is an 8-character field.

The MODNAME= keyword is required.

You can specify up to 1000 MODNAME= specifications.

PAGES= keyword

This keyword specifies the number of physical pages that are present in the indicated MOD name.

The format of the PAGES= keyword is PAGES=*value* where *value* is a 1- to 2-digit numeric.

The PAGES= keyword is required.

Notes:

- If all the MODs you want to process are single page MODs, no PAGECNTL entries are required. However, if any entries are coded, both keywords must be specified on the same control statement record.
- Multiple occurrences of this pairing are allowed. But you must specify at least one pair on one line.
- If you specify multiple MODNAME= statements, the value of the first MODNAME= is used and others are ignored. If you specify multiple PAGES= on one line, the value of the first PAGES= is used and others are ignored.

BTSAISR0 JCL requirements

The JCL for the BTSAISR0 module, which calls IMS Batch Terminal Simulator to process the extracted data by DLI ISRT calls, must satisfy certain JCL requirements.

Subsections:

- [“JCL statements” on page 174](#)
- [“CNTLCRDS control statement keywords” on page 175](#)

JCL statements

The following list summarizes the JCL statements for BTSAISR0 JCL.

BTSIN (Input)

The concatenation for this file includes the data set created by BTSAPRE0.

BTSOUT (Output)

The output report created by IMS Batch Terminal Simulator will be written.

The DCB for this data set is RECFM=FB,LRECL=133.

CNTLCRDS (Input)

Contains the CNTLCRDS control statements.

The DCB for this data set is RECFM=FB,LRECL=80.

FORMAT (Input)

Contains the MFS format library on which the IMS log was made.

IMS (For DLI execution) (Input)

Contains the PSB library to be used by IMS Batch Terminal Simulator.

The PSB is referenced in the PSB= keyword of the ./T control statement in BTSIN.

IMSACB (For DBB execution) (Input)

Contains the ACB library to be used by IMS Batch Terminal Simulator.

The ACB is referenced in the PSB= keyword of the ./T control statement in BTSIN.

SNAPTRX (Output)

If a U3501 abend occurs, this data set contains the last two TRXIN records that have been processed. These records are used to diagnose the cause of the U3501 abend.

The DCB for this data set is RECFM=VB,LRECL=32756.

STEPLIB (Input)

Describes the library that contains Playback.

SYSPRINT (Output)

Diagnostic messages and summary reports will be written.

The DCB for this data set is RECFM=FB,LRECL=133.

TRXIN (Input)

Contains the formatted extracts of the '01'x and '03'x records from BTSAPRE0.

The DCB for this data set is RECFM=VB,LRECL=32756.

CNTLCRDS control statement keywords

Playback processing can be influenced by the CNTLCRDS control statements that are read from the CNTLCRDS file.

Most control statements are optional. The default result of statement omission is discussed for each keyword.

Related reading: For syntax rules, see [“Syntax rules for the control statements” on page 177](#).

The following keywords apply to BTSAISR0.

DEFMOD= keyword

This keyword specifies the default MOD name that will be used if a record to be processed contains blank characters for the MOD name.

The format of the DEFMOD= keyword is DEFMOD=*modname* where *modname* is an 8-character field.

The DEFMOD= keyword is optional. If omitted, the value 'DFSMO3' will be used as a default MOD name.

PROCESS= keyword

This keyword specifies the number of records to be processed from the CROSSREF file.

The format of the PROCESS= keyword is PROCESS=*value* where *value* is a 1- to 8-digit numeric.

The PROCESS= keyword is optional. If omitted, all of the records from where processing begins are processed.

You can specify only one PROCESS= control statement.

Requirement: If the PROCESS= statement is used in BTSAISR0, the statement must also be used in the BTSAEDT0.

SKIP= keyword

This keyword specifies the number of records you want to skip in the TRXIN file prior to processing records.

The format of the SKIP= keyword is SKIP=*value* where *value* is a 1- to 8-digit numeral.

The SKIP= keyword is optional. If omitted, no records will be skipped prior to record processing.

You can specify only one SKIP= control statement.

Requirement: If the SKIP= statement is used in BTSAISR0, the statement must also be used in the BTSAEDT0.

BTSAEDT0 JCL requirements

The JCL for the BTSAEDT0 module, which edits the BTSOUT report that is created by IMS Batch Terminal Simulator, must satisfy certain JCL requirements.

Subsections:

- [“JCL statements” on page 176](#)
- [“CNTLCRDS control statement keywords” on page 176](#)

JCL statements

The following list summarizes the JCL statements for BTSAEDT0 JCL.

BTSOUT (Input)

Contains the output report created by IMS Batch Terminal Simulator.

The DCB for this data set is RECFM=FB,LRECL=133.

CNTLCRDS (Input)

Contains the CNTLCRDS control statements

The DCB for this data set is RECFM=FB,LRECL=80.

CROSSREF (Input)

Contains the formatted extracts of the '01'x and '03'x records from BTSAPRE0.

The DCB for this data set is RECFM=VB,LRECL=32756.

OUTFILE (Output)

A report containing the transaction screens will be written.

The DCB for this data set is RECFM=FB,LRECL=133.

STEPLIB (Input)

Describes the library that contains Playback.

SYSPRINT (Output)

Diagnostic messages and summary reports will be written.

The DCB for this data set is RECFM=FB,LRECL=133.

CNTLCRDS control statement keywords

Playback processing can be influenced by the CNTLCRDS control statements that are read from the CNTLCRDS file.

Most control statements are optional. The default result of statement omission is discussed for each keyword.

Related reading: For syntax rules, see [“Syntax rules for the control statements”](#) on page 177.

PROCESS= keyword

This keyword specifies the number of screens you want to process.

The format of the PROCESS= keyword is PROCESS=*value* where *value* is a 1- to 8-digit numeral.

The PROCESS= keyword is optional. If omitted, all of the records from where processing begins are processed.

Only one PROCESS= control statement is allowed.

Requirement: If the PROCESS= statement was used in BTSAISRO, it must also be used in the BTSAEDT0. The value associated with PROCESS= can be equal to or less than the value used in BTSAISRO.

You can specify the PROCESS= keyword in BTSAEDT0 even if it was not used in BTSAISRO.

SKIP= keyword

This keyword specifies the number of records you want to skip in the CROSSREF file prior to processing records.

The format of the SKIP= keyword is SKIP=*value* where *value* is a 1- to 8-digit numeral.

The SKIP= keyword is optional. If omitted, no records are skipped prior to record processing.

Only one SKIP= control statement is allowed.

Requirement: If the SKIP= statement was used in BTSAISRO, it must also be used in the BTSAEDT0. The value associated with SKIP= must be equal to the value used in BTSAISRO.

Syntax rules for the control statements

You must use the correct syntax when you code the CNTLCRDS and the PAGECNTL control statements for the Playback utility.

- Control statement records are 80 byte fixed length.
- Comment statements can be indicated with an asterisk (*) in column 1.
- Blank records are ignored.
- Control statements keywords are coded within the boundaries of columns 1 through 72.
- Keywords can start in any column.
- Keywords can be in any order.
- There can be no intervening blanks between keywords indicating a data value and the value itself.
- Multiple keywords are either:
 - Separated by one or more blanks.
 - Specified on multiple control statement records.
- Keywords for which multiple occurrences are allowed must have each occurrence specified on a separate control statement record.
- Individual keywords and their associated values cannot span or be continued on multiple control statement records.
- Keywords must be in upper case.
- For PAGECNTL, all control statements are optional. If no control statements are supplied, Playback treats each display as a single page screen.

Playback utility reports

Each Playback module creates a report that summarizes the process.

The following topics provide examples of Playback reports. The examples are organized by module.

BTSARCNO: RECON Query Summary report

The BTSARCNO module creates a RECON Query Summary report in the REPORT data set.

The following figure shows an example of the RECON Query Summary report.

```
BTS V4R1 PLAYBACK: RECON QUERY SUMMARY REPORT                                DATE: 2021/154  TIME: 14:08  PAGE: 1
BTS1002I CNTLCRDS:  SSID=SYS3
BTS1002I CNTLCRDS:  LOGTYPE=OLDS
BTS1002I CNTLCRDS:  STRTTIME=21153161850
BTS1002I CNTLCRDS:  STOPTIME=21153161930

NUMBER OF CNTLCRDS RECORDS READ   :      4
NUMBER OF SYSPRINT LINES READ     :     100
MAX. ALLOWABLE NUMBER OF LOG DSNS :     100
NUMBER OF LOG DSNS SELECTED       :      1

SELECTED LOG DSN=IMSTESTL.IMS01.OLDSP1
```

Figure 37. BTSARCNO: RECON Query Summary report

In this example:

- The IMS SSID was SYS3.
- OLDS type logs were requested.
- An STRTTIME and STOPTIME range was requested.
- There were four records read from the CNTLCRDS file.
- 100 records were read from the SYSPRINT file in order to determine the appropriate log data sets.
- The default number of log data sets (100) was the maximum number of logs allowed to be processed, based upon the selection criteria for STRTTIME and STOPTIME.
- One log data set was selected for processing.
- The log selected was IMSTESTL.IMS01.OLDSP1.

The following figure shows another example of this report.

```
BTS V4R1 PLAYBACK: RECON QUERY SUMMARY REPORT                                DATE: 2021/154  TIME: 14:08  PAGE: 1
BTS1002I CNTLCRDS:  SSID=SYS3
BTS1002I CNTLCRDS:  LOGTYPE=SLDS

NUMBER OF CNTLCRDS RECORDS READ   :      2
NUMBER OF SYSPRINT LINES READ     :      78
MAX. ALLOWABLE NUMBER OF LOG DSNS :     100
NUMBER OF LOG DSNS SELECTED       :      3

SELECTED LOG DSN=IMSVS.SLDSP.SYS3.D11153.T1618169.V00
SELECTED LOG DSN=IMSVS.SLDSP.SYS3.D11153.T1618488.V00
SELECTED LOG DSN=IMSVS.SLDSP.SYS3.D11153.T1619375.V00
```

Figure 38. BTSARCNO: RECON Query Summary report (second example)

In this example:

- The IMS SSID was SYS3.
- SLDS type logs were requested.
- No specific time range was selected.
- There were two records read from the CNTLCRDS file.
- 78 records were read from the SYSPRINT file in order to determine the appropriate log data sets.
- The default number of log data sets (100) was the maximum number of logs allowed to be processed, based upon the selection criteria for STRTTIME and STOPTIME.
- Three log data set were selected for processing.
- The log data set names are displayed in the report.

BTSALOG0: Summary report

The BTSALOG0 module creates a summary report in the REPORT data set.

The following figure shows an example of the report.

```
*****
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 1 *
*
*   BTS1002I: CNTLCRDS:  IMSLEVEL=15                                *
*   BTS1002I: CNTLCRDS:  USERID=ARS#USR1                          *
*   *****                                                         *
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 2 *
*
*   INPUT LOG DATA SET NAME(S)                                     *
*   HLQ.PRODIM.SYSV15.OLP00                                       *
*   *****                                                         *
*   LOG INFORMATION SUMMARY                                         *
*
*   FIRST LOG RECORD STCK (UTC) : 2021153 0161114                 *
*   LAST LOG RECORD STCK (UTC) : 2021153 0173152                 *
*   FIRST SELECTED LOG STCK (UTC) : 2021153 0161114                 *
*
*   TOTAL # OF LOG RECORDS READ : 608610                          *
*   # OF LOG RECORDS WRITTEN TO TRXOUT FILE : 436                  *
*   # OF MESSAGES CONTAINED IN TRXOUT FILE : 436                  *
*   # OF LOG RECORDS EXCLUDED FROM TRXOUT FILE : 0                 *
*   # OF SPA-ONLY 03 RECORDS EXCLUDED FROM TRXOUT : 225           *
*
*   NUMBER OF USERS WHO HAVE QUEUED MESSAGES : 276                 *
*   NUMBER OF MODS REFERENCED IN MESSAGES : 170                   *
*   MAXIMUM NUMBER OF CONCURRENT QUEUE MESSAGES : 0                *
*
*   *****                                                         *
*   LOG RECORD OCCURRENCES STATISTICS                               *
*
*   OCCURRENCES OF RECORD TYPE 01: 13506                          *
*   OCCURRENCES OF RECORD TYPE 03: 26393                          *
*   *****                                                         *
*   *****                                                         *
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 3 *
*
*   TRANSACTION LOG RECORDS STATISTICS                               *
*
*   TRANSACTION ARSTRX01 EXECUTED: 164                             *
*   TRANSACTION ARSTRX02 EXECUTED: 8                               *
*   TRANSACTION ARSTRX03 EXECUTED: 1                               *
*   TRANSACTION ARSTRX04 EXECUTED: 3                               *
*   TRANSACTION ARSTRX05 EXECUTED: 1                               *
*   *****                                                         *
*   *****                                                         *
*   *****                                                         *
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 4 *
*
*   USERID ACTIVITY STATISTICS                                     *
*
*   MESSAGES QUEUED BY USER ARSUSR01 : 308                        *
*   MESSAGES QUEUED BY USER ARSUSR02 : 20                         *
*   MESSAGES QUEUED BY USER ARSUSR03 : 7                          *
*   MESSAGES QUEUED BY USER ARSUSR04 : 415                        *
*   MESSAGES QUEUED BY USER ARSUSR05 : 24                         *
*   MESSAGES QUEUED BY USER ARSUSR06 : 16                         *
*   *****                                                         *
*   *****                                                         *
*   *****                                                         *
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 5 *
*
*   MESSAGE MODNAME LIST                                           *
*
*   MESSAGES REFERENCING MODNAME: ARSMOD01 : 13                   *
*   MESSAGES REFERENCING MODNAME: ARSMOD02 : 1                     *
*   MESSAGES REFERENCING MODNAME: ARSMOD03 : 7                     *
*   MESSAGES REFERENCING MODNAME: ARSMOD04 : 1443                  *
*   MESSAGES REFERENCING MODNAME: ARSMOD05 : 3                     *
*   MESSAGES REFERENCING MODNAME: ARSMOD06 : 1                     *
*   *****                                                         *
*   *****                                                         *
*   *****                                                         *
*   BTS V4R1 PLAYBACK: BTSALOG0 SUMMARY                               DATE: 2021/154  TIME: 14:09  PAGE: 6 *
*
*   TRANSACTION USAGE LIST                                         *
*
*   TRANSACTION ARSTRX01 BY USER ARSUSR01 EXECUTED: 31           *
*   TRANSACTION ARSTRX01 BY USER ARSUSR02 EXECUTED: 7             *
*   TRANSACTION ARSTRX01 BY USER ARSUSR03 EXECUTED: 4             *
*   TRANSACTION ARSTRX02 BY USER ARSUSR01 EXECUTED: 3             *
*   TRANSACTION ARSTRX02 BY USER ARSUSR03 EXECUTED: 1             *
*   TRANSACTION ARSTRX03 BY USER ARSUSR01 EXECUTED: 1             *
*   TRANSACTION ARSTRX03 BY USER ARSUSR04 EXECUTED: 31            *
*   TRANSACTION ARSTRX03 BY USER ARSUSR05 EXECUTED: 17            *
*   *****                                                         *
*   *****                                                         *
*   *****                                                         *
*   END OF IMS LOG SUMMARY REPORT                                   *
*   *****
```

Figure 39. BTSALOG0: Summary report

In this example:

- The log that was processed is from IMS 15.
- Only the records that are related to user ARS#USR1 were processed.
- The log data set name is 'HLQ.PRODIM.SYSV15.OLP00'.
- The first time on the log is 2021153 0161114.

- The last time on the log is 2021153 0173152.
- Processing began at time 2021153 0161114.
- Processing began at the first record on the log.
- 608620 records were read from the log.
- 436 log records were written for processing.
- The 436 log records represented 436 messages. In this case, each '01'x or '03'x was a complete message.
- No records were excluded from TRXOUT due to IGNORETERM matches.
- 225 '03'x records represented complete messages, but only contained SPA-related data, and were not included in TRXOUT.
- 276 users were queueing messages.
- 170 MODs were referenced in messages.
- Of the 608620 log messages:
 - 13506 were '01'x messages
 - 26393 were '03'x messages
- The codes and number of executions of transactions were identified (the list was abridged for this document).
- The user IDs and number of messages queued were identified (the list was abridged for this document).
- The MODs and number of messages referencing them were identified (the list was abridged for this document).
- Transaction codes and the number of times individual users executed them were identified (the list was abridged for this document).

BTSAPRE0: Summary report

The BTSAPRE0 module creates a summary report in the SYSPRINT data set.

The following figure shows an example of the summary report.

BTS V4R1 PLAYBACK: BTSAPRE0 SUMMARY REPORT		DATE: 2021/154	TIME: 14:07	PAGE: 1
NUMBER OF CNTLCRDS RECORDS READ	:	0		
NUMBER OF TRXIN RECORDS READ	:	436		
NUMBER OF TRXIN RECS SKIPPED	:	0		
NUMBER OF CROSSREF RECORDS WRITTEN	:	436		
NUMBER OF TRXOUT RECORDS WRITTEN	:	436		
NUMBER OF SINGLE-PAGE SCREEN DISPLAYS	:	436		
NUMBER OF MULTI-PAGE SCREEN DISPLAYS	:	0		

Figure 40. BTSAPRE0: Summary report

In this example:

- There were no CNTLCRDS control records included in the input.
- 436 records were read from the TRXIN file.
- No records were skipped.
- 436 records were written to the CROSSREF file. In this case, each of the QUEUE log records in the TRXIN file contained all of the data required for a transaction.
- 436 reformatted records were written to the TRXOUT file.
- 436 of the screen displays will be single-page displays.
- None of the screen displays will be multi-page displays.

The following figure shows another example of this report.

BTSA1002I CNTLCRDS: USERID=ARS#USR1
 BTSA1002I CNTLCRDS: USERID=ARS#USR2
 BTSA1002I CNTLCRDS: TRX=ARSTRX00

```

NUMBER OF CNTLCRDS RECORDS READ      :      3
NUMBER OF TRXIN RECORDS READ         :    50682
NUMBER OF TRXIN RECS SKIPPED         :      0
NUMBER OF CROSSREF RECORDS WRITTEN   :    1852
NUMBER OF TRXOUT RECORDS WRITTEN     :    2203
NUMBER OF SINGLE-PAGE SCREEN DISPLAYS :    1852
NUMBER OF MULTI-PAGE SCREEN DISPLAYS :      0

```

ACTION	TYPE	NAME	CROSSREF	TRXOUT
SELECTED	USERID	ARS#USR1	400	633
SELECTED	USERID	ARS#USR2	251	251
SELECTED	TRX	ARSTRX00	1201	1319

Figure 41. BTSAPRE0: Summary report (second example)

In this example:

- Activity related to user ARS#USR1 was requested.
- Activity related to user ARS#USR2 was requested.
- Activity related to transaction ARSTRX00 was requested.
- Three records were read from the CNTLCRDS file.
- 50682 records were read from the TRXIN file.
- No records were skipped.
- 1852 records were written to the CROSSREF file.
- 2203 reformatted records were written to the TRXOUT file.
- 1852 of the screen displays will be single-page displays.
- None of the screen displays will be multi-page displays.
- As a result of processing user ARS#USR1:

- 400 records were written to the CROSSREF file.

This action indicates that there will ultimately be 400 screens produced related to this user.

- 633 records were written to the TRXOUT file.

This action indicates that 633 queue log records will be involved to produce the 400 screens. In this case, some of the queue log records do not represent complete records; some of the screens are a result of multiple queue log records.

- As a result of processing user ARS#USR2:

- 251 records were written to the CROSSREF file.
- 251 records were written to the TRXOUT file.

In this case, each of the queue records represents a complete record.

- As a result of processing transaction ARSTRX00:

- 1201 records were written to the CROSSREF file.
- 1319 records were written to the TRXOUT file.

In this case, each of the queue records represents a complete record.

BTSAISR0: Summary report

The BTSAISR0 module creates a summary report in the SYSPRINT data set.

The following figure shows an example of the summary report.

```

NUMBER OF CNTLCRDS RECORDS READ      :      0
NUMBER OF TRXIN RECS READ             :     436
NUMBER OF TRXIN RECS SKIPPED          :      0
NUMBER OF 1ST MESSAGE RECORDS READ   :     436
NUMBER OF MESSAGE SEGMENTS INSERTED   :     436
NUMBER OF MESSAGES PURGED             :     436
NUMBER OF SPA-ONLY 01 TRXIN RECS     :      0
NUMBER OF SPA-TERMINATING 03 TRXIN RECS :      0

```

Figure 42. BTAISR0: Summary report

In this example:

- There were no CNTLCRDS control records included in the input.
- 436 records were read from the TRXIN file.
- No records from the TRXIN file were skipped.
- 436 records contained the first part of a queue message. In this case, each log record represented a complete message.
- 436 messages were inserted. In this case, each message consisted of a single segment.
- 436 messages were purged.
- No '01'x records contained only SPA-related data.
- No '03'x records contained only SPA-related data.

The following figure shows another example of this report.

```

BTS V4R1 PLAYBACK: BTAISR0 SUMMARY REPORT
BTA1002I CNTLCRDS: DEFMOD=DFSDSP01
DATE: 2021/154 TIME: 16:06 PAGE: 1

NUMBER OF CNTLCRDS RECORDS READ      :      1
NUMBER OF TRXIN RECS READ             :     2203
NUMBER OF TRXIN RECS SKIPPED          :      0
NUMBER OF 1ST MESSAGE RECORDS READ   :     1852
NUMBER OF MESSAGE SEGMENTS INSERTED   :     2091
NUMBER OF MESSAGES PURGED             :     1852
NUMBER OF SPA-ONLY 01 TRXIN RECS     :      199
NUMBER OF SPA-TERMINATING 03 TRXIN RECS :      181

```

Figure 43. BTAISR0: Summary report (second example)

In this example:

- The default MOD name of DFSDSP01 is used for those transactions with a blank value for MODNAME.
- One record was read from the CNTLCRDS file.
- 2203 records were read from the TRXIN file.
- No records from the TRXIN file were skipped.
- 1852 records contained the first part of a queue message.
- 2091 messages were inserted.
- 1852 messages were purged.
- 199 '01'x records contained only SPA-related data.
- 181 '03'x records contained only SPA-related data.

BTSAEDT0: Screen Image Summary report

The BTSAEDT0 module creates a summary report for the screen image in the SYSPRINT data set.

The following figure shows an example of the report.

B TSA1002I CNTLCRDS: *SKIP=20

```

NUMBER OF CNTLCRDS RECORDS READ :      1
NUMBER OF BTSOUT RECORDS READ   :   36067
NUMBER OF CROSSREF RECORDS READ :    436
NUMBER OF CROSSREF RECORDS SKIPPED:      0
NUMBER OF MODS PROCESSED       :    436
NUMBER OF SCREENS PRODUCED     :    436
NUMBER OF RECORD LINES WRITTEN :   23108

```

Figure 44. B TSAEDT0: Screen Image Summary report

In this example:

- The SKIP=20 control card was included as a comment.
- One record was read from the CNTLCRDS file.
- 36067 records were read from the BTSOUT file.
- 436 records were read from the CROSSREF file.
- No records from the CROSSREF file were skipped.
- 436 MODs from the CROSSREF file were processed.
- 436 screens were produced as a result of processing the TRXIN records.
- 23108 lines were written to produce the 436 screens.

The following figure shows another example of this report.

```

NUMBER OF CNTLCRDS RECORDS READ :      0
NUMBER OF BTSOUT RECORDS READ   :  135938
NUMBER OF CROSSREF RECORDS READ :   1852
NUMBER OF CRPSSREF RECORDS SKIPPED:      0
NUMBER OF MODS PROCESSED       :   1852
NUMBER OF SCREENS PRODUCED     :   1852
NUMBER OF RECORD LINES WRITTEN :   98156

```

Figure 45. B TSAEDT0: Screen Image Summary report (second example)

In this example:

- There were no CNTLCRDS control records included in the input.
- 135938 records were read from the BTSOUT file.
- 1852 records were read from the CROSSREF file.
- No records from the CROSSREF file were skipped.
- 1852 MODs from the CROSSREF file were processed.
- 1852 screens were produced as a result of processing the TRXIN records.
- 98156 lines were written to produce the 436 screens.

B TSAEDT0: Sample formatted screen

The B TSAEDT0 module creates formatted screen in the OUTFILE data set.

The following figure shows an example of the formatted screen.

```

USER: ARS#USR1   TERMINAL: TERM0001   PROCESSING DATE: 2021/153 TIME: 23:47
TRANSACTION: ARSTRX00   MID/MOD= ARSMOD02, CURSOR= L22C001, ACTION= INPUT
-----1-----2-----3-----4-----5-----6-----7-----8
UNCLASSIFIED
ACTION:                SAMPLE TRANSACTION
                        ECCN#: ARS#01A
TEST#: 3108 00   RST#: 9999999999   RST# EFF: 20/09/21   CCC#: 01N
CHG DATE: 20/09/21   BY: ARS010   REV IND: Y   REV DATE: 20/09/21   BY: ARS#USR1
TEST# DESC: SAMPLE ITEM
RST# DESC: SAMPLE COMP :NSPF   LOAD DATE: 20/09/21
PIE IMP:   PIE EXP:   ADD:   CVD:   FDA:   EPA:   DOT:   ASSIGN BY IPO:
PRIOR RST#:   FROM:   TO:
PRIOR RST#:   FROM:   TO:
PRIOR RST#:   FROM:   TO:
***** RST# INFORMATION *****
10/09/21   N
EFFECTIVE FROM:   21/01/01   RST# LAST CHANGED: 21/05/31
EFFECTIVE TO:    21/12/31   GSP STATUS:      A
COMPUTATION CODE:   7   U/M: 01- KG 02-   03-
CURRENT RATE:      0.03300000 (1)   0.25000000 (2)
CURRENT SPECIFIC:  0.00000000 (1)   0.00000000 (2)
OTHER RATES:       0.00000000 (1)   0.00000000 (2)

REQUEST:

PFKEYS: F1-MAIN MENU   F2-PAGE FWD   F3-PAGE BKWD   F9-CLEAR SCR
-----1-----2-----3-----4-----5-----6-----7-----8

```

Figure 46. B TSAEDT0: Sample formatted screen

In this example:

- The user ID associated with the screen display was ARS#USR1.
- The LTERM name associated with the screen display was TERM0001.
- The date and time of processing of the transaction was displayed.
- The transaction code associated with the screen display was ARSTRX00.
- The MOD name associated with the record was ARSMOD02.
- The log record from which the data was extracted was an Input queue record ('01'x).

If this record was an OUTPUT type of record, the report would have the following differences:

- "TRANSACTION: trrcode" would be replaced with "OUTPUT DEST: ltermname"
- "ACTION= INPUT" would be replaced with "ACTION= OUTPUT"

Part 5. Reference

These topics provide reference information that you need when using IMS Batch Terminal Simulator.

Topics:

- [Chapter 14, “Command reference,” on page 187](#)
- [Chapter 15, “Device feature indicator values,” on page 249](#)
- [Chapter 16, “BTS interface with a user-written BTSPIXT0 routine,” on page 251](#)
- [Chapter 17, “Playback utility record format,” on page 253](#)
- [Chapter 18, “How to read syntax diagrams,” on page 257](#)

Chapter 14. Command reference

IMS Batch Terminal Simulator supports simulator commands to define the transactions, simulator statements to supply transaction messages, and IMS commands to facilitate the testing activities.

Topics:

- [“Simulator commands” on page 187](#)
- [“Simulator statements” on page 239](#)
- [“IMS commands” on page 247](#)

Simulator commands

Simulator commands define the transactions to be performed, the logical terminal to be simulated, the format of the simulator statements and output listing, and the debugging aids that are used.

The `./T` command is the only simulator command that is required. The defaults for the other commands are assumed if not specified explicitly.

The following topics describe in detail each simulator command and its operands, and provide examples for each command.

Simulator command summary

IMS Batch Terminal Simulator provides many simulator commands to support testing of various applications. Each command has default keyword values that are defined internally by IMS Batch Terminal Simulator.

Subsections:

- [“Summary of simulator commands” on page 187](#)
- [“Default values and syntax specifications for simulator commands” on page 188](#)

Summary of simulator commands

The following table summarizes the simulator commands and shows their coded symbols.

Table 35. Simulator commands: Summary

Simulator command	Function
Transaction (<code>./T</code>)	Defines a transaction and the application program. Also used to define an alternate logical terminal.
COBOL (<code>./C</code>)	Defines the usage of interactive debug of some languages.
Device (<code>./D</code>)	Specifies the I/O PCB terminal-simulated device, the end-of-segment and end-of-message characters, whether a user-written output formatting routine or IMS Batch Terminal Simulator formatting is to be used, and the feature byte describing the features coded in the IMS MFS control blocks.
Environment (<code>./E</code>)	Defines the environment that the transaction is executed in.
Output (<code>./O</code>)	Specifies those items that are to be excluded from the IMS Batch Terminal Simulator output listing and TSO terminal display (for example, screen images, field attribute characters, application program statistics, and IMS Batch Terminal Simulator messages).
Patch (<code>./P</code>)	Specifies a modification to be made in main storage to a particular load module.

Table 35. Simulator commands: Summary (continued)

Simulator command	Function
Reader (./R)	Specifies whether the simulator statements are in character or hexadecimal format.
Snap (./S)	Specifies an area in main storage to be snapshot-dumped and the number of times for a particular PCB and IMS function code combination. It can also be used to set a given status code after a DL/I call.
SPA (./SPA)	Specifies an SPA to be initialized for passing to the application program.
Comment (./*)	Specifies the comments to be included in the printed output.
Debug commands	Specifies the debugging options to diagnose problems.

Default values and syntax specifications for simulator commands

The following table provides a summary of the commands and states the defaults of the keywords, indicating the length of their operands and whether their values are coded in EBCDIC character format or in hexadecimal notation. The maximum and minimum value for a given operand (where applicable) is also indicated.

Table 36. Simulator commands: Default values and syntax specifications

Simulator command	Keyword value	Default value	Rep	Length		Magnitude	
				Min.	Max.	Min.	Max.
./C		(N/A)					
./D	LTERM	IOPCB		1	8	-	-
	EOS	*		1	1	-	-
	EOM	\$		1	1	-	-
	DDOF	NONE		4	6	-	-
	TYPE			7	8	-	-
	SIZE			7	8	-	-
	LIMIT	0		1	1	-	-
	FEAT	7F		2	2	0	9
	TRSOSI	YES		1	3	-	-
	LUNAME	DFSLU		1	17	-	-
	TPNAME	DFSASYNC		1	64	-	-
	SYNCLVL	C		1	1	-	-
	CONVTYPE	M		1	1	-	-
	LINE	DELIMIT		1	7	-	-

Table 36. Simulator commands: Default values and syntax specifications (continued)

Simulator command	Keyword value	Default value	Rep	Length		Magnitude		
				Min.	Max.	Min.	Max.	
./E	APPLFE			1	8	-	-	
	ETO	NO		1	3	-	-	
	LCI	NO		1	3	-	-	
	MAXL	(X'003C')		4	4	-	-	
	SPACE	NO		1	3	-	-	
	REATTCH	NO		1	3	-	-	
	SSID	(required)		3	4	-	-	
	KATAKANA	NO		1	3	-	-	
	SYSRQ	NO		1	3	-	-	
	PA2	(required)		3	4	-	-	
	DT3270	NO		1	3	-	-	
	UGN	SYSDA		1	8	-	-	
	CONNECT	(N/A)		1	8	-	-	
	ERR	(N/A)		1	1	-	-	
	MLINE	60		dec	1	5	1	32767
	ECHKP	YES			1	3	-	-
	SCNDTRX	CONT			4	6	-	-
	USERID	blank			1	8	-	-
	USERIND	blank			1	1	-	-
	RCFGPN	blank			1	8	-	-
	IODATE	process date			8	8	-	-
	IOTIME	process time			8	8	-	-
	IOUDATE	process date			8	8	-	-
	IOUTIME	process time			15	51	-	-
	UIDDREG	blank			1	8	-	-
	SPLAPI	NO			1	3	-	-
	NOEDIT	NO			1	3	-	-
	IMSSUF				1	1	-	-
	FSEND	NO			1	3	-	-
	ENDROLB	NO			1	3	-	-
	PERMLOAD	NO			1	3	-	-
	IFIUTF	NO			1	3	-	-

Table 36. Simulator commands: Default values and syntax specifications (continued)

Simulator command	Keyword value	Default value	Rep	Length		Magnitude	
				Min.	Max.	Min.	Max.
./O	APS	YES		1	3	-	-
	SCREEN	INOUT		1	5	-	-
	ATR	YES		1	3	-	-
	DB	YES		1	3	-	-
	EATR	NO		1	3	-	-
	MSG	YES		1	3	-	-
	SPA	YES		1	3	-	-
	TSO	YES		1	3	-	-
	TSO Aid	NO		1	3	-	-
	TSODB	YES		1	6	-	-
	TSOMLVL	1		1	1	0	1
	TSOMSG	YES		1	6	-	-
	SQL	ALL		1	3	-	-
	TSOSQL	YES		1	6	-	-
	SQLHX	NO		1	3	-	-
	TSOSQLHX	NO		1	3	-	-
	IFI	YES		1	3	-	-
	TSOIFI	YES		1	3	-	-
	MQI	ALL		1	3	-	-
	TSOMQI	YES		1	6	-	-
	TSSLEN	114		1	4	0	9999
	SQLOBLN	(N/A)		1	4	0	9999
	ELAPTIME	NO		1	3	-	-
	REPORT	NO		1	3	-	-
	ICALOLEN	1000		1	4	0	9999
	./P	MBR	(required)	hex	1	8	-
CA		000000	hex	6	6	-	-
PA		(required)		6	6	-	-
PC		(X'00')	hex	1	137	-	-
PX		(X'00')		2	137	-	-
./R	IP	CHAR		3	4	-	-
./RETURN		(N/A)					

Table 36. Simulator commands: Default values and syntax specifications (continued)

Simulator command	Keyword value	Default value	Rep	Length		Magnitude	
				Min.	Max.	Min.	Max.
./S	MBR	(required)		1	8	-	-
	CA	000000	hex	6	6	-	-
	SA	(required)	hex	6	6	-	-
	BYTES	(required)	dec	1	5	1	32767
	PCB	(required)		1	8	-	-
	FUNC	(required)		2	4	-	-
	TIMES	1	dec	1	5	1	32767
	STCD			2	2	-	-
./SPA		(N/A)					
./T	TC	(required)		1	8	-	-
	MBR	(required)		1	8	-	-
	PSB	(PSB=MBR)		1	8	-	-
	PLAN	(PLAN=MBR)		1	8	-	-
	PLC	1	dec	1	5	0	32767
	SPA	0	dec	2	5	15	32767
	EM	(N/A)		1	8	-	-
	LANG	ASM		3	3	-	-
	TYPE	MSG		3	8	-	-
	DLITRACE	*		1	255	0	9
	SQLTRACE	*		1	255	0	B
	BCHKP	NO		1	3	-	-
	MQITRACE	*		1	255	0	B
	MSGTYPE	TRAN		4	6	-	-
	QNAME			1	48	-	-
	QMGRNAME			1	48	-	-
	REATTCH	NO		1	3	-	-

Note: For the ./T command, only the options for defining transactions are shown. See [“./T command”](#) on page 222 for the options for defining alternate logical terminals.

Syntax rules

You must use the correct syntax when you code IMS Batch Terminal Simulator simulator commands.

Syntax rules

The rules for coding simulator commands are as follows:

- All simulator commands begin with ./ and start in column 1. ./ is followed by the characters needed to form the particular command.
- When the BTSIN data set is in fixed or fixed-block format, the command keywords and their operands can be coded in any order in columns 5 - 71 of the first statement, and in columns 1 - 71 of any subsequent continuation. When the BTSIN data set is in variable or variable-block format, the command keywords and their operands can be coded in any order in columns 5 - 32752 of the statement.
- Each keyword specification must be preceded and followed by at least one blank space.
- When the BTSIN data set is in fixed or fixed-block format, a non-blank character in column 72 indicates continuation. The contents of column 72 are not used.
- The ./SPA statement has no keywords. The information to be passed to the scratchpad area (SPA) must start in column 7, and the first eight characters must be the transaction code. The end of the data must be terminated with the end-of-message indicator.

Exception: These syntax rules do not apply to the debug commands.

Conventions

The following graphic conventions are used throughout the description and summary:

- [] Optional
- { } Choose one
- _____ (**underscore**) Default assumed

./C command

Use the ./C command to define the usage of interactive debug of some languages.

The ./C command defines each application program that is to be run under control of COBOL interactive debug when IMS Batch Terminal Simulator is run under control of the BTS command processor (BTSCP).

Related reading: See [“Debugging with VS COBOL II interactive debug” on page 100](#).

Subsections:

- [“Usage notes” on page 192](#)
- [“Format” on page 192](#)
- [“Keywords and operands” on page 193](#)
- [“Example” on page 193](#)

Usage notes

The ./C command specification remains in effect throughout the IMS Batch Terminal Simulator run.

If more than one ./C command contains the same load module name, the latter specification is used.

Testing does not work under IMS Batch Terminal Simulator for dynamically linked COBOL subroutines. The symptom is that the ./C information is ignored or not processed.

You do not need to specify the ./C command when using Debug Tool under CODE/370.

Format

```
./C      COBTEST  load_module_name
```

Keywords and operands

COBTEST

Specifies the VS COBOL II interactive debug facility.

load_module_name

Defines the application program to be run under control of the COBOL interactive debug. The load module name must be a name defined by the MBR= operand of a ./T command.

Example

To run two COBOL application programs (APPLPGMA and APPB) under the control of the COBOL interactive debug, specify:

```
./T TC=TRANA MBR=APPLPGMA LAND=CBL
./T TC=TRANB MBR=APPB     LAND=CBL
./C COBTEST APPLPGMA
./C COBTEST APPB
./*
TRANA INPUT MESSAGE DATA $
```

./D command

Use the ./D command to define the device to be simulated as the I/O PCB logical terminal and to specify the device formatting facility that is to be used for logical terminal output.

Subsections:

- [“Usage note” on page 193](#)
- [“Format” on page 193](#)
- [“Keywords and operands” on page 194](#)
- [“Examples” on page 197](#)

Usage note

The values specified in the ./D command remain in effect until they are reset by another ./D command.

Format

For running with no special device formatting facility:

```
./D      [DDOF=      NONE]
         [EOM=      $ | character]
         [EOS=      * | character]
         [LINE=     DELIMIT | MSG | SEG]
         [LTERM=    IOPCB | name]
```

For running with the BTS-supplied 2740 Typewriter Output Formatting:

```
./D      DDOF=      27401z
         [EOM=      $ | character]
         [EOS=      * | character]
         [LINE=     DELIMIT | MSG | SEG]
         [LTERM=    IOPCB | name]
```

For running with the BTS-supplied 3270 Formatting Facility when using formats defined with symbolic types for 3270 devices:

```

./D    SIZE=          (lines,columns)
      TYPE=          symbolic_device_type
      [EOM=          $ | character]
      [FEAT=         7F | code]
      [LIMIT=        0 | count]
      [LINE=         DELIMIT | MSG | SEG]
      [LTERM=        IOPCB | name]
      [TRSOSI=       YES | NO]

```

For running with the BTS-supplied 3270 Formatting Facility when using formats defined as 3270-1 or 3270-2:

```

./D    DDOF=          3270mc
      [EOM=          $ | character]
      [FEAT=         7F | code]
      [LINE=         DELIMIT | MSG | SEG]
      [LTERM=        IOPCB | name]
      [TRSOSI=       YES | NO]

```

For running with an LU6.2 device:

```

./D    DDOF=          LU62
      [CONVTYPE=     M | B]
      [EOM=          $ | character]
      [EOS=          * | character]
      [LINE=         DELIMIT | MSG | SEG]
      [LUNAME=       DFSLU | luname | netwkid.luname]
      [SYNCLVL=     C | N]
      [TPNAME=       DFSASYNC | tpname]

```

Keywords and operands

CONVTYPE=

Specifies whether the conversation type is basic (B) or mapped (M). If not specified, the default value is M. If the DDOF=LU62 is not specified, the CONVTYPE= operand is ignored.

DDOF=

Specifies the device-dependent output formatter used for terminal output.

DDOF=NONE indicates that no special formatter is used. The output messages are presented as they appear in the user I/O area at the time of insertion to DL/I. The output is presented in the same format as it would appear at the device. The default value is DDOF=NONE.

DDOF=3270mc indicates that the BTS-supplied 3270 Formatting Facility is used to format output inserted to the logical terminal name specified by the LTERM= operand.

m

Is 1 for 3270-1 and 2 for 3270-2.

c

Is the limit count option and has the same values and meaning as the LIMIT operand.

DDOF=27401z indicates that the BTS-supplied 2740 Typewriter Output Formatting Facility is used to format output inserted to the logical terminal name specified by the LTERM= operand. The BTS-supplied 2740 Typewriter Output Formatting Facility offers line feed, carriage return, and skip-to-tab features. The type of output formatting is specified by the DDOF=27401z operand, where:

2740

Is the terminal type

1

Must always be 1, whether it is a 2740 Model 1 or Model 2.

z

Is one of the following values:

0

The output is to be formatted with the IMS Batch Terminal Simulator header and trailer, but without the I/O area of the application program.

1

The output is to be formatted with the IMS Batch Terminal Simulator header and trailer, and with the I/O area of the application program.

2

The output is to be formatted without the IMS Batch Terminal Simulator header and trailer, and without the I/O area of the application program.

Note: The BTS-supplied 2740 Typewriter Output Formatting Facility assumes a record length of 131 bytes on the BTSOUT data set (1 byte for carriage control plus 130 bytes for text). When the BTSOUT logical record length is not 131, formatted output might appear distorted.

DDOF=LU62 indicates that I/O PCB receives the incoming message from an LU6.2 destination.

EOM=

Specifies the graphic character that is interpreted by IMS Batch Terminal Simulator as the end-of-message indicator and is used to delimit input messages. The indicator is not included in the data passed to the application on GU or GN message calls. Comments can follow the end-of-message indicator.

EOS=

Specifies the graphic character that is interpreted by IMS Batch Terminal Simulator as the end of terminal segment indicator and is used to delimit input message segments. The indicator is not included in the data passed to the application on GU or GN message calls.

FEAT=

Specifies the device feature indicator values, in hexadecimal, that correspond to the device features that were specified in the message descriptor (MOD or MID) when the Message Format Service control blocks were generated.

IMS Batch Terminal Simulator uses the feature code specified with device type and format name to locate the correct IMSVS.FORMAT library member.

The default value of 7F corresponds to FEAT=IGNORE as coded in the Message Format Service control blocks.

Related reading:

- See [Chapter 15, “Device feature indicator values,”](#) on page 249 for the meaning of the device feature indicator values.
- For information about the indicator values and determining the hexadecimal indicator values that corresponds to the features that are specified in the control blocks that are being used, see *IMS Application Programming APIs*.

LIMIT=

Specifies the number of simulator statement errors allowed before the application is terminated. The default value is LIMIT=0. The limit count option is coded as follows:

0

The application program is to be terminated immediately upon detection of any statement error.

1 - 9

The limit count is multiplied by 10 resulting in a value in the range of 10 - 90. It is decremented by five for severe errors; otherwise, it is decremented by one. Severe errors also cause the rest of the current input message to be disregarded. When the limit count is decremented and a negative value is the result, the application is terminated. No more processing is allowed until a `./D` command is entered to reset the limit count value.

Examples of severe statement errors (message numbers with an A suffix) are:

- Unrecognized action keywords
- Invalid position reference
- Invalid IDCARD length
- Unmatched apostrophes

Other statement errors (message numbers ending with W) might include:

- An attempt to detect a undetectable field
- An attempt to enter data in a protected field

LINE=

Specifies how the input line is used.

DELIMIT

Indicates that the graphic characters that the EOM and EOS keywords specify are used to delimit the input line. The default value is `LINE=DELIMIT`.

MSG

Indicates that the entire input line, including any delimiters, is used as an input message. IMS Batch Terminal Simulator processes the line as if an end-of-message indicator is placed at the end of each line. The end-of-message indicator is not included in the data passed to the application on GU or GN message calls.

SEG

Indicates that the entire input line, including any delimiters, is used as an input segment. IMS Batch Terminal Simulator processes the line as if an end-of-terminal-segment indicator is placed at the end of each line. The end-of-terminal-segment indicator is not included in the data passed to the application on GU or GN message calls.

LTERM=

Specifies the logical terminal name for the I/O PCB. The default value is `LTERM=IOPCB`.

LUNAME=

Specifies the input LU name for LU6.2 conversation. If the LU name is a network qualified name, it can be up to 17 characters long and consists of the network ID of the original system followed by ':' and LU name (for example, `netwkid.luame`). The LU name and the network ID must be a 1- to 8-character name. The default value is `DFSLU`. If the `DDOF=LU62` is not specified, the `LUNAME=` operand is ignored.

SIZE=(lines,columns)

Specifies screen image size in lines (screen image height) and columns (screen image width). The two numeric values must be separated by a comma. For example, a 24 x 80 character screen image would be specified as `SIZE=(24,80)`.

SYNCLVL=

Specifies whether the APPC/IMS sync level is confirmed (C) or not (N). If not specified, the default value is C. If the `DDOF=LU62` is not specified, the `SYNCLVL=` operand is ignored.

TPNAME=

Specifies a 1- to 64-character name of the inbound TP. If not specified, the default name is `DFSASync`. If the `DDOF=LU62` is not specified, the `TPNAME=` operand is ignored.

TRSOSI=

Indicates the options for shift out (SO) and shift in (SI) character translation in the 3270 formatted screen image.

YES

Indicates that SO and SI characters are to be translated to X'40'. The default value is TRSOSI=YES.

NO

Indicates that SO and SI characters are not to be translated.

TYPE=3270-An

Specifies the symbolic device type, where *n* is a value in the range of 1 - 15. This operand must be entered exactly as coded in the TYPE keyword in the MFS DEV macro for the device that is being simulated.

Examples**Example 1 for ./D command**

Assume that the application checks the PCB name field of its I/O PCB as part of its processing logic, and that a message from TERM1 is followed by a message from TERM2. Also, assume that the data in the input message contains dollar signs (\$). Two ./D commands are required in your input stream:

<code>./T TC=TESTIT MBR=TEST</code>	1
<code>./D LTERM=TERM1 EOM=#</code>	2
<code>TESTIT \$99.99 CR #</code>	3
<code>./D LTERM=TERM2 EOM=#</code>	4
<code>TESTIT \$99.99 CR #</code>	5

where:

- 1** Default values of PSB=TEST, LANG=ASM, PLC=1 are acceptable.
- 2** No special formatting routine is used.
- 3** The first input message is from TERM1.
- 4** Defines the next terminal and restates EOM=#.
- 5** This input message is from TERM2.

Example 2 for ./D command

When TYPE=3270,1 or TYPE=3270,2 is used in the generation of your formats, then DDOF=32701c or DDOF=32702c are used for the ./D command.

Related reading: For a description of TYPE=3270 used in the DEV statement, see *IMS Application Programming APIs*.

<code>./D LTERM=BTS3270 DDOF=327021</code>
--

Example 3 for ./D command

Assume all the terminals for your application are 2740, and you want the complete report that is produced by your SUMMARY transaction to appear in your output. Your input stream might contain:

```

./T TC=SUMMARY MBR=SUMMARY 1
./D DDOF=274011 2
SUMMARY      ITEM1*
              ITEM2*
              ITEM3*
REPORT $

```

where:

- 1** Defines the transaction code and application to process it.
- 2** Specifies the BTS-supplied 2740 Output Formatting Facility, 1 for Model 1 or Model 2, and option 1 to include IMS Batch Terminal Simulator header, trailer, and user I/O area. Defaults of LTERM=IOPCB, EOS=*, and EOM=\$ are acceptable.

Example 4 for ./D command

Assume the application processes input from an IBM 3277 Model 2 and the input must be formatted by IMS Message Format Service before it is acceptable to the application program. The required ./D command is:

```
./D TYPE=3270-A2 SIZE=(24,80) LIMIT=0
```

The command explicitly states that the device for which a formatter is to be used is a 3270 with a 24 x 80 screen image size, and that the application is to be terminated if the input to the application and the available screen image formats disagree (as indicated by a limit count value of zero). The input stream might contain:

```

./T TC=SAMPLE MBR=SAMPLE 1
./D TYPE=3270-A2 SIZE=(24,80) LIMIT=0
/FORM SAMFORM $ 2
L01C10 '12345' 'ABC' 'SAMPLE' $ 3

```

where:

- 1** Defines the transaction code and the application to process it.
- 2** Specifies the format command for the screen image defined by the Message Format Service MOD named SAMFORM.
- 3** Specifies the data fields to fill in on the MID named SAMFORM screen image: ('12345' is the modifiable field that starts at or spans line 1 column 10).
'ABC' is the next modifiable field. 'SAMPLE' is the next modifiable field.

./E command

Use the ./E command to define the environment that the transaction is executed in.

Subsections:

- [“Format” on page 199](#)
- [“Keywords and operands” on page 199](#)
- [“Examples” on page 204](#)

Format

```
./E      [APPLFE=      name_of_the_front-end_routine]
        [CONNECT=   connection_name]
        [DT3270=    YES | NO]
        [ECHKP=     YES | NO]
        [ENDROLB=   YES | NO ]
        [ERR=       R | Q | A]
        [ETO=       NO | YES]
        [FSEND=     YES | NO]
        [IFIUTF=    YES | NO]
        [IMSSUF=    Suffix_character_for_DFSCLV0x]
        [IODATE=    date_of_iopcb]
        [IOTIME=    time_of_iopcb]
        [IOUDATE=   UTC_date_of_iopcb]
        [IOUTIME=   UTC_time_of_iopcb]
        [KATAKANA=  YES | NO]
        [LCI=       YES | NO]
        [MAXL=      maximum_number_of_lines_per_page]
        [MLINE=     maximum_number_of_lines_per_page]
        [NOEDIT=    YES | NO]
        [PA2=       PF-key]
        [PERMLOAD=  YES | NO ]
        [RCFGPN=    RACF_group_name_on_iopcb]
        [REATTCH=   YES | NO]
        [SCNDTRX=   CONT | NOCONT]
        [SPACE=     YES | NO]
        [SPLAPI=    YES | NO]
        [SSID=      ssid]
        [SYSRQ=     YES | NO]
        [UGN=       unit_group_name]
        [UIDDREG=   Userid_of_the_address_space_of_the_dependig_region]
        [USERID=    Userid_of_iopcb]
        [USERIND=   U | L | P | O]
```

Keywords and operands

APPLFE=

Specifies a 1- to 8-character name of an application front-end routine. For details, see *IMS System Definition*.

CONNECT=

Specifies the CONNECTION_NAME when running IMS Batch Terminal Simulator in batch IMS (that is, KW=DLI or DBB) with a Db2 application. The CONNECTION_NAME is set in the DDITV02 input data

set. If the CONNECT= operand is not specified, the TSO authorization ID or job name is set in the DDITV02 input data set. The CONNECTION_NAME value can be 1- to 8-characters long.

Related reading: For more information about the CONNECTION_NAME, see the *Db2 for z/OS Application Programming and SQL Guide*.

DT3270=

Specifies if MFS bypass is supported without basic input-editing for the inbound D/T3270 data stream.

YES

Indicates MFS bypass without basic input-editing for the inbound D/T3270 data stream.

NO

Indicates basic input-editing for the data. The default value is DT3270=NO.

ECHKP=

Specifies whether the ending checkpoint for all application programs is taken.

YES

Indicates that the ending checkpoint is taken for all application programs. The default value is ECHKP=YES.

NO

Indicates that the ending checkpoint is not taken for any application programs.

By setting ECHKP=NO, IMS Batch Terminal Simulator runs internally under KW=JBP. Therefore, do not specify ECHKP=YES under KW=JBP.

ENDROLB=

Specifies whether to issue a Rollback (ROLB) call at the completion of the application to rollback the updates made to the database.

YES

Issues a ROLB call at the completion of the application to rollback the updates made to the database.

When the application calls GU I/O PCB, CHKP, or SYNC, all the checkpoint calls issued to IMS are bypassed in order to backout the updates. If the ROLB call fails, the database updates made by the application are not rolled back.

NO

A ROLB call is not issued at the completion of the application. The default value is ENDROLB=NO.

ERR=

Specifies the region error option when IMS Batch Terminal Simulator is run in batch IMS (that is, KW=DLI or DBB) with a Db2 application. The region error option is set in the DDITV02 input data set.

Related reading: For more information about the region error option, see the *Db2 for z/OS Application Programming and SQL Guide*.

ETO=

Specifies whether ETO is available.

YES

Indicates that ETO is available.

NO

Indicates that ETO is not available. The default value is ETO=NO.

FSSSEND=

Specifies whether to request TSO/E to operate in full-screen mode. When full-screen mode is requested, the terminal operates in full-screen mode. This keyword is useful when the TSO CLIST displays the panel after returning from IMS Batch Terminal Simulator.

YES

Indicates that the request is made to TSO/E. The terminal operates in full-screen mode.

NO

Indicates that the request is not made. The terminal operates in line mode. The default value is FSEND=NO.

IFIUTF=

Specifies to convert the IFI information from UTF8 to EBCDIC when Db2 subsystem parameter UIFCIDS=YES is specified.

YES

Converts the IFI information from UTF8 to EBCDIC.

NO

Does not convert the IFI information. The default value is IFIUTF=NO

IMSSUF=

Specifies one alphanumeric suffix character that is appended to DFSCLV0x (the generated composite control block, nucleus, MFS device characteristics table, and security directory module names). It is effective only when the transaction code is BTSAISR0. It is ignored if either ./D DDOF= or ./D TYPE= is specified.

IODATE=

Specifies the date for the date field in IOPCB. The date on BTS IOPCB is replaced by this value. If ./E IOUDATE option is not specified, the default is the processing date. If ./E IOUDATE option is specified, the default is blank.

IOTIME=

Specifies the time for the date field in IOPCB. The time on BTS IOPCB is replaced by this value. If ./E IOUTIME option is not specified, the default is the processing time. If ./E IOUTIME option is specified, the default is blank.

IOUDATE=

Specifies the date for the UTC date field in IOPCB. The date on BTS IOPCB is replaced by this value. If ./E IODATE option is not specified, the default is the processing date. If ./E IODATE option is specified, the default is blank.

IOTIME=

Specifies the time for the UTC time field in IOPCB. The time on BTS IOPCB is replaced by this value. The default is the time set on processing if ./E IOTIME option is not specified. If ./E IOTIME option is specified, the default is blank.

KATAKANA=

Specifies whether to use Katakana¹ characters for I/O.

YES

Indicates that Katakana characters can be used for I/O.

NO

Indicates that Katakana characters cannot be used for I/O. The default is KATAKANA=NO.

LCI=

Indicates if lowercase input is required by the application.

YES

Indicates that lowercase input is required.

NO

Indicates that lowercase input is not required. The default value is LCI=NO.

MAXL=

Specifies the maximum number of lines per page for the BTSOUT data set. The value must be a 2 byte hexadecimal number. The default value is MAXL=003C.

MLINE=

Specifies the maximum number of lines per page for the BTSOUT data set. The value must be a decimal number. The default value is MLINE=60.

¹ Katakana is a character set of symbols; it is one of the two common Japanese phonetic alphabets.

NOEDIT=

Specifies whether IMS Batch Terminal Simulator can run FSS under the NOEDIT mode.

YES

FSS can be invoked under the NOEDIT mode.

NO

FSS cannot be invoked under the NOEDIT mode. The default value is NOEDIT=NO.

PA2=

Defines the PA2 function (NEXTMSG) for the FSS session in PFK when SYSRQ=YES is specified. With SYSRQ=YES, IMS Batch Terminal Simulator ignores PA2.

PERMLOAD=

Specifies whether to keep the application loaded after its run.

YES

Requests to keep the application loaded after its run.

The application is not deleted after its run; it remains to be loaded. If the application is requested again, the loaded application is used instead of loading the application again.

NO

Requests to delete the loaded application after its run. The default value is PERMLOAD=NO.

If the application was already loaded before this run and the PERMLOAD=NO option or no PERMLOAD= keyword is specified for this run, the application remains loaded even after this run.

RCFGPN=

Specifies a 1- to 8-character RACF® group name of the IO PCB. This value is set on the RACF group name field in IO PCB. The default is blank.

REATTCH=

Specifies if the IMS region controller is reattached for each transaction. When a PSB is immediately rescheduled in DBB batch execution, the same PCBs are passed to the application program unchanged from the earlier execution.

YES

Indicates that the reattaching to IMS region controller is required. You must specify REATTCH=YES when multiple PSBs are scheduled with DBRC.

NO

Indicates that the reattaching to IMS region controller is not required. The default is REATTCH=NO.

By setting REATTCH=YES, IMS Batch Terminal Simulator runs internally under KW=JBP. Therefore, do not specify REATTCH=NO under KW=JBP.

SCNDTRX=

Specifies the order in which the secondary transaction is scheduled when one primary transaction is scheduled many times on one JOB.

CONT

Indicates that a secondary transaction is scheduled after a primary transaction. The default value is SCNDTRX=CONT.

NOCONT

Indicates that a secondary transaction is not scheduled immediately after a primary transaction. If PLC=1 is specified with the ./T command, the secondary transaction is scheduled like as if SCNDTRX= CONT.

A secondary transaction is generally scheduled immediately after the primary transaction. When one primary transaction that kicks the secondary transaction is scheduled many times on one job, the order of the scheduling is as follows:

- PRIMARY1
- SECOND1
- PRIMARY1

- SECOND1

When `./E SCNDTRX=NOCONT` is specified, the secondary transaction is not scheduled immediately after the primary transaction. When one primary transaction that kicks the secondary transaction is scheduled many times on one job with `./E SCNDTRX=NOCONT`, the order of the scheduling is as follows:

- PRIMARY1
- PRIMARY1
- SECOND1
- SECOND1

SPACE=

Specifies whether the blank line is to be suppressed when `SCREEN=INOUT` and `ATR=NO` is specified with the `./O` command.

YES

Indicates that the blank line is suppressed to `BTSOUT` where the field attribute characters would otherwise be displayed in the screen image when `SCREEN=INOUT` and `ATR=NO` are specified with the `./O` command.

NO

Indicates that the blank line is written to `BTSOUT` where the field attribute characters would otherwise be displayed in the screen image when `SCREEN=INOUT` and `ATR=NO` are specified with the `./O` command. The default is `NO`.

SPLAPI=

Specifies whether Spool API functions are available.

YES

Spool API Functions are available. If `./E SPLAPI=YES` is specified, and if IMS Batch Terminal Simulator is run in an IMS online environment (that is, `KW=BMP`), IMS Batch Terminal Simulator can write data to the IMS Spool API for IMS Spool API functions. If IMS Batch Terminal Simulator is run in batch IMS (that is, `KW=DLI` or `DBB`), the `SPLAPI=YES` is ignored.

NO

Spool API functions are not available. The default value is `SPLAPI=NO`.

SSID=

Specifies `SSID` for running IMS Batch Terminal Simulator in batch IMS (that is, `KW=DLI` or `DBB`) with Db2 applications. `SSID` is your 3- to 4-character Db2 subsystem ID.

SYSRQ=

Specifies if TSO Disconnect and Reconnect support is activated.

YES

Indicates that TSO Disconnect and Reconnect support is available.

NO

Indicates that TSO Disconnect and Reconnect support is not available. The default value is `SYSRQ=NO`.

UGN=

Specifies the unit group name for the work data set when running IMS Batch Terminal Simulator in batch IMS (that is, `KW=DLI` or `DBB`) with the Db2 DL/I Batch Support. The default value is `UGN=SYSDA`.

UIDDREG=

Specifies the 1- to 8-character userid of the dependent region when an application program issues `INQY ENVIRON` call. The default is blank.

USERID=

Specifies a 1- to 8-character userid of the IO PCB. This value is set on the Userid field in IO PCB. The default is blank.

USERIND=

Specifies the Userid Indicator. The default is blank.

U

Userid

L

LTERM name

P

PSB name

O

Names other than the previous three types

Examples**Example 1 for ./E command**

You can set different values for the maximum number of lines per page for the BTSOUT data set. To set 90 lines per page, specify:

```
./E MAXL=005A      (the value is 2-byte hexadecimal)
```

or

```
./E MLINE=90      (the value is decimal)
```

Example 2 for ./E command

When a PSB is immediately rescheduled in DBB batch execution, the same PCBs are passed to the application program unchanged from the prior execution. To reattach the IMS region controller for each transaction and thereby get a fresh copy of the PSB, specify:

```
./E REATTCH=YES
```

This specification is required in the following cases:

- When an application program that relies on cleared PCB fields on entry is to be iteratively scheduled.
- When a COBOL program is invoked (via LINK, XCTL, LOAD, or scheduled by IMS Batch Terminal Simulator) immediately after the same program has been run under control of COBOL interactive debug. (COBOL interactive debug is not invoked if the same program is subsequently invoked again.)
- When Db2 application programs run consecutively and issue XRST calls. If the ./E REATTCH= option is not specified, IMS Batch Terminal Simulator is terminated with ABEND04E (rc=00D44054) when the second XRST call is issued.
- When one JOB executes multiple transactions that issue EXEC DLI commands.

Example 3 for ./E command

To run IMS Batch Terminal Simulator in batch IMS (that is, KW=DLI or DBB) with Db2 applications under Db2 DL/I Batch Support, the SSID must be specified. It is your 3- to 4-character Db2 subsystem ID. In addition, the unit group name for the fork data set must be specified to IMS Batch Terminal Simulator. If it is necessary to override the default unit group name SYSDA, specify:

```
./E SSID=ssid UGN=unit-group-name
```

Example 4 for ./E command

When TSO Disconnect and Reconnect support is active, IMS Batch Terminal Simulator does nothing when PA2 is entered. In this case, to define the PA2 function (NEXTMSG) for the FSS session in PFK, specify:

```
./E SYSRQ=YES PA2=PF5
```

or

```
./E SYSRQ=YES PA2=(PF5,PF12)
```

Example 5 for ./E command

To set USERID01 in the I/O PCB USERID field, specify:

```
./E USERID=USERID01
```

Example 6 for ./E command

To set L in the I/O PCB USERID Indicator field, specify:

```
./E USERIND=L
```

Example 7 for ./E command

To set RACFGPNM in the I/O PCB RACF GROUP NAME field, specify:

```
./E RCFGPN=RACFGPNM
```

Example 8 for ./E command

To set the date in the I/O PCB, specify:

```
./E IODATE=0099365F  
./E IOUDATE=1999365F
```

Example 9 for ./E command

To set the time in the I/O PCB, specify:

```
./E IOTIME=1545090F  
./E IOUTIME=12450902040136C
```

Example 10 for ./E command

To set UIDREG01 in the I/O area of INQY ENVIRON call, specify:

```
./E UIDDREG=UIDREG01
```

./O command

Use the ./O command to define the output to be collected and presented, and the options in effect when running in TSO foreground address space. IMS Batch Terminal Simulator output is presented in the BTSOUT data set and at the TSO terminal, depending on the options selected in the ./O command.

Subsections:

- [“Usage note” on page 205](#)
- [“Format” on page 205](#)
- [“Keywords and operands” on page 206](#)
- [“Examples” on page 211](#)

Usage note

The values specified on the ./O command remain in effect until they are reset by another ./O command, or until the end of the IMS Batch Terminal Simulator run.

Format

```
./O      [ATR=          YES | NO]  
        [APS=          YES | NO]  
        [EATR=         YES | NO]  
        [SCREEN=       INOUT | OUT | NO]
```

When running in TSO foreground address space, these additional parameters apply:

```
[TSO=          YES | NO]
[TSOAID=       YES | NO]
[TSOMLVL=      1 | 0]
```

The following parameters can be used to specify the call trace options for all transactions. These specifications can be overridden for a particular transaction by the call trace options of the ./T command.

```
[DB=           YES | NO | SGL]
[ELAPTIME=     YES | NO | ALL]
[ICALOLEN=     1000 | the_length_of_output_data_for_a_request_area_
                  or_a_response_area_on_an_ICAL_call]
[IFI=          YES | NO]
[MQI=          YES | ALL | NO]
[MSG=          YES | NO | SGL]
[REPORT=       YES | NO | T | F | G | C | I | S | M]
[SPA=          YES | NO]
[SQL=          YES | ALL | NO]
[TSODB=        YES | ALL | PROMPT | NO]
[TSOIFI=       YES | NO]
[TSOMQI=       YES | ALL | PROMPT | NO]
[TSOMSG=       YES | ALL | PROMPT | NO]
[TSOSQL=       YES | ALL | PROMPT | NO]
```

The following parameters can be used to specify the call trace option for all transactions. However, these specifications cannot be overridden by the ./T command call trace options.

```
[SQLHX=        YES | NO]
[SQLOBLN=      the_length_of_output_data_that_is_of_LOB_data_type]
[TSOSQLHX=     YES | NO]
[TSSLEN=       144 | the_length_of_the_area_for_a_TSS_call]
```

Keywords and operands

APS=

Indicates the print option for the application program statistics.

YES

Indicates that the statistics are to be printed. The summary shows the number of calls completed (regardless of status) by PCB name and function code for each application program scheduled. This summary is printed upon application program termination. The default value is APS=YES.

NO

Suppresses collecting and printing of application program statistics.

ATR=

Indicates the print option of the field attribute characters in the 3270 formatted screen image.

YES

Indicates that the hexadecimal field attribute characters are to be included in the screen image. A legend accompanies each screen image to assist in interpreting field attribute characters. The default value is ATR=YES.

NO

Suppresses printing of the field attribute characters.

DB=

Indicates the print option for the DL/I database call trace.

YES

Indicates that the DL/I database call trace is to be included in the print output. The default value is DB=YES.

NO

Suppresses the printing of the DL/I database call trace information.

SGL

Indicates that the DL/I database call trace is to be included as a single-line message to identify each call.

EATR=

Indicates the print option of the extended attribute characters in the 3270 formatted screen image.

YES

Indicates that the hexadecimal extended attribute characters are to be included wherever the print output contains 3270 screen images. The extended attribute character indicates color, extended highlighting, programmed symbols (PS), and Extended Graphic Character Set (EGCS). A legend accompanies each screen image to assist in interpreting extended attribute characters.

NO

Suppresses printing of the extended attribute characters. The default value is EATR=NO.

ELAPTIME=

Indicates whether to include the elapsed time of each call in the output.

YES

Indicates that the elapsed time of each call is to be included in the output.

When ELAPTIME=YES is specified, the elapsed time of each call in seconds ('ELAPSED SEC=x.xxxxxx', 25 bytes) is added to the trace output of each call. When the BTSOUT LRECL is more than 158, the output, which includes the elapsed time, is printed on a single line without being split by the default LRECL 133.

NO

Indicates that the elapsed time of each call is not to be included in the output. The default value is ELAPTIME=NO.

ALL

Indicates that the elapsed time and the start timestamp of each call are to be included in the output.

When ELAPTIME=ALL is specified, the elapsed time in seconds and the start timestamp of each call ('ELAPSED SEC=x.xxxxxx YYYY/MM/DD HH:MM:SS.SS', 48 bytes) are added to the trace output of each call. When the BTSOUT LRECL is more than 181, the output, which includes the elapsed time and the start timestamp, is printed on a single line without being split by the default LRECL 133.

ICALOLEN=

Specifies the length of the output data for a request area, a response area, or a control data area on an ICAL call.

Specify a 1- to 4-digit decimal number. The default is 1000 bytes. When the request length, the response length, or the control data length is equal to or less than the ICALOLEN value, all of the data is included in the output. When the length is longer than the ICALOLEN value, the output data is truncated to the ICALOLEN value. When ICALOLEN is 0, the data of the request, response, and control areas is not included in the output.

IFI=

Indicates the print option for the IFI call trace.

YES

Indicates that the IFI call trace is to be printed on BTSOUT as a single-line message to identify each call. The default value is IFI=YES.

NO

Suppresses printing of the IFI call trace information.

MQI=

Indicates the print option for MQI call trace.

YES

Indicates the MQI call trace is to be printed on BTSOUT as a single-line message to identify each call.

ALL

Indicates that the MQI call trace is to be printed on BTSOUT including the MQI message, if the MQI messages are applicable. The default value is MQI=ALL.

NO

Suppresses the printing of the MQI call trace information.

MSG=

Indicates the print option for the DL/I message call trace.

YES

Indicates that the DL/I message call trace is to be included in the print output. The default value is MSG=YES.

NO

Suppresses the printing of the DL/I message call trace.

SGL

Indicates that the DL/I message call trace is to be included as a single-line message to identify each call.

REPORT=

Indicates that program analysis reports are to be included in the output. The page is ejected for each report.

You can specify a REPORT keyword on multiple lines to request multiple reports (for example, ./O REPORT=F and ./O REPORT=C). When the TSOMLVL keyword is set to 0, the report output is not shown on the TSO terminal.

YES

Indicates to print all types of program analysis reports in the output.

T

Prints the program analysis report by PCB total in the output.

F

Prints the program analysis report by function code in the output.

G

Prints the program analysis report by function code and segment name in the output.

C

Prints the program analysis report by status code in the output.

I

Prints the program analysis report by ICAL OTMA descriptor in the output.

S

Prints the program analysis report by SQL statement in the output.

M

Prints the program analysis report by MQI function in the output.

NO

Indicates not to print any of the program analysis reports in the output. The default value is REPORT=NO.

SCREEN=

Indicates the print options for the 3270 formatted screen image.

INOUT

Indicates that both the input and output screen images are to be printed. The default is SCREEN=INOUT.

OUT

Indicates that only the output screen images are to be printed.

NO

Indicates that input and output screen images are not to be printed.

SPA=

Indicates the options for the SPA call trace.

YES

Indicates that the whole SPA that is returned to the I/O area is to be included in the DL/I message call trace. The default value is SPA=YES.

NO

Indicates that only the first 14 bytes of an SPA that is returned to the I/O area is to be included in the DL/I message call trace. The first 14 bytes include the transaction code.

SQL=

Indicates the print option for the SQL call trace.

YES

Indicates that the SQL call trace is to be printed on BTSOUT as a single-line message to identify each call.

ALL

Indicates that the SQL call trace is to be printed on BTSOUT including the IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable. The default value is SQL=ALL.

NO

Suppresses printing of the SQL call trace information.

SQLHX=

Indicates the print option for the hex representation of input and output variables in the SQL call trace. When the printing of input and output variables is suppressed by the SQL= operand, the SQLHX= operand is ignored.

YES

Indicates that the hex representation of input and output variables is to be printed on BTSOUT following the printing of input and output variables. When the data is defined as variable length, the length (LL) of the data is included in the BTSOUT hex representation.

NO

Suppresses printing of the hex representation of input and output variables. The default value is SQLHX=NO.

SQLOBLN=

Specifies the length of output data for LOB (large objects) data type that you want to print in the BTSOUT or that you want to display on the TSO terminal. The length of the LOB data type in SQLDA that is returned by Db2 is always 0. If this parameter is not specified, the LEN= operand on the BTSOUT shows 0, and LOB data is not printed in the BTSOUT or displayed on the TSO terminal.

If the length that is defined in the application program is smaller than SQLOBLN=*nn*, IMS Batch Terminal Simulator substitutes *nn* with the actual data length.

TSO=

Indicates whether the FSS option is wanted (YES) or not (NO). TSO=NO is the default unless IMS Batch Terminal Simulator is started in a TSO foreground address space, in which case TSO=YES is assumed. When TSO=NO, all other TSO operands of the ./O command are ignored.

Related reading: For information about the FSS option, see [“Full screen image support \(FSS\)”](#) on page 35.

TSOAID=

Indicates whether program function key simulation is required (YES) or not (NO). Either no PF keys are used, or the TSO terminal is equipped with the needed keys. The default value is TSOAID=NO.

When TSOAID=YES is coded, IMS Batch Terminal Simulator prompts the user for PF key simulation each time data is entered onto an IMS formatted screen image and Enter is pressed. You can respond to the prompting message by entering a value of PFK1 through PFK24 and pressing Enter. The appropriate terminal action is simulated by IMS Batch Terminal Simulator. If only Enter is wanted, press Enter again. When TSO=NO, the TSOAID operand is ignored.

TSODB=

Indicates the TSO terminal display options for the DL/I database call trace.

YES

Indicates that the DL/I database call trace is to be displayed as a single-line message to identify each call. The default value is TSODB=YES.

ALL

Indicates that the DL/I database call trace is to be displayed including the AIB, IOAREA, key feedback area, and SSA, if applicable.

PROMPT

Indicates that the DL/I database call trace is to be displayed as a single-line message to identify each call. A prompt is displayed so that you can request additional information.

NO

Suppresses the display of the DL/I database call trace information.

TSOIFI=

Indicates the TSO terminal display option for the IFI call trace.

YES

Indicates that the IFI call trace is to be displayed as a single-line message to identify each call. The default value is TSOIFI=YES.

NO

Suppresses the display of the IFI call trace information.

TSOMLVL=

Indicates the level of IMS Batch Terminal Simulator information to be presented at the TSO terminal.

1

Indicates that IMS screen images and all IMS Batch Terminal Simulator error messages and information messages are to be presented at the TSO terminal. Call trace information and statistics are displayed if requested. The default value is TSOMLVL=1.

0

Indicates that only IMS screen images and IMS Batch Terminal Simulator error messages are to be presented at the TSO terminal. Other IMS Batch Terminal Simulator output (information messages, call trace, and statistics) are not presented.

TSOMQI=

Indicates the TSO terminal display options for MQI call trace.

YES

Indicates that the MQI call trace is to be displayed as a single-line message to identify each call. The default value is TSOMQI=YES.

ALL

Indicates that the MQI call trace is to be displayed including the MQI message, if the MQI messages are applicable.

PROMPT

Indicates that the MQI call trace is to be displayed as a single-line message to identify each call. A prompting message is displayed to which you can respond requesting additional information.

NO

Suppresses the display of the MQI call trace information.

TSOMSG=

Indicates the TSO terminal display options for the DL/I message call trace.

YES

Indicates that the DL/I message call trace is to be displayed as a single-line message to identify each call. The default value is TSOMSG=YES.

ALL

Indicates that the DL/I message call trace is to be displayed including the AIB, IOAREA, or SPA, if applicable.

PROMPT

Indicates that the DL/I message call trace is to be displayed as a single-line message to identify each call. A prompt is displayed so that you can request additional information.

NO

Suppresses the display of the DL/I message call trace.

TSOSQL=

Indicates the TSO terminal display options for the SQL call trace.

YES

Indicates that the SQL call trace is to be displayed as a single-line message to identify each call. The default value is TSOSQL=YES.

ALL

Indicates that the SQL call trace is to be displayed including the IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.

PROMPT

Indicates that the SQL call trace is to be displayed as a single-line message to identify each call. A prompt is displayed so that you can request additional information.

NO

Suppresses the display of SQL call trace information.

TSOSQLHX=

Indicates the TSO terminal display option for the hex representation of input and output variables in the SQL call trace. When the display of input and output variables is suppressed by the TSOSQL= operand, the TSOSQLHX= operand is ignored.

YES

Indicates that the hex representation of input and output variables is to be displayed following the display of input and output variables. When the data is defined as variable length, the length (LL) of the data is included in the BTSOUT hex representation.

NO

Suppresses the display of the hex representation of input and output variables. The default value is TSOSQLHX=NO.

TSSLEN=

Specifies the length of the data specified by the second call parameter as the KFB= field in the BTSOUT when the TSS call is issued. Specify a 1- to 4-digit decimal number. The default is 114 bytes. The result from IMS Batch Terminal Simulator can be unexpected when TSSLEN=nn is larger than the actual length of the second call parameter.

Examples

Example 1 for ./O command

Assume that the application is run for the first time in a non-TSO environment and that a full IMS Batch Terminal Simulator trace output is wanted. No ./O command is required. The effective default command is:

```
./O APS=YES SCREEN=INOUT ATR=YES MSG=YES DB=YES SQL=ALL
```

Example 2 for ./O command

Assume a TSO environment for running an application in which you want to suppress DL/I call trace, Db2 SQL call trace, and IMS Batch Terminal Simulator information messages to the TSO terminal. The command is:

```
./O TSOMLVL=0
```

Example 3 for ./O command

For better performance for TSO execution, enter:

```
./O TSOMLVL=0 APS=NO SCREEN=NO MSG=NO DB=NO SQL=NO
```

Example 4 for ./O command

When a DL/I message call trace of IMS Batch Terminal Simulator is requested for application programs running in conversation mode, the scratchpad area is included in the call trace. To truncate the SPA call trace, enter:

```
./O SPA=NO
```

Example 5 for ./O command

To print the elapsed time of each call in a single line for each call, and to print program analysis reports, enter:

```
./O ELAPTIME=Y DB=S MSG=S REPORT=Y
```

Example 6 for ./O command

To print a program analysis report by function code and a program analysis report by status code, enter:

```
./O REPORT=F  
./O REPORT=C
```

./P command

Use the ./P command to define a patch to be applied to an application program. The patch is applied only one time, either at the next LOAD of the application program, or immediately if the application program is currently being run.

Subsections:

- [“Usage notes” on page 212](#)
- [“Format” on page 213](#)
- [“Keywords and operands” on page 213](#)
- [“Examples” on page 213](#)

Usage notes

You can specify multiple ./P commands.

A patch can also be applied to the BTS Common Area by specifying MBR=BTSCOM00.

Some specifications can be made by both the ./P and the ./E command. In such cases, it is probably easier so is better to use the ./E command than ./P command.

Format

```
./P      MBR=          application_program_load_module_name
        PA=          patch_address
        PC=          character_patch_text
        [CA=        000000 | CSECT_address]
        [PX=        00 | hexadecimal_patch_text]
```

Keywords and operands

CA=

Specifies the offset of the CSECT within the load module where the patch is to be applied and can be obtained from a link-editing map. The CSECT address must be a positive six-digit hexadecimal number and leading zeros must be provided. The default value is CA=000000.

MBR=

Specifies the name of the application program load module to which the patch is to be applied. The load module name must be a name defined by the MBR= operand of a ./T command.

PA=

Specifies the offset within the CSECT where the patch is to be applied. The patch address must be a positive six-digit hexadecimal number and leading zeros must be provided.

PC=

Provides the patch text in character format. The text is delimited by a single blank.

PX=

Provides the patch text in hexadecimal format, two characters per byte. The text is delimited by a single blank. Default patch text is PX=00.

Examples

Example 1 for ./P command

To copy (zap) the constant DC C'12345', defined at offset X'0DAC' in your single CSECT application program, to DC C'54321' for only the first transaction you process (the remaining transactions use the original value), your input stream might contain:

```
./T      TC=ZAPTEST MBR=MYPGM PLC=1
./P      MBR=MYPGM PA=000DAC PC=54321
ZAPTEST FIRST TRANSACTION      $
ZAPTEST SUBSEQUENT TRANSACTION $
```

IMS Batch Terminal Simulator puts the patch on a *patch queue* when the ./P command is processed. Then, as each application program is loaded, any patches on the queue are applied and dequeued. Because the patch is only applied once and PLC=1, a fresh copy of MYPGM is fetched for the second ZAPTEST transaction.

Example 2 for ./P command

To set a specified branch to no operation at offset X'010E' in the third CSECT of your application load module, when your link-editing map shows that CSECT begins at X'2C80', your patch statement might be:

```
./P      MBR=TESTPGM PA=00010E CA=002C80 PX=4700
```

or

```
./P      MBR=TESTPGM PA=002D8E PX=4700
```

Example 3 for ./P command

To change the constant DC C'X Y Z', which is defined at offset X'011C' in your load module to DC C'A B C', your patch statement might be:

```
./P MBR=PATCHES PA=00011C PX=C140C240C3
```

Example 4 for ./P command

To support applications that require lowercase input, modify the translation table in the BTS Common Area by specifying:

```
./P MBR=BTSCOM00 PA=000414 PX=40818283848586878889  
./P MBR=BTSCOM00 PA=000424 PX=40919293949596979899  
./P MBR=BTSCOM00 PA=000434 PX=40A1A2A3A4A5A6A7A8A9
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the LCI= parameter, see [“./E command” on page 198](#).

Example 5 for ./P command

To set USERID01 in the I/O PCB USERID field, specify:

```
./P MBR=BTSCOM00 PA=0000D8 PC=USERID01
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the USERID= parameter, see [“./E command” on page 198](#).

Example 6 for ./P command

To set the date in the I/O PCB, specify:

```
./P MBR=BTSCOM00 PA=0000C4 PX=0082365F  
./P MBR=BTSCOM00 PA=0000E8 PX=1982365F
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the IODATE= and IOUDATE= parameters, see [“./E command” on page 198](#).

Example 7 for ./P command

To set the time in the I/O PCB, specify:

```
./P MBR=BTSCOM00 PA=0000C8 PX=1245090F  
./P MBR=BTSCOM00 PA=0000EC PX=12450902040136C
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the IOTIME= and IOUTIME= parameters, see [“./E command” on page 198](#).

Example 8 for ./P command

The maximum number of lines per page for the BTSOUT data set defaults to 60. To set a different value, such as 80 lines per page, specify:

```
./P MBR=BTSCOM00 PA=000236 PX=0050
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the MAXL= and MLINE= parameters, see [“./E command” on page 198](#).

Example 9 for ./P command

When SCREEN=INOUT and ATR=NO is specified in the ./O command, a blank line is written to BTSOUT where the field attribute characters would otherwise be displayed in a screen image. To suppress the blank line, specify:

```
./P MBR=BTSCOM00 PA=000228 PX=01
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the SPACE= parameter, see [“./E command” on page 198](#).

Example 10 for ./P command

To activate BTS 3270 extended attribute support (if supported by your TSO terminal and your application), specify:

```
./P MBR=BTSCOM00 PA=00018B PX=10
```

PS/55 users can specify the following to activate BTS 3270 extended attribute support and to indicate that a PS/55 computer is being used:

```
./P MBR=BTSCOM00 PA=00018B PX=11
```

Example 11 for ./P command

When a PSB is immediately rescheduled in DBB batch execution, the same PCBs are passed to the application program unchanged from the prior execution. To reattach the IMS region controller for each transaction, and thus get a fresh copy of the PSB, specify:

```
./P MBR=BTSCOM00 PA=0004DE PX=01
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the REATTCH= parameter, see [“./E command” on page 198](#).

This specification is required in the following cases:

- When an application program that relies on cleared PCB fields on entry is to be iteratively scheduled.
- When a COBOL program is invoked (via LINK, XCTL, LOAD, or scheduled by BTS) immediately after the same program has been run under control of the COBOL interactive debug. (COBOL interactive debug is not invoked if the same program is subsequently invoked again.)
- When Db2 application programs run consecutively and issue XRST calls. If ./E REATTCH= option is not specified, IMS Batch Terminal Simulator is terminated with ABEND04E (rc=00D44054) when the second XRST call is issued.
- When one JOB executes multiple transactions that issue EXEC DLI commands.

Example 12 for ./P command

To run IMS Batch Terminal Simulator in batch IMS (that is, KW=DLI or DBB) with Db2 applications, the SSID must be specified as follows:

```
./P MBR=BTSCOM00 PA=000C14 PC=ssid
```

where ssid is your 3- to 4-character Db2 subsystem ID.

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the SSID= parameter, see [“./E command” on page 198](#).

Example 13 for ./P command

To set RACFGPNM in the I/O PCB RACF GROUP NAME field, specify:

```
./P MBR=BTSCOM00 PA=0000E0 PC=RACFGPNM
```

This specification can be made by use of the ./E command, which is better to use than the ./P command. For details about the RACFGPN= parameter, see [“./E command” on page 198](#).

Example 14 for ./P command

To support applications that require the input and output of Katakana characters, modify the translation table in the BTS Common Area by specifying the following commands:

```

./P MBR=BTSCOM00 PA=0003D4 PX=404142434445464748494A4B4C4D4E4F
./P MBR=BTSCOM00 PA=0003E4 PX=505152535455565758595A5B5C5D5E5F
./P MBR=BTSCOM00 PA=0003F4 PX=606162636465666768696A6B6C6D6E6F
./P MBR=BTSCOM00 PA=000404 PX=707172737475767778797A7B7C7D7E7F
./P MBR=BTSCOM00 PA=000414 PX=808182838485868788898A8B8C8D8E8F
./P MBR=BTSCOM00 PA=000424 PX=909192939495969798999A9B9C9D9E9F
./P MBR=BTSCOM00 PA=000434 PX=A0A1A2A3A4A5A6A7A8A9AAABACADAFAF
./P MBR=BTSCOM00 PA=000444 PX=B0B1B2B3B4B5B6B7B8B9BABBBBCBDBEBF

```

This specification can be made by use of the `./E` command, which is better to use than the `./P` command. For details about the `KATAKANA=` parameter, see [“./E command” on page 198](#).

Example 15 for `./P` command

To activate TSO Disconnect and Reconnect support, specify:

```
./P MBR=BTSCOM00 PA=000D01 PC=SYSRQ
```

With this specification, IMS Batch Terminal Simulator does nothing when PA2 is entered. To define the PA2 function (NEXTMSG) for the FSS session in PFK, specify:

```

./P MBR=BTSCOM00 PA=000D13 PX=D7C1F24040 (PA2 → PF1 )
./P MBR=BTSCOM00 PA=000D19 PX=D7C1F24040 (PA2 → PF2 )
./P MBR=BTSCOM00 PA=000D1F PX=D7C1F24040 (PA2 → PF3 )
./P MBR=BTSCOM00 PA=000D25 PX=D7C1F24040 (PA2 → PF4 )
./P MBR=BTSCOM00 PA=000D2B PX=D7C1F24040 (PA2 → PF5 )
./P MBR=BTSCOM00 PA=000D31 PX=D7C1F24040 (PA2 → PF6 )
./P MBR=BTSCOM00 PA=000D37 PX=D7C1F24040 (PA2 → PF7 )
./P MBR=BTSCOM00 PA=000D3D PX=D7C1F24040 (PA2 → PF8 )
./P MBR=BTSCOM00 PA=000D43 PX=D7C1F24040 (PA2 → PF9 )
./P MBR=BTSCOM00 PA=000D49 PX=D7C1F24040 (PA2 → PF10)
./P MBR=BTSCOM00 PA=000D4F PX=D7C1F24040 (PA2 → PF11)
./P MBR=BTSCOM00 PA=000D55 PX=D7C1F24040 (PA2 → PF12)

```

This specification can be made by use of the `./E` command, which is better to use than the `./P` command. For details about the `SYSRQ=` and `PA2=` parameters, see [“./E command” on page 198](#).

Example 16 for `./P` command

To support MFS bypass without basic input-editing for the inbound D/T3270 data stream, specify the following command:

```
./P MBR=BTSCOM00 PA=0004DF PX=01
```

In this case, the message output descriptor (MOD) must be `DFS.EDTN`.

This specification can be made by use of the `./E` command, which is better to use than the `./P` command. For details about the `DT3270=` parameter, see [“./E command” on page 198](#).

Example 17 for `./P` command

When running IMS Batch Terminal Simulator in batch IMS (that is, `KW=DLI` or `DBB`) with the `Db2 DL/I Batch Support`, the unit group name for the work data set must be specified to IMS Batch Terminal Simulator. The default unit group name is `SYSDA`. If you want to override the default, specify:

```
./P MBR=BTSCOM00 PA=000C18 PC=unit-group-name
```

where *unit-group-name* is your 1- to 5-character unit group name for the work data set.

This specification can be made by use of the `./E` command, which is better to use than the `./P` command. For details about the `UGN=` parameter, see [“./E command” on page 198](#).

Example 18 for `./P` command

When an `INQY` call is issued with the `NULL` subfunction, the application program obtains information related to the PCB. The default location and default status are as follows:

Terminal location:

```
'LOCAL '
```

Queue status:

```
'STARTED '
```

Session status:

'ACTIVE '

Transaction location:

'LOCAL '

Transaction status:

'STARTED '

Destination status:

'STARTED '

If you want to override the defaults, specify:

```
./P MBR=BTSCOM00 PA=000BD0 PC=terminal-location
./P MBR=BTSCOM00 PA=000BD8 PC=queue-status
./P MBR=BTSCOM00 PA=000BE0 PC=session-status
./P MBR=BTSCOM00 PA=000BE8 PC=transaction-location
./P MBR=BTSCOM00 PA=000BF0 PC=transaction-status
./P MBR=BTSCOM00 PA=000BF8 PC=destination-status
```

./R command

Use the ./R command to define the input format of subsequent simulator statements and the ./SPA command as characters or in hexadecimal. Normal input is in character format.

Subsections:

- [“Usage note” on page 217](#)
- [“Format” on page 217](#)
- [“Keyword and operands” on page 217](#)
- [“Example” on page 217](#)

Usage note

The format set by the ./R command applies to all subsequent simulator statements and the ./SPA command until it is reversed by another ./R command.

Format

```
./R [IP= CHAR | HEX]
```

Keyword and operands

IP=

Specifies the format of all subsequent simulator statements and commands.

CHAR

Indicates that data is in character format (one character or column per byte of data). The default value is IP=CHAR.

HEX

Indicates that data is in hexadecimal format (two characters or columns per byte of data).

Example

To get the number 12345 in packed format, the desired SPA includes the transaction code followed by the packed number. Your input stream might include:

```
./T TC=TRNCODE MBR=CONPGM SPA=11
./R IP=HEX
./SPA E3D9D5C3D6C4C54012345C5B
./R IP=CHAR
TRNCODE INPUT MESSAGE DATA $
```

1

2

where:

1

is the setup for hex SPA.

2

is the return to character input.

All simulator statement data, including the end-of-message indicator X'5B', must be in hex until the input format is reset to CHAR (character) by the next ./R command.

./RETURN command

Use the ./RETURN command to reformat the current IMS screen image on the TSO terminal. This command is intended to be used when the TSO terminal is formatted with an IMS screen image, and you want to enter a simulator command before entering data into the screen image.

Usage notes

You can use the ./RETURN command only with FSS.

If the ./RETURN command is issued when no screen image is available, an error message is displayed on the terminal.

Format

```
./RETURN
```

Keyword

The ./RETURN command has no keywords.

Related tasks

[Reformatting the screen image](#)

You can reformat the current IMS screen image on the TSO terminal by using the ./RETURN command.

./S command

Use the ./S command to define some special action to be taken at a particular point during the processing of the named program load module. The special action can be either the issuing of a SNAP macro or the substituting of a given PCB status code on a designated DL/I call.

Subsections:

- [“Usage note” on page 218](#)
- [“Format” on page 219](#)
- [“Keywords and operands” on page 219](#)
- [“Examples” on page 220](#)

Usage note

Multiple ./S commands can be entered, but only one special action is taken per call.

To use the `./S` command, you must specify the BTSSNAP data set. The `./S` command writes a snap dump in the BTSSNAP data set.

Format

To use the `./S` command to issue a SNAP macro, specify the command in the following format:

```
./S    BYTES=      n
      FUNC=       function_code
      MBR=        application_program_load_module_name
      PCB=        pcbname
      SA=         storage_address
      [CA=       000000 | CSECT_address]
      [TIMES=    1 | t]
```

To use the `./S` command to substitute a status code, specify the command in the following format:

```
./S    FUNC=       function_code
      MBR=        load_module_name
      PCB=        pcbname
      STCD=       xx
      [TIMES=    1 | t]
```

Keywords and operands

BYTES=

Specifies the length of the area beginning at the entry point of the load module name plus the storage address and the CSECT address that is to be snapshot dumped. Boundary alignment is handled by the OS/390 and z/OS SNAP macro facility.

CA=

Specifies the offset from the entry point of the CSECT within the load module where the snap storage area is located and can be obtained from a link-editing map. For COBOL programs, the entry point might not be the same as the origin. The CSECT address must be a positive six-digit hexadecimal number and leading zeros must be provided. The default value is `CA=000000`.

FUNC=

Specifies the function code of the DL/I call that triggers the special action. The function code is specified as two to four alphanumeric characters, and can be any valid DL/I function code. IMS Batch Terminal Simulator takes special action only when the specified function code is issued with the named *pcbname* (`PCB=`) in the named load module (`MBR=`).

MBR=

Specifies the name of the application program load module for which the special action is applicable. The load module name must be a name defined by the `MBR=` operand of a `./T` command.

PCB=

Specifies which PCB triggers the special action. For a database PCB, *pcbname* is the database name field from the PCB, or it is the logical terminal name field (in other words, it is the same as the `PCB=pcbname` field printed in the IMS Batch Terminal Simulator call trace messages).

SA=

Specifies the storage address offset within the CSECT where the snap storage area begins. The storage address must be a positive six-digit hexadecimal number and leading zeros must be provided.

STCD=

Defines the status code to be substituted for the actual status code on return from the call. *xx* can be any value except blank. If `STCD=00` is coded, it is assumed that blanks are intended, and the status

code is set to X'4040'. The call trace message for this call shows the actual status code returned by DL/I. A BTS0073I message follows the call trace, showing the status code that is substituted and returned to the application program.

For Fast Path applications, to set the status code byte in the field search argument (FSA) after a FLD call, specify STCD=y (where y is a single alphanumeric character) and SA=disp (where disp is the displacement in the call IOAREA at which to store y). This specification causes a status code 'FE' to be returned to the application program in the PCB, and y is stored at the specified displacement in the IOAREA of the caller.

TIMES=

Defines a 1- to 5-digit decimal number specifying the maximum number of times the special action is to be taken. The action (snapshot dump or status substitution) is taken the first t times that the MBR=, PCB=, and FUNC= conditions are met. The default value is TIMES=1.

Examples

Example 1 for ./S command

A snapshot dump of a 62 byte work area at offset '1F20' in the PARTRAN application program is required when a GNP call is issued in the PART database. The input stream might include:

```
./T    TC=PARTRAN MBR=PARTRAN
./S    MBR=PARTRAN SA=001F20 BYTES=62 PCB=PART          X
      FUNC=GNP TIMES=5
```

Example 2 for ./S command

A status code of AI is wanted on the first GU to the UNIT database processed by the ABC transaction. The input stream might contain:

```
./T    TC=ABC MBR=UNITPROC
./S    MBR=UNITPROC PCB=UNIT FUNCTION=GU STCD=AI
```

Example 3 for ./S command

A status code of A5 is wanted on the fifth ISRT to the I/O PCB while the XYZ transaction is being processed. The input stream might contain:

```
./T    TC=XYZ MBR=XYZPGM . .
./D    LTERM=IOPCB . .
./S    MBR=XYZPGM PCB=IOPCB          X
      FUNC=ISRT TIMES=4 STCD=00
./S    MBR=XYZPGM PCB=IOPCB          X
      FUNC=ISRT TIMES=1 STCD=A5
```

The first ./S command causes no special action on the first four ISRT calls (STCD=00 causes substitution of blanks for the already blank status code on those calls).

./SPA command

Use the ./SPA command to define the contents of a scratchpad area (SPA) for conversational processing. IMS Batch Terminal Simulator provides an initialized scratchpad area to the first processing program of the conversation and is initialized with transaction code and binary zeros. If the execution is to begin at a point other than the start of the conversation, and a nonzero SPA is required, then the SPA contents can be provided through the ./SPA command.

Subsections:

- [“Usage note” on page 221](#)
- [“Format” on page 221](#)
- [“Keywords” on page 221](#)
- [“Examples” on page 221](#)

Usage note

The `./SPA` command is used only when the program is scheduled.

The `./SPA` command can continue in the same manner as other simulator commands and simulator statements. The end-of-message indicator defined by the `EOM` keyword on the `./D` command must be used in place of the default value (`$`). The end-of-message indicator must be hexadecimal if an `./R IP=HEX` command precedes the `./SPA` command.

Format

```
./SPA ttttttttdata$
```

Keywords

ttttttt

Is the eight-character transaction code and must begin in column 7 of the `./SPA` command. It is the transaction code of the transaction to be passed to this scratchpad area on its initial GU call. *ttttttt* must be hexadecimal if an `./R` command precedes the `./SPA` command.

data

Is the data to be inserted following the transaction code in the scratchpad area. Data must be hexadecimal if an `./R` command precedes the `./SPA` command.

\$

Indicates end of text.

Examples

Example 1 for `./SPA` command

You need a data field in the scratchpad area, and the desired SPA is:

```
RENAME JANE DOE SMITH
```

Then your input stream might include:

```
./T TC=RENAME MBR=NAMER SPA=24  
./D EOM=#  
./SPA RENAMEbbJANEbbDOEbbSMITHbb#
```

Example 2 for `./SPA` command

You are testing conversational processing and the program module that processes the message before you is not working yet. Assume that the SPA, when you get it, looks like:

```
YOURTC 0017E5999+ bbbb
```

Then your input stream might include:

```
./T TC=YOURTC MBR=YOURPGM SPA=20  
./R IP=HEX          Hex spa definition to follow  
./SPA E8D6E4D9E3C340400017E5999C4040404040405B  
./*   Y O U R T C  
./R IP=CHAR        Readable input again  
YOURTC INPUT MESSAGE DATA $
```

In this case, IMS Batch Terminal Simulator queues the SPA defined by the `./SPA` command awaiting a `YOURTC` input message. When IMS Batch Terminal Simulator reads the `YOURTC` message:

- It schedules `YOURPGM` to process the `YOURTC` transaction.
- It returns the queued SPA in the I/O area when `YOURPGM` issues a GU by using its I/O PCB.
- It returns `YOURTC INPUT MESSAGE DATA` when `YOURPGM` issues a GN by using its I/O PCB.

Example 3 for ./SPA command

When a conversational application on REXX uses SPA, SPA data must be defined by using MAPDEF. If SPA data defined by MAPDEF is of the variable-data type or the packed-decimal-data type, you must initialize the SPA area for each data type by using the ./SPA command.

SPA is defined as follows in application programs:

```
spa_map = "spa.txn C 8 : spa.pack P.2 6:",
          "spa.var V 5: spa.zone Z.2 6 : spa.bnṛ B 1 :"  
  
Address REXXIMS "MAPDEF spa spa_map REPLACE "
```

Then your input stream might include:

```
./T TC=TRXPTEST LANG=ASM MBR=REXXP010 SPA=100  
./R IP=HEX  
./SPA E3D9E7D7E3C5E2E300000000000C0005000000000F0F0F0F0F001  
./R IP=CHAR  
TRXPTEST DATA $
```

./T command

Use the ./T command to define each primary and secondary transaction name. Also use this command to define any alternate logical terminal.

Subsections:

- [“Usage note” on page 222](#)
- [“Format” on page 222](#)
- [“Keywords and operands” on page 224](#)
- [“Examples” on page 230](#)

Usage note

Multiple ./T commands can be entered. If more than one ./T command contains the same TC operand, the latter specification is used.

Format

For primary or secondary transaction names:

```

./T   MBR=      membername
      TC=      transaction_code
      [BCHKP=  YES | NO]
      [DLITRACE= nn...n]
      [EM=     editmodulename]
      [LANG=   ASM | CBL | PLI]
      [MQITRACE= nn...n]
      [MSGTYPE= TRAN | MQTMC2]
      [PLAN=   db2_plan_name]
      [PLC=    1 | count]
      [PSB=    psbname]
      [QNAME=  triggered_queue_name]
      [QMGRNAME= queue_manager_name]
      [REATTCH= YES | NO]
      [SPA=    0 | size]
      [SQLTRACE= nn...n]
      [TYPE=   MSG | IFP | MDB | BMP | DLI | JBP | JMP]

```

For alternate logical terminal names that use formats defined with symbolic types for 3270 devices:

```

./T   SIZE=    (lines,columns)
      TC=      ltermname
      TYPE=    symbolic_device_type
      [FEAT=   7E | code]
      [TERM=   STAT|DYNM]

```

For alternate logical terminal names that use formats defined as 3270-1, 3270-2, 3270-P1, or 3270-P2:

```

./T   MDL=     1 | 2 | P1 | P2
      TC=      ltermname
      [FEAT=   7E | code]
      [TERM=   STAT | DYNM]

```

For alternate logical terminal names that use formats defined for DEV TYPE=SCS1:

```

./T   TC=      ltermname
      TYPE=    SCS1
      [FEAT=   7E | code]
      [SIZE=   (,132) | (,columns)]
      [TERM=   STAT | DYNM]

```

For the LU6.2 descriptor:

```

./T      TC=          ltermname
        TYPE=        LU62
        [CONVTYPE=   M | B]
        [LUNAME=     DFSLU | luname | netwkid.luname]
        [MODE=       DFSMODE | modename]
        [SIDE=       side_information]
        [SYNCLVL=    C | N]
        [TPNAME=     DFSASYNC | tpname]

```

Keywords and operands

BCHKP=

Specifies if a beginning checkpoint for an application program is taken.

YES

Indicates that a beginning checkpoint is taken for all application programs.

NO

Indicates that a beginning checkpoint is not taken for any application programs. The default value is BCHKP=NO.

CONVTYPE=

Specifies whether the conversation type is basic (B) or mapped (M). If not specified, the default value is CONVTYPE=M. If TYPE=LU62 is not specified, the CONVTYPE= operand is ignored.

DLITRACE=

Specifies the DL/I call trace options for each PCB for the specified PSB. The number of options specified can be more than or less than the number of PCBs for the PSB. The first PCB for a TP PSB is the I/O PCB.

The TSO-related options are ignored if ./O TSO=NO is specified or assumed. The TSO-related options are also ignored if ./O TSOMLVL=0 is specified. The specified DLITRACE options override the DL/I call trace specifications of the ./O command. Valid DL/I call trace options are:

*

The call trace options specified by the ./O command for this PCB are used.

0

DL/I call trace for this PCB is suppressed.

1

DL/I call trace for this PCB is printed on BTSOUT.

2

DL/I call trace for this PCB is displayed on the TSO terminal as a single-line message to identify each call.

3

DL/I call trace for this PCB is printed on BTSOUT and is displayed on the TSO terminal as a single-line message (options 1 and 2).

4

DL/I call trace for this PCB is displayed on the TSO terminal including the AIB, IOAREA, key feedback area, and SSA, if applicable.

5

DL/I call trace for this PCB is printed on BTSOUT and is displayed on the TSO terminal including the AIB, IOAREA, key feedback area, and SSA, if applicable (options 1 and 4).

8

DL/I call trace for this PCB is displayed on the TSO terminal as a single-line message to identify each call. A prompt is displayed so that the user can request additional trace information for the call.

Related reading: See [“Displaying the call trace information” on page 96](#) for details of this option.

9

DL/I call trace for this PCB is printed on BTSOUT and is displayed on the TSO terminal as a single-line message to identify each call. A prompt is displayed so that the user can request additional trace information for the call (options 1 and 8).

Related reading: See [“Displaying the call trace information” on page 96](#) for details of this option.

EM=

Specifies the member name of the user-written transaction code (input) edit routine that is called to edit each input message segment. *editmodulename* must be from one to eight alphanumeric characters in length. *editmodulename* must name a module that resides on a library accessible by LOAD EP=*editmodulename*. If the EM= operand is not specified, no user edit routine is invoked.

FEAT=

Specifies the device feature indicator values for the alternate logical terminal defined by this ./T command.

Feature code is a two digit hexadecimal value that must correspond to the device features specified in the message output descriptor (MOD) when the Message Format Service (MFS) control block was generated. IMS Batch Terminal Simulator uses the value feature code, in combination with device type and format name, to locate the correct IMSVS.FORMAT library module.

The default value of 7F corresponds to FEAT=IGNORE as coded in the Message Format Service control blocks.

Related reading:

- See [Chapter 15, “Device feature indicator values,” on page 249](#) for the meaning of the device feature indicator values.
- For information about the indicator values and determining the hexadecimal indicator values that corresponds to the features specified in the control blocks being used, see *IMS Application Programming APIs*.

LANG=

Specifies the programming language of the module named by the MBR= operand.

ASM

Assembler Language (default)

CBL

COBOL

PLI

PL/I

Specify ASM or CBL to run IMS C, C++, REXX, or Java applications.

The LANG= operand does not have an influence on an IMS Batch Terminal Simulator execution. It is merely printed on a BTSOUT or a TSO terminal.

LUNAME=

Specifies the LU name of the partner for LU6.2 conversation with a partner application program. If the LU name is a network-qualified name, it can be up to 17 characters long and consist of the network ID of the original system followed by X' and the LU name (for example, *netwkid.luname*). The LU name and the network ID must be a 1- to 8-character name. The default value is LUNAME=DFSLU. If TYPE=LU62 is not specified, the LUNAME= operand is ignored.

MBR=

Specifies the load module name of the application program that processes the transaction named by the TC= operand. IMS Batch Terminal Simulator uses *membername* to load the application program (with LOAD EP=*membername*).

MDL=

Is the 3270 model number and must be specified to define an alternate logical terminal for which output is to be formatted. Valid values are:

1

3277 Model 1

2

3277 Model 2

P1

3284/3286 Model 1, 3287, or 3289 printer

P2

3284/3286 Model 2, 3287, or 3289 printer.

MODE=

Specifies a 1- to 8-character name of VTAM mode table entry. The default value is MODE=DFSMODE. If TYPE=LU62 is not specified, the MODE= operand is ignored.

MQITRACE=

Specifies the MQI call trace options for each MQI call issued by the application program. The number of options specified can be more than or less than the number of MQI calls issued. The specified MQITRACE options override the call trace options of the ./O command.

The TSO-related options are ignored if ./O TSO=NO is specified or assumed. The TSO-related options are also ignored if ./O TSOMLVL=0 is specified.

Valid MQI call trace options are:

Uses the call trace options of the ./O command for this MQI call.

0

MQI call trace for this MQI call is suppressed.

1

MQI call trace for this MQI call is printed on BTSOUT as a single-line message.

2

MQI call trace for this MQI call is printed on BTSOUT including MQI messages, if the MQI messages are applicable.

3

MQI call trace for this MQI call is displayed on the TSO terminal as a single-line message.

4

The same as specifying options 1 and 3.

5

The same as specifying options 2 and 3.

6

MQI call trace for this MQI call is displayed on the TSO terminal including MQI messages, if the MQI messages are applicable.

7

The same as specifying options 1 and 6.

8

The same as specifying options 2 and 6.

9

MQI call trace for this MQI call is displayed on the TSO terminal as a single-line message. A prompting message is displayed to which you can respond requesting additional trace information for the call.

Related reading: See [“Displaying the call trace information” on page 96](#) for a detailed description of this option.

A

The same as specifying options 1 and 9.

B

The same as specifying options 2 and 9.

MSGTYPE=

Indicates the input message format passed with a GU call.

TRAN

Indicates that the input message format is the transaction message for IMS. The default value is MSGTYPE=TRAN.

MQTMC2

Indicates that the input message format is the MQTMC2 structure of the trigger message passed by CSQQTRMN for IBM MQ.

PLAN=

Specifies a 1- to 8-character Db2 plan name. If there is no plan name, the member name is used as the plan name. This function is supported only in a BTS batch environment.

PLC=

Specifies the process limit count for this transaction. For PLC=*n*, a QC status code is returned to the application program on the 1+*n*th IOPCB GU call. The exceptions are:

- For nonconversational secondary transactions, the PLC is ignored and a QC status code is not returned until the message queue is exhausted.
- For conversational processing, /EXIT terminates the application regardless of the PLC or the state of the message queue.

The PLC value for a primary transaction must be a 1- to 5-digit numeric value greater than zero. If you specify PLC=0 for the primary transaction, the status code QC is returned for the first GU call. You can specify PLC=0 for a secondary transaction to indicate that no processing of the secondary transaction is desired. The default value is PLC=1.

PSB=

Specifies the alphanumeric name of the PSB to be used when processing the transaction named by the TC= operand. The *psbname* must be the same as the name supplied for the PSBGEN for this application. If not specified, IMS Batch Terminal Simulator assumes *psbname* is the same as the *membername* specified for the MBR=*membername* operand.

QMGRNAME=

Specifies the name of the queue manager for the IBM MQ batch adapter.

The maximum length of the name is 48 bytes. This parameter is valid when MSGTYPE=MQTMC2 is specified.

If the QMGRNAME= operand not specified, IMS Batch Terminal Simulator uses the default queue manager name defined by the CSQBDEFV module.

QNAME=

Specifies the name of the triggered queue. The maximum length of the name is 48 bytes. The QNAME= operand is valid when MSGTYPE=MQTMC2 is specified.

If the QNAME= operand is specified, IMS Batch Terminal Simulator issues the MQI calls internally by using the IBM MQ batch adapter, and sets the following values in the trigger message:

- Name of triggered queue
- Name of process object

- Trigger data
- Environment data
- User dataQueue manager name

If the QNAME= operand is not specified, these values are blank.

When the internal MQI call fails, the message BTS0115I is returned on BTSOUT.

REATTCH=

Specifies whether to reattach the IMS region controller for the transaction in DBB batch execution only at the beginning and end of this transaction. This option is effective when you have also specified KW=DBB.

YES

Indicates that the reattach to the IMS region controller is needed for the transaction under KW=DBB at the beginning and end of this transaction. You must specify REATTCH=YES when multiple PSBs are scheduled with DBRC.

NO

Indicates that the reattach to the IMS region controller is not needed for the transaction under KW=DBB at the beginning and end of this transaction. The default value is REATTCH=NO.

SIDE=

Specifies a 1- to 8-character name of identifying side information entry. If TYPE=LU62 is not specified, the SIDE= operand is ignored.

SIZE=(lines,columns)

Specifies screen image size in lines (screen image height) and columns (screen image width). The two numeric values must be separated by a comma. For example, a 24 x 80 character screen image would be specified as SIZE=(24,80).

SPA=

Defines the size, in bytes, of the scratchpad area for the transaction named by the TC= operand. Size is specified as a 2- to 5-digit decimal number. A nonzero operand defines this transaction as conversational. The default value is SPA=0.

SQLTRACE=

Specifies the SQL call trace options for each SQL call issued by the application program. The number of options specified can be more than or less than the number of SQL calls issued. The specified SQLTRACE options override the call trace options of the ./O command.

The TSO-related options are ignored if ./O TSO=NO is specified or assumed. The TSO-related options are also ignored if ./O TSOMLVL=0 is specified.

SQLTRACE for Db2 Data Capture:

Specifies the SQL call trace options for each SQL call issued by the application program and the Db2 Changed Data Capture Exit routine (DB2CDCEX). See [Example 9 for ./T command](#) for details.

Valid SQL call trace options are as follows:

The call trace options of the ./O command for this SQL call are used.

0

SQL call trace for this SQL call is suppressed.

1

SQL call trace for this SQL call is printed on BTSOUT as a single-line message.

2

SQL call trace for this SQL call is printed on BTSOUT including IFI SQL statement information, input/output variables, and Db2 messages, if the Db2 messages are applicable.

3

SQL call trace for this SQL call is displayed on the TSO terminal as a single-line message.

- 4** SQL call trace for this SQL call is printed on BTSOUT as a single-line message and is displayed on the TSO terminal as a single-line message (options 1 and 3).
- 5** SQL call trace for this SQL call is printed on BTSOUT including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable. The trace is also displayed on the TSO terminal as a single-line message (options 2 and 3).
- 6** SQL call trace for this SQL call is displayed on the TSO terminal including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.
- 7** SQL call trace for this SQL call is printed on BTSOUT as a single-line message; it is displayed on the TSO terminal including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable (options 1 and 6).
- 8** SQL call trace for this SQL call is printed on BTSOUT including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable. The trace is also displayed on the TSO terminal including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable (options 2 and 6).
- 9** SQL call trace for this SQL call is displayed on the TSO terminal as a single-line message. A prompt is displayed so that you can request additional trace information for the call.
Related reading: See [“Displaying the call trace information” on page 96](#) for details on this option.
- A** SQL call trace for this SQL call is printed on BTSOUT as a single-line message and is displayed on the TSO terminal as a single-line message. A prompt is displayed so that you can request additional trace information for the call (options 1 and 9).
Related reading: See [“Displaying the call trace information” on page 96](#) for details on this option.
- B** SQL call trace for this SQL call is printed on BTSOUT including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable. The trace is also displayed on the TSO terminal as a single-line message. A prompt is displayed so that you can request additional trace information for the call (options 2 and 9).
Related reading: See [“Displaying the call trace information” on page 96](#) for details on this option.

SYNCLVL=

Specifies whether the APPC/IMS sync level is confirmed (C) or not (N). If not specified, the default value is SYNCLVL=C. If TYPE=LU62 is not specified, the SYNCLVL= operand is ignored.

TC=

Specifies the transaction code name of a primary or secondary transaction. The name must be 1- to 8-alphanumeric characters long. Messages with this transaction code are generated from simulator statements, or they are generated by processing programs and inserted via DL/I calls.

ltermname is the 1- to 8-character alternate logical terminal name of a destination to which messages are sent via a DL/I ISRT specifying an alternate PCB. It is associated with an LU 6.2 device when used to specify the LU6.2 descriptor.

TERM=

Specifies whether the terminal is a dynamic terminal or a static terminal.

STAT

Indicates that it is a static terminal. The default value is TERM=STAT.

DYNM

Indicates that it is a dynamic terminal.

TPNAME=

Specifies a 1- to 64-character name of the partner TP. The default is TPNAME=DFSASYNC. If TYPE=LU62 is not specified, the TPNAME= operand is ignored.

TYPE=

Specifies the type of application program being defined or the alternate logical terminal type.

When used to specify the type of application program being defined, valid values are:

MSG

Stands for the message processing program (MPP). Application program TP calls are handled by IMS Batch Terminal Simulator. The default value is TYPE=MSG.

IFP

Stands for the IMS Fast Path (IFP). Application program TP calls are handled by IMS Batch Terminal Simulator. IFP has an effect only on an INQY call.

MDB

Stands for message-driven batch message processing program (BMP). Application program TP calls are handled by IMS Batch Terminal Simulator.

BMP

Stands for a non-message-driven (NMD) batch message processing program. Application program TP calls are handled by IMS Batch Terminal Simulator.

JBP

Stands for a Java non-message-driven (NMD) message processing program. Application program TP calls are handled by IMS Batch Terminal Simulator.

JMP

Stands for Java message processing program (JMP). Application program TP calls are handled by IMS Batch Terminal Simulator.

DLI

Stands for the batch processing program. All application program calls are passed directly to DL/I regardless of the PCB address specified in the call.

When used to specify alternate logical terminal types, the valid values are:

3270-An

Specifies the symbolic device type, where *n* is a value in the range of 1 - 15. This operand must be entered exactly as coded in the TYPE keyword in the MFS DEV macro for the device being simulated.

SCS1

Indicates that the device is an SCS1 type printer.

When used to specify the LU 6.2 descriptor, the valid value is:

LU62

Indicates that the device is an LU 6.2 device.

Examples**Example 1 for ./T command**

You run a batch application named PAYROLL. Your input stream might contain the following statements:

```
./T TC=DUMMY MBR=PAYROLL TYPE=DLI
DUMMY $
```

IMS Batch Terminal Simulator requires a simulator statement with the transaction code to force scheduling of the application.

Example 2 for ./T command

Your PL/I application named PARTRAN processes the ADDPART and DELPART transactions. The required definitions are as follows:

```
./T TC=ADDPART MBR=PARTRAN LANG=PLI
./T TC=DELPART MBR=PARTRAN LANG=PLI
```

Example 3 for ./T command

Your application uses the following PSB:

```
PCB TYPE=TP,NAME=TRAN2
PCB TYPE=TP,LTERM=OUTPUT1
PCB TYPE=DB,...
SENSEG NAME=...
PSBGEN LANG=COBOL,PSBNAME=APPLPGM1
```

Assume APPLPGM1 processes TRAN1 transactions coming from a 3270 Model 2. Further assume that OUTPUT1 is an alternate output destination for an IBM 3270 Model 1, and that the application programs APPLPGM1 and APPLPGM2 process TRAN1 and TRAN2, respectively. Your input stream might contain the following statements:

```
./D TYPE=3270-A2 SIZE=(24,80) LIMIT=1
./T TC=TRAN1 MBR=APPLPGM1 PSB=APPLPGM1 LANG=CBL PLC=5
./T TC=TRAN2 MBR=APPLPGM2 PSB=APPLPGM2 LANG=CBL
./T TC=OUTPUT1 MDL=1
```

Example 4 for ./T command

A BMP application uses the PSB in [Example 3 for ./T command](#). The name of the program is APGM1. Assume that OUTPUT1 is an alternate output destination for a 3270 printer. The MFS format used with output to this destination has been defined as TYPE=SCS1. Your input stream might contain the following statements:

```
./D DDOF=327020
./T TC=ATRAN1 MBR=APGM1 LANG=CBL PSB=APPLPGM1 TYPE=BMP
./T TC=TRAN2 MBR=APPLPGM2 LANG=CBL
./T TC=OUTPUT1 TYPE=SCS1
ATRAN1 $
```

Example 5 for ./T command

You are running IMS Batch Terminal Simulator with FSS in a TSO environment and you want to print DL/I call trace on BTSOUT for the I/O PCB and you want to suppress the display of the call trace on the TSO terminal. For all other PCBs, you want to display the DL/I call trace as a single line identifying each call PCB and suppress printing of the trace on BTSOUT. None of the PSBs has more than 12 PCBs. Your input stream might contain the following statements:

```
./T TC=TRAN1 MBR=APPLPGM1 DLITRACE=122222222222
./T TC=TRAN2 MBR=APPLPGM2 DLITRACE=122222222222
./T TC=TRAN3 MBR=APPLPGM3 DLITRACE=122222222222
```

Example 6 for ./T command

You have a PSB named APPLPGM that has 12 or more PCBs. You want to use various DL/I call trace options. Your input stream might contain the following statements:

```
./T TC=TRAN MBR=APPLPGM DLITRACE=238*39144005
```

The following DL/I call trace options have been specified:

PCB1 (option 2)

Displays the DL/I call trace on the TSO terminal as a single line to identify the call.

PCB2 and PCB5 (option 3)

Prints the DL/I call trace on BTSOUT and displays it on the TSO terminal as a single line to identify the call.

PCB3 (option 8)

Displays the DL/I call trace on the TSO terminal as a single line to identify the call and to prompt for additional information.

PCB4 (option *)

Uses the call trace options as specified by the ./O command.

PCB6 (option 9)

Prints the DL/I call trace on BTSOUT and displays it on the TSO terminal as a single line to identify the call and to prompt for additional information.

PCB7 (option 1)

Prints the DL/I call trace on BTSOUT.

PCB8 and PCB9 (option 4)

Displays the DL/I call trace on the TSO terminal.

PCB10 and PCB11 (option 0)

Suppresses the DL/I call trace.

PCB12 (option 5)

Prints the DL/I call trace on BTSOUT and displays it on the TSO terminal.

PCB13...PCBnn

Defaults to the DL/I call trace options as specified by the ./O command.

Example 7 for ./T command

You are running IMS Batch Terminal Simulator with FSS in a TSO environment and you want to print the SQL call trace on BTSOUT for the first SQL call and you want to suppress the display of the call trace on the TSO terminal. For all other SQL calls, you want to display the SQL call trace as a single line identifying the SQL statement and to suppress printing of the call trace on BTSOUT. None of the application programs issues more than 12 SQL calls. Your input stream might contain the following statements:

```
./T TC=TRAN1 MBR=APPLPGM1 SQLTRACE=122222222222
./T TC=TRAN2 MBR=APPLPGM2 SQLTRACE=122222222222
./T TC=TRAN3 MBR=APPLPGM3 SQLTRACE=122222222222
```

Example 8 for ./T command

You have an application program named APPLPGM, which issues 13 or more SQL statements. You want to use various SQL call trace options. Your input stream might contain the following statements:

```
./T TC=TRAN MBR=APPLPGM SQLTRACE=238*691470A5B
```

The following SQL call trace options have been specified:

SQL call 1 (option 2)

Prints the SQL call trace on BTSOUT including IFI SQL statement information, input/output variables, and Db2 messages, if the Db2 messages are applicable.

SQL call 2 (option 3)

Displays the SQL call trace on the TSO terminal as a single-line message.

SQL call 3 (option 8)

Displays the SQL call trace on the TSO terminal and prints the call trace on BTSOUT, both including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.

SQL call 4 (option *)

Uses the call trace options specified by the ./O command.

SQL call 5 (option 6)

Displays the SQL call trace on the TSO terminal, including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.

SQL call 6 (option 9)

Displays the SQL call trace on the TSO terminal as a single-line message to identify the call and to prompt for additional information.

SQL call 7 (option 1)

Prints the SQL call trace on BTSOUT as a single-line message.

SQL call 8 (option 4)

Displays the SQL call trace on the TSO terminal and prints the call trace on BTSOUT, both as a single-line message.

SQL call 9 (option 7)

Displays the SQL call trace on the TSO terminal, including IFI SQL statement information, input/output variables, and Db2 messages, if the Db2 messages are applicable. Also prints the call trace on BTSOUT as a single-line message.

SQL call 10 (option 0)

Suppresses the SQL call trace.

SQL call 11 (option A)

Displays the SQL call trace on the TSO terminal as a single line to identify the call and to prompt for additional information. Also prints the call trace on BTSOUT as a single-line message.

SQL call 12 (option 5)

Prints the SQL call trace on BTSOUT including IFI SQL statement information, input/output variables, and Db2 messages, if the Db2 messages are applicable. Also displays the call trace on the TSO terminal as a single-line message.

SQL call 13 (option B)

Displays the SQL call trace on the TSO terminal as a single-line message to identify the call and to prompt for additional information. Also prints the call trace on BTSOUT including IFI SQL statement information, input/output variables, and Db2 messages, if the Db2 messages are applicable.

All other SQL calls

Defaults to the call trace options specified by the `./O` command.

Example 9 for `./T` command

You have an application program named APPLPGM1 and a Db2 changed-data-capture exit routine (DB2CDCEX). As shown in the following figure, APPLPGM1 issues five or more SQL statements and DB2CDCEX issues three SQL statements. The third SQL call in APPLPGM1 updates the Db2 table which is defined for data capture.

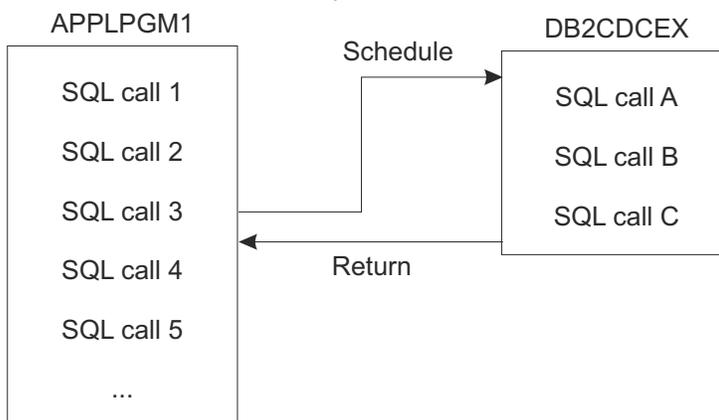


Figure 47. SQL call flow

You want to use various SQL call trace options. Your input stream might contain the following statements:

```
./T TC=TRAN MBR=APPLPGM1 SQLTRACE=238*6914
```

The following SQL call trace options have been specified:

SQL call 1 (option 2)

Prints the SQL call trace on BTSOUT including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.

SQL call 2 (option 3)

Displays the SQL call trace on the TSO terminal as a single-line message.

SQL call 3 (option 8)

Displays the SQL call trace on the TSO terminal and prints the call trace on BTSOUT, both including IFI SQL statement information, input and output variables, and Db2 messages, if the Db2 messages are applicable.

SQL call A (option *)

Uses the call trace options specified by the ./O command.

SQL call B (option 6)

Displays the SQL call trace on the TSO terminal, including IFI SQL statement information, input and output variables, and Db2 messages.

SQL call C (option 9)

Displays the SQL call trace on the TSO terminal, as a single-line message to identify the call to prompt for additional information.

SQL call 4 (option 1)

Prints the SQL call trace on BTSOUT as a single-line message.

SQL call 5 (option 4)

Displays the SQL call trace on the TSO terminal and prints the call trace on BTSOUT, both as a single-line message.

All other SQL calls

Defaults to the call trace options specified by the ./O command.

Example 10 for ./T command

You have an application program named APPLPGM, which issues 13 or more MQI calls. You want to use various MQI call trace options. Your input stream might contain the following statement:

```
./T TC=TRAN MBR=APPLPGM MQITRACE=238*691470A5B
```

The following MQI call trace options have been specified:

MQI call 1 (option 2)

Prints the MQI call trace on BTSOUT including MQI messages, if the MQI messages are applicable.

MQI call 2 (option 3)

Displays the MQI call trace on the TSO terminal as a single-line message.

MQI call 3 (option 8)

Displays the MQI call trace on the TSO terminal and prints the call trace on BTSOUT, both including MQI messages, if the MQI messages are applicable.

MQI call 4 (option *)

Uses the call trace options specified by the ./O command.

MQI call 5 (option 6)

Displays the MQI call trace on the TSO terminal including MQI messages, if the MQI messages are applicable.

MQI call 6 (option 9)

Displays the MQI call trace on the TSO terminal as a single-line message to identify the call and to prompt for additional information.

MQI call 7 (option 1)

Prints the MQI call trace on BTSOUT as a single-line message.

MQI call 8 (option 4)

Displays the MQI call trace on the TSO terminal and prints the call trace on BTSOUT, both as a single-line message.

MQI call 9 (option 7)

Displays the MQI call trace on the TSO terminal including MQI message, if the MQI messages are applicable. Also prints the call trace on BTSOUT as a single-line message.

MQI call 10 (option 0)

Suppresses the MQI call trace.

MQI call 11 (option A)

Displays the MQI call trace on the TSO terminal as a single-line message to identify the call and to prompt for additional information. Also prints the call trace on BTSOUT as a single-line message.

MQI call 12 (option 5)

Prints the MQI call trace on BTSOUT, including MQI messages, if applicable. Also displays the call trace on the TSO terminal as a single-line message.

MQI call 13 (option B)

Displays the MQI call trace on the TSO terminal as a single-line message to identify the call and to prompt for additional information. Also prints the call trace on BTSOUT including MQI messages, if the MQI messages are applicable.

All other MQI calls

Defaults to the call trace options specified by the `./O` command.

Example 11 for `./T` command

When ETO is available and TERM1 is a dynamic terminal, your input stream must contain the following statements:

```
./T TC=TERM1 TERM=DYNM
./E ETO=YES
```

`./*` command

Use the `./*` command to supply comments and documentation of execution on the output listing.

Format

```
./*  comments
```

One blank must follow the asterisk.

Keyword

The `./*` command has no keywords.

Debug commands

IMS Batch Terminal Simulator provides diagnostic capabilities to help you solve problems in the IMS Batch Terminal Simulator system and in application program errors which can be hard to diagnose. This capability consists of debug commands, macros, and a trace table.

Subsections:

- [“Debug commands” on page 235](#)
- [“Examples” on page 236](#)
- [“Debug command trace table” on page 238](#)

Debug commands

The following debug commands can be used for diagnosing problems. All commands start in column 1.

ABEND

This command immediately causes a user 4091 abend.

DEBUG

This command turns on the DEBUG function if it has been turned off by the DEBUGOFF command. The SNAP data is written to the BTSDEBUG data set.

Note: With the DEBUG function activated, you get a U4089 abend dump of the BTS region controller task if the IMS region controller subtask terminates abnormally. With the DEBUG function off, the BTS region controller task terminates normally regardless of the return code from the IMS region controller subtask.

DEBUGOFF

This command turns off the DEBUG function when the DEBUG function is activated by the allocation of a data set with the ddname of BTSDEBUG. The DEBUG function can be turned back on with the DEBUG command.

MSGABENDxxx

This command causes a user 4091 abend after IMS Batch Terminal Simulator issues message xxx. The message number (xxx) must be specified as three digits with leading zeros. There must not be any blanks between MSGABEND and xxx. For example, to cause an abend after message BTS0015W, code MSGABEND015.

TPBUF

This command causes the printing of the TP buffers that are passed to MFS and received from MFS. These buffers contain information that would be received from a terminal or sent to a terminal. The TP buffer information is printed in the BTSOUT data set with a TPLINE= indicator.

TPBUFOFF

This command turns off the TPBUF function.

Examples

The following examples show how to use the debug command by using the sample JCL procedures that are provided in the BTS JCL library (SBTSJCL0).

Example for the ABEND command

The following figure shows the use of the ABEND command in the BTSSAMP1 JCL. The specification causes abend U4091 to occur after transactions PART and DSPALLI are processed.

```
//G.BTSIN DD *
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./T TC=ADDPART LANG=CBL MBR=DFSSAM04
./T TC=ADDINV LANG=CBL MBR=DFSSAM04
./T TC=DLETINV LANG=CBL MBR=DFSSAM04
./T TC=DLETPART LANG=CBL MBR=DFSSAM04
./T TC=CLOSE LANG=CBL MBR=DFSSAM05
./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
PART AN960C10$
DSPALLI AN960C10$
ABEND
DSPINV AN960C10,28009126$
ADDPART AB960C10,RIVET,74$
ADDINV AB960C10,8009126A$
DSPINV AB960C10,8009126A$
DLETINV AB960C10,8009126A$
DLETPART AB960C10$
```

Figure 48. Debug command example: ABEND command

Example for the DEBUGOFF and the DEBUG commands

The following figure shows the use of the DEBUGOFF and the DEBUG commands in the BTSSAMP1 JCL. The specifications cause traces of transactions PART and DLETPART to be generated in data set BTSDEBUG. To generate the trace, you must specify the BTSDEBUG data set.

```

//G.BTSIN DD *
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./T TC=ADDPART LANG=CBL MBR=DFSSAM04
./T TC=ADDINV LANG=CBL MBR=DFSSAM04
./T TC=DLETINV LANG=CBL MBR=DFSSAM04
./T TC=DLETPART LANG=CBL MBR=DFSSAM04
./T TC=CLOSE LANG=CBL MBR=DFSSAM05
./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
PART AN960C10$
DEBUGOFF
DSPALLI AN960C10$
DSPINV AN960C10,28009126$
ADDPART AB960C10,RIVET,74$
ADDINV AB960C10,8009126A$
DSPINV AB960C10,8009126A$
DLETINV AB960C10,8009126A$
DEBUG
DLETPART AB960C10$

```

Figure 49. Debug command example: **DEBUGOFF** and **DEBUG** commands

Example for the **MSGABEND** command

The following figure shows the use of the **MSGABEND** command in the **BTSSAMP1 JCL**. The specifications cause abend **U4091** to occur after message **BTS0020I** is issued while processing transaction **PART**.

```

//G.BTSIN DD *
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./T TC=ADDPART LANG=CBL MBR=DFSSAM04
./T TC=ADDINV LANG=CBL MBR=DFSSAM04
./T TC=DLETINV LANG=CBL MBR=DFSSAM04
./T TC=DLETPART LANG=CBL MBR=DFSSAM04
./T TC=CLOSE LANG=CBL MBR=DFSSAM05
./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
MSGABEND020
PART AN960C10$
DSPALLI AN960C10$
DSPINV AN960C10,28009126$
ADDPART AB960C10,RIVET,74$
ADDINV AB960C10,8009126A$
DSPINV AB960C10,8009126A$
DLETINV AB960C10,8009126A$
DLETPART AB960C10$

```

Figure 50. Debug command example: **MSGABEND** command

Example for the **TPBUF** and **TPBUFOFF** commands

The following figure shows the use of the **TPBUF** and **TPBUFOFF** commands in the **BTSSAMP2 JCL**. The specifications cause the **TP BUFFER** of the first **MOD SCREEN SAMOUX** to be generated in **BTSOUT**.

```

./T TC=PART      MBR=DFSSAM02 LANG=CBL
./T TC=DSPINV   MBR=DFSSAM03 LANG=CBL
./T TC=ADDPART  MBR=DFSSAM04 LANG=CBL
./T TC=ADDINV   MBR=DFSSAM04 LANG=CBL
./T TC=DLETINV  MBR=DFSSAM04 LANG=CBL
./T TC=DLETPART MBR=DFSSAM04 LANG=CBL
./T TC=CLOSE    MBR=DFSSAM05 LANG=CBL
./T TC=DISBURSE MBR=DFSSAM06 LANG=CBL
./T TC=DSPALLI  MBR=DFSSAM07 LANG=CBL
./D LTERM=BTS3270 DDOF=327021
TPBUF
/FORMAT SAMOXX
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 3).
L03C30 'AN960C10' PFK2 $ ENTER 'DSPALLI' TRANS CODE BY PF KEY 2
L04C30 '28009126' PFK3 $ USE PREMODIFIED FIELD TO ENTER PART NO
./D LTERM=BTS3270 DDOF=327016
./O ATR=NO
TPBUFOFF
/FORMAT SAMOXX
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 3).
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 4).
./* POS REF INPUT DATA IMPLIED POS DATA ACTION END-OF MSG
L01C09 'AN960C10' L02C05 'PART ' ENTER $
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
./D LTERM=BTS3270 DDOF=327016
./O ATR=YES
/FORMAT SAMOXX
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 3).
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 4).
L01C09 'AN960C10' L02C05 'PART ' ENTER $
PA1 $          PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
/*

```

Figure 51. Debug command example: TPBUF and TPBUFOFF commands

Debug command trace table

The trace table is written to BTSDEBUG data set when the DEBUG command is specified in the BTSIN data set. Every BTS module puts a trace entry in the trace table as the initial entry. In addition, some IMS Batch Terminal Simulator modules put a trace entry as the Data Format with a message.

This section contains Product-sensitive Programming Interface information.

Each trace entry has one of two formats, depending on whether a message was specified. The CSECT initial entry trace does not use a message.

Trace entry for CSECT initial entry

The following table shows the formats of the trace CSECT initial entry.

Table 37. Formats of the trace CSECT initial entry

OFFSET10	OFFSET16	LENGTH	Explanation
0	0	5	Last 5 characters of called module name
5	5	1	'R' BTSRC000 TCB in control 'P' BTSPC000 TCB in control
6	6	2	Offset of the calling module
8	8	1	Snap identifier
9	9	3	Base register of the called program

Table 37. Formats of the trace CSECT initial entry (continued)

OFFSET10	OFFSET16	LENGTH	Explanation
12	C	5	Last 5 characters of the calling program name (when the calling program is IMS Batch Terminal Simulator)
12	C	(4)	First 4 characters of the calling program name (when the calling program is not IMS Batch Terminal Simulator)
17 (16)	11 (10)	3 (4)	Register 14 of the calling program
20	14	4	Register 1 of the called program
24	18	8	<ul style="list-style-type: none"> • Data pointed to by called program Register 1 or if called module = 'WRT00' • Data = first 8 non-blank message characters with leading "BTS0" suppressed • Only if register 1 is positive; else blanks

Note: The values in parentheses are used when the calling program is not IMS Batch Terminal Simulator. The trace table is provided for supporting user applications that run AMODE=31 and RMODE=ANY.

Trace entry for data format

The following table shows the formats of the trace data format entry.

Table 38. Formats of the trace data format entry

OFFSET10	OFFSET16	LENGTH	Explanation
0	0	5	Last 5 characters of name of module issuing trace
5	5	1	'D'
6	6	2	Displacement of trace call in issuing module
8	8	1	Snap identifier
9	9	3	Base register of module issuing trace
12	C	4	Unused
16	10	4	Register 14 of module issuing trace
20	14	4	Register 1 of module issuing trace
24	18	8	First 8 bytes pointed to by Register 1

Simulator statements

Simulator statements define the transaction messages. There are two types of simulator statements: unformatted-mode simulator statements and formatted-mode simulator statements.

When the full screen image support (FSS) function is used in the TSO foreground in interactive mode, you can enter simulator statements and transaction messages directly through the TSO terminal, as well as coding them in BTSIN before invoking FSS. IMS Batch Terminal Simulator first reads the simulator statements in BTSIN, then waits for direct input. For example, you can specify the simulator statements

that are commonly used in every run in BTSIN, and then specify only the simulator statements that are specific to a run directly through the TSO terminal.

When 3270-formatting is used in a batch job (that is, when the /FORMAT command is specified in the BTSIN data set), IMS Batch Terminal Simulator expects the formatted-mode simulator statements.

When 3270-formatting is not used in a batch mode or when FSS is not used in interactive mode, IMS Batch Terminal Simulator expects the unformatted-mode simulator statements.

Unformatted-mode simulator statements

Unformatted-mode simulator statements are used when the full screen image support (FSS) function or 3270-formatting is not used.

The first simulator statement of a message must begin with the transaction code to which it is related. The transaction code is specified beginning in column one of the first statement. Subsequent simulator statements, which define dependent segments and subsequent messages for conversational processing, must not start with a transaction code, because each simulator statement for teleprocessing applications represents a line or segment of input from a terminal.

For batch applications, only one simulator statement is coded, in addition to the command statements. That simulator statement is in fact a *dummy* statement and serves only to cause the initiation of the application program. As in the case of *real* simulator statements, a *dummy* statement of a batch application must also begin with the transaction code, as specified on the ./T command statement, to which it is related.

Syntax rules

The rules for coding the simulator statements are as follows:

- All simulator statements are coded beginning in column 1 and continuing through column 71.
- A non-blank character in column 72 indicates continuation. The contents of column 72 are not used.
- The simulator statement ends with either an end-of-segment indicator or an end-of-message indicator. This indicator can also be coded as either a single-EBCDIC character or as two hexadecimal digits.

Example

The following figure shows an example of BTSIN input. In this example, lines that start with ./T are the simulator commands, and the line that starts with PART and the subsequent lines are the unformatted-mode simulator statements. Each simulator statement ends with a dollar sign (\$), which is the end-of-message indicator.

```
//G.BTSIN DD *
./T TC=PART LANG=CBL MBR=DFSSAM02 PSB=DFSSAM02
./T TC=DSPINV LANG=CBL MBR=DFSSAM03
./T TC=ADDPART LANG=CBL MBR=DFSSAM04
./T TC=ADDINV LANG=CBL MBR=DFSSAM04
./T TC=DLETINV LANG=CBL MBR=DFSSAM04
./T TC=DLETPART LANG=CBL MBR=DFSSAM04
./T TC=CLOSE LANG=CBL MBR=DFSSAM05
./T TC=DISBURSE LANG=CBL MBR=DFSSAM06
./T TC=DSPALLI LANG=CBL MBR=DFSSAM07
PART AN960C10$
DSPALLI AN960C10$
DSPINV AN960C10,28009126$
ADDPART AB960C10,RIVET,74$
ADDINV AB960C10,8009126A$
DSPINV AB960C10,8009126A$
DLETINV AB960C10,8009126A$
DLETPART AB960C10$
/*
```

Figure 52. Example of a simulator statement

Formatted-mode simulator statements

Formatted-mode simulator statements are used when the 3270-formatting is used in batch mode.

Subsections:

- [“Format” on page 241](#)
- [“Example” on page 242](#)

Format

The format of the formatted-mode simulator statements is as follows. Each field must be separated by one or more blanks.

```
{[LxCy|CyLx] ['data'] [terminal-action]} . . . {terminal-action|eom}
```

Only columns 1 - 71 can be used for a simulator statement. A non-blank character in column 72 indicates continuation.

x and y

x and *y* are 1- to 3-digit numbers that indicate, respectively, the line (L) and the column (C) location at which data is entered on the screen image. This field is optional. The default is the field containing the cursor.

If the user codes the location of the field attribute character on a formatted-mode simulator statement, IMS Batch Terminal Simulator places data in the first data position of the field unless that field is protected.

Values for *x* and *y* must not exceed the screen image size as coded in the SIZE keyword or implied by the 3270 model.

If the location given is anywhere within a field, IMS Batch Terminal Simulator assumes that the first position of the field was the intended target. It is not possible to specify a change to part of a field. Entering data in a field implies filling that field and moving the cursor to the first position in the next unprotected field.

'data'

'*data*' is the data entered onto the screen image using the terminal keyboard, or the operator identification card reader. The single quotes serve as identifiers for this optional field on the simulator statement. Quotes within the data field must be represented by two single quotes (for example, 'six o'clock').

Note: Entered data strings are always left-aligned. If the data string is shorter than the device field, the rest of the field is erased. If the data string is longer than the device field, it is truncated and a message is issued.

terminal-action

terminal-action is the terminal function key, the Operator Identification Card Reader, the selector pen used by the terminal operator, or the special aid-to-testing pseudo-terminal key, PAX. PAX can be used to cause every page in a multi-paged message to be *displayed*.

A terminal action that causes transmission of a message to IMS ends the simulator statement. If no terminating action is encountered before the end-of-message indicator (eom), the ENTER terminal action is assumed. (In the IMS Batch Terminal Simulator output listing, the ENTER terminal action is indicated in the 3270 formatted output heading by ACTION=\$ENTER for that statement.)

Related reading: For the keywords that can be specified in this field of the simulator statement, see [“Terminal action keywords” on page 242](#).

eom

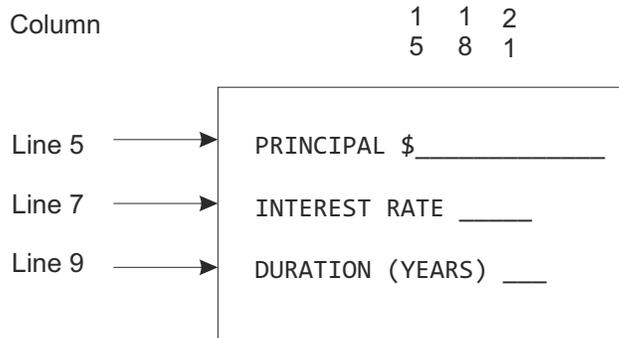
eom is the end-of-message indicator specified explicitly or implicitly in the ./D command or defaulted to \$.

Notes:

1. Comments can be entered after the "eom" indicator.
2. When PEN or ERASIN is specified, "eom" is required.

Example

Each simulator statement for the IBM 3270 Display Station constitutes a complete terminal input sequence. For example, suppose an application uses the following screen image, and the terminal operator is to supply the figures for the principal, interest rate, and duration at the indicated places on the screen:



For this particular screen, only one formatted-mode simulator statement would be necessary, namely the following:

```

L05C15      '500'  L07C18      '4.5'  L09C21      '1'  ENTER  $
^         ^         ^         ^         ^         ^
line    column  keyed-in data  terminal action  end-of-message
indicator

```

As shown in the example, formatted-mode simulator statements can contain data entries for more than one location at a time. The simulator statement must, however, end in a terminal action keyword that causes the transmission of a message to IMS, followed by an end-of-message indicator. In the preceding example, ENTER simulates the terminal operator pressing Enter, and \$ is the end-of-message indicator specified in the ./D command.

Terminal action keywords

For input, IMS Batch Terminal Simulator supports 3270 terminal operator actions. The terminal operator actions can be described with *terminal action keywords*.

The following keywords correspond to 3270 terminal operator actions and are valid specifications for the terminal action field in a formatted-mode simulator statement.

- PAX
- PA1
- PA2
- PA3
- PEN
- PFK1
- PFK2
- PFK3
- PFK4
- PFK5
- PFK6
- PFK7
- PFK24
- PFK8
- PFK9
- PFK10
- PFK11
- PFK12
- PFK13
- PFK14
- PFK15
- PFK16
- PFK17
- PFK18
- PFK19
- PFK20
- PFK21
- PFK22
- PFK23
- PFK24
- CLEAR
- ENTER
- IDCARD
- ERASIN

When CLEAR, PA1–PA3, or PAX are specified, location references, data entries, and other terminal action fields must not be specified on the same simulator statement.

For CLEAR, ENTER, and IDCARD keywords, you can code only the first four characters of the keywords.

Subsections:

- “CLEAR, ENTER, IDCARD keywords” on page 243
- “PAx and PFKx keywords” on page 244
- “ERASIN and PEN keywords” on page 244

CLEAR, ENTER, IDCARD keywords

The following table shows the resulting actions for the CLEAR, ENTER, and IDCARD keywords.

Keyword	Resulting action
CLEA[R]	This terminal action erases the screen image buffer. All output messages are dequeued, and IMS Batch Terminal Simulator now accepts only standard simulator statements.
ENTE[R]	This keyword represents the Enter key. All fields modified before the ENTER action (or having the MOD field attribute) are sent to IMS.
IDCA[RD]	<p>This terminal action is used to simulate the operator identification card reader. 4 - 128 bytes of operator identification card data, plus 1 byte for the field attribute character, are contained in the data field immediately preceding the IDCARD action keyword. The field attribute character for card reader data is filled in by IMS Batch Terminal Simulator in the first byte of the card reader data field. You must provide a blank in front of the data that would actually be read from the card. The makeup of the IMS Batch Terminal Simulator card reader data field is shown the following table.</p> <p>Other data fields can precede the IDCARD data field on the simulator statement, but the card reader data field and action keyword (IDCARD) must be the last items on the statement.</p>

The following table summarizes the operator identification card reader data field information.

Field	Length	Contents	Meaning
1	1	b	Reserved for the field attribute character inserted by IMS Batch Terminal Simulator.
2	1–126	ID number	1 - 126 operator identification character codes.
3	1	Either " or @	End of record (EOR) End of inquiry (EOI)
4	1	LRC Character	A longitudinal redundancy check character that is used by the program for comparison for a parity check.

Because the operator identification card can contain 4 - 128 characters, the cursor location on the simulator statement must point to an unprotected field that can accommodate one field attribute character (blank) plus the operator identification card data, for example, a maximum of 129 characters.

In the following example, the field at line 2 column 2 is unprotected and can accommodate 10 characters. The operator identification card, however, contains only six characters.

```
L02C02 ' 102"7' IDCARD
```

The blank (field attribute character) preceding 102"7 is not sent to the application program.

PAX and PFKx keywords

The following table summarizes the resulting actions for the PAX and PFKx keywords.

Table 41. Resulting actions for PAX and PFKx keywords

Keyword	Resulting action
PAX	The PAX terminal action represents the number of PA1 and PA2 simulator statements needed to display and dequeue all the physical pages of all the messages in the output queue. (PAX is intended primarily as a supplementary testing aid.) After PAX has been specified, the screen image is cleared and unformatted.
PA1	The PA1 keyword displays the next physical page of the message. When PA1 is used against the last page of a message, one of the following event occurs: <ul style="list-style-type: none">• If the message was specified with PAGE=NO, the message is dequeued and the first page of the next message on the queue is displayed.• If the message is specified with PAGE=YES, an error message is issued and the first page of the message is displayed again.
PA2	The PA2 keyword dequeues the current message. The first page of the next message on the queue is then displayed. Note: If PA1 or PA2 has been issued and the output queue is empty, the screen image is cleared and unformatted.
PA3	The PA3 keyword informs the user that the copy-to-printer command has been issued.
PFK1–PFK11 and PFK13–PFK24	These keywords simulate the action obtained from program function keys 1 – 11 and 13 – 24. Modified fields on the screen are sent.
PFK12	This keyword acts the same as keywords PFK1 – PFK11, but in addition issues a message indicating that the copy-to-printer command has been requested.

ERASIN and PEN keywords

The preceding keywords always result in an IMS action. The ERASIN keyword does not result in an IMS action. The PEN keyword causes message transmission only when the field detected is an immediate detectable field.

The following table shows the resulting actions for the ERASIN and PEN keywords.

Table 42. Resulting actions for ERASIN and PEN keywords

Keyword	Resulting action
ERAS[IN]	This terminal action nullifies all unprotected fields on the current screen. It does not send messages.
PEN	This keyword represents the selector light pen and indicates that the light pen is used to point at a specified pen-detectable field. Whether a message is sent depends on the type of the particular field.

The following table shows the field types and their resulting actions.

Table 43. Resulting actions for each field type

Field type	Resulting action
Immediate detectable (First field character=blank, null or &)	All fields modified before the PEN terminal action (or having the MOD attribute) are sent to IMS.

Table 43. Resulting actions for each field type (continued)

Field type	Resulting action
Deferred detectable (First field character=?)	The question mark (?) is changed to a greater than symbol (>), and the field is marked as modified. The cursor position is not affected. Data transmission is deferred until a terminating terminal action is encountered (for example, a PEN action on an immediate detectable field).
Not detectable	An error message is issued and the request is ignored.

The following example assumes that the first two fields pointed to are deferred detectable, and the third field is immediate detectable:

```
L03C15 PEN    L05C15 PEN    L23C03 PEN
```

This simulator statement sends the contents of all three fields.

Programmed symbol and EGCS fields

IMS Batch Terminal Simulator supports the Extended Graphic Character Set (EGCS) attribute specification or the programmed symbol (PS) attribute specification for an MFS field.

IMS Batch Terminal Simulator adds the shift-in (SI) character and the shift-out (SO) character for an EGCS field. The SO character is placed before and the SI character after the EGCS field. When there is enough space for SO and SI characters between EGCS fields, IMS Batch Terminal Simulator tries to use the blank area in the data field. When there is not enough space before or after the EGCS field, the text is unreadable because SO and SI fields are not added.

The formatted simulator statement can be used to input data to a field that has been defined as an EGCS or a PS field. The simulator statement must have an X'0E' character at the start of the data and an X'0F' character at the end of the data. The data must not include any characters with a hex value of less than X'40'.

A hex representation of a valid data specification is:

```
70ABCD07
DE1234FD
```

Where the EGCS or PS data is X'A1B2C3D4'.

In some extreme cases, it might be necessary for a field to span one or more simulator statement cards. A field larger than 70 characters is a case where spanning of cards is necessary. To continue a field onto another card, indicate the end of the current card with an X'0F' character in column 70, 71, or 72, and begin the next card with an X'0E' character. For example:

Column	0	777
	1	012
Line 170..... 0 C DE F 3	
Line 2	0..... 0C E F3	
Line 3	0.....07 E FD	

Mixed mode fields

The PS/55 computer allows a mixed-mode attribute specification for a field which can contain any combination of DBCS (Kanji) or EBCDIC characters.

The formatted simulator statement can be used to input data to a field which has been defined as mixed mode. The simulator statement must have a 'shift out' (SO) character to indicate the start of a DBCS (Kanji) subfield and a 'shift in' (SI) character to indicate the end of a DBCS (Kanji) subfield, X'0E' and X'0F' respectively.

A hex representation of a valid data specification is:

```
7CCC0ABCD0CCC7
D523E1234F493D
```

Where the data is EBCDIC subfield C'EBC' followed by DBCS (Kanji) subfield X'A1B2C3D4' followed by EBCDIC subfield "DIC".

In some extreme cases, it might be necessary for a field to span one or more simulator statement cards. The rules for continuing a DBCS (Kanji) subfield onto another card are the same for mixed mode as for programmed symbols or EGCS.

Operator logical paging

When a message specifies PAGE=YES (MFS specification), operator logical paging can be used.

Format

The simulator statement can be in the following format. Each field must be separated by one or more blanks.

```
[LxCy|CyLx] ['= {+|-}n'] ENTER
```

x and y

x and y indicate the line and column of a field that become the first field of the first input segment.

+

The plus sign (+) indicates the request of a succeeding page.

-

The minus sign (-) indicates the request of a preceding page.

n

n is a 1- to 3-digit decimal number that indicates the relative displacement of the requested page from the currently displayed logical page.

Note: The number of characters following the equal sign (=) cannot exceed four. If a shorter field is wanted, make sure that a blank space follows the last digit.

Related reading: For a complete description of paging, see *IMS Application Programming APIs*.

Examples

For example, the following simulator statement displays a logical page five times removed from the page that is currently displayed:

```
'=+5' ENTER $
```

Note: Line 01 column 02 is assumed.

For the preceding page:

```
'=-1' ENTER $
```

Note: Line 01 column 02 is assumed.

For the field at line 6, column 1, as the first field in the input segment:

```
L06C01 '=+1' ENTER $
```

The PAX command is a sort of *keyboard macro* that lets the user simulate the terminal operator request that every physical page of a multi-paged message displays (not just the first page).

Note: After PAX is specified on a simulator statement, the message is no longer available for subsequent paging requests. The screen image is cleared and unformatted.

IMS commands

IMS Batch Terminal Simulator supports several IMS commands. You can use these IMS commands to control the behavior of the IMS Batch Terminal Simulator job.

IMS /EXIT command

The /EXIT command can be used for terminating IMS Batch Terminal Simulator while a transaction is in progress.

IMS /EXIT cards can be included in your IMS Batch Terminal Simulator input stream. IMS Batch Terminal Simulator ends the particular conversation at that point exactly as IMS does and continues processing any other transactions in the input stream.

If a ./R (reader) command is specified with IP=HEX, the /EXIT command must be coded in hexadecimal, that is, 61C5E7C9E3. It does not require an end-of-segment or end-of-message indicator.

The following figure shows example to specify /EXIT command in BTSIN:

```
./T TC=CONV1 MBR=PROG1 SPA=100
./T TC=CONV2 MBR=PROG2 SPA=50
CONV1 MESSAGE1
MESSAGE2
MESSAGE3
/EXIT
CONV2 MESSAGE1
/EXIT
```

IMS /FORMAT command

Use the /FORMAT command to cause a specific format to be displayed on a physical terminal by using the IMS Message Format Service (MFS). The /FORMAT command starts the full screen image support (FSS).

The /FORMAT command is applicable only for 3270 applications. You can specify /FOR instead of /FORMAT.

The IMS /FORMAT command is used to format the screen image. A /FORMAT command must be entered only while the 3270 is in unformatted mode; otherwise, the results are unpredictable. Subsequent input is assumed to consist of *formatted-mode* simulator statements. In either case, a message output descriptor (MOD) controls the output formatting. The MOD name is specified in the Message Format Service Utility MSG macro statement.

The following example shows how to specify the /FORMAT command in BTSIN:

```

//G.BTSIN DD *
./T TC=PART MBR=DFSSAM02 LANG=CBL
./T TC=DSPINV MBR=DFSSAM03 LANG=CBL
./T TC=ADDPART MBR=DFSSAM04 LANG=CBL
./T TC=ADDINV MBR=DFSSAM04 LANG=CBL
./T TC=DLETINV MBR=DFSSAM04 LANG=CBL
./T TC=DLETPART MBR=DFSSAM04 LANG=CBL
./T TC=CLOSE MBR=DFSSAM05 LANG=CBL
./T TC=DISBURSE MBR=DFSSAM06 LANG=CBL
./T TC=DSPALLI MBR=DFSSAM07 LANG=CBL
./D LTERM=BTS3270 DDOF=327021
/FORMAT SAMOUX
PA1 $ PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 2).
PA1 $ PAGE TO NEXT PAGE (IN THIS CASE PHYSICAL PAGE 3).
L03C30 'AN960C10' PFK2 $ ENTER 'DSPALLI' TRANS CODE BY PF KEY 2
L04C30 '28009126' PFK3 $ USE PREMODIFIED FIELD TO ENTER PART NO

```

IMS /RESET command

The /RESET command eliminates the preset mode that was established by the /SET command.

You can include IMS /RESET cards in your IMS Batch Terminal Simulator input stream. The transaction code is reset for an application.

Related reading: For more information about the /RESET command, see *IMS Operations and Automation*.

The following figure shows example to specify /RESET command in BTSIN:

```

./T TC=TRAN1 MBR=PGOG1
./T TC=TRAN2 MBR=PROG2
/SET TRAN1
/FORMAT MODNAME1
CxxLyy 'MESSAGE1' ENTER
/RESET
/SET TRAN2
/FORMAT MODNAME2

```

IMS /SET command

The /SET command establishes the destination of all messages that are entered into this terminal. The destination can be another terminal or a particular transaction code.

IMS /SET cards can be included in your IMS Batch Terminal Simulator input stream. The transaction code is preset for an application.

Related reading: For more information about the /SET command, see *IMS Operations and Automation*.

The following figure shows example to specify /SET command in BTSIN:

```

./T TC=TRAN1 MBR=PGOG1
./T TC=TRAN2 MBR=PROG2
/SET TRAN1
/FORMAT MODNAME1
CxxLyy 'MESSAGE1' ENTER
/RESET
/SET TRAN2
/FORMAT MODNAME2

```

Chapter 15. Device feature indicator values

This technical reference summarizes the device feature indicator values.

To the application program, the following input devices can enter messages that can look identical regardless of how they were entered:

- Print Line 120 (P.L. 120)
- Print Line 126 (P.L. 126)
- Print Line 132 (P.L. 132)
- Data Entry Keyboard (DEK)
- Program Function Keys (PFK)
- Selector Light Pen Detect (SLPD)
- Operator Identification Card Reader (OICR)
- Dual Platen (DUAL)
- User-defined features for the SCS1 and SCS2 devices and DPM programs

The following table shows the indicator values for the device features.

Table 44. Indicator values for device features

Device features	Indicator values (Hex)
P.L. 120	40
P.L. 126	50
P.L. 132	60
DEK	C8
PFK	C4
SLPD	C2
OICR	C1
IGNORE	7F
DEK,SLPD	4A
DEK,OICR	C9
DEK,SLPD,OICR	4B
PFK,SLPD	C6
PFK,OICR	C5
PFK,SLPD,OICR	C7
SLPD,OICR	C3
DUAL	C1
P.L. 132,DUAL	61
NO FEATURES (3270)	40

Related reading: For more information about device feature indicator values, see *IMS Application Programming APIs*.

Chapter 16. BTS interface with a user-written BTSPIXT0 routine

IMS Batch Terminal Simulator interfaces with a user-written BTSPIXT0 routine.

The coding sample for BTSPIXT0 is supplied as the member BTSPIXT0 and is loaded into the BTS JCL library (SBTSJCL0) during the installation.

This topic contains Product-sensitive Programming Interface information.

Subsections:

- [“BTSPIXT0 call conditions” on page 251](#)
- [“Registers at entry to BTSPIXT0” on page 251](#)
- [“Registers on return from BTSPIXT0” on page 251](#)

BTSPIXT0 call conditions

The user-written routine called BTSPIXT0 must be included in the BTS load module BTSRC000.

If the IMS application program supplies DFS.EDTN in the MOD name parameter for the output message, the basic edit routine is bypassed and BTSPIXT0 is called.

IMS Batch Terminal Simulator maintains a flag in the CTB (bit CTB6TRNI in the CTBFLAG6 field) to indicate 3270 MFS bypass, nonconversational, and no preset destination.

Registers at entry to BTSPIXT0

R1

This register points to the data length field of an unedited message.

R7

This register points to the communication terminal block (CTB).

R9

This register points to the communication line block (CLB).

Registers on return from BTSPIXT0

R1

This register points to the data length field of an edited message. If Register 15 contains a value of 12, R1 contains a message number.

R15

This register is a return code where:

0

Means that the message has been edited successfully. The message is queued by BTS3270Q unless the TPAID byte is PA1, PA2, PA3, or CLEAR. In these cases, the message is dequeued.

12

Means that the message has not been edited successfully. The message is dequeued, and the message identified by Register 1 is displayed.

Related reading: See [“Implementing IMS user-written routines” on page 49](#) for the user message table DFSCMTU0.

16

Means that the message has been edited successfully. The message is queued by BTS3270Q. If the flag CTBFLAG6 does not have the bit CTB6TRNI, the message is dequeued.

Others

Means that the message has not been edited successfully. The message is dequeued.

Chapter 17. Playback utility record format

The Playback utility generates output files that contain records that were extracted from the BTSALOG0 and the BTSAPRE0 modules.

The following technical reference topics explain the format of the records that are created by the Playback utility.

Topics:

- [“Extract file record and CROSSREF file record formats” on page 253](#)
- [“MSGLIST file record format” on page 255](#)

Extract file record and CROSSREF file record formats

The records that are created by BTSALOG0 and BTSAPRE0 conform to certain record format.

This topic contains Product-sensitive Programming Interface information.

Subsections:

- [“BTSALOG0 record format” on page 253](#)
- [“BTSAPRE0 record format” on page 254](#)

BTSALOG0 record format

The following table describes the format of the record created by BTSALOG0.

Table 45. Record format created by BTSALOG0

Displacement (Decimal)	Length (Byte)	Contents
0	2	Length of data area
2	2	Reserved
4	1	Log type (sort field)
5	1	Flag byte
6	8	MFS MOD name
14	8	User ID
22	8	Input destination ID
30	8	Output destination ID
38	8	Input IMS system ID (sort field)
46	16	Input store clock time (sort field)
62	8	Target IMS system ID (sort field)
70	16	Target store clock time (sort field)
86	2	Reserved
88	4	Message DRRN (sort field)
92	4	Record DRRN
96	4	Eye-catcher "TRX#"

Table 45. Record format created by BTSALOG0 (continued)

Displacement (Decimal)	Length (Byte)	Contents
100	2	Length of user data
102	2	Reserved
104	*	User data area

BTSAPRE0 record format

The following table describes the format of the record created by BTSAPRE0.

Table 46. Record format created by BTSAPRE0

Displacement (Decimal)	Length (Byte)	Contents
0	2	Length of record
2	2	Reserved
4	8	Sequence number of record
12	4	Number of screens for this MOD
16	2	Length of data area
18	2	Reserved
20	1	Log type
21	1	Flag byte
22	8	MOD name
30	8	User ID
38	8	Input destination ID
46	8	Output destination ID
54	8	Input IMS system ID
62	16	Input store clock time
78	8	Target IMS system ID
86	16	Target store clock time
102	2	Reserved
104	4	Message DRRN
108	4	Record DRRN
112	4	Eye-catcher "TRX#"
116	2	Length of user data
118	2	Reserved
120	*	User data area

MSGLIST file record format

The MSGLIST file records that are created by BTSALOG0 conform to certain record format.

This technical reference topic explains the format of the MSGLIST records that are created by BTSALOG0.

This topic contains Product-sensitive Programming Interface information.

The following table describes the format of the MSGLIST record created by the BTSALOG0.

Table 47. MSGLIST record

Displacement (Decimal)	Length (Byte)	Contents
0	8	Message type 'INPUT' or 'OUTPUT'
8	1	Reserved
9	1	Message flag ' ' for first part, 'C' for subsequent part
10	1	Reserved
11	8	User ID
19	1	Reserved
20	8	Node name
28	1	Reserved
29	8	Input destination
37	1	Reserved
38	8	Output destination
46	1	Reserved
47	8	IMS system ID
55	16	Store clock time

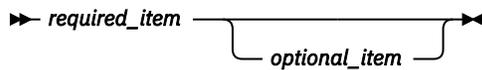
Chapter 18. How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

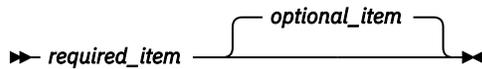
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
 - The >>--- symbol indicates the beginning of a syntax diagram.
 - The ---> symbol indicates that the syntax diagram is continued on the next line.
 - The >--- symbol indicates that a syntax diagram is continued from the previous line.
 - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

▶▶ *required_item* ▶▶

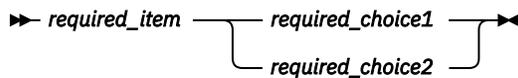
- Optional items appear below the main path.



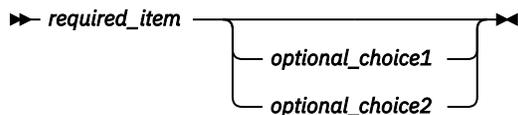
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



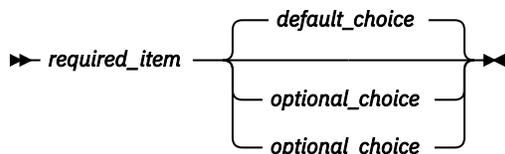
- If you can choose from two or more items, they appear vertically, in a stack. If you must choose one of the items, one item of the stack appears on the main path.



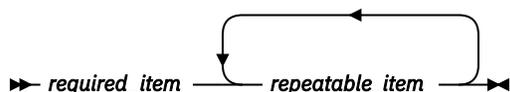
If choosing one of the items is optional, the entire stack appears below the main path.



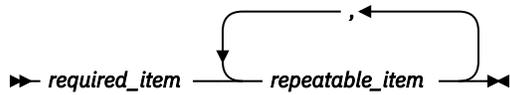
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.

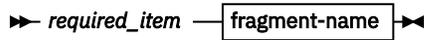


If the repeat arrow contains a comma, you must separate repeated items with a comma.

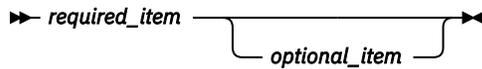


A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



fragment-name



- A b symbol indicates one blank position.
- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses; for example, (1).

Part 6. Troubleshooting

IMS Batch Terminal Simulator provides resources that can be used to help you troubleshoot and diagnose IMS Batch Terminal Simulator problems.

Topics:

- [Chapter 19, “Troubleshooting IMS Batch Terminal Simulator problems,” on page 261](#)
- [Chapter 20, “Messages and codes,” on page 263](#)

Chapter 19. Troubleshooting IMS Batch Terminal Simulator problems

The information in this section can be used to help you troubleshoot and diagnose IMS Batch Terminal Simulator problems.

Topics:

- [“Tips for troubleshooting IMS Batch Terminal Simulator problems” on page 261](#)
- [“Gathering diagnostic information” on page 261](#)

Tips for troubleshooting IMS Batch Terminal Simulator problems

The information in this section can be used to help you troubleshoot common IMS Batch Terminal Simulator problems.

If the IMS Batch Terminal Simulator job ends with return code of 08 and if the following message is issued, that user abend is likely caused by an application environment error. Ensure that your application environment (for example, IBM Language Environment) has no errors, and rerun the job. If the error persists, contact IBM Software Support.

```
BTS0021E TRANSACTION ABNORMALLY TERMINATED.  COMPLETION CODE WAS Uxxxx.
```

Before contacting IBM Software Support, run the IMS application without using IMS Batch Terminal Simulator to determine whether the problem is caused by IMS Batch Terminal Simulator.

Gathering diagnostic information

Before you report a problem with IMS Batch Terminal Simulator to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all IMS Batch Terminal Simulator problems:

- A clear description of the problem and the steps that are required to re-create the problem
- The version of IMS, Db2, and IBM MQ that you are using and the version of the operating system that you are using
- A complete log of the job
- Whether the IMS application runs without problems when IMS Batch Terminal Simulator is not used
- A clear description of the situation when this problem happened:
 - Did this problem happen while testing your application for the first time?
 - Did this problem happen while testing your application after several times? For example, did it happen on the first run after you modified your applications or system resources?

In the latter case, provide information about the resources that you have modified.

IBM accepts Authorized Program Analysis Reports (APARs) describing any situation where the installation of IMS Batch Terminal Simulator causes an exposure to the system integrity of MVS.

Chapter 20. Messages and codes

This reference section provides detailed information about the return codes, abend codes, and messages that might be issued during the execution of IMS Batch Terminal Simulator.

Topics:

- [“IMS Batch Terminal Simulator return codes” on page 263](#)
- [“IMS Batch Terminal Simulator abend codes” on page 263](#)
- [“IMS Batch Terminal Simulator messages” on page 268](#)
- [“IMS Batch Terminal Simulator resource adapter messages” on page 291](#)
- [“Playback utility return codes” on page 302](#)
- [“Playback utility abend codes” on page 302](#)
- [“Playback utility messages” on page 303](#)

IMS Batch Terminal Simulator return codes

At the end of the IMS Batch Terminal Simulator run, the IMS Batch Terminal Simulator job step finishes with the maximum return code that is returned by the application program. When the BTSISRIO DD is specified, the return code is set to 0 or 8.

The following list explains the return codes of IMS Batch Terminal Simulator.

0

The user application programs successfully ended.

8

The user application programs ended abnormally. The abend code is accompanied by BTS0021E messages.

```
BTS0021E TRANSACTION ABNORMALLY TERMINATED. COMPLETION CODE WAS Uxxxx.
```

Others return codes

The user application programs ended with the maximum return codes of the user application programs in the job.

IMS Batch Terminal Simulator abend codes

IMS Batch Terminal Simulator issues an abend code before it ends abnormally (abend).

For each abend code, the following information is provided when applicable:

Explanation:

The Explanation section explains what the abend means; why it occurred; what caused it; what its variable entry fields are.

System action:

The System action section explains what is happening as a result of the condition that caused the abend; whether the system is waiting for responses.

User response:

The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

0113

Explanation

This IMS user abend might be due to an incorrect value for the IMS PARDLI parameter for BMP execution.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See the description of the PARDLI parameters in *IMS System Definition*. Change the value of the PARDLI parameter from 0 to 1.

0200

Explanation

A GET type function was issued using the AIB Interface that was expecting data to be returned in the I/O area. However, the length of the I/O area was too small to receive the data.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Increase the size of the I/O area to allow the data to be returned to the application program.

0260

Explanation

The application program has more than 18 addresses in a parameter list for an IMS call.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Correct and rerun the application program.

0430

Explanation

Module BTSDVBI0 determined that DL/I buffering services cannot be initialized. Message BTS0430I is issued, and the reason code in the message defines the reason for the failure. This problem can occur only when an IMS DBB region is running.

For additional information about abend U430, see *IMS Messages and Codes*.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Contact IMS support personnel for an explanation or correction of the error. To circumvent this problem, modify the JCL or TSO CLIST to run an IMS region type other than DBB.

0476

Explanation

The application program has an invalid PCB address in the call parameter list.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Correct and rerun the application program.

0513

Explanation

A buffer overlay error occurred within IMS.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

First, verify that the device types specified on the ./T cards and ./D cards match the device types specified for IMS (for example, DEV macro of MFS source, and terminal macro for IMSGEN). If there is no conflict between the device types, and the abend U0513 does not occur when IMS is running without IMS Batch Terminal Simulator, contact IMS Batch Terminal Simulator support personnel.

4074

Explanation

A nonzero return code was returned from the extended user exit (BTSEEXIT) program.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check whether there is any other information related to the abend and contact the owner of the extended user exit program if necessary.

4075**Explanation**

A nonzero return code was returned after the character conversion module (CUNLCNV) of z/OS Unicode Services was loaded.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check whether the character conversion services of z/OS Unicode Services are available, make all the necessary corrections, and rerun the job.

4076**Explanation**

An internal API of IMS Tools Base Distributed Access Infrastructure returned an error.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Rerun the application. If the abend persists, contact IMS Batch Terminal Simulator support personnel.

4077**Explanation**

An internal API of IMS Tools Base Distributed Access Infrastructure returned an error.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Rerun the application. If the abend persists, contact IMS Batch Terminal Simulator support personnel.

4078**Explanation**

An internal API of IMS Tools Base Distributed Access Infrastructure returned an error.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Rerun the application. If the abend persists, contact IMS Batch Terminal Simulator support personnel.

4079**Explanation**

The IMS system being run is IMS/ESA® 5.1 or earlier, or the Db2 system using this IMS Batch Terminal Simulator run is Db2 4.1 or earlier. This abend is preceded by the message BTS0088A or BTS0112A.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See message BTS0088A or BTS0112A.

4080**Explanation**

LRECL for BTSOUT is too small to print the screen image properly when EGCS has been specified in an MFS format.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Correct and rerun the application program.

4081**Explanation**

A nonzero return code was received from IMS module DFSBBLD0. In the abend dump, register 10 contains the return code. The problem can occur only when an IMS DBB region is running.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Contact IMS support personnel for an explanation or correction of the error. To circumvent the problem,

modify the JCL or TSO CLIST to run an IMS region type other than DBB.

4082**Explanation**

The IMS call parameter list in the application program contains only one address.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Correct and rerun the application program.

4083**Explanation**

A nonzero return code was returned following a load of the IMS module DFSBSCD0, which contains the current IMS release level.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check whether the IMS load-module library is available, make all the necessary corrections, and rerun the job.

4084**Explanation**

A nonzero return code was returned following a load of the Db2 module DSNHDECP, which contains the current Db2 release level.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check whether the Db2 load-module library is available, make all the necessary corrections, and rerun the job.

4085**Explanation**

Either a nonzero return code was returned by the Db2 message formatting routine, or a nonzero return

code was returned when the Db2 message formatting routine DSNTIAR was being loaded.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Locate message BTS0131E or BTS0132E and follow the User response section of the message. The return code from the Db2 message formatting routine is set to Register 15. For details, see the *Db2 for z/OS Application Programming and SQL Guide*.

4086**Explanation**

A nonzero return code (except 4 and 8) was returned by the Db2 Instrumentation Facility Component Application Program Interface (IFC API). In the abend dump, register 15 contains the return code and register 14 contains the reason code.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

For details, see the *Db2 for z/OS Administration Guide*. Correct and rerun the application program.

4087**Explanation**

A DYNALLOC for the DDITV02/DDOTV02 data set has failed. This abend is preceded by the message BTS0109A, BTS0110A, or BTS0111A.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See message BTS0109A, BTS0110A, or BTS0111A.

4088**Explanation**

As a result of an internal CHKP call, a blank status code was not returned and this CHKP call did not complete.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See message BTS0098W. This message has a status code that was returned by the internal CHKP call. See the status code explanations in *IMS Application Programming APIs*. Analyze the cause of the status code and rerun the job.

4089

Explanation

IMS Batch Terminal Simulator received a nonzero completion code from IMS. This abend was issued to give additional documentation to the problem.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Determine the cause of the original problem. Correct and rerun the application program.

4090

Explanation

The SYNAD exit routine received control from a QSAM PUT macro.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check the BTSOUT DD statement, making certain that it accurately describes the IMS Batch Terminal Simulator output data set. Make the necessary corrections and rerun the job.

4091

Explanation

This abend was requested by the BTS MSGABENDxxx or ABEND command.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Use as appropriate.

4092

Explanation

A TPUT or a TGET to the TSO terminal has failed. If possible, the user receives an error message at the terminal. This message indicates the type of error that has occurred.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Contact TCAM or VTAM support personnel for an explanation or correction of the error.

4093

Explanation

An error internal to the BTS 3270 formatting facility has occurred.

This abend might also occur if IMS Batch Terminal Simulator processing is interrupted in an unsupported way.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Examine the remaining output for further errors. Correct and rerun the application. If the error persists, inform the IMS Batch Terminal Simulator support personnel.

4094

Explanation

IMS Batch Terminal Simulator was unable to open the BTSOUT data set.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check that the BTSOUT data set is accurately specified, make all necessary corrections, and rerun the job.

User response

Check that the data sets are specified accurately on their respective DD statements, and rerun the application.

Explanation

IMS Batch Terminal Simulator was unable to open the FORMAT, QIOPCB, QALTPCB, or BTSEXITO data set.

System action

IMS Batch Terminal Simulator ends abnormally.

IMS Batch Terminal Simulator messages

Use the information in these messages to help you diagnose and solve IMS Batch Terminal Simulator problems.

Message format

IMS Batch Terminal Simulator messages adhere to the following format:

```
BTSnnnnx
```

Where:

BTS

Indicates that the message was issued by IMS Batch Terminal Simulator

nnnn

Indicates the message identification number

x

Indicates the severity of the message:

A

Indicates that operator intervention is required before processing can continue.

E

Indicates that an error occurred, which might or might not require operator intervention.

I

Indicates that the message is informational only.

W

Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

Explanation:

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

System action:

The System action section explains what the system will do in response to the event that triggered this message.

User response:

The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

When simulator command `./O TSOMLVL=0` has been specified, messages with the suffix *I* are not displayed on the TSO terminal.

BTS0000W *nnnn* MESSAGE MISSING.

Explanation

The BTS message routine (BTSMWRT0) was called to issue the diagnostic message (*nnnn*) which is not in the internal message table.

User response

This message might appear when an error is detected in the IMS code and is signaled to IMS Batch Terminal Simulator on the return from the IMS module. IMS Batch Terminal Simulator attempts to print an error message equivalent to the DFS *nnnn* message that would be generated by IMS in a non-BTS environment. If IMS Batch Terminal Simulator has no equivalent message in its internal table (within module BTSMWRT0), the message number *nnnn* might give a clue if the error was encountered in an IMS module. If the error was encountered in a BTS module (3270 formatting not involved), then the message table might be incomplete or an internal error has caused an incorrect message number to be generated; inform the IMS Batch Terminal Simulator support personnel.

BTS0001I COMMAND DISREGARDED.

Explanation

This message is displayed immediately following the messages that indicate why the command was not processed.

User response

Look for other messages immediately preceding this message on the output listing and act accordingly.

BTS0002I **INPUT [RECORD | CHGTABLE]:**
[input record from BTSIN | input record from BTSCHTBL]

Explanation

input record from BTSIN shows each record in the BTSIN data set.

input record from BTSCHTBL shows each record in the environment specification table (BTSCHTBL).

User response

None. This message is informational.

BTS0003I **TRANSACTION NOT QUEUED.**
PLC=0.

Explanation

This message is issued when an application program inserts (ISRT or PURG) a message to an alternate PCB for which the destination is a transaction and that transaction has a process limit count set to zero. An insert of a scratchpad with a changed transaction code also causes this message to be printed under the same condition.

User response

If the transaction is to be queued and subsequently processed, set the PLC keyword of the `./T` command to a nonzero positive number.

BTS0004W **NO TRANSACTION INFORMATION SUPPLIED. UNABLE TO SCHEDULE TRANSACTION: *transaction code***

Explanation

The user has not supplied a `./T` command to provide IMS Batch Terminal Simulator with the information it needs to process the transaction. The transaction code in question is printed at the end of the message.

This message is also issued if the first character in the input message is a blank.

System action

IMS Batch Terminal Simulator continues processing.

User response

Include a `./T` command for the transaction in the input stream.

BTS0005I **END OF BTS RUN.**

Explanation

This message signals the end of the IMS Batch Terminal Simulator run.

User response

None. This message is informational.

BTS0006I **TRANSACTION STARTED:**
transaction code

Explanation

IMS Batch Terminal Simulator has selected a transaction (transaction code is appended) from either the input stream or the message queue. The `./T` information was found.

User response

None. This message is informational.

BTS0007I **BTS V4R1 SIMULATION STARTED.**
TIME=hh:mm:ss, DATE=yyyy.ddd.
IMS VERSION=xx.y.

Explanation

This message indicates the time and date of the run, and the IMS version that is being used. xx.y shows the version of IMS.

User response

None. This message is informational.

BTS0008I **END OF INPUT DATA SET**
ENCOUNTERED.

Explanation

BTSIN has reached the end of file.

User response

None. This message is informational.

BTS0009W **SYNAD ERROR OCCURRED**
DURING READ OF INPUT DATA
SET.

Explanation

An uncorrectable input or output error was detected by the operating system for the input (BTSIN) data set. IMS Batch Terminal Simulator stops processing the current transaction at this point and does not issue a dump.

System action

IMS Batch Terminal Simulator returns an AQ status code to the application program if the message GU was in process. Otherwise, IMS Batch Terminal Simulator simply ends.

User response

Check the JCL or the BTS CLIST to see that the BTSIN data set is correctly specified and rerun the job.

BTS0010W **INVALID BTS COMMAND:**

Explanation

The simulator command specified is invalid. The command is omitted from further consideration and processing continues as far as practical.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator command and rerun the job.

BTS0011I **CONTINUING WITH NEXT**
TRANSACTION.

Explanation

IMS Batch Terminal Simulator continues processing by attempting to schedule the next transaction.

User response

None. This message is informational.

BTS0012W **UNRESOLVED COMMAND MODULE**
ENTRY POINT.

Explanation

IMS Batch Terminal Simulator program error. An unresolved entry point was detected for a BTS command module.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Notify the system support group of the IMS Batch Terminal Simulator program error.

BTS0013W **EXPECTED CONTINUATION CARD**
NOT FOUND BEFORE END OF
INPUT DATA SET.

Explanation

The last card in the input (BTSIN) data set contained a non-blank character in column 72. All input is processed as far as practical.

System action

IMS Batch Terminal Simulator ends normally.

User response

Either add the expected continuation card or delete the non-blank character from column 72 and rerun the job.

**BTS0014I FUNCTION NOT FOUND IN BTS
FUNCTION CODE TABLE.****Explanation**

A function code unknown to IMS Batch Terminal Simulator is used by the application program, and IMS Batch Terminal Simulator is unable to update statistics for this function.

User response

If the function code used was invalid, correct the program and rerun the job. If the function code is valid, notify the IMS Batch Terminal Simulator support personnel to add the function to the appropriate IMS Batch Terminal Simulator tables.

**BTS0015W INVALID KEYWORD (*keyword*);
OPERAND (*operand*).****Explanation**

An invalid keyword was specified on a simulator command.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the keyword or operand.

**BTS0016W REQUIRED KEYWORD MISSING
(*keyword*)****Explanation**

A required keyword is missing from a simulator command.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Code the missing keyword and a valid operand and rerun the job.

BTS0017W DUPLICATE KEYWORD (*keyword*)**Explanation**

A keyword has been specified more than once on a simulator command.

System action

The command is processed provided no other errors are detected using the operand of the first keyword detected. The duplicate keyword operands are ignored.

User response

Remove any duplicate keywords from the command.

**BTS0018W OPEN FAILED FOR INPUT (BTSIN)
DATA SET.****Explanation**

The BTSIN data control block could not be successfully opened by IMS Batch Terminal Simulator.

System action

IMS Batch Terminal Simulator ends normally.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for BTSIN. Correct and rerun the job. See *z/OS DFSMS Macro Instructions for Data Sets* for the probable cause of the OPEN errors.

**BTS0019W INVALID RETURN CODE
ENCOUNTERED: *aaaaaaaa*,
*bbbbbbbb*****Explanation**

An invalid return code was encountered by module *aaaaaaaa*. The return code was generated by *bbbbbbbb*.

System action

IMS Batch Terminal Simulator continues processing.

User response

Examine the remaining output for further errors. Correct and rerun the job. If the error persists, inform the IMS Batch Terminal Simulator support personnel. This message should never appear in a production environment.

**BTS0020I STATISTICS REPORT FOR
TRANSACTION: *transaction code*****Explanation**

This message is written prior to the first line of the application statistics report.

User response

None. This message is informational.

BTS0021E **TRANSACTION ABNORMALLY
TERMINATED. COMPLETION CODE
WAS xxxxx.**

Explanation

An application program has ended abnormally. xxxxx is the return code from the IMS region controller (DFSRR00).

System action

IMS Batch Terminal Simulator end abnormally.

User response

Correct the application and rerun the job.

BTS0022I **THE FOLLOWING MESSAGE WILL
CAUSE BTS TO ABEND:**

Explanation

Debug command MSGABEND was specified. This message is followed by the message that causes IMS Batch Terminal Simulator to end abnormally.

User response

None. This message is informational.

BTS0023I **EXIT COMPLETED.**

Explanation

A /EXIT command was detected as the first message segment for an application program. The transaction ends and IMS Batch Terminal Simulator continues processing.

User response

None. This message is informational.

BTS0024I **BTS OUTPUT LRECL NOT 131 FOR
2740 FORMATTER.**

Explanation

This message is a warning. The 2740 device-dependent output formatter requires a logical record length (LRECL) of 131 on the BTSOUT data set (1 for carriage control and 130 for simulator text). With LRECL other than 131, the formatter output might appear distorted.

User response

Modify the procedure and rerun the job.

BTS0025W **EPA OF THE REQUESTED DDOF
UNRESOLVED.**

Explanation

The user is attempting to utilize a particular device-dependent output formatter. The DDOF keyword operand defined a module whose entry point address is not resolved in the BTS load module.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Link edit the load module, including all appropriate modules, and rerun the job.

BTS0026W **OPEN FAILED FOR SNAP OUTPUT
DATA SET.**

Explanation

A SNAP dump has been requested by the ./S command. IMS Batch Terminal Simulator has attempted, unsuccessfully, to open the BTSSNAP output data set.

System action

IMS Batch Terminal Simulator continues processing.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for BTSSNAP. Correct and rerun the job. See *z/OS DFSMS Macro Instructions for Data Sets* for the probable cause of the OPEN errors.

BTS0027W **OS SNAP MACRO RETURN
CODE=xx, SNAP ID=yyy.**

Explanation

Return code xx was returned to IMS Batch Terminal Simulator from the execution of the OS SNAP macro. The IMS Batch Terminal Simulator user SNAP ID that was being processed at the time was yyy.

System action

IMS Batch Terminal Simulator continues processing.

User response

See the *z/OS MVS Programming: Assembler Services Guide* for a discussion of the SNAP macro and the cause of the error. Correct and rerun the job.

BTS0028W **WRITE LENGTH ERROR: *length* field**

Explanation

IMS Batch Terminal Simulator has printed a DL/I call trace with an invalid length.

System action

The message is discarded and IMS Batch Terminal Simulator continues processing.

User response

Notify the IMS Batch Terminal Simulator support personnel of the error.

BTS0029W **PROGRAMMED SYMBOLS SHOULD BEGIN WITH A X'OE' CHARACTER AND END WITH A X'OF' CHARACTER.**

Explanation

Using MFS, a programmed symbol set might be specified for a particular field. Input data for such a field must be preceded with an X'OE' character to designate *shift out of EBCDIC* and then followed with an X'OF' character to designate *shift into EBCDIC*.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues the processing.

User response

Correct the error and rerun the application.

BTS0030W **UNKNOWN INPUT FIELD TYPE.**

Explanation

A field on the 3270 formatted-mode simulator statement has been encountered that is none of the following fields:

- A data entry enclosed in quotes
- A position reference of the form: LxCy or CyLx
- A valid terminal action keyword
- The end-of-message indicator specified in the ./D command

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator statement and rerun the application.

BTS0031I **MODNAME: *module name***

Explanation

The indicated module name is used to format the subsequent output message. The module name was obtained from the format name parameter in the first ISRT call for the message, or (if this parameter was not specified) from the modname field of the IOPCB.

User response

None. This message is informational.

BTS0032I **OUTPUT QUEUE NOT EMPTY AT A/P RETURN. NUMBER OF MESSAGES: *xxxx***

Explanation

All output messages created by an application program between a GU and either the next GU or application program return have not been accessed.

User response

If the pages are to be displayed, insert PAX or selective paging (LxCy '=+n' ENTER eom) simulator statements and rerun the application.

BTS0033W **ERROR IN /FORMAT COMMAND.**

Explanation

A syntax error was found in a /FORMAT command and the command has been ignored. The command specification must be coded as /FOR or /FORMAT and must be followed by at least one blank space and the 1- to 8-character message output descriptor (MOD) name.

System action

The command is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the /FORMAT statement or message input descriptor (MID) format and rerun the application.

BTS0034I THE SCREEN IS CLEARED AND UNFORMATTED.

Explanation

This message is issued when CLEAR or CLEA has been specified on a simulator statement. Subsequent input messages are not formatted.

User response

None. This message is informational.

BTS0035W OPEN FAILED FOR FORMAT DATA SET.

Explanation

OPEN failed for the Format Control Block Library (IMSVS.FORMAT) data set.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for FORMAT. Correct and rerun the job. See *z/OS DFSMS Macro Instructions for Data Sets* for the probable causes of the OPEN errors.

BTS0036W TERMINATING ERROR IN CARD NO. xx COL. NO. yyy

Explanation

The field beginning in the indicated card and column contains a syntax error as noted in a preceding message.

System action

Data following the invalid field is not scanned and the simulator statement is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the syntax error and rerun the application.

BTS0037W INVALID SO/SI ENCOUNTERED. LINE NO. xx, COLUMN NO. yyy

Explanation

The line and the column point to the attribute character of a field in the internal screen image. IMS Batch Terminal Simulator is processing a simulator

statement that would update this field. This field has a "mixed mode" attribute. The text of the simulator statement contains an invalid combination of shift-out (SO) and shift-in (SI) characters, X'0E' and X'0F', respectively.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator statement or the MFS format and rerun the application.

BTS0038W INVALID DETECT INDICATOR. LINE NO. xx, COLUMN NO. yyy

Explanation

The line and the column point to the attribute character of a field having an invalid designator character. For immediate selector pen detectable fields, the designator character must be blank or null. For deferred selector pen detectable fields, the designator character must be either a question mark (?), a greater-than symbol (>), or an ampersand (&).

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct the specification for the field in the IMS Message Format Service Utility control statement and rerun the application.

BTS0039W TRIED PEN DETECT IN NON-DETECTABLE FIELD LINE NO. xx, COLUMN NO. yyy

Explanation

The field having its attribute character at line xx and column yyy is undetectable, and a selector pen detectable action was attempted. The action is ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct either the simulator statement or the field characteristic definition and rerun the application.

BTS0040W GO/NOGO COUNT EXHAUSTED.**Explanation**

The limit count specified in the ./D command has been reached.

System action

Subsequent input to the application is disregarded until the next ./D command is encountered in the BTS input stream.

User response

Resolve the discrepancies between the input to the application and the available screen formats, and rerun the application.

Note: The limit count is decremented by 5 for each severe error (error message numbers ending in A), and by 1 for each warning (error message numbers ending in W).

BTS0041I SET COMMAND COMPLETED**Explanation**

A /SET command was successfully processed.

User response

None. This message is informational.

BTS0042W UNBALANCED QUOTES.**Explanation**

An odd number of quotes was encountered in the simulator statement data entry field. Quotes within the data field must be represented by double quotes, for example: 'O' 'Neil'.

System action

The input message is discarded and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator statement and rerun the application.

BTS0043W UNABLE TO LOAD ERROR MESSAGE OUTPUT DESCRIPTION.**Explanation**

Block fetch has failed for the requested message output description or device output description.

System action

The output message is discarded and IMS Batch Terminal Simulator continues processing.

User response

Verify that the DOF FEATURE option agrees with the FEAT= operand coded on the ./D or ./T command, that the correct version of the IMS format blocks are being used, that MOD and DOF compilation date and time match, that the IMS error default output descriptions are in the format data set, and that no I/O error has occurred. If necessary, rebuild the format blocks.

BTS0044W OPEN FAILED FOR QALTRAN DATA SET.**Explanation**

OPEN failed for the QALTRAN work data set.

System action

IMS Batch Terminal Simulator continues processing.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for QALTRAN. Correct and rerun the job. See *z/OS DFSMS Macro Instructions for Data Sets* for the probable causes of the OPEN errors.

BTS0045W KANJI INPUT NOT PERMITTED IN THIS FIELD. LINE NO. xx, COL NO. yyy**Explanation**

The line and the column point to an attribute character of a field in the internal screen image. IMS Batch Terminal Simulator is processing a simulator statement that would update this field. This field does not have "mixed mode" or EGCS attribute. The text of the simulator statement contains invalid shift-out (SO) and shift-in (SI) characters, X'0E' and X'0F', respectively.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues the processing.

User response

Correct the simulator statement or the MFS format and rerun the application.

BTS0046W **INPUT DATA TRUNCATED. LINE
NO. xx COLUMN NO. yyy.**

Explanation

Input data has been truncated to fit the field beginning at the indicated line and column number.

System action

IMS Batch Terminal Simulator continues processing.

User response

If truncation is not desired, correct the simulator statement or respecify the length of the field.

BTS0047W **TRIED TO ENTER DATA IN
PROTECTED FIELD LINE NO. xx,
COLUMN NO. yyy**

Explanation

The field having its attribute character at line xx and column yyy is protected. An attempt was made to enter data into this field. The action has been ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

Either correct the field specification, or specify a valid position reference for the data entry.

BTS0048W **INVALID POSITION REFERENCE.**

Explanation

Either an invalid position reference has been specified in a simulator statement, or the screen image has no attribute characters. A valid position reference must have the form LxCy or CyLx and be followed by at least one blank space. x and y are 1- to 3-digit decimal numbers.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues the processing.

User response

If no attribute characters can be found on the screen image and ATR=YES was specified, insert a /FORMAT command for a valid message output descriptor and rerun the application.

BTS0049I **SEGMENT EDIT EXIT REQUEST
MESSAGE ECHO.**

Explanation

A segment edit exit DFSMExxx was called and has returned the return code 16. This is a request to echo the message.

User response

None. This message is informational.

BTS0050W **SEGMENT LENGTH GREATER
THAN Q BLKSIZE OR LRECL.**

Explanation

The BLKSIZE or LRECL in the QIOPCB DD or QALTPCB DD statement must be at least as large as the largest output segment.

System action

The output segment is discarded and IMS Batch Terminal Simulator continues processing.

User response

Increase the BLKSIZE, LRECL, or both operand values and rerun the application.

BTS0051W **MID AND DIF ARE INCOMPATIBLE
- INPUT IGNORED.**

Explanation

The MID and DIF in the format control block library were not processed by the Message Format Language Utility at the same time and are not usable for online editing.

System action

The input is discarded and IMS Batch Terminal Simulator continues processing.

User response

Recompile the input MSG and FMT. Ensure that the FMT includes a DEV statement for the proper device type and features. Ensure also that the phase 4 output is inserted into the online format data set.

BTS0052W **NO INPUT MESSAGE CREATED.**

Explanation

Editing of input data has resulted in a message with no data. This message is discarded and IMS Batch Terminal Simulator awaits new input data.

System action

IMS Batch Terminal Simulator continues processing.

User response

Make the necessary corrections, and rerun the application.

**BTS0053I /FOR COMMAND RECEIVED BUT
 NOT SINGLE SEGMENT MESSAGE.**

Explanation

The /FOR command has been formatted as an input message but is not a single segment message. The /FORMAT command is processed.

User response

Correct the format control blocks for the /FOR command.

**BTS0054I TERMINATING ACTION IN CARD
 NO. xx COL NO. yyy**

Explanation

This message is issued when a terminating action such as an immediate PEN or PFKn is encountered, causing the immediate transmission of a message. Only fields in the simulator statement up to and including this terminal action field are transmitted as input to the application.

User response

Check the simulator statement to be sure that the intended application input was transmitted; otherwise, no response is needed.

BTS0055I AUDIBLE ALARM SOUNDED.

Explanation

The alarm on the 3270 Display Station was triggered in an output operation.

User response

Check the SCA of the active format. If necessary, correct the simulator statement and rerun the application.

**BTS0056W UNABLE TO LOCATE MESSAGE
 DESCRIPTION - INPUT IGNORED.**

Explanation

Either an I/O error has occurred while attempting to fetch the message input descriptor block or the device input descriptor block, or the blocks were not in the format control block library.

System action

The input is discarded and IMS Batch Terminal Simulator continues processing.

User response

Check that the DIF exists in the library and that the FEATURE option agrees with the FEAT= operand coded on the ./D or ./T command. Make the necessary corrections and rerun the application.

BTS0057W INVALID PAGE REQUEST.

Explanation

The format of the request-for-new-page entry is improper. The page request has been ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct and rerun the application.

**BTS0058W PAGE REQUESTED NOT
 CONTAINED IN CURRENT
 MESSAGE.**

Explanation

A request for a page previous to the first or subsequent to the last of this message was made. The page request was ignored and subsequent input data processed.

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct and rerun the application.

**BTS0059W MULTIPLE PAGE MESSAGE NOT IN
 PROCESS, REQUEST IGNORED.**

Explanation

A paging request was made while a message was being viewed which was not formatted as a multiple-page message. The request was ignored, and processing continued.

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct and rerun the application.

BTS0060I RESET COMMAND COMPLETED

Explanation

A /RESET command was successfully processed.

User response

None. This message is informational.

BTS0061W INVALID SCREEN ADDRESS.

Explanation

An address in an output message exceeds the screen image capacity.

System action

The results are unpredictable, and subsequent output is ignored.

User response

Check format descriptions for incorrect linkages.

BTS0062W IDCARD ERROR.

Explanation

The IDCARD simulator statement has been ignored. IDCARD data must be 5 through 129 characters in a text string. The first character is used to store the hardware-generated attribute character, and the rest is considered to be the card contents. The data must not have been truncated.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues the processing.

User response

Check the field specification to make sure that the field has a length long enough to contain the Operator Identification Card data plus one attribute character. If the field specification is correct, correct the simulator statement, making sure that the first position in the data entry field is a blank space and that it is followed by the card data. (The data entry field on the simulator statement must be enclosed in single quotes.)

BTS0063W OPEN FAILED FOR QIOPCB OR QALTPCB.

Explanation

OPEN failed for QIOPCB or QALTPCB data sets.

System action

The output is discarded and IMS Batch Terminal Simulator continues processing.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for QIOPCB and QALTPCB. Correct and rerun the job. See *z/OS DFSMS Macro Instructions for Data Sets* for the probable causes of the OPEN errors.

BTS0064I COPY COMMAND ISSUED.

Explanation

A copy request was made using PA3, PFK12, or the corresponding bit in the SCA or DSCA field. IMS Batch Terminal Simulator simulates the COPY command by issuing this message.

User response

None. This message is informational.

BTS0065W INPUT IGNORED.

Explanation

Data was encountered between the terminating action and the end of the input statement. Data subsequent to the terminating action was ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

Check the simulator statement for untransmitted application input, and restructure the simulator statement as required for complete data transmission. Otherwise, place subsequent (untransmitted) data on a separate simulator statement when transmission is desired.

BTS0066I **ONE MESSAGE DEQUEUED IN PAX SIMULATION.**

Explanation

All physical pages for this message are displayed and the message is dequeued.

User response

None. This message is informational.

BTS0067I **NO MORE MESSAGES. SCREEN CLEARED.**

Explanation

The output message queue is empty and IMS Batch Terminal Simulator is now in an unformatted mode. The last screen image is no longer available for additional input.

User response

None. This message is informational.

BTS0068W **FORMATTER TERMINATED - INTERNAL ERROR CODE xxxxx**

Explanation

An error internal to the 3270 formatting facility occurred. IMS Batch Terminal Simulator is now in unformatted mode.

System action

IMS Batch Terminal Simulator continues processing.

User response

None.

Note: Examine the remaining output for further errors. Correct and rerun the application. If the error persists, inform the IMS Batch Terminal Simulator support personnel.

BTS0069E **INTERNAL ERROR. ABEND U4093 FOLLOWS.**

Explanation

An error internal to the 3270 formatting facility has occurred, and further processing is impossible.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

None.

Note: Examine the remaining output for further errors. Correct and rerun the application. If the error persists, inform the IMS Batch Terminal Simulator support personnel.

BTS0070W **MIXED MODE KANJI SUBFIELD LENGTH IS ODD. LINE NO. xx COL NO. yyy**

Explanation

The line and the column point to an attribute character of a field in the internal screen image. IMS Batch Terminal Simulator is processing a simulator statement that would update this field. This field has the "mixed mode" attribute. The start and end of a DBCS (Kanji) subfield in the text of the simulator statement are indicated by shift-out (SO) and shift-in (SI) characters, X'0E' and X'0F', respectively. An odd number of bytes have been specified.

System action

The simulator statement is ignored and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator statement.

BTS0071W **DFS290I NO MESSAGES AVAILABLE FOR OUTPUT.**

Explanation

No messages were available for output. The request is ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

None.

BTS0072W **DFS293A INVALID ID CARD
READER INPUT.**

Explanation

Input from a 3270 operator identification card reader was invalid. No EOR or EOI character was found to end the data input, or the LRC check failed.

System action

IMS Batch Terminal Simulator continues processing.

User response

Ensure that entry is made into a field large enough to contain all card data. Ensure that its IDCARD was specified correctly.

BTS0073I **STATUS CODE IN IOPCB CHANGED
TO = xx**

Explanation

The xx is the status code placed in the PCB requested by the ./S command.

User response

None. This message is informational.

BTS0074I **APPLICATION PGM ISSUED ROLL
CALL**

Explanation

This message is self-explanatory.

User response

None. This message is informational.

BTS0076W **QALTRAN BLKSIZE LESS THAN
SEGMENT LENGTH - nnnn**

Explanation

The BLKSIZE in the QALTRAN DD statement must be 26 bytes larger than the maximum length for a message segment that includes the LLZZ prefix field. nnnn is the length of the segment plus BTS header information to be inserted.

System action

The input is discarded and IMS Batch Terminal Simulator continues processing.

User response

Increase the BLKSIZE operand value in the QALTRAN DD statement to at least nnnn and rerun the application. For additional information, see “Customizing the BTS cataloged procedure” on page 45, under the description of the ALTSEG parameter for additional information.

BTS0077I **OUTPUT FOR ALTERNATE LOGICAL
TERMINAL xxxxxxxx**

Explanation

This message is self-explanatory.

User response

None. This message is informational.

BTS0078I **BUFFER xxxx FOR ALTERNATE
TERMINAL yyyyyyyy**

Explanation

This message is self-explanatory.

User response

None. This message is informational.

BTS0079W **DFS272 FORMAT BLOCK LEVEL
ERROR - INPUT IGNORED**

Explanation

This message is issued only while in MFS TEST mode. The DIF block that was fetched to process the input was not at the same level as the DOF block. See IMS message DFS272 for more information.

System action

IMS Batch Terminal Simulator continues processing.

User response

Correct format control block.

BTS0080W **CONFLICTING KEYWORDS**

Explanation

A simulator command contained mutually exclusive keywords. For example, the TYPE and MDL keywords of the ./T command occurred together in a simulator command.

System action

The command is ignored, and IMS Batch Terminal Simulator continues processing.

User response

Correct the simulator command.

BTS0081W	CHKP/XRST I/O AREA LIMITED TO 32K - BTS LIMITATION (I/O AREA_x)
-----------------	--

Explanation

This message is a warning. A checkpoint restart call was made for an area greater than 32,767 bytes, the maximum area that IMS Batch Terminal Simulator allows. *x* is the I/O area number.

System action

The area greater than 32,767 bytes is not printed, and IMS Batch Terminal Simulator continues processing.

User response

None.

BTS0082W	DFS574 UNEXPECTED DATA RECEIVED - INPUT IGNORED
-----------------	--

Explanation

MFS has received a field from the 3270 device that was not defined in the device input descriptor. The input is ignored.

System action

IMS Batch Terminal Simulator continues processing.

User response

Retry the input.

BTS0083W	EGCS FIELD BEGINS ON AN EVEN NUMBERED COLUMN.
-----------------	--

Explanation

An EGCS field begins on an even-numbered column. The field attributes and position are specified in the MFS format definition. Even column field specification is valid for the PS/55 computer, but is invalid for the IBM 3270.

System action

IMS Batch Terminal Simulator continues processing.

User response

Check your MFS format definition. Correct it if necessary, and rerun the application.

BTS0084W	NO RESPONSE, CONVERSATION TERMINATED
-----------------	---

Explanation

The application program ended without issuing an ISRT to the IOPCB.

User response

Correct and rerun the application.

BTS0088E	BTS CANNOT RUN UNDER THIS IMS RELEASE LEVEL
-----------------	--

Explanation

This IMS release level is not supported by IMS Batch Terminal Simulator.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check the STEPLIB data sets in the JCL or the TASKLIB data sets in the TSO CLIST and rerun the JOB.

BTS0089W	DFS273 QUEUE ERROR ON MESSAGE DELETION
-----------------	---

Explanation

See *IMS Messages and Codes*.

System action

IMS Batch Terminal Simulator continues processing.

User response

See *IMS Messages and Codes* for the meaning of the IMS error message.

BTS0090W	DFS291 INPUT MUST BEGIN FROM FIRST PHYSICAL PAGE
-----------------	---

Explanation

See *IMS Messages and Codes*.

System action

IMS Batch Terminal Simulator continues processing.

User response

See *IMS Messages and Codes* for the meaning of the IMS error message.

BTS0091W **DFS296 PROGRAM FUNCTION KEY
LITERAL ALLOWED ONLY ONCE
PER MESSAGE**

Explanation

See *IMS Messages and Codes*.

System action

IMS Batch Terminal Simulator continues processing.

User response

See *IMS Messages and Codes* for the meaning of the IMS error message.

BTS0092W **DFS298 INPUT MESSAGE
CANCELLED BY SEGMENT EDIT
EXIT**

Explanation

See *IMS Messages and Codes*.

System action

IMS Batch Terminal Simulator continues processing.

User response

See *IMS Messages and Codes* for the meaning of the IMS error message.

BTS0093W **DFS060 INVALID MESSAGE KEY
nnn**

Explanation

The Segment Edit Exit return code was 12. The displayed number (*nnn*) was passed to the BTS message generator module. The message could not be displayed for one of the following reasons:

- User message table DFSCMTU0 was not link-edited into BTS load module BTRC000.
- User message table DFSCMTU0 does not contain an entry for this message number.
- User message table entry length is greater than BTRCOUT LRECL.
- User message table entry length is greater than 100 bytes.
- User message table entry length is less than 5 bytes.

System action

IMS Batch Terminal Simulator continues processing.

User response

Include user message table DFSCMTU0 in the BTS load module BTRC000 and ensure that DFSCMTU0 contains a valid entry for this message number.

BTS0094W **DFSUxxxx aaaaaaaaa**

Explanation

The Segment Edit Exit return code was 12. User message number *xxxx* was requested. Message *aaaaaaaa* was found in user message table DFSCMTU0.

System action

IMS Batch Terminal Simulator continues processing.

User response

None.

BTS0095I **ALTERNATE TERMINAL OUTPUT
NOT FORMATTED.**

Explanation

3270 formatting was not specified on the last entered *./D* command.

System action

The alternate terminal output is discarded and IMS Batch Terminal Simulator continues processing.

User response

If formatting of the alternate terminal output is desired, enter a *./D* command with either the DDOF=3270mc operand or the TYPE=3270-An and SIZE=(*ll,cc*) operands.

BTS0096I **DB2=[VxRy | xx.y], SQL CALLS =
nnnnn**

Explanation

This message shows the Db2 version (*VxRy* or *xx.y*) and the total number (*nnnnn*) of Db2 SQL calls. The message is followed by Db2 SQL call statistics information, which lists the number of calls made for each Db2 SQL call type.

User response

None. This message is informational.

BTS0097W **BTSIN IS NOT FIXED OR
VARIABLE BLOCK**

Explanation

The record format of BTSIN is not fixed, fixed-block, variable, or variable-block as required. See the BTSIN requirements in the following topics:

- For batch mode, “[Step 2: Specifying the IMS Batch Terminal Simulator data sets](#)” on page 69.
- For interactive mode, “[Step 2: Specifying the IMS Batch Terminal Simulator data sets](#)” on page 88.

System action

IMS Batch Terminal Simulator ends with the message.

User response

Change the BTSIN data set to fixed, fixed-block, variable, or variable-block and rerun the job.

BTS0098W **INVALID STATUS CODE ON
INTERNAL CHKP CALL. STATUS
CODE = xx**

Explanation

As a result of an internal CHKP call, the status code xx was returned instead of a blank status code.

System action

After this message appears, IMS Batch Terminal Simulator ends with ABENDU4088.

User response

See the status code explanation in *IMS Application Programming APIs*. Analyze the cause of the status code and rerun the JOB.

BTS0099I **SSM PARAMETER IGNORED.**

Explanation

IMS Batch Terminal Simulator ignored the SSM value. See “[Applications that access Db2 databases](#)” on page 114 for details.

System action

The value for SSM is ignored and IMS Batch Terminal Simulator continues processing.

User response

None. This message is informational.

BTS0100I **ATTACHING DFSRRC00,
PARM=(aaa,bbbbbbbb,ccccccc,...)**

Explanation

The execution parameters shown as (aaa,bbbbbbbb,ccccccc,...) are passed to the IMS region controller (DFSRRC00) when attached by the BTS region controller (BTSRC000).

For an explanation of the IMS execution parameters, see *IMS System Definition*.

User response

If execution parameter bbbbbbbb or cccccccc is not correct, check the ./T commands you have supplied. If other parameters are not correct, check the BTS TSO CLIST. Make the necessary corrections, and rerun the application.

BTS0101A **ENTER NULL LINE TO OBTAIN
IMS-SCREEN FOR PCB(xxxxxxxx)**

Explanation

When a BTS message is displayed on the TSO screen image, you are prompted to respond when ready to view the next IMS screen image.

System action

IMS Batch Terminal Simulator waits for a response from you.

User response

Press Enter.

BTS0102W **CAUTION: DATABASE UPDATES
MAY BE IN PROCESS.**

Explanation

You have pressed the PA1 key which interrupts the processing.

When running IMS batch (that is, KW=DLI or KW=DBB), if IMS database updates are in process and execution is not resumed, the databases are likely to contain errors. The databases might need to be recovered or rebuilt.

When running an IMS online dependent region (that is, KW=BMP), if execution is not resumed, you might bring down the IMS control region.

System action

Issue message BTS0103A.

User response

None.

BTS0103A	HIT ENTER TO RESUME PROCESSING. (HIT PA1 AGAIN TO EXIT.)
-----------------	---

Explanation

Following message BTS0102W, you can resume execution or permanently interrupt execution.

System action

IMS Batch Terminal Simulator waits for a response from you.

User response

Press Enter to resume execution. Press the PA1 key to permanently interrupt execution.

BTS0104I	UNABLE TO OBTAIN IFI- INFORMATION. DB2 RETURN CODE xxxxxxxx, REASON CODE yyyyyyyy
-----------------	--

Explanation

IMS Batch Terminal Simulator tried to obtain IFI SQL statement information using the Db2 Instrumentation Facility Component Application Program Interface (IFC API), but received an error return code of 4 or 8. This message does not indicate any error in the application program being tested.

System action

The IFI SQL statement information trace is stopped and IMS Batch Terminal Simulator continues processing.

User response

Take one of the following actions:

- For the Db2 return code (xxxxxxx) and reason code (yyyyyyy), read *Db2 for z/OS Codes* that correspond to your Db2 environment and investigate the cause of the error. Check first that the error was not caused by Db2 monitor trace class 1 not being activated. Correct the error and rerun the application.
- Ignore this message if you do not need to obtain the IFI SQL statement information.

See “Db2 call trace listing” on page 144 for a description of the IFI SQL statement.

BTS0105I	SEGMENT DATA USER EXIT STARTED.
-----------------	--

Explanation

This message signals that the DL/I changed data user exit routine has been called. The trace information of IMS DL/I calls and Db2 SQL calls issued from the user exit is written between this message and the BTS0106I message.

User response

None. This message is informational.

BTS0106I	SEGMENT DATA USER EXIT ENDED.
-----------------	--------------------------------------

Explanation

This message signals that the DL/I changed data user exit routine has been ended. This message is followed by the trace information of the DL/I update call that called the user exit.

User response

None. This message is informational.

BTS0107I	DB2 CHANGED DATA CAPTURE EXIT STARTED.
-----------------	---

Explanation

This message signals that the Db2 changed data capture exit routine (DB2CDCEX) has been called. The trace information of IMS DL/I calls and Db2 SQL calls issued from the user exit (DB2CDCEX) is written between this message and the BTS0108I message.

User response

None. This message is informational.

BTS0108I	DB2 CHANGED DATA CAPTURE EXIT ENDED.
-----------------	---

Explanation

This message signals that the Db2 changed data capture exit routine (DB2CDCEX) has been ended. This message is followed by the trace information of the SQL update call that called the user exit (DB2CDCEX).

User response

None. This message is informational.

BTS0109E **DYNAMIC UNALLOCATION FAILED FOR DDITV02 DATA SET. RETURN CODE xx, S99ERROR yyyy, S99INFO zzzz.**

Explanation

Return code xx was returned to IMS Batch Terminal Simulator from the execution of the OS DYNALLOC macro.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See the *z/OS MVS Authorized Assembler Services Guide* for a discussion of the DYNALLOC macro and the cause of the error. Notify the system support group of the IMS Batch Terminal Simulator program error.

BTS0110E **DYNAMIC ALLOCATION FAILED FOR DDITV02 DATA SET. RETURN CODE xx, S99ERROR yyyy, S99INFO zzzz.**

Explanation

Return code xx was returned to IMS Batch Terminal Simulator from the execution of the OS DYNALLOC macro.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

See the *z/OS MVS Authorized Assembler Services Guide* for a discussion of the DYNALLOC macro and the cause of the error. If the return code (xx) is 04 and the error reason code (yyyy) is 021C, see the .P command and correct the unit group-name. Otherwise, notify the system support group of the IMS Batch Terminal Simulator program error.

BTS0111E **DYNAMIC ALLOCATION FAILED FOR DDOTV02 DATA SET. RETURN CODE xx, S99ERROR yyyy, S99INFO zzzz.**

Explanation

Return code xx was returned to BTS from the execution of the OS DYNALLOC macro.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Check the *z/OS MVS Programming: Authorized Assembler Services Guide* for a discussion of the DYNALLOC macro and the cause of the error. Notify the system support group of the BTS program error.

BTS0112E **BTS CANNOT RUN WITH THIS DB2 RELEASE LEVEL**

Explanation

This Db2 release level is not supported by IMS Batch Terminal Simulator.

System action

IMS Batch Terminal Simulator ends abnormally.

User response

Rerun the application with Db2 at the supported release level.

BTS0113I **EXTERNAL SUBSYSTEM CALL IS ISSUED**

Explanation

This message signals that the application program has issued the call to the external subsystem other than the Db2 subsystem and the IBM MQ subsystem.

System action

If you are running IMS Batch Terminal Simulator in IMS online environment (that is, KW=BMP), the call is passed to the IMS. If you are running IMS Batch Terminal Simulator in an IMS batch environment (that is, KW=DLI or KW=DBB), the call is discarded.

User response

None. This message is informational.

BTS0114I **MQI CALL IS ISSUED IN BTS BATCH ENVIRONMENT WITHOUT WEBSPHERE MQ BATCH ADAPTER (FUNC=funcrcode)**

Explanation

The application program has issued a call to the IBM MQ subsystem in BTS batch environment (that is,

KW=DLI/DBB) without an IBM MQ batch adapter. The *funccode* is the function code of the MQI call.

System action

The MQI call is discarded.

User response

If you need the MQI call trace information in the batch environment, check the STEPLIB data sets in JCL or the TASKLIB data sets in the TSO CLIST, and rerun the JOB.

BTS0115I **UNABLE TO OBTAIN
THE TRIGGERED QUEUE
INFORMATION. *reason***

Explanation

IMS Batch Terminal Simulator was unable to obtain the triggered queue information for the trigger message. *reason* is one of the following reasons:

FUNC=*funccode*, MQCC=*mqcc*, MQRC=*mqrc*

This message is issued when a nonzero return code is returned for the internal MQI call, which uses the IBM MQ batch adapter. *Funccode* is the function code of the internal MQI call; *mqcc* is the completion code; and *mqrc* is the reason code.

LOAD ERROR FOR *modname*

This message is issued when a LOAD fails for the module CSQBxxxx. *Modname* is the IBM MQ stub module name.

System action

The following values in the trigger message are blank, and IMS Batch Terminal Simulator continues processing:

- Name of triggered queue
- Name of process object
- Trigger data
- Environment data
- User data
- Queue manager name

User response

Take one of the following actions:

- Ignore this message if you do not need to obtain the triggered queue information.
- For the first form of this message, see *IBM MQ Reference* for the MQI return code, and investigate

the cause of the error. Correct the error and rerun the job.

- For the second form of this message, determine if the IBM MQ load-module library is available, make all the necessary corrections, and rerun the job.

BTS0116W **UNSUPPORTED TYPE OF
CALLABLE SERVICE IS INVOKED.**

Explanation

The type of IMS callable services invoked by the user exit of IMS exit routines is not supported by IMS Batch Terminal Simulator. IMS Batch Terminal Simulator supports only Get/Free storage services.

System action

The IMS callable service is discarded and IMS Batch Terminal Simulator continues processing.

User response

None.

BTS0117W **OPEN FAILED FOR ACBLIB DATA
SET.**

Explanation

The BTSACB data set could not be successfully opened by IMS Batch Terminal Simulator. IMS Batch Terminal Simulator might report an incorrect length on the BTSOUT or TSO screen when ISRT call for the DEDB is issued.

System action

IMS Batch Terminal Simulator continues processing.

User response

Check the DD statement in the JCL or the ALLOC statement in the CLIST for BTSACB. For the probable causes of the OPEN errors, see *z/OS DFSMS Macro Instructions for Data Sets*.

BTS0118W **UNABLE TO OBTAIN DMCB FROM
ACBLIB. *reason***

Explanation

IMS Batch Terminal Simulator was unable to obtain the DEDB DMCB for the indicated DBDNAME. IMS Batch Terminal Simulator might report an incorrect length on the BTSOUT or TSO screen when an ISRT call for the DEDB is issued. *reason* is one of the following reasons:

System action

IMS Batch Terminal Simulator does not load DFSCLV0x, and ignores ./E IMSSUF=.

User response

If necessary, specify either ./E IMSSUF=, ./D DDOF=, or ./D TYPE=, and rerun the job.

BTS0125W LOAD FAILED FOR DFSCLV0x

Explanation

DFSCLV0x load error. x is the suffix code specified by ./E IMSSUF=.

System action

IMS Batch Terminal Simulator runs with a default device type (DEV TYPE=(3270,2),FEAT=IGNORE).

User response

If necessary, correct the value of ./E IMSSUF=, and rerun the job.

BTS0126W UNABLE TO FIND LTERM: *lterm* IN DFSCLV0x

Explanation

The first LTERM name in TRXIN records does not exist in DFSCLV0x, when the transaction BTSAISR0 runs with ./E IMSSUF=. x is the suffix code specified by ./E IMSSUF=.

System action

IMS Batch Terminal Simulator ignores ./E IMSSUF=, and runs with a default device type (DEV TYPE=(3270,2),FEAT=IGNORE).

User response

Correct the value of ./E IMSSUF= in BTSIN or SKIP= in CNTLCRDS, and rerun the job.

BTS0127W OPEN FAILED FOR xxxxxxxx

Explanation

OPEN failed for the CNTLCRDS or TRXIN data sets when the transaction BTSAISR0 ran with ./E IMSSUF=.

System action

IMS Batch Terminal Simulator ignores ./E IMSSUF=, and runs with a default device type (DEV TYPE=(3270,2),FEAT=IGNORE).

User response

Check the DD statement in the JCL stream. If necessary, correct the DD statement and rerun the job.

BTS0128W LTERM IS NOT IDENTIFIED FROM TRXIN.

Explanation

TRXIN DD is empty or all data is skipped by SKIP= in the CNTLCRDS data set. IMS Batch Terminal Simulator cannot identify the LTERM name that is used to find the screen device type.

System action

IMS Batch Terminal Simulator ignores ./E IMSSUF= and runs with a default device type (DEV TYPE=(3270,2),FEAT=IGNORE).

User response

Check the TRXIN data set and rerun the job.

BTS0129I IMSSUF=x PROCESS STARTED.

Explanation

Gets TYPE, SIZE, and FEAT definitions automatically, because device type is not specified by ./D.

System action

IMS Batch Terminal Simulator continues processing.

User response

None. This message is informational.

BTS0130I EGCS FIELD(S) ARE NOT SHOWN CORRECTLY.

Explanation

Not enough space is available to put shift-out (SO) and shift-in (SI) characters for the EGCS and the DBCS fields in the screen. Therefore, the text is not readable.

User response

None. This message is informational.

BTS0131E DSNTIAR FAILED. RC=xx

Explanation

The Db2 message formatting routine (DSNTIAR) returned return code xx.

System action

After issuing this message, IMS Batch Terminal Simulator ends with an abend code of U4085.

User response

The return code from the Db2 message formatting routine is set to Register 15. For more information, see the *Db2 for z/OS Application Programming and SQL Guide*.

BTS0132E LOAD FAILED FOR DSNTIAR.

Explanation

An error occurred while loading DSNTIAR.

System action

After issuing this message, IMS Batch Terminal Simulator ends with an abend code of U4085.

User response

Correct the JCL stream by specifying the Db2 SDSNLOAD library in the JOBLIB or STEPLIB.

**BTS0133I AN MQCONN CALL WAS
INTERNALLY CHANGED TO AN
MQCONN CALL.**

Explanation

The application program issued an MQCONN call to the IBM MQ subsystem in a BTS batch environment (KW=DLI or KW=DBB). However, because IMS Batch Terminal Simulator does not support MQCONN calls in BTS batch environments, the MQCONN call was internally changed to an MQCONN call.

System action

Processing continues.

User response

If you want to issue an MQCONN call, you must run the application in a BTS online environment (KW=BMP).

BTS0134W MQCNO IS INVALID.

Explanation

The application program issued an MQCONN call to the IBM MQ subsystem in a BTS batch environment (KW=DLI or KW=DBB). However, the structure

identifier field (StrucId) in the MQCNO structure specifies an invalid MQCNO_STRUC_ID.

System action

Subsequent MQI calls are ignored and processing continues.

User response

Correct the invalid specification, and rerun the job.

**BTS0135I BTS LOADED THE APPLICATION
FRONT-END ROUTINE *routine*
THAT IS SPECIFIED BY THE
APPLFE KEYWORD.**

Explanation

The specified application front-end routine is loaded.

User response

None. This message is informational.

**BTS0136I BTS CALLED THE APPLICATION
FRONT-END ROUTINE *routine* FOR
INITIALIZATION PROCESSING.**

Explanation

The indicated application front-end routine is called for initialization processing.

User response

None. This message is informational.

**BTS0137I BTS CALLED THE APPLICATION
FRONT-END ROUTINE *routine* FOR
TRANSACTION PROCESSING.**

Explanation

The indicated application front-end routine is called for transaction processing.

User response

None. This message is informational.

**BTS0138I BTS CALLED THE APPLICATION
FRONT-END ROUTINE *routine* FOR
SHUTDOWN PROCESSING.**

Explanation

The indicated application front-end routine is called for shutdown processing.

User response

None. This message is informational.

BTS0139I **BTS ISSUED ROLB FOR
ENDROLB=YES OPTION**

Explanation

IMS Batch Terminal Simulator issued a ROLB call to rollback the database updates because the ENDROLB=YES option is active.

User response

None. This message is informational.

BTS0140W **UNICODE CONVERSION SERVICES
FAILED. RETURN CODE X'xx',
REASON CODE X'yy', SOURCE
CCSID sssss, TARGET CCSID tttt**

Explanation

The character conversion module (CUNLCNV) of z/OS Unicode Services returned the indicated return code and reason code. IMS Batch Terminal Simulator called this module to convert characters from the source CCSID to the target CCSID.

System action

IMS Batch Terminal Simulator continues processing without character conversion.

User response

See the *z/OS Unicode Services User's Guide and Reference* for the meaning of the return code and the reason code. If the error persists, notify the system support group of the IMS Batch Terminal Simulator program error.

BTS0141W **BTSISRIO IS NOT VARIABLE
FORMAT.**

Explanation

The record format of BTSISRIO is not variable or variable-block as required.

System action

IMS Batch Terminal Simulator discards the output and continues processing.

User response

Change the record format of the BTSISRIO data set to variable or variable-block and rerun the job.

BTS0429I **VSAM [SHOWCB | GENCB] ERROR,
RETURN CODE X'yy'**

Explanation

For an explanation of the DFS0429I message, see *IMS Messages and Codes*.

BTS0430I **UNABLE TO INITIALIZE DL/I
BUFFERING SERVICES, REASON
CODE xx**

Explanation

For an explanation of the DFS0430I message, see *IMS Messages and Codes*.

BTS0431I **STORAGE NOT AVAILABLE FOR
VSAM BUFFER POOL, RETURN
CODE X'yy'**

Explanation

For an explanation of the DFS0431I message, see *IMS Messages and Codes*.

BTS0432I **VSAM SHARED RESOURCE POOLS
CANNOT BE BUILT, RETURN CODE
X'yy'**

Explanation

For an explanation of the DFS0432I message, see *IMS Messages and Codes*.

BTS0500E **COMMUNICATION TIMEOUT
ERROR FROM DISTRIBUTED
ACCESS INFRASTRUCTURE.**

Explanation

IMS Batch Terminal Simulator received a communication timeout error from an internal API of IMS Tools Base Distributed Access Infrastructure.

System action

IMS Batch Terminal Simulator ends with an abend code of U4077.

User response

Ensure the Distributed Access Infrastructure connection between the client and IMS Batch Terminal Simulator and rerun the application.

BTS0501E THE DISTRIBUTED ACCESS INFRASTRUCTURE SERVER IS SHUTTING DOWN.**Explanation**

IMS Batch Terminal Simulator received a connection error from an internal API of Distributed Access Infrastructure. The Distributed Access Infrastructure server is shutting down.

System action

IMS Batch Terminal Simulator ends with an abend code of U4076, U4077, or U4078.

User response

Restart the Distributed Access Infrastructure server and rerun the application.

BTS0502E DISTRIBUTED ACCESS INFRASTRUCTURE XCF ERROR.**Explanation**

IMS Batch Terminal Simulator received an XCF error from an internal API of Distributed Access Infrastructure.

System action

IMS Batch Terminal Simulator ends with an abend code of U4077 or U4078.

User response

Ensure that the Distributed Access Infrastructure server is running and rerun the application.

BTS0503E ABEND DUE TO DISTRIBUTED ACCESS INFRASTRUCTURE INTERNAL ERROR.**Explanation**

IMS Batch Terminal Simulator received an error from an internal API of Distributed Access Infrastructure. This error condition is different from the other Distributed Access Infrastructure API errors that are described by BTS0500E, BTS0501E, and BTS0502E messages.

System action

IMS Batch Terminal Simulator ends with an abend code of U4076, U4077, or U4078.

User response

Notify the IMS Batch Terminal Simulator support personnel of the error.

IMS Batch Terminal Simulator resource adapter messages

Use the information in these messages to help you diagnose and solve IMS Batch Terminal Simulator resource adapter problems.

Message format

IMS Batch Terminal Simulator resource adapter messages adhere to the following format:

```
AIInnnnx
```

Where:

AII

Indicates that the message was issued by IMS Batch Terminal Simulator resource adapter.

nnnn

Indicates the message identification number

x

Indicates the severity of the message:

E

Indicates that an error occurred, which might or might not require operator intervention.

I

Indicates that the message is informational only.

W

Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

Explanation:

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

User response:

The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

AIIO003E ***method_name* error. Failed to connect to host *host_name*, port *port_number*. *Java_exception***

Explanation

The IMS TM resource adapter was unable to connect to the host and port combination. *java_exception* indicates the reason for the failure to connect.

User response

Examine the exception to determine the reason for the failure to connect to the host.

AIIO005E ***method_name* error. A communication error occurred during sending or receiving the IMS message. *clientID=client_ID java_exception***

Explanation

IMS TM Resource Adapter was unable to successfully complete a send and receive interaction with the target IMS Connect. The Java exception indicates the reason for the failure to complete the interaction.

User response

Use the client ID to help analyze the trace data from the different components that are involved. The client ID is the ID of the connection on which the communication exception occurred. Examine the exception to determine the reason for the failure.

AIIO007E ***method_name* error. The *property_name* property value *property_value* is not supported.**

Explanation

The value specified for the property is not supported

User response

Provide a supported value for the named property.

AIIO008E ***method_name* error. The value *property_value* of the *property_name* property exceeds the maximum allowable length of *max_property_length*.**

Explanation

The length of the value that is supplied for the property exceeds *max_property_length*, the maximum length that is allowed for values for *property_name*.

User response

For the named property, provide a value that does not exceed *max_property_length*.

AIIO009E ***method_name* error. The *property_name* property value *property_value* is not valid.**

Explanation

The value *property_value* specified for the property *property_name* is not valid.

User response

Provide a valid value for the *property_name* property. See the online help for the valid value for this property.

AIIO010E ***method_name* error. The method was invoked on an invalid IMSConnection instance.**

Explanation

The *method_name* method was invoked on an invalid IMSConnection instance. The method was most likely issued on an IMSConnection instance that was already closed. If the method name is lazyEnlist, an attempt was made to enlist a connection in the current transaction that could not be enlisted.

User response

If the method name is lazyEnlist, ensure that your application is not using non-managed connections in

a managed environment. Otherwise, ensure that the IMSConnection instance is not already closed before you attempt to use it or close it.

AII0011E *method_name* error. The method invoked on an invalid IMSInteraction instance.

Explanation

The method was most likely issued on an IMSInteraction instance that was already closed.

User response

Ensure that the IMSInteraction instance is not already closed before you attempt to use it or close it.

AII0012E *method_name* error. The value provided for HostName is null or an empty string.

Explanation

The method was invoked using a null or empty HostName parameter.

User response

Provide a valid HostName parameter. In a managed environment, the property value is specified when you are configuring a connection factory to be used by WebSphere Application Server. In a non-managed environment, the property value is specified in your Java application.

AII0014E *method_name* error. The input record contains no data.

Explanation

method_name was invoked with an input record that contained no data.

User response

Verify that the input record that you provide is not empty.

AII0015E *method_name* error. An unexpected error occurred while building the input message. *Java_exception*

Explanation

An unexpected internal error was encountered while building the input message.

User response

Contact IBM Software Support and provide the information on the Java exception that was thrown.

AII0016E *method_name* error. The message was encoded using an unsupported code page. *Java_exception*

Explanation

The message cannot be encoded using the code page that was specified or the code page was not found.

User response

Ensure that the local Java client has code pages 1047 and 037 installed and that they are not corrupted

AII0017E *method_name* error. An invalid value is provided for TraceLevel.

Explanation

An invalid trace level was specified.

User response

Specify a valid trace level. Optionally, this exception can be ignored because the default trace level is for this connection factory. In this case, the connection factory is still usable but the trace level is the default trace level.

AII0018E *method_name* error. The value provided for PortNumber is null.

Explanation

The method was invoked using a null port number.

User response

Provide a valid PortNumber parameter. In a managed environment, the property value is specified when you configure a Connection Factory to be used by WebSphere Application Server. In a non-managed environment, the property value is specified in your Java application.

AII0024E *method_name* error. Invalid segment length (LL) of *LL_value* in the input object. *Java_exception*

Explanation

The input message provided by the Java program for the IMS application program contains a value for its

segment length that is negative, 0, or greater than the number of bytes of data in the message segment.

User response

Provide the correct value for the segment length of the input message.

AII0025E *method_name* error. Invalid segment length (LL) of *LL_value* in the message.

Explanation

The output message provided by the IMS application program contains a value for its segment length that is negative, 0, or greater than the number of bytes of data in the message segment.

User response

Ensure that your IMS application program provides valid lengths for the segments of its output message.

AII0026E *method_name* error. An error was encountered while processing the IMS message.
Source_method_name:source_exception

Explanation

An error occurred while processing the IMS transaction input or output message. The exception provides additional information regarding the cause of the error.

User response

Examine the exception for information regarding the cause of the error.

AII0030E *method_name* error.
Source_method_name:source_exception

Explanation

A runtime error or exception was detected in the named method during the interaction.

User response

Examine the exception for information regarding the cause of the error.

AII0031E *method_name* error. Protocol violation. The interaction verb *interaction_verb* is not allowed for

the current state *current_state*.
Java_exception

Explanation

The interaction attempted by the application resulted in a protocol violation.

User response

Ensure that you are using an appropriate value for the *interactionVerb* property. Check the IMS TM resource adapter documentation for supported values of the *interactionVerb* property.

AII0035E *method_name* error. Local transaction is not supported.

Explanation

Local transactions are not supported by the IMS TM resource adapter.

User response

Ensure that your Java application uses classes and methods that are appropriate for the level of support that is provided by the BTS resource adapter.

AII0037E *method_name* error. *ResultSet* is not supported.

Explanation

ResultSets are not supported by the IMS TM resource adapter.

User response

Ensure that your Java application uses classes and methods that are appropriate for the level of support that is provided by the BTS resource adapter.

AII0039E *method_name* error. Not in **CONNECT** state.

Explanation

The sequence of interactions between the IMS TM resource adapter and IMS Connect is not valid. The current state of the protocol used for the interactions between the IMS TM resource adapter and IMS Connect is not **CONNECT** as it needs to be at this point in the interactions.

User response

This error is most likely an error in the IMS TM resource adapter or in IMS Connect. Contact IBM Software Support.

AIIO041E *method_name error. An invalid interactionSpec intefactionSpec was specified.*

Explanation

An invalid InteractionSpec object was passed to the execute method of class com.ibm.connector2.ims.ico.IMSInteraction.

User response

Ensure that the InteractionSpec object that you pass to the execute method of class com.ibm.connector2.ims.ico.IMSInteraction is of type com.ibm.connector2.ims.ico.IMSInteractionSpec.

AIIO042E *method_name error. The input is not of type Streamable.*

Explanation

The Input object provided to the execute method of com.ibm.connector2.ims.ico.IMSInteraction for the input parameter either was null or did not implement the interface javax.resource.cci.Streamable.

User response

Ensure that you provide a valid javax.resource.cci.Record object for the input parameter to the execute method. For example, ensure that this object implements the interfaces javax.resource.cci.Record and javax.resource.cci.Streamable.

AIIO043E *method_name error. The output is not of type Streamable.*

Explanation

The output object provided to the execute method of com.ibm.connector2.ims.ico.IMSInteraction either was null or did not implement the interface javax.resource.cci.Streamable. This exception occurs most likely when an application is written to use the J2EE Connector Architecture Common Client Interface (CCI).

User response

Ensure that you provide a valid output object to the execute method

AIIO044E *method_name error. RecordFactory is not supported by the BTS Resource Adapter.*

Explanation

RecordFactory is not supported by the IMS TM resource adapter.

User response

Ensure that your Java application uses classes and methods that are appropriate for the level of support that the BTS resource adapter provides.

AIIO045E *method_name error. Invalid type of ConnectionRequestInfo.*

Explanation

An invalid ConnectionRequestInfo object was passed to an IMS TM resource adapter method

User response

This is most likely an error in the BTS resource adapter. Contact IBM Software Support.

AIIO049E *method_name error. The security credentials passed to getConnection do not match existing security credentials.*

Explanation

The security credentials in the request do not match the security credentials of the IMSManagedConnection instance that is used to process the request.

User response

Contact IBM Software Support.

AIIO050E *method_name error. Invalid RACF user id specified in SSLKeyStoreName or SSLTrustStoreName when specifying a RACF keystore or truststore.*

Explanation

The user ID specified in the SSLKeyStoreName or the SSLTrustStoreName property for the RACF keystore or truststore is not valid.

User response

Specify a valid user ID and ensure that the RACF user ID is less than 8 characters long.

AIIO053E ***method_name error. The client ID value is not valid. The prefix HWS is reserved by the IMS TM resource adapter.***

Explanation

The value specified for the clientID property is not valid. The prefix HWS is reserved by the IMS TM resource adapter.

User response

Provide a valid value for the clientID property. The value must be 8 characters long and cannot begin with a blank field or the reserved prefix HWS. Valid characters are A - Z, 0 - 9, @, #, and \$.

AIIO054E ***method_name error. Invalid ConnectionSpec.***

Explanation

The BTS resource adapter was unable to cast the provided connectionSpec to type IMSConnectionSpec. The BTS resource adapter works only with an IMSConnectionSpec or a derivative of IMSConnectionSpec as its connectionSpec.

User response

Ensure that the connectionSpec used by your application is an IMSConnectionSpec object or inherits from IMSConnectionSpec.

AIIO055E ***method_name error. Failed to cast the connection object to IMSConnection.***

Explanation

The IMS TM resource adapter was unable to cast the connection object allocated by the ConnectionManager for this connection to type IMSConnection. The BTS resource adapter works only with an IMSConnection or a derivative of IMSConnection as its connection object.

User response

Contact IBM Software Support.

AIIO057E ***method_name error. Invoked with invalid connection handle.***

Explanation

The application is in an illegal state. The connection handle (IMSConnection instance) used for this interaction is not valid. The application might be attempting to use a connection handle that is either for a previously used connection or for the wrong connection if the application has more than one connection open.

User response

Ensure that the application is using the valid IMSConnection instance for that connection.

AIIO058E ***method_name error. Local Option is not supported with the BTS resource adapter.***

Explanation

You can use Local Option to communicate with IMS Connect only if your application is using the IMS TM resource adapter with commit mode 1.

User response

Ensure that your application is selected with commit mode 1. If you plan to run your application with commit mode 0, correct your application to use TCP/IP communication.

AIIO059E ***method_name error. SYNC_END_CONVERSATION interaction with Commit Mode 0 is not supported.***

Explanation

Interaction SYNC_END_CONVERSATION with commit mode 0 is not supported.

User response

The BTS resource adapter supports the interaction combinations of SYNC_END_CONVERSATION with commit mode 1 and SYNC_SEND_RECEIVE with commit mode 0.

AIIO060E ***method_name error. Error loading Local Option native library: libname=library_file_name. source_exception.***

Explanation

The Local Option native library cannot be found in any of the directories listed in the LIBPATH environment

variable, which is expected behavior if Local Option has never been configured.

User response

Option communication is not supported by BTSRA so this error is expected if you inadvertently specify an IMSConnectName in your J2C connection factory custom properties. This error will go away when you remove the IMSConnectName from the J2C connection factory custom properties.

AIIO062E *method_name* error. Error loading Local Option native method: libfilename=*library_file_name*, methodname=*native_method_name*. *source_exception*.

Explanation

The Local Option native method cannot be found. Local Option communication is not supported by BTSRA.

User response

Remove the IMSConnectName from the J2C connection factory custom properties to resolve this error.

This error occurs if an IMSConnectName in your J2C connection factory custom properties was specified and a version other than 11 of the Local Option native library was installed. The library was loaded but failed when one of its methods was called.

AIIO063E *method_name* error. Exception thrown in native method *source_exception*.

Explanation

An internal error occurred in the Local Option native method.

User response

Remove the IMSConnectName from the J2C connection factory custom properties to resolve this error.

AIIO064E *method_name* error. Invalid security credential.

Explanation

WebSphere Application Server did not provide a security credential that is supported by the BTS resource adapter

User response

Ensure that you have the correct level of WebSphere Application Server for z/OS installed. Configure WebSphere Application Server for z/OS to provide a security credential that is supported by BTS resource adapter (PasswordCredential for TCP/IP connections). UToken GenericCredential for Local Option connects is not supported because BTSRA does not support Local Option connections.

AIIO065E *method_name* error. Error obtaining credential data from the security credential *source_exception*.

Explanation

A security-related error occurred in obtaining the credential data from the security credential provided by the application server.

User response

Ensure that you correctly set up security for your application server so that the user associated with the calling program is authorized to extract the data from a security credential.

AIIO071E *method_name* error. A communication error occurred when processing the XA *command_type* operation. *java_exception*

Explanation

A communication failure during the processing of a global transaction can occur because of a TCP/IP or socket failure or if IMS Connect is down. The connection in error is not reused.

User response

XA operations (global transactions in distributed environments) are not supported by BTSRA.

AIIO072E *method_name* error. The associated UR for the Xid is not found.

Explanation

During transaction processing a unit of recovery (UR) that was tied to a specific X/Open identifier (Xid) was eliminated by manual intervention or an error in IMS Connect or Recovery Resource Services (RRS).

User response

XA operations (global transactions in distributed environments) are not supported by BTSRA.

AII0073E *method_name error. RRS is not available.*

Explanation

Recovery Resource Services (RRS) has been brought down or communication between RRS and IMS Connect has ended.

User response

Global transactions are not supported by BTSRA.

AII0074E *The RRS `rss_routine` call returns with a return code `rss_routine_code`.*

Explanation

During the processing of the global transaction the following Recovery Resource Services (RRS) error message was passed in by IMS Connect.

User response

Global transactions are not supported by BTSRA.

AII0075E *method_name error. The transaction branch may have been heuristically completed. `rrs_exception`*

Explanation

An RRS error has been passed in by IMS Connect that indicates that the processing of your transaction might have been left in a heuristic situation, where part of the transaction was committed and part of it encountered an error during the commit phase.

User response

Global transactions are not supported by BTSRA.

AII0076E *method_name error. An internal error occurred. `rrs_exception`*

Explanation

An internal error occurred while trying to extract information about a Recovery Resource Services (RRS) error message from IMS Connect. The exception is an AII0074E error message indicating the RRS routine and return code associated with the error.

User response

Global transactions are not supported by BTSRA.

AII0077E *method_name error. The transaction has already rolled back. `rrs_exception`*

Explanation

A Recovery Resource Services (RRS) error has been passed by IMS Connect and indicates that the attempt to roll back a transaction has been made a second time on the same unit of recovery (UR).

User response

Global transactions are not supported by BTSRA.

AII0078E *method_name error. A valid user-specified clientID is required for interactions on a dedicated persistent connection.*

Explanation

A valid, user-specified value is required for the clientID property when the commit mode is 0, and the interaction is using a dedicated persistent socket connection.

User response

Provide a valid value for the clientID property. The value must be 8 characters long and cannot begin with a blank field or the reserved prefix HWS. Valid characters are A - Z, 0 - 9, @, #, and \$.

AII0079E *method_name error. IMS returned a DFS message: `DFS_message`*

Explanation

IMS returned a DFS message instead of the output of the IMS transaction. This exception is thrown if the interaction uses the value `IMS_REQUEST_TYPE_IMS_TRANSACTION` for the `imsRequestType` property.

User response

Find the explanation and response that corresponds to the DFS message in the IMS Messages and Codes documentation, and then address the problem in IMS.

AII0083E *method_name error. `SYNC_SEND_RECEIVE`, and `SYNC_SEND`, interactions with Commit Mode 0 are not valid*

within the scope of a global transaction.

Explanation

Global transactions requires SYNC_LEVEL_SYNCPOINT and are only valid with commit mode 1.

User response

To use commit mode 0, ensure that your application is configured as a non-transactional application. To run your interactions within the scope of a global transaction, commit mode must be 1.

AII0084E *method_name* error. An unexpected internal BTS Resource Adapter error occurred. *source_methodsource_exception*

Explanation

This exception occurs if Java 2 security is enabled, and the user associated with the calling program or any program in the current call stack is not authorized to execute the method.

User response

Ensure that you have correctly set up security for your application server so that the user associated with the calling program, along with any programs in the current call stack at the time of the exception, are authorized to execute the method. Alternatively, turn off Java 2 security checking in the application server.

AII0085E *method_name* error. Protocol violation. A user-specified clientID is not allowed for interactions on a shareable persistent connection.

Explanation

Because the connection factory is configured for a shareable persistent socket, a user-specified client ID is not allowed.

User response

For a shareable persistent socket connection factory, the BTS resource adapter provides a generated client ID. A user-specified client ID is not allowed. To determine if you are using a shareable persistent socket, check for a value of FALSE for the CMODedicated property of the connection factory that is used by the interaction.

AII0086E *method_name* error. Invalid value was specified for CommitMode property.

Explanation

The CommitMode value you have specified in the commitMode property field is invalid.

User response

Ensure that your application has set a valid value for commitMode property.

AII0087E *method_name* error. Protocol violation. Commit Mode 1 is not allowed for interactions on a dedicated persistent connection.

Explanation

Because the connection factory is configured for a dedicated persistent socket, commit mode 1 is not allowed.

User response

Determine if you are using a dedicated persistent socket by checking for a value of TRUE for the CMODedicated property of the connection factory that is used by the interaction.

AII0113E *method_name* error. Socket timeout has occurred for this interaction. The socket timeout value specified was *socket_timeout_value* milliseconds. *source_exception:exception_reason*

Explanation

The timeout value for the BTS resource adapter to receive a response from BTS is greater than the value that is specified for socket timeout.

User response

Ensure that the time value of socket timeout is sufficient for the BTS resource adapter to receive a response from BTS.

AII0114E *method_name* error. The socket timeout value of *socket_timeout_value* milliseconds is not valid. *source_exception:exception_reason*

Explanation

The value that was specified for socket timeout is not valid.

User response

Review the exception reason provided. Ensure a positive numerical value was given for the socket timeout property.

AIIO115E *method_name* error. A TCP error occurred.

Explanation

This is an error in the underlying protocol.

User response

Contact your network administrator.

AIIO116E *method_name* error. A Common Client Interface error occurred.

Explanation

An error occurred in the underlying protocol.

User response

Contact IBM Software Support.

AIIO117E *method_name* error. Protocol violation. Commit Mode 1 is not allowed for SYNC_SEND interactions.

Explanation

The BTS resource adapter supports only commit mode 0 for SYNC_SEND interactions.

User response

Correct either the commit mode or the interaction verb. Commit mode 0 is required for SYNC_SEND interactions. Commit mode 1 is valid with SYNC_SEND_RECEIVE and SYNC_END_CONVERSATION interactions.

AIIO118E *method_name* error. Protocol violation. IMS request type 2(IMS_REQUEST_TYPE_IMS_COMMAND) is not allowed for SYNC_SEND nor SYNC_END_CONVERSATION interactions.

Explanation

The value 2 (IMS_REQUEST_TYPE_IMS_COMMAND) that was specified for the imsRequestType property is invalid.

User response

The value of 2 (IMS_REQUEST_TYPE_IMS_COMMAND) for the ImsRequestType property is valid only with the SYNC_SEND_RECEIVE interaction. The value of 1 (IMS_REQUEST_TYPE_IMS_TRANSACTION) is required for SYNC_SEND, and SYNC_END_CONVERSATION interactions.

AIIO121E *method_name* error. Invalid reRoute name value. Prefix HWS is reserved for use by IMS TM Resource Adapter.

Explanation

The value that is specified for the reRouteName property contains the prefix HWS, which is reserved for use by the IMS TM resource adapter.

User response

Do not provide a value for the reRouteName property because reroute is not supported.

AIIO122E *method_name* error. Invalid reRoute value. When purgeAsyncOutput value is true, reRoute value cannot be true.

Explanation

The value for the reRoute property is invalid because the value for the purgeAsyncOutput property is TRUE, or the default value (TRUE) is used for the purgeAsyncOutput property.

User response

Set the purgeAsyncOutput property to FALSE if you want to set the reRoute property to TRUE.

AIIO123E *method_name* error. A Sync Level value of *sync_level* is not supported for Commit Mode *commit_mode* interactions.

Explanation

The value specified for IMSInteractionSpec.syncLevel is not supported for commit-then-send (commit mode 0) interactions. Confirm (1) is the only supported

value of IMSInteractionSpec.syncLevel for commit-then-send (commit mode 0) interactions.

User response

Specify the value of confirm (1) for the IMSInteractionSpec.syncLevel property, or accept the default Sync Level value of confirm (1), for commit-then-send (commit mode 0) interactions.

AIIO124E *method_name* error. **SYNC_SEND_RECEIVE interactions with Commit Mode *commit_mode* and Sync Level *sync_level* are not supported with Local Option. Local Option**

Explanation

Commit-then-send (commit mode 0) interactions are not supported with Local Option. Send-then-commit (commit mode 1) interactions with Sync Level confirm (1) are not supported with Local Option.

User response

Do not use Local Option for this functionality.

AIIO127E *method_name* error. **Protocol violation. The Mode *mode* is not allowed for the current state *state*. *java_exception***

Explanation

The BTS resource adapter is in an illegal state.

User response

This error might be an error in the BTS resource adapter. Contact IBM Software Support.

AIIO128E *method_name* error. **The Sync Level property value of *sync_level* given is invalid. Sync Level NONE (0) and Sync Level CONFIRM (1) are the only values supported by the setSyncLevel(int) method.**

Explanation

A non-zero or non-one value was specified as to input to the setSyncLevel(int) method.

User response

Specify either 0 or 1 as a sync level property value input to the setSyncLevel(int) method.

AIIO131E *method_name* error. **The property *property_name* is not supported for Interaction Verb *interaction_verb*. The property *property_name* can only be specified for Interaction *interaction*.**

Explanation

The ignorePURGCall property is not supported for SYNC_END_CONVERSATION interactions.

User response

Use the SYNC_SEND or SYNC_SEND_RECEIVE interaction on a shareable persistent socket connection. If want to execute a SYNC_END_CONVERSATION interaction, do not specify TRUE for the IMSInteractionSpec ignorePURGCall property.

AIIO200E *method_name* error. **An error occurred while loading the BTS configuration XML file. The original error was: *original_exception_message*.**

Explanation

The configuration XML file is invalid or was not specified.

User response

In the WebSphere Application Server administrative console, edit the IMS Batch Terminal Simulator resource adapter (BTSRA), and specify a valid BTSTConfigFile XML in the BTSRA J2C connection factory. The XML file is exported from the IMS Batch Terminal Simulator Eclipse interface and specifies the runtime environment and configurations for an IMS application.

AIIO201E *method_name* error. **An error occurred while loading the BTS configuration XML file. The original error was: *original_exception_message*.**

Explanation

The configuration XML file is invalid or was not specified.

User response

In the WebSphere Application Server administrative console, edit the IMS Batch Terminal Simulator

resource adapter (BTSRA), and specify a valid BTSSConfigFile XML in the BTSRA J2C connection factory. The XML file is exported from the IMS Batch Terminal Simulator Eclipse interface and specifies the runtime environment and configurations for an IMS application.

AIIO202E *method_name* error. The *function_name* support provided by TMRA is not available in BTSRA.

Explanation

The client application on WebSphere Application Server made a request that is not supported by

the IMS Batch Terminal Simulator resource adapter (BTSRA).

User response

When testing applications with BTSRA, avoid using TMRA functions that are directly related to IMS Connect or IMS OTMA. In general, IMS Batch Terminal Simulator does not support TMRA functions that are related to IMS Connect or IMS OTMA, such as resumeTpipe, asynchronous output, or reroute.

Playback utility return codes

The Playback utility modules generate return codes to indicate the success or failure of a job.

The following return codes are issued by the Playback utility.

0

Job successfully ended.

8

Job ended with an error message.

Playback utility abend codes

The Playback utility issues an abend code before it ends abnormally (abend).

For each abend code, the following information is provided when applicable:

Explanation:

The Explanation section explains what the abend means; why it occurred; what caused it; what its variable entry fields are.

System action:

The System action section explains what is happening as a result of the condition that caused the abend; whether the system is waiting for responses.

User response:

The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

3501

Explanation

An unexpected IMS status code was returned during a DLI call.

System action

The Playback utility terminates abnormally. This abend code is accompanied by messages BTSA1021E.

User response

Provide appropriate actions associated with message BTSA1021E.

Playback utility messages

Use the information in these messages to help you diagnose and solve the Playback utility problems.

Message format

The Playback utility messages adhere to the following format:

```
BTSA $nnnnx$ 
```

Where:

BTSA

Indicates that the message was issued by the Playback utility

$nnnn$

Indicates the message identification number

x

Indicates the severity of the message:

A

Indicates that operator intervention is required before processing can continue.

E

Indicates that an error occurred, which might or might not require operator intervention.

I

Indicates that the message is informational only.

W

Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

Explanation:

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

System action:

The System action section explains what the system will do in response to the event that triggered this message.

User response:

The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

BTSA1001E CROSSREF AND BTSOUT FILES ARE OUT OF SYNC.

Explanation

End of file was reached in the CROSSREF file prior to processing the last screen display from the BTSOUT file.

System action

Processing terminates with return code 8.

User response

Determine the reason for the out-of-sync condition and resubmit the job. One possible condition is that the SKIP parameter might have been used to start at a

point other than the first record in the CROSSREF file without an equivalent SKIP being taken in the TRXIN file during BTSAPRE0 processing.

BTSA1002I *ddname: input statement image*

Explanation

The message displays the data contained on a control statement record that was read from the file associated with the indicated ddname.

System action

Processing continues.

User response

None. This message is informational.

BTSA1003E **BTS PROCESSING FAILURE
DETECTED IN BTSOUT FILE.**

Explanation

IMS Batch Terminal Simulator was unable to create screen images for all of the input it was attempting to process. IMS Batch Terminal Simulator error messages were encountered by the BTSAEDT0.

System action

Processing terminates with return code 8. This message will be accompanied by other messages that will better describe the problem.

User response

Determine and correct the problem.

BTSA1004E **FAILURE MAY BE RELATED TO
MODNAME *modname* IN RECORD
sequence# IN CROSSREF FILE.**

Explanation

See the explanation associated with message BTSA1009E

where:

modname represents the MOD name.

sequence# represents the sequence number of the record.

System action

Processing terminates with return code 8.

User response

See the user response for message BTSA1009E for appropriate actions.

BTSA1005E **OPEN FAILURE ON *ddname* DATA
SET.**

Explanation

The indicated data set could not be opened.

ddname represents the DD name from the JCL.

System action

Processing terminates with return code 8.

User response

Supply a DD statement for the indicated data set, or remove the request for the data set from the control cards.

BTSA1006E **INVALID NUMERIC
ENCOUNTERED.**

Explanation

The value associated with a keyword should be numeric; however, non-numeric data was found.

System action

Processing terminates with return code 8.

User response

Correct the invalid keyword and resubmit the job.

BTSA1007E **MUTUALLY EXCLUSIVE
KEYWORDS *keyword1* AND
keyword2 HAVE BEEN
ENCOUNTERED.**

Explanation

The two indicated keywords have mutually exclusive functions.

System action

Processing terminates with return code 8.

User response

Select either keyword (not both) and resubmit the job.

BTSA1008E **REQUIRED KEYWORD *keyword*
HAS NOT BEEN SPECIFIED.**

Explanation

The indicated keyword is required.

System action

Processing terminates with return code 8.

User response

Specify a value for the keyword and resubmit the job.

BTSA1009E **FAILURE MAY BE RELATED TO
RECORD *sequence#* IN TRXIN
FILE CREATED DURING BTSAPREO
PROCESSING.**

Explanation

IMS Batch Terminal Simulator was not able to successfully complete using the BTSIN statements created by BTSAPRE0. Error message BTS0004W, generated by IMS Batch Terminal Simulator, indicated that IMS Batch Terminal Simulator was unable to schedule additional transactions. The failure might have been initiated by the indicated or preceding record in the indicated file.

System action

Processing terminates with return code 8.

User response

Determine and correct the problem. One of the following might be in error:

- A restart might have been attempted where the SKIP= specification for TRXIN records in BTSAPRE0 was inconsistent with the SKIP= specification for CROSSREF records in BTSAEDT0.
- A PROCESS= parameter might have been used in the BTSAISR0 without a corresponding PROCESS= parameter being used in the subsequent BTSAEDT0.
- The PAGES= value associated with a MODNAME= specification in BTSAPRE0 might be incorrect.

**BTSA1010E BTSIN CONTROL STATEMENTS
 MAY BE INVALID.**

Explanation

The BTSAEDT0 has terminated without formatting any transaction screens. One or more of the control statements included in the BTSIN file might be invalid.

System action

Processing terminates with return code 8.

User response

Determine the error and correct the control statements. One possible cause of the problem is that BTSAISR0, the initial transaction being processed by BTSAEDT0, does not have a corresponding TC=parameter on the ./T statement supplied in the BTSIN to BTSAEDT0.

**BTSA1011E MAXIMUM NUMBER OF *xxxxx* IN
 ARRAY EXCEEDED: *value***

Explanation

The maximum capacity of the indicated array has been exceeded. See the following table for an explanation of the message.

<i>xxxxx</i>	<i>value</i>	Explanation
LOG DSNS	100 (default)	Log data set name in RECON
IGNORE TERM TERMIN TRX USERID	500	Keywords in CNTLCRDS control statement
TRX	2500	TRANSACTION that was specified in '08' log
RACFUSER	5000	USERID that was specified in '01' log
MODNAMES	2500	MODNAME that was specified in '01' log
TRXUSE	10000	'07' log
UOWS	100	Inflight UOW (unit of work) for x'01' and x'03'

System action

Processing terminates with return code 8.

User response

See the following table to determine the action to take.

<i>xxxxx</i>	<i>value</i>	User response
LOG DSNS	100 (default)	1. Change the default value by specifying the MAXLOGS= keyword. 2. Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the RECON.
IGNORE TERM TERMIN TRX USERID	500	Up to 500 keyword specifications are allowed in CNTLCRDS control statement. Delete unnecessary keyword specifications.
TRX	2500	Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the OLDS.
RACFUSER	5000	Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the OLDS.
MODNAMES	2500	Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the OLDS.
TRXUSE	10000	Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the OLDS.
UOWS	100	Use the STRTTIME= keyword, the STOPTIME= keyword, or both to limit the duration that is required to read the OLDS.

If the problem persists, contact IBM Software Support.

**BTSA1012E INVALID VALUE *value* WAS
 SPECIFIED FOR KEYWORD
 *keyword***

Explanation

The value that was specified for the indicated keyword was syntactically incorrect for the keyword.

System action

Processing terminates with return code 8.

User response

Correct the keyword specification and resubmit the job.

**B TSA1013E NO LOG DATA SETS FOUND
MATCHING SELECTION CRITERIA**

Explanation

No entries in RECON match the combination of SSID, STRTTIME, or STOPTIME that were specified in CNTLCRDS.

System action

Processing terminates with return code 8.

User response

Determine what the proper selection criteria should be and resubmit the job. The SYSPRINT output from BTSARCN0 can be used to determine what logs and what subsystems are available.

**B TSA1014E STRTTIME CANNOT BE LATER
THAN STOPTIME**

Explanation

Values were specified for STRTTIME and STOPTIME, but the time specified for STOPTIME is earlier than that for STRTTIME.

System action

Processing terminates with return code 8.

User response

Correct the STRTTIME or STOPTIME keyword values and resubmit the job.

B TSA1015E CALL TO DBRC FAILED

Explanation

DBRC was called from BTSARCN0, but the access did not complete successfully.

System action

Processing terminates with return code 8.

User response

Determine the problem and resubmit the job. The error might be due to the following condition:

The SYSIN input must be the character string 'LIST.LOG'; however, the SYSIN input might be invalid or omitted.

The SYSPRINT output from BTSARCN0 can be used to determine the problem.

**B TSA1016E PREVIOUS CONTROL STATEMENT
CONTAINED NO VALID
KEYWORDS**

Explanation

A control statement was encountered that did not contain any valid keywords.

System action

Processing terminates with return code 8.

User response

Correct or remove the control statement and resubmit the job.

**B TSA1017E TIMESTAMP CONTROL
STATEMENTS MUST INCLUDE
CENTURY**

Explanation

The STRTTIME or STOPTIME keywords contained values that did not begin with the century portion of a date.

System action

Processing terminates with return code 8.

User response

Correct the control statement and resubmit the job. See [“BTSARCN0 JCL requirements”](#) on page 165 for the correct syntax for values for these keywords.

**B TSA1018I FEWER LOGS MATCHED
SELECTION CRITERIA THAN
WERE REQUESTED**

Explanation

The LOGCOUNT= parameter specified a number of logs to be selected; however, few log data sets matched time-related selection criteria.

System action

Processing continues.

User response

None. This message is informational.

BTSA1019E NUMERIC VALUE IS TOO LARGE

Explanation

A numeric parameter was encountered in which the associated value exceeded the allowable size.

System action

Processing terminates with return code 8.

User response

Correct the value associated with the keyword and resubmit the job.

BTSA1020E IMS RESLIB IS NOT CONCATENATED IN STEPLIB

Explanation

DBRC module DSPURX00 could not be found in any of the data sets concatenated as STEPLIB. This module resides in the IMS load module library.

System action

Processing terminates with return code 8.

User response

Add the IMS load module library to the STEPLIB DD concatenation and resubmit the job.

BTSA1021E INVALID STATUS: *sc* RETURNED ON *type* CALL.

Explanation

An unexpected IMS status code was returned during a DL/I call.

where:

- *sc* represents the status code returned from IMS
- *type* is one of the following call types:

- XRST
- CHKP
- ISRT
- PURG

This message is followed by message BTSA1009E, which might identify the TRXIN record in process when the abend occurred.

System action

Processing terminates with abend code 3501.

User response

Provide appropriate action in response to the status codes as described in IMS manuals and resubmit the job.

BTSA1022W THE SAME KEYWORD IS SPECIFIED IN 1 RECORDS: *keyword*

Explanation

The same keyword is specified in CNTLCRDS. You cannot specify this keyword multiple times in one CNTLCRDS.

System action

Processing continues. If you specified the same keyword more than once in a line, the first value is used. If the same keyword is specified more than once in multiple lines, the last value is used.

User response

Correct the indicated keyword.

BTSA1023W THE SAME VALUE IS SPECIFIED IN [IGNORETERM AND TERMID | IGNOREMOD AND MODNAME]. [TERMID | MODNAME]: *value*

Explanation

The same value is specified for IGNORETERM and TERMID or IGNOREMOD and MODNAME.

System action

Processing continues. The value of TERMID or MODNAME is ignored.

User response

Correct the *termid* or *modname*.

BTSA1024E **INVALID KEYWORD IS SPECIFIED: *keyword***

Explanation

Either the keyword specified in CNTLCRDS or the PAGECNTL DD statement is incorrect.

System action

Processing ends with a return code 8.

User response

Correct the indicated keyword and resubmit the job.

BTSA1025I **LAST 5-DIGITS OF THE STRTTIME OR THE STOPTIME KEYWORD IS IGNORED.**

Explanation

The value of the STRTTIME or the STOPTIME keyword is a 19-digit number. Because the time precision in log records is a 14-digit number, the last five digits of the STRTTIME or the STOPTIME keywords are ignored.

System action

Processing continues.

User response

None. This message is informational.

BTSA1026E **THE STRTTIME OR THE STOPTIME DIGIT IS INCORRECT.**

Explanation

The value of the STRTTIME or STOPTIME keyword is specified incorrectly. Only 14-digit numbers and 19-digit numbers are allowed for the time precision.

System action

Processing ends with return code 8.

User response

Specify a correct digit number on the STRTTIME or the STOPTIME keyword.

BTSA1027I **THE STRTTIME OR THE STOPTIME DIGIT IS NOT SAME WITH TIMESTAMP IN LIST.LOG.**

Explanation

The specified digit number on the STRTTIME or the STOPTIME keyword is not the same as the time stamp in LIST.LOG. IMS Batch Terminal Simulator adjusts the STRTTIME or the STOPTIME value to the digit number of the time stamp in LIST.LOG and continues processing.

System action

Processing continues.

User response

None. This message is informational.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive

Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of IMS Batch Terminal Simulator.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IMS Batch Terminal Simulator. This information is identified where it occurs by an introductory statement to a topic or section.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] 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 IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

Special Characters

- ./* command [235](#)
- ./C command
 - examples [192](#)
- ./D command
 - CONVTYPE= [193](#)
 - DDOF= [193](#)
 - EOM= [193](#)
 - EOS= [193](#)
 - examples [193](#)
 - FEAT= [193](#)
 - LIMIT= [193](#)
 - LINE= [193](#)
 - LTERM= [193](#)
 - LUNAME= [193](#)
 - SIZE= [193](#)
 - SYNCLVL= [193](#)
 - TPNAME= [193](#)
 - TRSOSI= [193](#)
 - TYPE= [193](#)
 - using [72](#), [91](#), [94](#), [241](#)
- ./E command
 - APPLFE= [198](#)
 - CONNECT= [198](#)
 - DT3270= [198](#)
 - ECHKP= [198](#)
 - ENDROLB= [198](#)
 - ERR= [198](#)
 - ETO= [198](#)
 - examples [198](#)
 - FSEND= [198](#)
 - IFIUTF= [198](#)
 - IMSSUF= [198](#)
 - IODATE= [198](#)
 - IOTIME= [198](#)
 - IOUDATE= [198](#)
 - IOUTIME= [198](#)
 - KATAKANA= [198](#)
 - LCI= [198](#)
 - MAXL= [198](#)
 - MLINE= [198](#)
 - NOEDIT= [198](#)
 - PA2= [198](#)
 - PERMLOAD= [198](#)
 - RCFGPN= [198](#)
 - REATTCH= [198](#)
 - SCNDTRX= [198](#)
 - SPACE= [198](#)
 - SPLAPI= [198](#)
 - SSID= [198](#)
 - YSRQ= [198](#)
 - UGN= [198](#)
 - UIDDREG= [198](#)
 - USERID= [198](#)
 - USERIND= [198](#)
- ./O command

./O command (*continued*)

- APS= [205](#)
- ATR= [205](#)
- DB= [205](#)
- EATR= [205](#)
- ELAPTIME= [205](#)
- examples [205](#)
- ICALOLEN= [205](#)
- IFI= [205](#)
- MQI= [205](#)
- MSG= [205](#)
- REPORT= [205](#)
- SCREEN= [205](#), [212](#)
- SPA= [205](#)
- SQL= [205](#)
- SQLHX= [205](#)
- SQLOBLN= [205](#)
- TSO= [205](#)
- TSO Aid= [205](#)
- TSODB= [205](#)
- TSOIFI= [205](#)
- TSOMLVL= [91](#), [205](#)
- TSOMQI= [205](#)
- TSOMSG= [205](#)
- TSOSQL= [205](#)
- TSOSQLHX= [205](#)
- TSSLEN= [205](#)

./P command

- CA= [212](#)
- examples [212](#)
- MBR= [212](#)
- PA= [212](#)
- PC= [212](#)
- PX= [212](#)

./R command

- examples [217](#)
- IP= [217](#)

./RETURN command [97](#), [218](#)

./S command

- BYTES= [218](#)
- CA= [218](#)
- examples [218](#)
- FUNC= [218](#)
- MBR= [218](#)
- PCB= [218](#)
- SA= [218](#)
- STCD= [218](#)
- TIMES= [218](#)

./SPA command

- examples [220](#)

./T command

- BCHKP= [222](#)
- CONVTYPE= [222](#)
- DLITRACE= [222](#)
- EM= [222](#)
- examples [222](#)
- FEAT= [222](#)

./T command (continued)

LANG= [222](#)
LUNAME= [222](#)
MBR= [212](#), [218](#), [222](#)
MDL= [222](#)
MODE= [222](#)
MQITRACE= [222](#)
MSGTYPE= [222](#)
PLAN= [222](#)
PLC= [222](#)
PSB= [222](#)
QMGRNAME= [222](#)
QNAME= [222](#)
REATTCH= [222](#)
SIDE= [222](#)
SIZE= [222](#)
SPA= [222](#)
SQLTRACE= [222](#)
SYNCLVL= [222](#)
TC= [222](#)
TERM= [222](#)
TPNAME= [222](#)
TYPE= [27](#), [222](#)
using [118](#)

/EXIT command [5](#), [120](#), [222](#), [246](#), [247](#)

/FORMAT command [5](#), [33](#), [247](#)

/RESET command [5](#), [248](#)

/SET command [5](#), [248](#)

&IOBLK [45](#)

&IOSEG [45](#)

Numerics

31 bit mode [59](#)

3270

extended attributes

examples [212](#)

field attribute character [36](#)

formatted-mode simulator statements

format of [241](#)

formatting

description of [33](#)

output listing [33](#)

terminal operator actions [242](#)

3278 Model 5 [60](#)

A

ABEND [235](#)

abend codes

playback utility [302](#)

accessibility [17](#)

alarm [35](#)

ALLOC [88](#)

ALLOCD5 [88](#)

alternate

logical terminal [72](#), [91](#), [94](#)

terminals, with FSS [102](#)

ALTSEG [45](#)

AMODE=31 [59](#)

APPC applications [122](#)

APPLCTN macro [22](#)

APPLFE= [198](#)

Application Interface Block interface [27](#)

applications

3270 [121](#)

APPC [122](#)

Db2 [47](#)

DEDB [118](#)

events, tracing [35](#)

IBM MQ [48](#)

Java [47](#), [113](#)

REXX [47](#), [113](#)

sync point [27](#)

APS= [205](#)

ASM application [222](#)

assembler language [14](#)

ATR= [205](#)

ATTENTION EXIT routine [128](#)

attribute, field [36](#), [245](#)

audible alarm [35](#)

AUTH call [28](#)

B

backout [59](#), [60](#)

batch

applications [74](#), [240](#)

execution [22](#), [87](#)

batch mode

tutorial [65](#)

BCHKP= [222](#)

benefits [7](#)

BMP

application [222](#)

region [22](#), [87](#)

BTS0004W message [163](#)

BTSAEDT0 JCL [176](#)

BTSAISR0 JCL [174](#)

BTSALOGO JCL [168](#)

BTSAPRE0

JCL [171](#)

sort dependency [158](#)

BTSARCN0 JCL [165](#)

BTSCLIST [46](#)

BTSCP [87](#), [100](#), [192](#)

BTSCSI00 [50](#)

BTSDEBUG data set

batch mode [69](#)

interactive mode [88](#)

BTSDEFDF

macro statements [125](#)

BTSDFTL [125](#)

BTSIN data set

batch mode [69](#)

interactive mode [88](#)

BTSISRIO data set

batch mode [69](#)

interactive mode [88](#)

BTSOUT data set

batch mode [69](#)

call trace [205](#)

interactive mode [88](#)

lines per page [212](#)

screen image [36](#), [205](#), [212](#)

BTSPC000 [20](#)

BTSPCM00 [20](#)

BTSPIXTO [49](#), [251](#)
BTSPUNCH data set
 interactive mode [88](#)
BTSRA [111](#)
BTSRC000 [20](#)
BTSSAILO [128](#)
BTSSAIL1 [128](#)
BTSSAIL2 [128](#)
BTSSAMP1 [65](#)
BTSSAMP3 [79](#)
BTSSNAP data set
 batch mode [69](#)
 interactive mode [88](#)
BTSTSOAT [128](#)
BTSTSOST [20](#)
BTSTSSEO [130](#)
bypass, MFS [49](#)
BYTES= [218](#)

C

C applications [14](#)
C++ applications [14](#)
CA= [212](#), [218](#)
call trace [205](#)
call trace listing
 Db2 [144](#)
 DLI [142](#)
 MQI [148](#)
call trace, prompting [96](#), [205](#)
callable services [50](#)
CALLTYPE(BTSCP) [87](#)
CALLTYPE(CALL) [87](#)
CALLTYPE(TEST) [87](#)
card, operator identification [242](#)
cataloged procedure
 BTS [45](#)
 BTSBMP [45](#)
 BTSJBP [45](#)
CEEOPPTS [76](#), [98](#)
CHECKPOINT ID [60](#)
CHKP call [27–29](#)
CHNG call [29](#)
CLEAR [96](#), [242](#)
CLIST
 description of [87](#)
 statement parameters [87](#)
CMD call [28–30](#)
COBOL
 application [5](#), [14](#), [77](#), [87](#), [100](#), [222](#)
COBOL (./C) [192](#)
COBOL interactive debug [5](#), [77](#), [87](#), [100](#), [192](#), [212](#)
COBTEST [87](#), [100](#), [192](#)
color [205](#)
commands
 ./C [192](#)
 ./D [193](#)
 ./E [198](#)
 ./O [205](#)
 ./P [212](#)
 ./R [217](#)
 ./RETURN [97](#), [218](#)
 ./S [218](#)
 ./SPA [220](#)

commands (*continued*)
 ./T [222](#)
 debug
 ABEND [235](#)
 DEBUG [235](#)
 DEBUGOFF [235](#)
 MSGABEND [235](#)
 TPBUF [235](#)
 TPBUFOFF [235](#)
 rules for coding [191](#)
 summary of [187](#)
 under TSO [102](#)
comment (./*) [235](#)
configuration checklist [43](#)
configuration, minimum system [41](#)
configuring [43](#)
CONNECT= [198](#)
CONNECTION_NAME [116](#)
continuation [191](#), [241](#)
control statement keywords
 BTS AEDT0 [176](#)
 BTS AISR0 [174](#)
 BTS AL0G0 [168](#)
 BTS APRE0 [171](#)
 BTS ARCN0 [165](#)
 syntax rules [177](#)
conversational processing [120](#), [220](#), [222](#)
CONVTYPE= [193](#), [222](#)
cookie policy [309](#)
CRC [116](#)
CROSSREF file record, Playback [253](#)
CTLBLKS [22](#)
cursor location [33](#)

D

data sets
 batch mode
 BTSDEBUG [69](#)
 BTSIN [69](#)
 BTSOUT [69](#)
 BTSSNAP [69](#)
 BTSOUT
 call trace [205](#)
 Debug Tool [98](#)
 interactive mode
 BTSDEBUG [88](#)
 BTSIN [88](#)
 BTSOUT [88](#)
 BTSPUNCH [88](#)
 BTSSNAP [88](#)
 QALTPCB [45](#)
 QALTRAN [45](#)
 QIOPCB [45](#)
data tags, modified [35](#)
DATABASE macro [22](#)
database recovery [60](#)
Database Recovery Control (DBRC) [60](#)
DB= [205](#)
Db2
 application [5](#)
 applications [47](#), [114](#)
 authorization ID [144](#)
 plan name [222](#)

- Db2 (*continued*)
 - SQL call trace [144](#)
 - SSID [212](#)
- Db2 DL/I Batch Support [5](#), [114](#), [144](#)
- Db2 SQL call trace [212](#)
- DBB region [22](#), [68](#), [87](#)
- DBCS (Kanji) [246](#)
- DBD [22](#)
- DBRC [60](#)
- DD statements
 - batch mode [69](#)
 - Debug Tool [76](#)
 - IMS [70](#)
- DDITV02 [114](#)
- DDOF= [193](#)
- DDOTV02 [114](#)
- debug
 - batch mode [75](#)
 - interactive mode [98](#)
- DEBUG [235](#)
- Debug Tool
 - batch mode [76](#)
 - CODE/370 [192](#)
 - customize [46](#)
 - interactive mode [98](#)
- Debug Tool for z/OS [5](#)
- DEBUGOFF [235](#)
- DEDB [118](#)
- default values
 - simulator commands [187](#)
- deferred detectable field [242](#)
- defining
 - contents of SPA [220](#)
 - device to be simulated [193](#)
 - patch [212](#)
 - primary transactions [222](#)
 - secondary transaction [118](#), [222](#)
 - snap [218](#)
 - terminals [94](#)
 - transactions [94](#), [222](#)
 - transactions and terminals
 - batch mode [72](#)
 - interactive mode [91](#)
- DEFMOD= [175](#)
- DEL (Delete Character) [5](#)
- DELETE [88](#)
- DEQ call [28](#)
- device
 - feature indicator values [193](#), [222](#), [249](#)
 - formatting
 - facility [193](#)
 - support [14](#)
- Device (./D) [193](#)
- DFS.EDT [49](#)
- DFS.EDTN [49](#)
- DFSCMTU0 [49](#)
- DFSCSI00 [50](#)
- DFSCSMB0 [50](#)
- DFSESL [114](#)
- DFSMExxx [49](#)
- DFSPIXT0 [49](#)
- DFSRRCO0 [20](#), [22](#)
- DFSVSAMP [70](#), [90](#)
- disability [17](#)

- Distributed Access Infrastructure
 - configuration [53](#)
 - TLS and SSL [53](#)
- DLI
 - application [222](#)
 - commands [27](#), [198](#), [212](#)
 - commands limitations [27](#)
 - region [22](#), [68](#), [87](#)
 - synchronization [27](#)
- DLITRACE= [222](#)
- documentation
 - accessing [16](#)
 - sending feedback [16](#)
- documentation changes [3](#)
- DT3270= [198](#)
- DUP (Duplicate) [5](#)

E

- EATR= [205](#)
- ECHKP= [198](#)
- Eclipse interface
 - architecture [9](#)
 - installing [54](#)
 - launch configuration [107](#)
 - migrating [54](#)
 - plugins [54](#)
 - server runtime environment [105](#)
 - testing an application [108](#)
 - tutorial [105](#)
- edit module [222](#)
- edit routine, MFS [49](#)
- EGCS (Extended Graphic Character Set) [14](#), [37](#), [205](#), [242](#)
- ELAPTIME= [205](#)
- EM= [222](#)
- END [96](#)
- end of job (EOJ) [60](#)
- end-of-message indicator (eom) [193](#), [240](#), [241](#)
- end-of-segment indicator (eos) [193](#), [240](#), [241](#)
- ENDROLB= [198](#)
- ENTER [96](#), [242](#)
- Environment (./E) [198](#)
- environment parameters (./E command) [51](#)
- eom (end-of-message) [241](#)
- EOM= [193](#)
- EOS= [193](#)
- ERASE (ERASIN) [5](#), [242](#)
- ERASE INPUT key [96](#)
- ERR= [198](#)
- ESMT [116](#)
- ETO= [198](#)
- execution
 - batch mode [68](#)
 - interactive mode [86](#)
- exit routines
 - application front-end routines [130](#)
 - attention exit routine [128](#)
 - BTSEEXIT [131](#)
 - output formatting [125](#)
 - SAIL/ESA [128](#)
 - TSS call [130](#)
- Exporting configurations
 - resource adapter configuration file [109](#)
 - server runtime [109](#)

extended attributes, 3270 [139](#), [205](#), [212](#)
Extended Graphic Character Set (EGCS) [14](#), [37](#), [205](#), [242](#)
Extract file record, Playback [253](#)

F

Fast Path [22](#), [59](#)
FEAT= [193](#), [222](#)
feature code [222](#), [249](#)
features [7](#)
field attribute [36](#), [245](#)
field edit routine, MFS [49](#)
format control blocks [5](#)
formatted mode [33](#)
formatting modes [33](#)
FREE [88](#)
FSS [35](#)
FSEND= [198](#)
full screen image support (FSS)
 call trace [96](#)
 flow [36](#)
 limitations [37](#)
 option [205](#)
 overview [35](#)
 simulating terminal actions [96](#)
 terminal session example [79](#)
FUNC= [218](#)
function codes [27](#)

G

GCMD call [28–30](#)
GDDM [49](#)
GN call [29](#)
GO subcommand [100](#)
Graphic Data Display Manager (GDDM) [49](#)
GSCD call [28](#), [30](#)
GU call [29](#)

H

highlighting [205](#)

I

I/O PCB
 date-time
 examples [212](#)
 output formatting [125](#)
 terminal [72](#), [91](#), [94](#), [193](#)
 USERID
 examples [212](#)
IBM Debug Tool for z/OS [26](#), [46](#)
IBM Developer for z Systems [5](#)
IBM MQ
 applications [116](#)
 MQI Call Trace [116](#)
 trigger message [116](#)
IBM MQ applications [48](#)
IBM z/OS Debugger [5](#), [26](#), [46](#)
ICALOLEN= [205](#)
IDCARD [5](#), [242](#)
identification card reader (IDCARD) [5](#)

IEFRDER [59](#), [70](#), [90](#)
IFI SQL statement information [144](#), [222](#)
IFI= [205](#)
IFIUTF= [198](#)
IGNOREMOD= [172](#)
IGNORETERM=
 BTSALOGO [169](#)
 BTSAPREO [172](#)
IGNORETYPE= [173](#)
immediate detectable field [242](#)
immediate pen detectable field [37](#)
IMS
 Batch Backout [59](#), [60](#)
 call interception [20](#)
 logging [59](#), [60](#)
 Message Format Service [35](#)
 region controller
 examples [212](#)
 region types [22](#)
IMS callable services [50](#)
IMS calls
 function codes [27](#)
IMS commands
 /EXIT [247](#)
 /FORMAT [247](#)
 /RESET [248](#)
 /SET [248](#)
IMS Message Format Service [121](#)
IMSLEVEL= [169](#)
IMSSUF= [198](#)
INIT call [28](#)
input
 description of [9](#)
 edit [222](#)
 lowercase
 examples [212](#)
 terminal [9](#)
input and output [9](#)
input edit, MFS [49](#)
input stream
 example [118](#)
INQY call [28–30](#)
INS MODE (Insert Mode) [5](#)
installing IMS Batch Terminal Simulator [41](#)
interactive mode
 tutorial [79](#)
introduction [3](#)
IOBF [70](#), [90](#)
IODATE= [198](#)
IOTIME= [198](#)
IOUDATE= [198](#)
IOUTIME= [198](#)
IP= [217](#)
ISRT call [29](#)

J

Java applications [14](#), [47](#), [113](#)
JCL statements [68](#)

K

Kanji [246](#)

Katakana [212](#)
KATAKANA= [198](#)
keyboard shortcuts [17](#)
keywords
 list of [187](#)
KW
 BMP [87](#)
 DBB [87](#)
 DLI [87](#)
 JBP [87](#)

L

L CALL [96](#)
L IOAR [96](#)
LANG= [222](#)
LCI= [198](#)
LCLTIME= [170](#)
legal notices
 cookie policy [309](#)
 notices [309](#)
 programming interface information [309](#)
 trademarks [309](#)
limit count option
 coding [193](#)
LIMIT= [193](#)
limitations
 Playback utility [158](#)
limitations for APPC support function [122](#)
limitations, FSS [37](#)
LINE= [193](#)
LINKLIST [114](#)
LIT [116](#)
load modules [20](#)
log data set [59](#), [70](#), [90](#)
LOGCOUNT= [166](#)
LOGTYPE= [166](#)
lowercase [212](#)
LTERM= [193](#)
LUNAME= [193](#), [222](#)

M

machine requirements [41](#)
macro statements, BTSDEFDF [125](#)
main storage requirements [60](#)
MAXL= [198](#)
MAXLOGS= [166](#)
MBR= [212](#), [218](#), [222](#)
MDL= [222](#)
message input descriptor (MID) [33](#)
message output descriptor (MOD) [33](#)
messages
 playback utility [303](#)
messages and codes [263](#)
messages, BTSRA [291](#)
MFS
 operator logical paging [246](#)
 Segment Edit Routine [49](#)
MFS bypass [49](#)
MFS edit routine [49](#)
MID/MOD [35](#)
minimum system configuration [41](#)

mixed mode [37](#), [246](#)
MLINE= [198](#)
MODE= [222](#)
modified
 data tags [35](#)
 fields [35](#)
MODNAME= [173](#)
MQI call trace [116](#)
MQI= [205](#)
MQITRACE= [222](#)
MSG application [222](#)
MSG= [205](#)
MSGABEND [235](#)
MSGLIST [170](#)
MSGLIST file record, Playback [255](#)
MSGTYPE= [222](#)
MVS system integrity [60](#)

N

NOEDIT= [198](#)
NORECON [167](#)
notices [309](#)

O

online
 dependent region [22](#), [87](#)
 execution [22](#), [87](#)
operator identification card reader [241](#)
operator logical paging [246](#)
output
 formatting [125](#)
Output (./O) [205](#)
output listing
 3270 formatting [33](#)
 screen image [36](#)
overview [5](#)

P

PA= [212](#)
PA1, use of [35](#), [96](#), [244](#)
PA2, use of [244](#)
PA2= [198](#)
PA3, use of [244](#)
PAGECNTL control statements [171](#)
Patch (./P) [212](#)
PAX
 command [241](#), [246](#)
PC= [212](#)
PCB fields, cleared
 examples [212](#)
PCB= [218](#)
PEN [5](#), [242](#)
pen detectable field [37](#)
performance [22](#), [205](#)
PERMLOAD= [198](#)
PFK1–PFK24, use of [244](#)
PFK12, use of [244](#)
PFnn keys [96](#), [244](#)
PL/I applications [14](#), [222](#)
PLAN [116](#)

PLAN= [222](#)
Playback
 BTS0004W error [163](#)
 input [165](#)
 limitations [158](#)
 messages [303](#)
 modules [156](#)
 overview [155](#)
 process flow [156](#)
 reports [178](#)
 run [161](#)
 syntax rules [177](#)
PLC= [222](#)
plugins
 Eclipse interface [54](#)
primary transactions [118](#)
process flow
 Playback [156](#)
process limit count (PLC) [222](#)
PROCESS=
 B TSAEDT0 [177](#)
 B TSAISR0 [175](#)
 B TSAPRE0 [173](#)
PROG [116](#)
Program analysis reports [149](#)
program controller [20](#)
programmed symbols [14](#), [205](#), [242](#)
programming interface information [309](#)
prompting, call trace [96](#)
PS/55 [14](#), [37](#), [212](#), [246](#)
PSB [22](#)
PSB scheduling
 examples [212](#)
PSB= [222](#)
PSBGEN [27](#)
PURG call [29](#)
PX= [212](#)

Q

QALTPCB data set [45](#)
QALTRAN data set [45](#)
QIOPCB data set [45](#)
QMGRNAME= [222](#)
QNAME= [222](#)
queues [60](#)

R

RACF group name [212](#)
Rational Developer for System z [5](#)
RCFGPN= [198](#)
Reader (./R) [217](#)
reader comment form [16](#)
REATTCH= [198](#), [222](#)
RECON query summary report
 BTSARCNO [178](#)
recovery, database [60](#)
region controller [20](#)
REO [116](#)
REPORT= [205](#)
requirements
 machine [41](#)

requirements (*continued*)
 main storage [60](#)
 virtual storage [60](#)
resource adapter
 architecture [9](#)
 configuration file [109](#)
 installing [54](#)
 J2C connection factory [55](#)
 supported functions [111](#)
 testing applications [111](#)
resource information [52](#)
restrictions [59](#)
return codes
 Playback utility [302](#)
REXX applications [14](#), [47](#), [113](#)
RLSE, BTSOUT SPACE [45](#)
ROLB call [28](#)
ROLL call [28](#)
ROLS call [28](#)
RTT [116](#)
runtime modes [24](#)

S

SA= [218](#)
SAIL/ESA [128](#)
sample formatted screen
 B TSAEDT0 [183](#)
SCNDTRX=
 scheduling [118](#)
scratchpad area (SPA) [191](#), [220](#)
screen image [36](#)
screen image summary report
 B TSAEDT0 [182](#)
screen readers and magnifiers [17](#)
SCREEN=
 line skip [212](#)
SCS1 [14](#), [222](#)
secondary transactions [19](#), [118](#)
segment edit routine, MFS [49](#)
segment search argument (SSA) [59](#)
selector pen (PEN) [5](#)
service information [16](#)
session example
 FSS from TSO terminal [79](#)
SETO call [28](#)
SETS call [29](#)
SETU call [29](#)
shift in (SI) [246](#)
shift out (SO) [246](#)
SI (shift in) [246](#)
SIDE= [222](#)
simulating terminal actions with FSS [96](#)
simulator commands
 default values and syntax [187](#)
simulator statements
 batch mode [74](#)
 examples [240](#), [241](#)
 formatted mode [241](#)
 logical paging [246](#)
 unformatted mode [240](#)
site default [51](#)
SIZE= [193](#), [222](#)
SKIP=

SKIP= *(continued)*
 BTS AEDT0 [177](#)
 BTS AISR0 [176](#)
 BTS ALOG0 [170](#)
 BTS APRE0 [173](#)
 Snap (./S) [218](#)
 snap dumps [69, 88](#)
 SO (shift out) [246](#)
 SO/SI handling [193](#)
 software requirements [41](#)
 SPA (./SPA) [220](#)
 SPA call trace [205](#)
 SPA= [205, 222](#)
 SPACE, BTSOUT [45](#)
 SPACE= [198](#)
 SPLAPI= [198](#)
 SQL call trace [114, 144, 212](#)
 SQL= [205](#)
 SQLHX= [205](#)
 SQLOBLN= [205](#)
 SQLTRACE=
 overview [144](#)
 SSID= [167, 198](#)
 SSL, *See* Distributed Access Infrastructure
 SSN [116](#)
 statements
 CLIST [87](#)
 JCL [68](#)
 STCD= [218](#)
 STOPTIME=
 BTS ALOG0 [170](#)
 BTS ARCN0 [167](#)
 STRTIME=
 BTS ALOG0 [170](#)
 BTS ARCN0 [167](#)
 summary of changes [3](#)
 summary report
 BTS AEDT0 [182](#)
 BTS AISR0 [181](#)
 BTS ALOG0 [179](#)
 BTS APRE0 [180](#)
 BTS ARCN0 [178](#)
 support information [16](#)
 support of IMS calls [27](#)
 SYNC call [29](#)
 sync point
 application program [27](#)
 SYNCLVL= [193, 222](#)
 syntax diagrams
 how to read [257](#)
 syntax rules
 CNTLCRDS [177](#)
 PAGECNTL [177](#)
 Playback [177](#)
 SYSRQ= [198](#)
 system flow [20](#)

T

TC=
 alternate logical terminal [72, 91, 94](#)
 primary and secondary transactions [118](#)
 technotes [16](#)
 TERM= [222](#)

TERM=DYMN [222](#)
 TERMID= [171](#)
 terminal
 action [241](#)
 alternate logical [72, 91, 94](#)
 defining
 batch mode [72](#)
 interactive mode [91](#)
 displays [79](#)
 I/O PCB [72, 91, 94](#)
 input [9](#)
 terminal action
 keywords [242](#)
 termination of a simulation run [60](#)
 termination prior to the end of job [60](#)
 terminology [19](#)
 TEST REQ (Test Request) [5](#)
 testing
 application example [121](#)
 TIMES= [218](#)
 tips [163](#)
 TLS, *See* Distributed Access Infrastructure
 TMRA support [111](#)
 TPBUF [235](#)
 TPBUFOFF [235](#)
 TPNAME= [193, 222](#)
 trace entry
 CSECT initial entry [235](#)
 data format [235](#)
 trace table [235](#)
 tracing
 application program events [35](#)
 trademarks [309](#)
 transaction (./T) [222](#)
 transaction codes [74, 240, 241](#)
 transaction screen formatting [155](#)
 transactions
 defining
 batch mode [72](#)
 interactive mode [91](#)
 translation table [212](#)
 TRSOSI= [193](#)
 TRX=
 BTS ALOG0 [171](#)
 BTS APRE0 [173](#)
 TRXLINK [173](#)
 TSO
 call trace [205](#)
 COBTEST [100](#)
 command list (CLIST) [87](#)
 Command List (CLIST) [93](#)
 extended attributes
 examples [212](#)
 full screen image support (FSS) [79, 205](#)
 simulator commands [102](#)
 test command, using [100](#)
 TSO CLIST
 run [93](#)
 TSO command list [46](#)
 TSO controller (BTSTSOST) [20](#)
 TSO/E [41](#)
 TSO= [205](#)
 TSOAID= [205](#)
 TSODB= [205](#)

TSOIFI= [205](#)
TSOMLVL= [91](#), [205](#)
TSOMQI= [205](#)
TSOMSG= [205](#)
TSOSQL= [205](#)
TSOSQLHX= [205](#)
TSS call [130](#), [205](#)
TSSLEN= [205](#)
tutorial
 batch mode [65](#)
 interactive mode [79](#)
 setup [52](#)
TYPE= [193](#), [222](#)

U

UGN= [198](#)
UIDDREG= [198](#)
unformatted mode [33](#)
unit group-name [114](#)
UNIT= [167](#)
user-written exit routines [125](#)
USERID, I/O PCB
 examples [212](#)
USERID=
 BTSALOG0 [171](#)
 BTSAPRE0 [174](#)
USERIND= [198](#)

V

virtual storage requirements [60](#)
VOLSER [168](#)

X

XRST call [29](#), [30](#), [198](#), [212](#)



Product Number: 5655-BT4

SC19-3230-04

