Tivoli

Information Management for z/OS

Operation and Maintenance Reference

Version 7.1

SC31-8749-00

Tivoli

Information Management for z/OS

Operation and Maintenance Reference

Version 7.1

SC31-8749-00

Tivoli Information Management for z/OS Operation and Maintenance Reference

Copyright Notice

Trademarks

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Contents

Preface

This book describes and illustrates the BLX-SP commands for use by the operator and the utilities for installing, defining, and maintaining data sets required for using the Tivoli® Information Management for z/OS licensed program.

There may be references in this publication to versions of Tivoli Information Management for z/OS's predecessor products. For example:

- TME 10™ Information/Management Version 1.1
- Information/Management Version 6.3, Version 6.2, Version 6.1
- | Tivoli Service Desk for OS/390® Version 1.2

The version number of Information/Management was reset to Version 1.1 with the TME 10 Information/Management release (1997).

Who Should Read This Guide

This book is intended for:

- System administrators
- Database administrators
- System analysts or programmers, who provide procedures for using Tivoli Information Management for z/OS

You should be familiar with the contents of the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* and the *OS/390 MVS™: System Commands* manuals before using this book.

Prerequisite and Related Documentation

The library for Tivoli Information Management for z/OS Version 7.1 consists of these publications. For a description of each, see "The Tivoli Information Management for z/OS Library" on page 193.

Tivoli Information Management for z/OS Application Program Interface Guide, SC31-8737-00

Tivoli Information Management for z/OS Client Installation and User's Guide, SC31-8738-00

Tivoli Information Management for z/OS Data Reporting User's Guide, SC31-8739-00

Tivoli Information Management for z/OS Desktop User's Guide, SC31-8740-00

Tivoli Information Management for z/OS Diagnosis Guide, GC31-8741-00

Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications, SC31-8744-00

Tivoli Information Management for z/OS Integration Facility Guide, SC31-8745-00

Tivoli Information Management for z/OS Licensed Program Specification, GC31-8746-00

Tivoli Information Management for z/OS Master Index, Glossary, and Bibliography, SC31-8747-00

Tivoli Information Management for z/OS Messages and Codes, GC31-8748-00

Tivoli Information Management for z/OS Operation and Maintenance Reference, SC31-8749-00

Tivoli Information Management for z/OS Panel Modification Facility Guide, SC31-8750-00

Tivoli Information Management for z/OS Planning and Installation Guide and Reference, GC31-8751-00

Tivoli Information Management for z/OS Problem, Change, and Configuration Management, SC31-8752-00

Tivoli Information Management for z/OS Program Administration Guide and Reference, SC31-8753-00

Tivoli Information Management for z/OS Reference Summary, SC31-8754-00

Tivoli Information Management for z/OS Terminal Simulator Guide and Reference, SC31-8755-00

Tivoli Information Management for z/OS User's Guide , SC31-8756-00

Tivoli Information Management for z/OS World Wide Web Interface Guide, SC31-8757-00

Note: Tivoli is in the process of changing product names. Products referenced in this manual may still be available under their old names (for example, TME 10 Enterprise Console instead of Tivoli Enterprise Console®).

What This Guide Contains

This book lists the commands you use in BLX-SP and utilities you use to manage the Tivoli Information Management for z/OS data sets or database.

- Chapters 2 through 9 contain the BLX-SP operator commands, listed in alphabetical order.
- Chapters 10 through 35 contain the Tivoli Information Management for z/OS utilities, listed in alphabetical/numerical order.

This product is enabled for DBCS support. As a result, this book uses the following terms:

- DBCS (double-byte character set)
- SBCS (single-byte character set).
- Mixed data

The term *mixed data* refers to data strings that can contain only DBCS data, only SBCS data, or any combination of DBCS and SBCS data. SBCS data is the same as EBCDIC data. The term *mixed case data* refers to data strings that can contain uppercase, lowercase, or a combination of uppercase and lowercase SBCS data.

Typeface Conventions

This guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold Entries that you must use literally, choices, or options that you select appear in **bold**. The names of titles or screen objects in graphical windows also appear in bold.

Italics Variables and values that you must provide appear in *italics*. New terms also appear in italics.

`Monospace`
Code examples, output, and messages are in `monospace` font.

The syntax symbols used throughout this document are as follows:

```
MODIFY|F procname,COMMANDname
      ,KEYWORD=[data]
```

MODIFY|F
All BLX-SP operator commands begin with the MVS MODIFY command. You can use either the complete word or its abbreviation F.

procname
Indicates the procedure that starts the BLX-SP.

COMMANDname
Uppercase letters designate the minimum truncation of the BLX-SP operator command name.

KEYWORD=
Enter a keyword value with at least the characters shown in uppercase, include the equal sign (=), and replace a value shown in all lowercase with your own data.

brackets []
Indicate optional entries.

braces {}
Specify alternatives; you must choose one.

bar |
Specifies alternatives; you must choose one.

The panels as presented in this book are not meant to be exact replicas of the way a panel might appear on the screen. The information on the panels is correct, but the spacing is not always exact. The panels shown are examples of the panels as shipped. Changes made during installation are not taken into consideration. Therefore, you may notice differences in your panels.

Contacting Customer Support

For support inside the United States, for this or any other Tivoli product, contact Tivoli Customer Support in one of the following ways:

- Send e-mail to **support@tivoli.com**
- Call 1-800-TIVOLI8
- Navigate our Web site at **http://www.support.tivoli.com**

For support outside the United States, refer to your Customer Support Handbook for phone numbers in your country. The Customer Support Handbook is available online at **http://www.support.tivoli.com**.

When you contact Tivoli Customer Support, be prepared to provide identification information for your company so that support personnel can assist you more readily.

| The latest downloads and fixes can be obtained at **http://www.tivoli.com/infoman**.

1

Operation and Maintenance

One of the main duties of a Tivoli Information Management for z/OS system administrator is to ensure the availability and the integrity of the end user's data. Other tasks include expanding your database, and preventing or recovering from the loss of data. Tivoli Information Management for z/OS provides utility programs to maintain, update, and rebuild data sets.

Tivoli Information Management for z/OS system administrators must also know how to use BLX-Service Provider (BLX-SP) operator commands to alter the characteristics of the BLX-SP to the best advantage of their installation. Commands can be entered to control VSAM I/O operations (in a non-sysplex environment) and to perform other tasks such as sending notifications to users.

BLX-SP Operator Commands

The BLX-SP is a task that runs in its own address space. Start the BLX-SP with the MVS* START operator command. Stop the BLX-SP with the MVS STOP operator command.

When you want to change the operating characteristics of the BLX-SP, use the BLX-SP operator commands described in this book. Use the MVS MODIFY operator command to issue each BLX-SP operator command. The MODIFY command is included in the syntax description of the BLX-SP operator commands.

For more information about MVS commands, refer to the *OS/390 MVS: System Commands* manual. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for more information about setting up the BLX-SP.

In a non-sysplex environment, the BLX-SP operator commands perform on only one BLX-SP at a time. If you run a command against a BLX-SP in a sysplex with shared databases, and sysplex mode is enabled, the command can act on one or all BLX-SPs in the sysplex, depending on the command and how the command is run.

Utility Programs

You can use the utility programs to maintain, update, and rebuild data sets. Each utility is described in this book.

If you run multiple BLX-SP subsystems on one z/OS system, each BLX-SP must have a load library that contains a unique BLXSSINM module and it must be concatenated to the STEPLIB DD statement before the main Tivoli Information Management for z/OS load library.

Note: The session member being used must have a value on the CAS keyword of the BLGPARMS macro which matches the value used in the BLXSSINM module.

If you are using a Tivoli Information Management for z/OS utility, then one of the following must be true:

- The load library that contains the BLXSSINM load module for the desired subsystem must be allocated in the STEPLIB for the JCL you use to run the utility.
- The load library that contains the BLXSSINM load module for the desired subsystem must be specified in the link list.
- The BLXSSINM load module for the desired subsystem must be in the link pack area.

2

ADDVDEF Command

Non-Sysplex Environment Only

The information in this chapter does not apply to your environment if sysplex mode is enabled for Tivoli Information Management for z/OS in a parallel sysplex (BLX-SP parameter SYSPLEX=YES). You cannot use the ADDVDEF command when sysplex mode is enabled.

For a BLX-SP to work with a data set, the data set does not require a VSAM resource definition. Tivoli Information Management for z/OS automatically specifies and uses nonshared resource (NSR) processing for data sets. However, you could use the ADDVDEF command to provide temporary additional VSAM resource definitions for a new data set, if desired.

You define the VSAM resource being added by use of the BLXDSN, BLXNSR, and BLXGEN macros and the VSAM BLDVRP macro. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information about these macros and defining VSAM resources.

If you want to add a VSAM resource definition for a new data set, you can connect the data set to an existing local shared resource (LSR) pool that you define in the new definition. When you add a data set to an existing LSR pool, be sure that the LSR pool satisfies the buffer requirements of the data set being added. Data sets defined in the VSAM definition can also be defined to use the NSR. If you do not specifically connect the data set to a new or existing VSAM resource pool, then NSR is used. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information on how to set up the buffer pools.

The VSAM definitions that you add to a BLX-SP using the ADDVDEF command are temporary. The next time you stop the BLX-SP, the definitions are lost unless you run the ADDVDEF command again, or unless you have added the definitions in the ADDVDEF definition module to the VSAM resource definition module. (The sample VSAM resource definition module shipped with Tivoli Information Management for z/OS is BLXVDEF.) When you want a permanent definition, update your VSAM resource definition module to include the new definition. (You use this same VSAM resource definition module to start a BLX-SP.) Otherwise, enter the ADDVDEF command each time you start the BLX-SP. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for more information on the BLX-SP parameters member, BLXPRM, and the VSAMRESOURCES parameter which specifies the name of the VSAM resource definition module that Tivoli Information Management for z/OS uses when the BLX-SP is started.

The number of NSR placeholders specified on the PLACES keyword of the BLXNSR macro is ignored in the added definition. When the value in the added definition differs from the value in the existing definition, the new value is ignored. No warning messages are issued.

Syntax

The syntax for the ADDVDEF command is as follows:

```
MODIFY|F   procname,ADDVDEF
                ,PARM=modname
```

procname
Specifies the name of the started catalog procedure used to start the BLX-SP.

PARM=modname
Specifies the name of the module containing the VSAM resource definitions you are adding.

Use the ADDVDEF command to add the VSAM resources defined in *modname* to the specified BLX-SP.

If you successfully run the ADDVDEF command for a module, do not run the ADDVDEF command for the same module again until you stop and restart the BLX-SP. If you do, any changes you made to the specifications will not be accepted. (You can run the ADDVDEF command for a different module without stopping and restarting the BLX-SP.)

Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information on setting up the VSAM resource definitions referred to here as *modname*.

Adding New Definitions

You can add new VSAM data sets and new VSAM LSR pools during BLX-SP operation. VSAM resource definitions added in this way are temporary and effective only until BLX-SP operation ends.

Enter the ADDVDEF command before the user’s address space allocation of data sets, which are identified in the VSAM resource definition module being added. **Data sets allocated by a user’s address space are dynamically included in the BLX-SP existing resource definition.** Data sets are allocated using NSR if the data set is not defined in an existing definition at the time of user’s address space allocation.

To add new definitions, use the following procedure:

1. Prepare a VSAM resource definition module as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*. This definition must contain only those data sets and resource pools that you are adding to the BLX-SP VSAM resource definition.
2. Assemble and link-edit the VSAM resource definition from step 1 and place the resultant load module in the search path of the active BLX-SP.
3. Enter an ADDVDEF command that identifies the VSAM resource definition member containing the new data set definitions and, optionally, new resource pool definitions.

The ADDVDEF command fails if:

- It duplicates any existing VSAM:
  - Data set name
  - Data set logical name
  - LSR pool ID
- You try to add a data set using SHARE=YES to a BLX-SP that is not set up to share databases.

Use the BLX-SP QUERY command to verify whether a data set is already allocated and thereby avoid using duplicate names. If the data set logical name begins with BLXLxxxx (where xxxx is 0001, 0002, etc.), the data set was dynamically allocated. Otherwise, the logical name comes from the BLDVRP macro specification for the data set in the VSAM resource definition.

Scenario Using the ADDVDEF Command

Use the ADDVDEF command to extend your existing VSAM resource definition. The following scenario determines the:

- Additional resources that an imaginary installation needs
- Steps that are required to define and attach new data sets to the VSAM resource pools

Lance attended a meeting for product users where the demonstration of an order entry application based on the Management application impressed him. He decided to design and install a similar order entry system for his organization. He asked Yvonne to design and install the order entry system.

To proceed with this new application, Lance and Yvonne want a new database so that they do not interfere with their installation’s current applications. They also want a read panel data set and a write panel data set, but they plan to use the common production dictionary. The team defines these data sets with the same attributes as the existing VSAM data sets of their installation.

They use the following AMS LISTCAT commands to ensure the values they define are the same as those chosen by AMS to assign to the data sets:

```
LISTCAT ENTRY('BLM.ORDER.PANELS') ALL /* New appl Panel Data Set  */
LISTCAT ENTRY('BLM.TEAM.PANELS')  ALL /* Write Panel Data Set     */
LISTCAT ENTRY('BLM.ORDER.SDDS')   ALL /* New application SDDS     */
LISTCAT ENTRY('BLM.ORDER.SDIDS')  ALL /* New application SDIDS    */
```

Lance and Yvonne decide to allocate two new resource pools for their SDDS and SDIDS. They use LSR for their read panel data set, but they want to allow the write panel data set to use NSR.

The information extracted with the LISTCAT command provides the values shown in Table 1.

Table 1. LSR Definition Worksheet Example

Data Set Name	Data CI Size	Data CIs	Index CI Size	Index CIs
Read panel	2048	300*	1536	3*
Write panel	2048	10*	1536	3*
Order/Entry SDDS	4096	220*	512	5*

Table 1. LSR Definition Worksheet Example (continued)

Data Set Name	Data CI Size	Data CIs	Index CI Size	Index CIs
Order/Entry SDIDS	1024	160*	1536	3*
Note: *The sizes are estimated because these are newly defined data sets.				

With this information, Lance and Yvonne determine what additional LSR pools and sizes are required. The first LSR pool the team defines is for the SDDS, which is required for the new application. They enter the following data:

```
*
*     RESOURCE POOL NINE (FOR ORDER ENTRY DATABASE SDDS)
*
*        RESOURCE POOL NINE (SDDS DATA BUFFER POOLS)
*
LSRD9  BLDVRP BUFFERS=(4096(50)),        SDDS DATA BUFFERS            X
               KEYLEN=7,                 KEYLENGTH OF SDDS            X
               STRNO=20,                 MAINTAIN 20 POSITIONS        X
               SHRPOOL=9,                IDENTIFY POOL                X
               TYPE=(LSR,DATA),          DATA BUFFERS ONLY            X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
*
*        RESOURCE POOL NINE (SDDS INDEX BUFFER POOLS)
*
LSRI9  BLDVRP BUFFERS=(512(5)),          SDDS INDEX BUFFERS AND       X
               SHRPOOL=9,                POOL IDENTIFIER              X
               TYPE=(LSR,INDEX),         INDEX BUFFERS ONLY           X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
```

The next LSR pool they define is for the SDIDS required for the new application. They enter the following:

```
*
*     RESOURCE POOL TEN (FOR ORDER ENTRY DATABASE SDIDS)
*
*        RESOURCE POOL TEN (SDIDS DATA BUFFER POOLS)
*
LSRD10 BLDVRP BUFFERS=(1024(100)),       SDIDS DATA BUFFERS           X
               KEYLEN=18,                KEYLENGTH OF SDIDS           X
               STRNO=20,                 MAINTAIN 20 POSITIONS        X
               SHRPOOL=10,               IDENTIFY POOL                X
               TYPE=(LSR,DATA),          DATA BUFFERS ONLY            X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
*
*        RESOURCE POOL TEN (SDIDS INDEX BUFFER POOLS)
*
LSRI10 BLDVRP BUFFERS=(1536(11)),        SDIDS INDEX BUFFERS          X
               SHRPOOL=10,               POOL IDENTIFIER              X
               TYPE=(LSR,INDEX),         INDEX BUFFERS ONLY           X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
```

Note: The numbers of buffers defined in these examples is higher than the data set currently needs. This higher number of buffers allows for the anticipated growth of these small example data sets.

Yvonne and Lance decide to connect their read panel data set to the shared resource pool already defined for BLX-SP for panel data sets. They do not define more shared resource pools because they used NSR for their write panel data set.

Order/Entry VSAM Resource Definition Member

Lance and Yvonne decide to define a separate VSAM resource definition member for their application rather than change the installation member. They decide to activate their definition with the ADDVDEF command using the name OEVDEF for their VSAM resource definition module. Also, Lance and Yvonne prefix the logical names assigned by the BLXDSN macros with OE, so that they can use truncated names on the QUERY command. (In this example, OEPREAD is the logical data set name for data set BLM.ORDER.) Lance further qualifies the names by using P and S in the labels as illustrated in the following example.

```
OEVDEF  CSECT
*
* DEFINE PLACEHOLDERS REQUIRED FOR NONSHARED RESOURCES
*
         BLXNSR  PLACES=20
*
* DEFINE VSAM DATA SETS TO BLX-SP AND CONNECT VSAM RESOURCES
*
*  DEFINE DATA SETS USING LOCAL SHARED RESOURCES (LSR)
*
OEPREAD  BLXDSN   DSN=BLM.ORDER.PANELS,                         X
               LSR=(3,3)
OESDDS   BLXDSN   DSN=BLM.ORDER.SDDS,                           X
               LSR=(9,9)
OESDIDS  BLXDSN   DSN=BLM.ORDER.SDIDS,                          X
               LSR=(10,10)



*
*  DEFINE DATA SETS USING NONSHARED RESOURCES (NSR)
*
OEPWRIT  BLXDSN   DSN=BLM.TEAM.PANELS
*
* SPECIFY DEFINITION END
*
*        BLXGEN
*
*
*     RESOURCE POOL NINE (FOR ORDER ENTRY DATABASE SDDS)
*
*        RESOURCE POOL NINE (SDDS DATA BUFFER POOLS)
*
```

```
LSRD9  BLDVRP BUFFERS=(4096(50)),        SDDS DATA BUFFERS            X
              KEYLEN=7,                  KEYLENGTH OF SDDS            X
              STRNO=20,                  MAINTAIN 20 POSITIONS        X
              SHRPOOL=9,                 IDENTIFY POOL                X
              TYPE=(LSR,DATA),           DATA BUFFERS ONLY            X
              RMODE31=ALL,                                            X
              MODE=24,                                                X
              MF=L
*
*        RESOURCE POOL NINE (SDDS INDEX BUFFER POOLS)
*


LSRI9  BLDVRP BUFFERS=(512(5)),          SDDS INDEX BUFFERS AND       X
              SHRPOOL=9,                 POOL IDENTIFIER              X
              TYPE=(LSR,INDEX),          INDEX BUFFERS ONLY           X
              RMODE31=ALL,                                            X
              MODE=24,                                                X
              MF=L
*
*     RESOURCE POOL TEN (FOR ORDER ENTRY DATABASE SDIDS)
*
*        RESOURCE POOL TEN (SDIDS DATA BUFFER POOLS)
*


LSRD10 BLDVRP BUFFERS=(1024(100)),        SDIDS DATA BUFFERS          X
              KEYLEN=18,                 KEYLENGTH OF SDIDS           X
              STRNO=20,                  MAINTAIN 20 POSITIONS        X
              SHRPOOL=10,                IDENTIFY POOL                X
              TYPE=(LSR,DATA),           DATA BUFFERS ONLY            X
              RMODE31=ALL,                                            X
              MODE=24,                                                X
              MF=L
*
*        RESOURCE POOL TEN (SDIDS INDEX BUFFER POOLS)
*


LSRI10 BLDVRP BUFFERS=(1536(11)),        SDIDS INDEX BUFFERS          X
              SHRPOOL=10,                POOL IDENTIFIER              X
              TYPE=(LSR,INDEX),          INDEX BUFFERS ONLY           X
              RMODE31=ALL,                                            X
              MODE=24,                                                X
              MF=L
          END
```

When they add this definition using the ADDVDEF command

```
F BLX1PROC,ADDVDEF,PARM=OEVDEF
```

they use all of the provided functions of this command. Their efforts produce the following results:

- The write panel data set is added to the BLX-SP definition and given the logical name they specified.
- The read panel data set is added to the BLX-SP definition, given the specified name, and connected to an existing resource pool in the BLX-SP definition (LSR pool 3).
- The SDDS and SDIDS data sets are added to the BLX-SP definition, given the specified names, and connected to newly defined resource pools for the BLX-SP definition (LSR pools 9 and 10).

- LSR pools 9 and 10 are defined as new resource pools added to the existing BLX-SP VSAM resource definition.

3

BRDCST Command

The BRDCST command enables the operator to broadcast a message to all users connected to the BLX-SP or only those users of a particular file or data set.

Syntax

The syntax for the BRDCST command is as follows:

```
MODIFY|F   procname,BRDCST
           ,DATA={([MSG,]text)|(REFRESH[,dsname])}
           [,{FILE=abbbbbbb | DATASET=xxxx   | TABLES}]
           [,SYSPLEX]
```

procname
Specifies the name of the started catalog procedure that identifies the BLX-SP.

DATA=

MSG Specifies that information is sent to Tivoli Information Management for z/OS users connected to the BLX-SP. The value ″MSG″ is optional. If you specify MSG, it is removed from the message that displays on the user’s terminal. The data is transferred to a user following the user’s next service call to the BLX-SP. If you do not specify FILE or DATASET, the data is transferred to all users connected to the BLX-SP.

Note: If the text of the message that you are sending to users begins with a BRDCST keyword, for example ″Refresh″, then you must use the syntax `DATA=('MSG,Refresh.....')`.

text Specifies the data the Tivoli Information Management for z/OS user is to view. The text can consist of up to 128 characters. The data is transferred to each connected application following its next service call to the BLX-SP. If you do not specify FILE or DATASET, the data is transferred to all users connected to the BLX-SP. When the value specified for data has imbedded blanks, then enclose the character string (including ″MSG″) in single quotation marks. The value DATA cannot exceed 240 characters including blanks and the enclosing quotation marks.

REFRESH
Specifies that the users’ address spaces discard any panels stored in local panel buffers and obtain new copies of the panels from the BLX-SP. Use this keyword after you change panels and copy them to a read panel data set. You can REFRESH the users’ panel buffers so that the next BLX-SP request refreshes the panel buffers on connected applications, so that the users get the updated panels when appropriate.

dsname

Specifies the 1- to 44- character fully qualified data set name of the data set you want to refresh. Only the Tivoli Information Management for z/OS users accessing this panel data set get refreshed buffers. If you omit this parameter, all buffers for all panel data sets are refreshed.

FILE=abbbbbbb

Specifies the logical file name of a data set. The value specified is from 1 to 8 characters in length. The first character must be alphabetic. The remaining characters must be alphanumeric. The broadcast data goes only to users who are using the data set identified by FILE.

DATASET=

Specifies a data set name. The value specified can be 1 to 44 characters in length. The value must be a valid MVS data set name. The broadcast data goes only to users who are using the data set identified by DATASET.

TABLES

Specifies that the PIDTs and data model records currently maintained in storage are to be refreshed, so that users get the updated PIDTs and data model records when appropriate. After using BRDCST to refresh the tables, the updated PIDTs and data model records should be available the next time they are accessed through the HLAPI.

SYSPLEX

Specifies that the message should be broadcasted to all users connected to all BLX-SP servers in the sysplex. The keyword SYSPLEX is optional. If SYSPLEX is not specified, the message is sent only to the users connected to the BLX-SP specified in *procname*. If SYSPLEX is specified but the BLX-SP is not in sysplex mode, an error message is returned.

To check the mode setting of the BLX-SP, refer to the SYSPLEX parameter of the BLX-SP parameters member (BLXPRM). The BLX-SP SYSPLEX parameter will be SYSPLEX=YES or SYSPLEX=NO. BLX-SP parameters members are described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

Note: If the BLX-SP is in sysplex mode, you cannot use the FILE keyword of the BRDCST command. Use the DATASET keyword instead.

Scenario Using the BRDCST Command

Use the BRDCST command to warn all of your Tivoli Information Management for z/OS users to stop using the system, because you are going to reinitialize the system for some reason.

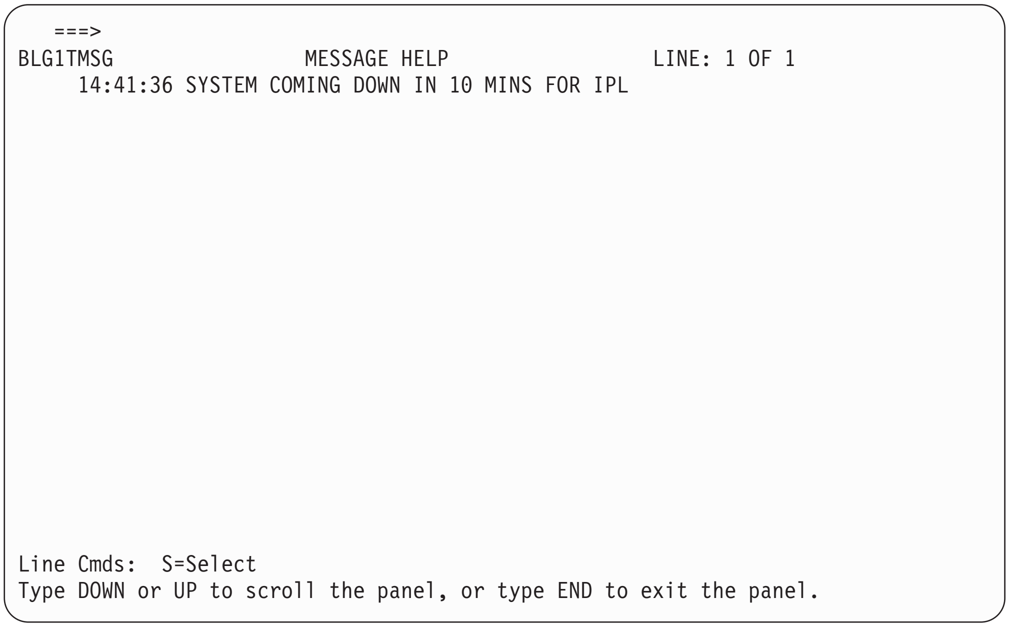
Enter the command on your MVS master console command line.

```
F BLX1PROC,BRDCST,DATA=('SYSTEM COMING DOWN IN 10 MINS FOR IPL')
```

The BLX-SP returns a message to the system operator that the BRDCST command has finished:

```
14.40.47 BLX03250I The BRDCST command completed processing.
```

Your users see the data you entered as a message on panel BLG1TMSG the next time they do something that causes an interaction with BLX-SP.



```
   ===>
BLG1TMSG                MESSAGE HELP                 LINE: 1 OF 1
     14:41:36 SYSTEM COMING DOWN IN 10 MINS FOR IPL
















Line Cmds:  S=Select
Type DOWN or UP to scroll the panel, or type END to exit the panel.
```

Scenario Using the BRDCST Command With MSG Keyword

The following example of the BRDCST command using the MSG keyword produces the same results as the scenario just described.

```
F BLX1PROC,BRDCST,DATA=('MSG, SYSTEM COMING DOWN IN 10 MINS FOR IPL')
```

The resulting message is the same.

Scenario Using the BRDCST Command With REFRESH Keyword

Lance and Yvonne, the team in our ADDVDEF example (see “Scenario Using the ADDVDEF Command” on page 5), decide to change some panels they used in the order entry application. They change the panels in BLM.ORDER.PANELS, and they want to be sure that the updated panels are used by the Tivoli Information Management for z/OS users.

Lance enters the command on his MVS master console.

```
F BLX1PROC,BRDCST,DATA=(REFRESH,BLM.ORDER.PANELS)
```

The BLX-SP returns a message telling him that the BRDCST command is completed for the REFRESH:

```
12.53.42 BLX03250I The BRDCST command completed processing.
```

The Tivoli Information Management for z/OS users now have use of the corrected panels.

4

FREE Command

Use the FREE command to do either of the following:

- In a non-sysplex environment: To physically deallocate a VSAM data set that was allocated by BLX-SP for users' address spaces.
- When sysplex mode is enabled: To block users from accessing a VSAM data set.

Since the operation of the FREE command varies slightly in these settings, a description is provided for each environment.

In a non-sysplex environment (BLX-SP parameter SYSPLEX=NO)

The FREE command physically deallocates a VSAM data set that was allocated by the BLX-SP for user's address spaces. When the FREE command is issued for an open data set, the REALLOC command must be issued to reallocate (or unblock) the data set before any user can access it. Active users of the data set have to use the QUIT command and then reinvoke Tivoli Information Management for z/OS before they can reaccess the data set.

Users who are not actively using the data set do not have to stop and restart Tivoli Information Management for z/OS to access the data set. However, these users cannot access the data set after the REALLOC command is issued until another user (or utility), allocates the data set.

In a non-sysplex environment, the FREE command has three options: NORMAL, QUIESCE, and FORCE. The NORMAL option prohibits additional allocation of the data set by users' address spaces. The QUIESCE option prohibits additional transactions within the data set. The FORCE option provides immediate action (the data set is freed) with no regard to its impact on the users' address spaces or the data integrity of the data set being freed.

After a data set is allocated by a user's address space, the BLX-SP maintains shared allocation of the data set until the BLX-SP operation is stopped. If no users' address spaces have the data set currently allocated, the data set is still left open. If another job requires allocation of the data set, you must use the FREE command to release the data set. You must use the REALLOC command after the job has completed to notify the BLX-SP that the data set is again available for its use.

To run a utility that requires exclusive access against a VSAM data set, use the FREE command on each BLX-SP that accesses the data set, and then run the REALLOC command with the UTIL keyword on each BLX-SP on which you want to run the utilities.

In a sysplex environment (BLX-SP parameter SYSPLEX=YES)

In a sysplex, the BLX-SP functions are somewhat different. The BLX-SP does not handle VSAM I/O processing; rather, it simply keeps track of which users have the

VSAM data sets. The FREE command blocks further transactions with the data set rather than physically deallocates it. When the FREE command is issued for an open data set, the REALLOC command must be issued to unblock the data set before any user can access it. Active users of the data set have to use the QUIT command and then reinvoke Tivoli Information Management for z/OS before they can reaccess the data set.

The NORMAL, QUIESCE, and FORCE options are not available with this command. Although you cannot enter these options, the default processing performed by the FREE command is equivalent to the NORMAL option. The data set is freed and additional users are blocked from performing transactions with it.

After a data set is allocated by a user's address space, the data set remains allocated until the user quits out of Tivoli Information Management for z/OS, at which time it is closed and freed. If another BLX-SP running in non-sysplex mode or a job running a non-Tivoli Information Management for z/OS program (such as IDCAMS REPRO) requires exclusive allocation of the data set, you must use the FREE command to release it and then use the REALLOC command to make it available for use again after that job or BLX-SP has completed.

If you are running a utility that requires exclusive access against a VSAM data set, FREE the data set to keep users from accessing the data set, run the utility, and then REALLOC the data set. (You do not have to run REALLOC against the freed data set before running the utility.) The UTIL keyword of the REALLOC command does not apply when sysplex mode is enabled.

Syntax

The syntax for the FREE command is as follows:

```
MODIFY|F   procname,FREE
           {,FILE=lname|,DATASET=dsname}
           [,OPTION={NORMAL|QUIESCE|FORCE}]
```

procname
Specifies the name of the started catalog procedure that identifies the BLX-SP.

If sysplex mode is enabled, the FREE command request affects all Tivoli Information Management for z/OS users on BLX-SPs running in sysplex mode on each system in a sysplex, regardless of the BLX-SP procname specified.

FILE=lname
Specifies the full logical name of the data set you are freeing. This keyword is used only in a non-sysplex environment. The FILE keyword is not valid if sysplex mode is enabled.

Truncated logical names are not supported by the FREE command. The logical name assigned to the data set by BLX-SP is usually the label specified for the BLXDSN macro in the VSAM resource definition member that initiates the BLX-SP. A name is assigned by the BLX-SP operation for use as the logical name when the label on the BLXDSN macro is not coded or a BLXDSN macro is not specified for the data set. Use the BLX-SP QUERY command to determine which data sets are currently defined, and to determine the logical names assigned to them.

Specify a value for either the FILE keyword or the DATASET keyword parameters, but not both. The FILE keyword parameter is mutually exclusive from the DATASET keyword parameter.

DATASET=dsname

Specifies the 1- to 44-character fully qualified data set name of the data set you are freeing. Use the BLX-SP QUERY command to determine which data sets are currently being accessed and their names.

In a non-sysplex environment, specify a value for either the DATASET keyword or the FILE keyword parameters, but not both. When sysplex mode is enabled, only the DATASET keyword is valid.

OPTION=

Indicates what option is used to free the data set. OPTION is valid only in a non-sysplex environment.

When you omit the OPTION keyword, the default is NORMAL.

NORMAL

Indicates that no additional allocations of the data set by users' address spaces are permitted. The data set is freed after all active users' address spaces have ended their Tivoli Information Management for z/OS sessions. This is the default value.

The default processing performed when sysplex mode is enabled is equivalent to this option.

QUIESCE

Indicates that no additional transactions can be started in the data set. The data set is freed after all current transaction activity is completed.

FORCE

Indicates that the data set is freed immediately with no regard to data integrity. Use this option in *extreme* conditions only. All transactions active within the data set are interrupted. This option can result in partial records being left in the database.

Scenario Using the FREE Command – Non-Sysplex Environment

Lance decides to FREE one of the data sets he created for the order entry application. He gave the BLM.ORDER.PANELS data set the label OEPREAD when he and Yvonne created the application's VSAM resource definition member (see "Scenario Using the ADDVDEF Command" on page 5). He enters the FREE command on the master console, using the FILE parameter:

```
F BLX1PROC,FREE,FILE=OEPREAD
```

The BLX-SP returns a message that the FREE command has completed processing. Because Lance specified no OPTION for the command, the default value NORMAL was used. With this option, the data set is freed only after all active user's address spaces on the BLX-SP terminate their Tivoli Information Management for z/OS sessions. Lance does not see this message until no more users are using the data set.

```
12.53.42 BLX03250I The FREE command completed processing.
```

If Lance is unsure about the label he gave the data set, he can use the FREE command with the DATASET parameter and the fully qualified data set name.

```
F BLX1PROC,FREE,DATASET=BLM.ORDER.PANELS
```

The BLX-SP returns the same message.

```
12.53.42 BLX03250I The FREE command completed processing.
```

Scenario Using the FREE Command – Sysplex Environment

Lance’s IT organization decided to exploit sysplex services. Lance needs to FREE the BLM.ORDER.PANELS data set he created for the order entry application. He enters the FREE command on the master console:

```
F BLX1PROC,FREE,DATASET=BLM.ORDER.PANELS
```

The existing access requests are allowed to complete uninterrupted. Any new attempts by new users to access the data set are immediately rejected, even if the data set has not been freed yet. The data set is freed after all users quit out of Tivoli Information Management for z/OS across the entire sysplex. After all users quit out of Tivoli Information Management for z/OS, the BLX-SP returns a message:

```
11.36.42 BLX03250I The FREE command completed processing.
```

After the data set is freed, any new attempts to access the data set are rejected.

5 MAILQ Command

Use the MAILQ command to change the settings of BLX-SP queues used in the processing of electronic mail notices to users when a record is created or updated. The MAILQ command enables you to:

- Query the BLX-SP mail queues to list the queue names, the warning and maximum limits currently in effect for a queue, and the current count of mail items on a queue.
- Change the warning and maximum limits for a specified queue. Only limits can be changed. Queue names cannot be changed through use of this command.

For more information on e-mail notification, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Syntax

The syntax for the MAILQ command is as follows:

```
MODIFY|F   procname,MAILQ
           {,MODIFYQ=(queuename,warning_limit,maximum_limit)}
           [,QUERY]
```

Only one keyword, MODIFYQ or QUERY, can be specified on the command at a time.

procname
Specifies the name of the started catalog procedure that identifies the BLX-SP that is running.

queuename
Specifies the name of the queue that you want to modify. The queue name can be 1 to 8 alphabetic or numeric characters. The default names are MAILQ1, MAILQ2, and MAILQ3, since up to three queue names can be specified for the BLX-SP. To determine your queue names, use the QUERY keyword or refer to the BLX-SP parameters member in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

warning_limit
Resets the warning limit for the specified mail queue. Valid values can be between 0 (no mail queueing) and 32 767. This limit specifies the number of mail items that can be on the queue before a message is issued to the operator to warn that the limit has been reached. The default is 0.

maximum_limit
Resets the maximum limit for the specified mail queue. Valid values can be between 0

(no mail queueing) and 32 767. This limit specifies the number of mail items that can exist on the queue before the next queued item is discarded (the queue is full). The default is 0.

QUERY
Queries the mail queues and displays a list of the queue names, warning and maximum limits in effect for a queue, and the current count of mail items on a queue.

All three values, queue name, warning limit, and maximum limit, must be specified on the command to modify the limits.

Scenario Using the MAILQ Command

Yvonne decides to modify the queue she set up for e-mail notification of changes to problem records. First, she wanted to query the original settings. She enters:

```
F BLX1PROC,MAILQ,QUERY
```

and views the results on the console:

```
BLX03402I MAIL QUEUE STATISTICS
BLX03403I Queue Name: MAILQ1  Warning: 5  Maximum: 10  Current: 0
BLX03403I Queue Name: MAILQ2  Warning: 5  Maximum: 10  Current: 0
BLX03403I Queue Name: MAILQ3  Warning: 5  Maximum: 10  Current: 0
BLX03250I The MAILQ command completed processing.
```

She originally specified a warning limit of 5 mail items and maximum limit of 10 items. After a week she determined that the limits were set too low. She enters the MAILQ command on the master console to set the new limits to 10 (warning) and 15 (queue is full), using the MODIFYQ parameter:

```
MODIFY BLX1PROC,MAILQ,MODIFYQ=(MAILQ1,10,15)
```

The BLX-SP returns a message that the EMAIL command has completed processing. The modifications you make to the mail queues are only in effect until the BLX-SP is stopped. To make a permanent change, modify the BLX-SP parameters member MAILQ. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for details on using the BLX-SP parameters.

6

QUERY Command

Use the QUERY command to obtain information about data sets accessed by the BLX-SP, or in the case of a sysplex, to get information about the status of data sets in the sysplex.

You can obtain information for the following:

In a non-sysplex environment

- A specified data set
- A group of data sets
- All data sets accessed by the BLX-SP

You can obtain several types of data set information with the QUERY command:

- Data set users
- Data set status
- Data set I/O statistical summary
- VSAM related information

In a sysplex environment

- A specified data set
- All VSAM data sets in the sysplex

You can obtain the following type of data set information with the QUERY command:

- Data set status

When sysplex mode is enabled, only the BLX-SP under which a user accessed a data set has the latest information regarding the status of the data set. Therefore, if you issue the QUERY command for a data set from a BLX-SP that did not previously have the data set accessed, the output from the QUERY command may not reflect the actual status of the data set.

Note that while data set user information is not available via the QUERY command when sysplex mode is enabled, you can use the MVS Display GRS command instead to determine the users of one or more data sets. See “Scenario Using the Query Command – Sysplex Environment” on page 25 for an example of this.

Syntax

The syntax for the QUERY command is as follows:

```
MODIFY|F   procname,QUERY
           [,{FILE=lname|DATASET=dsname}]
           [,USER=userid]
           [,TYPE={USERS|IO|VSAM|STATUS}]
           [,RESET={YES|NO}]
```

procname

Specifies the name of the started catalog procedure that identifies the BLX-SP.

If sysplex mode is enabled, the QUERY command queries the entire sysplex. The command can be issued from any BLX-SP in a sysplex for any Tivoli Information Management for z/OS data set.

FILE=lname

Specifies the 1- to 8-character logical name of the data set about which you want information. The FILE keyword is not valid if sysplex mode is enabled.

By using a 1- to 7-character truncated logical name followed by a period, you can request that a group of data sets be reported. All data sets matching the truncated logical name are reported. For example, FILE=MGMT. causes all data sets with logical names beginning with MGMT to be reported.

Usually, the label specified for the BLXDSN macro in the VSAM resource definition member used to initiate the BLX-SP is the logical name assigned to the data set by the BLX-SP. A name is assigned by BLX-SP for use as the logical name when the label on the BLXDSN macro is not coded or a BLXDSN macro is not specified for the data set. Determine the assigned logical names of all BLX-SP data sets by first using the BLX-SP QUERY command without parameters.

In a non-sysplex environment, specify the FILE keyword parameter or the DATASET keyword parameter, but not both. The FILE keyword parameter is mutually exclusive from the DATASET keyword parameter. In a sysplex, only the DATASET keyword is valid.

DATASET=dsname

Specifies the 1- to 44-character fully qualified data set name of the data set about which you want information. Truncated data set names are not supported.

In a non-sysplex environment, specify the DATASET keyword parameter or the FILE keyword parameter, but not both.

In a sysplex, only the DATASET keyword is valid. If you omit the DATASET keyword, the status of all VSAM data sets in the sysplex is displayed in the output.

USER=userid

Specifies the 1- to 8-character user identifier of the user about whom you want information. The USER keyword is not valid if sysplex mode is enabled.

When you specify TYPE=VSAM or TYPE=USERS, then the USER keyword is ignored.

TYPE=

Indicates the type of information requested about the data set. If sysplex mode is enabled, the TYPE keyword is not valid. If sysplex mode is not enabled, all TYPE values described here are supported.

When you omit the TYPE keyword, STATUS is the default.

USERS

Requests user information about data sets accessed by the BLX-SP.

When you include either the FILE or DATASET keyword, only the users of the specified data set are returned. When you specify a truncated value for the FILE keyword, the group of data sets beginning with the truncated logical name are reported. When you omit both the FILE and DATASET keyword parameters, all users of all BLX-SP-defined data sets are returned. RESET and USER keyword values are ignored if specified. User information reported for a data set includes:

- Data set logical name
- Data set name
- List of users having the data set allocated

IO

Requests a statistical summary of logical VSAM requests for a BLX-SP data set. I/O statistics do not directly correlate to physical I/O, but rather to completed VSAM requests. VSAM might perform several physical I/O requests, or none at all, to perform the requested operation. Specify either the FILE or DATASET keyword to obtain the summary for a particular data set. When you specify a truncated value for the FILE keyword, the group of data sets beginning with the truncated logical name is reported. When you omit both the FILE and DATASET keyword parameters, all data sets are reported.

Use the RESET keyword to clear these statistics after they are reported.

When the USER keyword is not specified, the statistical summary information reported consists of all requests made for the data set. The values reported are the number of:

- Gets
- Puts
- Erases
- Retries
- Logical opens
- Logical closes
- Points
- End requests

When the USER keyword is specified, the statistical summary information reported contains only those values for the identified user. The values reported are the same ones, but only for the identified user.

VSAM

Requests VSAM related information available to the BLX-SP. When you omit both the FILE and DATASET keyword parameters, only statistical information of VSAM resource pools defined to the BLX-SP is returned. When you specify a truncated value for the FILE keyword, data sets beginning with the truncated logical name are reported. RESET and USER keyword values are ignored if specified. Information reported for each VSAM resource pool includes:

- Number of active placeholders
- Placeholder wait count
- Maximum placeholders used

When the data set is physically OPEN and either the FILE or DATASET keyword is also specified, the following additional information is reported:

- High used relative byte address (RBA)
- High allocated RBA
- Control area splits

- Control interval splits
- Number of extents
- Number of strings
- Gets requiring I/O operations
- Gets not requiring I/O operations

STATUS

Requests status information of BLX-SP data sets. When you include either the FILE or DATASET keyword, only the status of the specified data set is returned. When you specify a truncated value for the FILE keyword, then data sets beginning with the truncated logical name are reported. When you include the USER keyword, the status of all data sets that the user allocated is returned. When you omit the FILE, DATASET, and USER keyword parameters, the status of all BLX-SP data sets is returned. The RESET keyword value is ignored if specified.

Status information reported for a data set includes:

- Data set logical name
- Data set name
- Number of user allocations
- Number of active users
- Resource pool ID
- Maximum placeholder wait count
- Maximum placeholders used

Status information reported for a user includes status for only those data sets allocated by the user.

RESET=

Indicates the disposition of the statistical summary information requested about the data sets. The RESET keyword is ignored if you do not specify TYPE=IO also. The RESET keyword is not valid if sysplex mode is enabled.

When you omit the RESET keyword, the default value is NO.

NO

Requests that statistical summary information of BLX-SP data sets not be cleared (zeroed) after it is reported.

YES

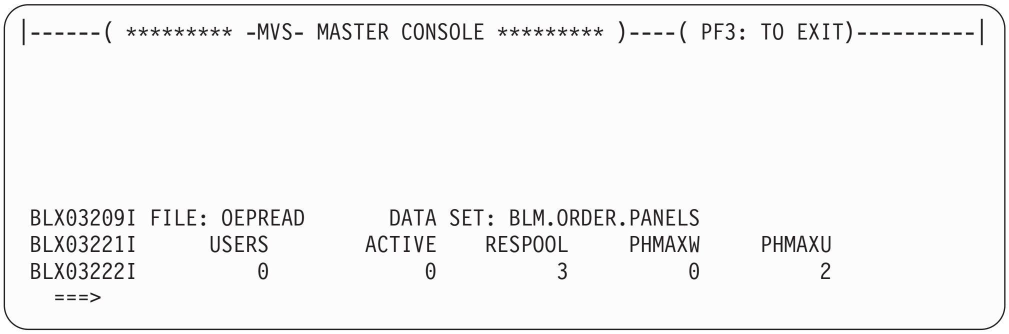
Requests that statistical summary information of BLX-SP data sets be cleared (zeroed) after it is reported.

Scenario Using the QUERY Command – Non-Sysplex Environment

Lance wants to find out the status of a file. Using the logical name of a data set he defined in an earlier example (see “Scenario Using the ADDVDEF Command” on page 5), this QUERY command finds the STATUS of a particular FILE:

```
F BLX1PROC,QUERY,FILE=OEPREAD,TYPE=STATUS
```

The BLX-SP returns messages with the status of the data set he named with the FILE parameter.



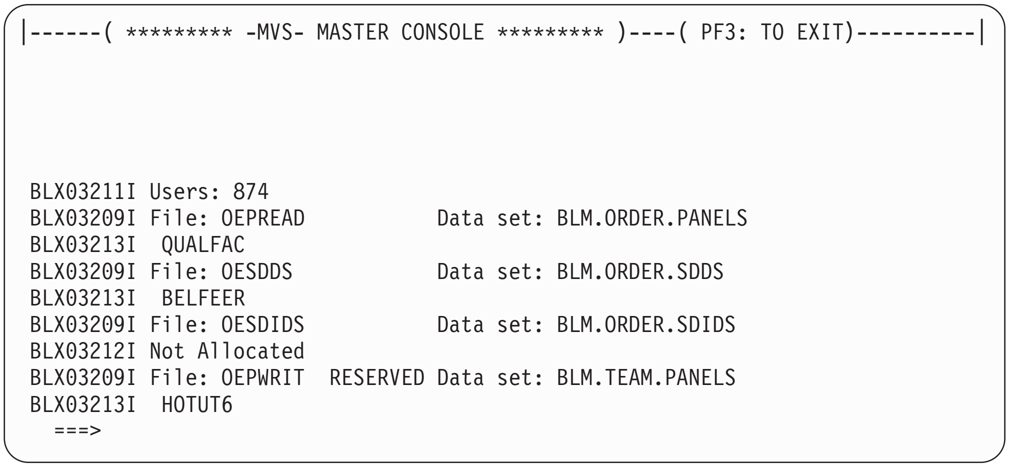
```
|------( ********* -MVS- MASTER CONSOLE ********* )----( PF3: TO EXIT)----------|






BLX03209I FILE: OEPREAD       DATA SET: BLM.ORDER.PANELS
BLX03221I      USERS        ACTIVE    RESPOOL     PHMAXW     PHMAXU
BLX03222I          0             0          3          0          2
  ===>
```

Lance can use the QUERY command with the USERS option on the TYPE parameter to determine which users have a particular data set allocated. As with previous examples, assume that the data sets created in the ADDVDEF example (see "Scenario Using the ADDVDEF Command" on page 5) are the only ones in the database. Lance enters a QUERY command with no FILE or DATASET parameter to see who has each of the data sets in the database allocated:

```
F BLX1PROC,QUERY,TYPE=USERS
```

The BLX-SP returns a list of the data sets with the IDs of the users who allocated them.

Note that the BLM.TEAM.PANELS data set is *reserved* for use by utilities only. The BLX03213I message indicates what job is using the data set. See "REALLOC Command" on page 33 for information about reserving a data set for use by utilities.



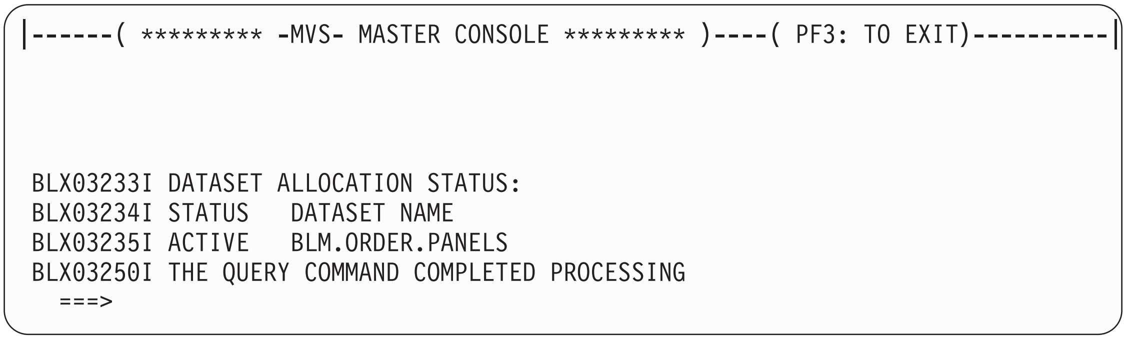
```
|------( ********* -MVS- MASTER CONSOLE ********* )----( PF3: TO EXIT)----------|





BLX03211I Users: 874
BLX03209I File: OEPREAD           Data set: BLM.ORDER.PANELS
BLX03213I  QUALFAC
BLX03209I File: OESDDS            Data set: BLM.ORDER.SDDS
BLX03213I  BELFEER
BLX03209I File: OESDIDS           Data set: BLM.ORDER.SDIDS
BLX03212I Not Allocated
BLX03209I File: OEPWRIT  RESERVED Data set: BLM.TEAM.PANELS
BLX03213I  HOTUT6
  ===>
```

Scenario Using the Query Command – Sysplex Environment

Example 1: Steve wants to find the status of a particular data set in his sysplex. He enters the following command:

```
F BLX1PROC,QUERY,DATASET=BLM.ORDER.PANELS
```

The output generated by the QUERY command is:



```
|------( ********* -MVS- MASTER CONSOLE ********* )----( PF3: TO EXIT)----------|




 BLX03233I DATASET ALLOCATION STATUS:
 BLX03234I STATUS   DATASET NAME
 BLX03235I ACTIVE   BLM.ORDER.PANELS
 BLX03250I THE QUERY COMMAND COMPLETED PROCESSING
   ===>
```

The following output is generated by the QUERY command:

ACTIVE
A status of ACTIVE indicates the data set is available for usage by Tivoli Information Management for z/OS.

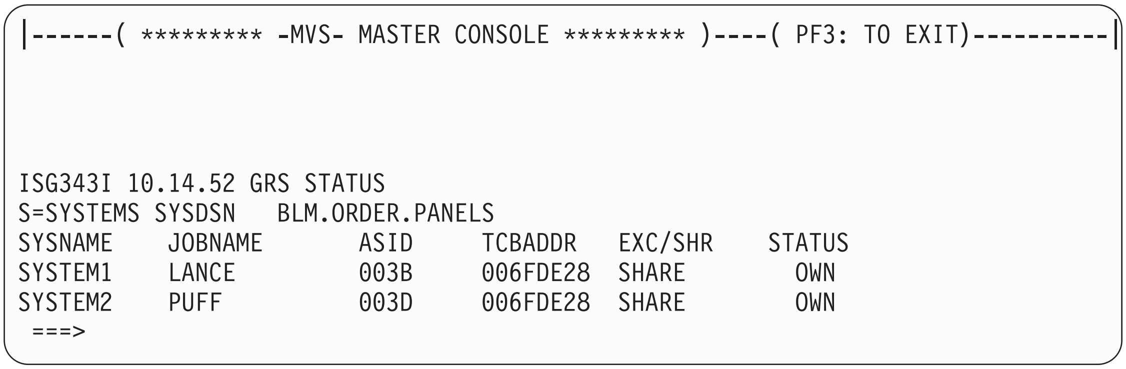
INACTIVE
A status of INACTIVE indicates the data set is unavailable to Tivoli Information Management for z/OS users. The FREE operator command was used to make the data set unavailable.

PENDING
A status of PENDING indicates there is an active FREE request for the data set that has not completed yet. Access to the data set is denied for any new allocation attempts by Tivoli Information Management for z/OS users.

Example 2: Pete wants to find out which users have a particular dataset allocated in his sysplex. Because he has sysplex mode enabled, he can’t do this with the QUERY command, so he enters this command instead:

```
D GRS,RES=(SYSDSN,BLM.ORDER.PANELS)
```

The output generated shows which users/jobs are holding a SYSDSN enqueue against the data set, indicating that they have it allocated:

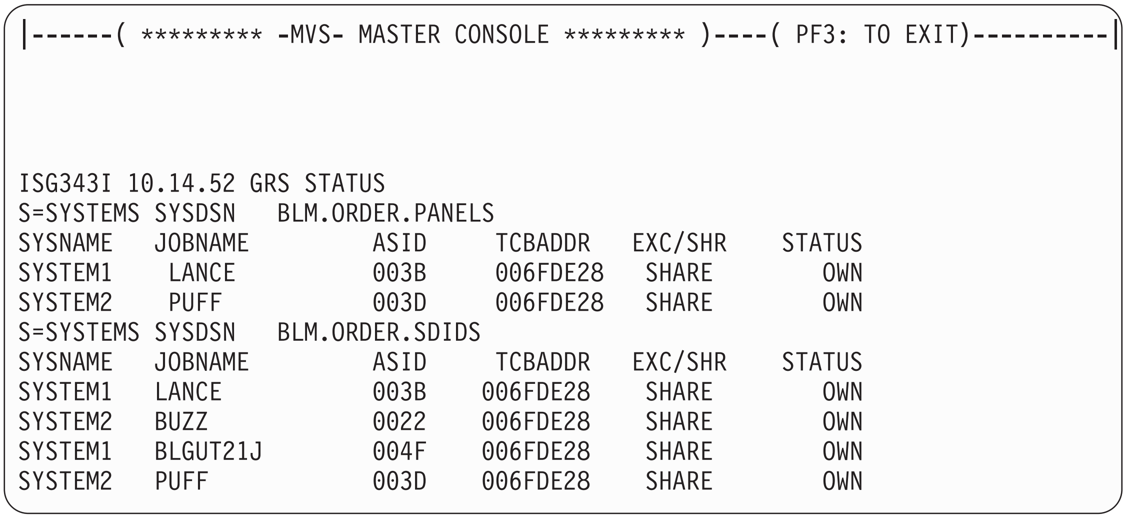


```
|------( ********* -MVS- MASTER CONSOLE ********* )----( PF3: TO EXIT)----------|




ISG343I 10.14.52 GRS STATUS
S=SYSTEMS SYSDSN   BLM.ORDER.PANELS
SYSNAME    JOBNAME       ASID     TCBADDR   EXC/SHR    STATUS
SYSTEM1    LANCE         003B     006FDE28  SHARE        OWN
SYSTEM2    PUFF          003D     006FDE28  SHARE        OWN
 ===>
```

Example 3: If Pete is interested in querying the users of all datasets with a high level qualifier of BLM.ORDER, he enters this command:

```
D GRS,RES=(SYSDSN,BLM.ORDER.*)
```

An example of the results of this command is:



```
|------( ********* -MVS- MASTER CONSOLE ********* )----( PF3: TO EXIT)----------|




ISG343I 10.14.52 GRS STATUS
S=SYSTEMS SYSDSN   BLM.ORDER.PANELS
SYSNAME   JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SYSTEM1    LANCE          003B     006FDE28   SHARE        OWN
SYSTEM2    PUFF           003D     006FDE28   SHARE        OWN
S=SYSTEMS SYSDSN   BLM.ORDER.SDIDS
SYSNAME   JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SYSTEM1   LANCE           003B    006FDE28    SHARE        OWN
SYSTEM2   BUZZ            0022    006FDE28    SHARE        OWN
SYSTEM1   BLGUT21J        004F    006FDE28    SHARE        OWN
SYSTEM2   PUFF            003D    006FDE28    SHARE        OWN
```

More information on the Display GRS command can be found in the *OS/390 MVS System Commands* document.

7

RDR Command

You can use the RDR command to maintain and monitor remote data resources. A remote data resource is a named area in the Tivoli Information Management for z/OS BLX-SP that enables multiple user address spaces being supported by a single BLX-SP address space to share data. Items can be placed in a remote data resource by terminal simulator EXECs (TSXs) running in any user address space. For example, a remote data resource can be used to temporarily store and retrieve data that can later be written to a file by a TSX or sent to a batch job.

A remote data resource is defined by issuing the OPENRRES or GETRDATA control lines from within a TSX. For an overview of remote data resources, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*. For more information on using TSX control lines used to create and manage a remote data resource, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

The RDR command enables you to do the following:

- Query one or all of the remote data resources that have been defined for a BLX-SP.
- Set the maximum number of data entries that can be contained by a remote data resource. You can also set the minimum and warning limits for writing alert messages to the operator console.
- Change the state of a resource to released, draining, or closed.

Note: Once a remote data resource is defined in Tivoli Information Management for z/OS, it remains in the system until the BLX-SP is restarted.

Syntax

The syntax for the RDR command is as follows:

```
MODIFY|F   procname,RDR
               [,Query[=name]]
               [,Drain=name]
               [,Flush=name]
               [,Release=name]
               [,Set=(name,minimum,warning,maximum)]
```

A maximum of one keyword can be specified on the RDR command. If no keyword is specified, the default is QUERY.

procname
Specifies the name of the started catalog procedure that identifies the BLX-SP.

Query[=name]

Requests that the state and statistics be displayed either for the named remote data resource or, if the name value is omitted, for all defined remote data resources. This is an optional keyword.

Drain=name

Requests that the named remote data resource be changed from its current state to the draining state. If a resource is in a draining state, a TSX may remove existing data entries from it, one data entry at a time. No new data entries can be added to the resource. If there are no entries in the resource, the state is changed directly to closed. DRAIN is valid only when the current state is open or released. If the request is successfully processed, the new state and statistics are displayed for the resource. DRAIN is an optional keyword.

Flush=name

Requests that the named remote data resource be changed from its current state to the closed state. FLUSH forces immediate termination of all processing for a resource. Any data entries in the resource are flushed. This keyword is valid only when the current state of the resource is open, draining, or released. If the request is successfully processed, the new state and statistics are displayed for the resource. FLUSH is an optional keyword.

Release=name

Requests that the named remote data resource be changed from its current state to the released state. Data entries may continue to be added to the resource. RELEASE is valid only when the current state of the resource is open. If the request is successfully processed, the new state and statistics are displayed for the resource. RELEASE is an optional keyword.

Set=(name,minimum,warning,maximum)

Requests that the alert values of the named remote data resource be changed to the specified values.

If you specify a value with the SET keyword, all three values for minimum, warning, and maximum are required. Each value cannot exceed 999999999. When non-zero values are entered, the value for minimum should be less than the value for warning, and the warning value should be less than the value for maximum.

minimum

Specifies the lower threshold for reactivating a warning message after it has been initially triggered. If you are specifying a minimum allowed item count, the value must be greater than 0 and less than the value specified for maximum allowed item count.

warning

Specifies the threshold for generating a warning message. The value must be 0 (a message is never issued) or greater than or equal to the value specified for minimum and less than or equal to the value specified for maximum allowed item count.

maximum

Specifies the largest number of data entries that can be contained by the resource.

SET is not dependent upon the state of the resource. If the request is successfully processed, the new state and statistics are displayed for the resource. SET is an optional keyword.

Scenario Using the RDR Command

Lance wants to monitor the operation of a TSX that removes data entries from a remote data resource. Unsure of the exact spelling of the resource name, he queries all the defined resources:

```
F BLX1PROC,RDR,QUERY
```

and views the results on the MVS master console:

```
BLX03420I Name      State    Current   Minimum   Warning   Maximum
BLX03421I RDR2      DRAINING       5       512       768      1024
BLX03422I RDR45A    OPEN           0       512       768      1024
BLX03422I RDR503    RELEASED      21       512       768      1024
```

Lance notes that the particular resource of interest is named RDR45A and that it is open, but currently contains no data entries. Therefore, he decides to close the resource, thus ending the TSX that is waiting for data, in order to perform some maintenance work. To do this, he enters the command:

```
F BLX1PROC,RDR,FLUSH=RDR45A
```

The resource is closed and the new state and statistics are displayed on the MVS master console:

```
BLX03420I Name      State    Current   Minimum   Warning   Maximum
BLX03422I RDR45A    CLOSED         0       512       768      1024
```

A little while later the batch job running the TSX, accessing the RDR45A remote data resource, determines that the resource has been closed and ends.

8

REALLOC Command

The REALLOC command reallocates a data set after it was freed by the FREE command. You can reallocate the data set so that the general user community can access the data set.

If you need to run a Tivoli Information Management for z/OS utility program against a VSAM data set and sysplex mode is not enabled, you can specify a UTIL keyword to enable the utility to run without users accessing the data set. When sysplex mode is enabled, there is no need to specify a UTIL keyword to run a utility. You can run the utility after the data set is freed but before it is reallocated.

In a non-sysplex environment, you cannot use the FREE and REALLOC commands on data sets that are not defined in the VSAM Resource Definition module until those data sets have been accessed through a Tivoli Information Management for z/OS session and dynamically allocated. In sysplex mode, to use the FREE and REALLOC commands, the data sets need only be accessed by a user or job through a Tivoli Information Management for z/OS session and dynamically allocated.

Syntax

The syntax for the REALLOC command is as follows:

```
MODIFY|F   procname,REALLOC
               {,FILE=lname|,DATASET=dsname}
               [,UTIL]
```

procname
Specifies the name of the started catalog procedure that identifies the BLX-SP.

If sysplex mode is enabled, the REALLOC command request affects all Tivoli Information Management for z/OS users on each system in a sysplex, regardless of the BLX-SP procname specified.

FILE=lname
Specifies the logical name of the data set you want to reallocate (in a non-sysplex environment). The FILE keyword is not valid if sysplex mode is enabled.

The REALLOC command does not support truncated logical names. The logical name assigned by the BLX-SP is usually specified for the BLXDSN macro in the VSAM Resource Definition member that initiates the BLX-SP. The data set could have been added using the ADDVDEF command. If the label on the BLXDSN macro is not coded or a BLXDSN macro is not specified for the data set, BLX-SP assigns the logical name. Use the BLX-SP QUERY command to determine the logical names currently assigned.

Specify a value for either the FILE keyword or the DATASET keyword parameters, but not both.

DATASET=dsname

Specifies the 1- to 44-character fully qualified data set name of the data set you want to reallocate. Use the BLX-SP QUERY command to determine the data sets currently defined and their names.

In a non-sysplex environment, specify a value for either the DATASET keyword or the FILE keyword parameters, but not both. When sysplex mode is enabled, only the DATASET keyword is valid.

UTIL

When the UTIL keyword is specified in a non-sysplex environment, the specified data set is reallocated, but only Tivoli Information Management for z/OS utility programs can access the data set.

The UTIL keyword is not supported when sysplex mode is enabled; if specified, an error message will be returned.

The REALLOC command, without the UTIL keyword, reallocates the data set for general use. This action clears away the effect of the UTIL keyword specified on a previous call to the REALLOC command.

To use the UTIL keyword in a non-sysplex environment, follow these steps:

1. Use the FREE operator command to free the appropriate data sets.
2. Use the REALLOC operator command specifying the UTIL keyword to reallocate the appropriate data sets. This reserves them in Tivoli Information Management for z/OS for use by the utilities. It does not, however, reserve them for use only by Tivoli Information Management for z/OS. Use the BLX-SP QUERY command to verify that the data set is reserved.
3. Run the desired Tivoli Information Management for z/OS utilities.
4. Use the REALLOC operator command, with no UTIL keyword, for the appropriate data sets to make each one available to the general Tivoli Information Management for z/OS user community.

To run a utility when sysplex mode is enabled, follow these steps:

- Use the FREE operator command to free the appropriate data sets.
- Run the desired Tivoli Information Management for z/OS utilities.
- Use the REALLOC operator command, with no UTIL keyword, for the appropriate data sets to make each one available to the general Tivoli Information Management for z/OS user community.

Scenario Using the REALLOC Command (File Keyword) – Non-Sysplex Environment

Lance wants to reallocate a file for general use. Sysplex mode is not enabled. He enters the command on his MVS Master Console.

```
F BLX1PROC,REALLOC,FILE=OEPREAD
```

The BLX-SP returns a message that the REALLOC command is completed.

```
12.53.42 BLX03250I The REALLOC command completed processing.
```

Scenario Using the REALLOC Command (Data Set Keyword)

| This scenario is the same for sysplex and non-sysplex environments. You can use the command with the fully qualified data set name. For example:

```
F BLX1PROC,REALLOC,DATASET=BLM.ORDER.PANELS
```

The BLX-SP returns the same message as before.

```
12.53.42 BLX03250I The REALLOC command completed processing.
```

| **Note:** The data set is not reopened until you perform a function that opens the data set, such as starting a Tivoli Information Management for z/OS session.

9

TL (trace/log) Command

Use the TL command to do the following:

- Manage the trace and log data sets.
- Compile information created by functions in the BLX-SP address space.
- Place the compiled information in the log data set.
- Place the compiled information in the trace data set.
- Continue data accumulation in a new data set.
- Close and free the log data set.
- Close and free the trace data set.
- Turn on or turn off the trace data set.
- Turn on or turn off the log data set.
- Specify the number of lines to write.
- Turn on or turn off individual trace points.
- Specify the SYSOUT data set class.
- Time the close and the free operation.
- Review your current setup.

The trace data set and the log data set are system data sets. You need not specify their names.

The trace data set contains trace information about the operation of the BLX-SP address space. Use this information for analysis of problems in the BLX-SP. This information is in machine-readable form for additional processing.

You can turn on or turn off individual trace points dynamically. Before data is generated, you must turn trace on, and you must set trace points.

The log data set is a log of the initial operating characteristics of the BLX-SP address space and messages that pertain to the BLX-SP address space. The messages appear in the operator's log when appropriate.

The log data set is either active or inactive. Before data is generated, you must turn on trace, and you must set trace points.

Syntax

The syntax for the TL command is as follows:

```
MODIFY|F   procname,TL
           [,TRACE={ON|OFF|SWAP}]
           [,LOG={ON|OFF|SWAP}]
           [,TRACELINES=n]
           [,LOGLINES=n]
           [,TRACESYSOUT=class]
           [,LOGSYSOUT=class]
           [,TRACETOD={x|(x,[y,[z]])}]
           [,LOGTOD={x|(x,[y,[z]])}]
           [,QUERY={TRACE|LOG|*}]
           [,TRACEPOINTS={x|(x,y,z,...)}]
```

procname

Specifies the name of an active BLX-SP process.

TRACE=

Specifies the setting of:

ON Set trace on

OFF Set trace off

SWAP

Set trace to swap

SWAP performs four steps:

1. Closes the current trace data set. The close function can be based on a maximum number of allowable lines, or on a time-of-day clock value, or both.
2. Frees the current trace data set. The close function can be based on a maximum number of allowable lines, or on a time-of-day clock value, or both.
3. Allocates another trace data set to open.
4. Activates the new trace data set.

LOG=

Specifies the setting of:

ON Set log on

OFF Set log off

SWAP

Set log to swap

SWAP performs four steps:

1. Close the current log data set. The close function can be based on a maximum number of allowable lines, or on a time-of-day clock value, or both.
2. Free the current log data set. The free function can be based on a maximum number of allowable lines, or on a time-of-day clock value, or both.
3. Allocate a new log data set.

4. Activate the new log data set.

TRACELINES=n

Specifies line value of the trace data set.

- **n** valid values are integers between 50 and 200 000. When the **n** limit is reached, the current trace data set is closed and freed, and a new trace data set is allocated and activated.
- A value of 0 specifies that there is no limit on the number of lines recorded in the trace data set. Also, a value of 0 specifies that automatic swapping of data sets based on line count does not occur.

LOGLINES=n

Specifies line value of the log data set.

- Valid values for **n** are integers between 50 and 200 000. When the **n** limit is met, the current log data set is closed and freed, and a new log data set is allocated and activated.
- Value of 0 specifies that there is no limit on the number of lines recorded in the log data set.

  Also, a value of 0 specifies that automatic swapping of data sets based on line count does not occur.

TRACESYSOUT=class

Specifies the SYSOUT class to which the current trace data set is to be allocated.

LOGSYSOUT=class

Specifies the SYSOUT class to which the current log data set is to be allocated.

TRACETOD={x|(x,[y,[z]])}

Specifies from one to three time-of-day values, or activates or deactivates existing time-of-day values. When more than one value is specified, the values must be enclosed in parentheses.

Each time-of-day value is specified as follows:

```
HH:MM:SS
```

- Hours are based on a 24-hour clock and must be between 00 and 23.
- Minutes must be between 00 and 59.
- Seconds must be between 00 and 59.
- Midnight is 00:00:00.

Use the operands ON and OFF to activate or deactivate existing time-of-day values. The existing time-of-day values can be set by the system administrator during Tivoli Information Management for z/OS installation. They are the default values for your system. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for more information on how to set these time-of-day values.

At the specified time of day, the trace data set is closed and freed and a new trace data set is allocated and activated,regardless of the value in the TRACELINES keyword.

LOGTOD={x|x,[y,[z]])}

Specifies from one to three time-of-day values, or activates or deactivates existing time-of-day values. If more than one value is specified, the values must be enclosed in parentheses.

Each time-of-day value is specified as follows:

```
HH:MM:SS
```

- Hours are based on a 24-hour clock and must be between 00 and 23.
- Minutes must be between 00 and 59.
- Seconds must be between 00 and 59.
- Midnight is 00:00:00.

Use the operands ON and OFF to activate or deactivate existing time-of-day values. The existing time-of-day values can be set by the system administrator during Tivoli Information Management for z/OS installation. They are the default values for your system. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for more information on how to set these time-of-day values.

At the specified time of day, the log data set is closed and freed, and a new log data set is allocated and activated, regardless of the value in the LOGLINES keyword.

QUERY=

Specifies that messages about the TL operating characteristics be sent to the requesting operator's console. The operating characteristics indicate the current status or settings of the other keywords in the MODIFY command.

The QUERY keyword is mutually exclusive from all other keywords.

TRACE

The operating characteristics detailed in the console messages.

- State of the trace data set
- The current trace data set trace points
- The state of the current trace data set trace points
- Current and maximum number of lines written to the trace data set, the SYSOUT class
- Times of day for swapping

LOG

The operating characteristics detailed in the console messages.

- State of the log data set
- Current and maximum number of lines written to the log data set, the SYSOUT class
- Times of day for swapping

***** Enter an asterisk for both log and trace information.

TRACEPOINTS={x|(x,y,z,...)}

Specifies one or more trace points. The values X, Y, and so forth, specify decimal integers between 1 and 64 inclusive. Each number represents a predefined trace point to turn on. Use a value of 0 to turn off all currently set trace points.

The value of 0 is mutually exclusive from all other values.

When you set a trace point, the trace data set can be either active or inactive:

- If it is active, tracing begins immediately.
- If it is inactive, tracing is delayed until you make the trace data set active (TRACE=ON).

When tracing begins, the internal timer limit of 3 minutes begins. After 3 minutes, tracing stops, and the current trace points are set inactive. Start another 3-minute interval for the current trace points by specifying TRACE=OFF, then TRACE=ON, or by specifying TRACEPOINTS=(x,y,z,...) again.

Because trace points can generate a lot of data and can result in decreased performance of the BLX-SP, the 3-minute timer limits the amount of data recorded. In fact, values set for the TRACELINES and TRACETOD keywords can cause swapping of the trace data set.

Specify either TRACE=OFF or TRACEPOINTS=0 to stop tracing before the 3-minute limit expires.

Trace messages, not yet recorded in a trace data set when the trace is completed, are written into a trace data set.

Defining a Trace Point

The trace point defined for Tivoli Information Management for z/OS is the *program call service request trace point*. The trace point traces information associated with requests to the BLX-SP. The following figure is a DSECT that maps the data recorded in the trace data set for this trace point.

```
TRACE1    DSECT
ASA       DC      CL1    ASA control character (blank).
RSVD1     DC      CL1    Reserved
TP        DC      FL2    Trace point number (1).
TOD1      DC      CL8    TOD clock timestamp:  date/time that this
*                           trace point message was processed by TL.
MSGS      DC      FL4    Number of trace point messages on the
*                           trace queue to be processed.
FMC       DC      FL4    Function matrix code of this service request.
FC        DC      FL4    Function code of this service request.
ASID      DC      FL2    Requesting ASID (address space).
RSVD2     DC      CL6    Reserved.
TODSTART  DC      CL8    TOD clock timestamp when the service request was
*                           received by the BLX-SP's address space.
TODEND    DC      CL8    TOD clock timestamp indicating the service
*                           request was completed by the BLX-SP's
*                           address space.
```

Figure 1. Program Call Service Request Trace Point

Trace points can produce a great deal of output, which can impact the speed of the BLX-SP address space when servicing users' address spaces. However, trace points are not normally turned on until performance data is needed for analysis. Generally, in normal operating mode the log is active, but trace points are not. In addition, a 3-minute limit is clocked by BLX-SP for trace points; After 3 minutes the trace points are turned off. At that point, you must specify the trace points again for additional tracing, or again turn TRACE on to continue collecting data using the same trace points.

Scenario Using the TL Command

To see the current parameters for the LOG, at the command line of the MVS master console type

```
F BLX1PROC,TL QUERY=LOG
```

and press Enter.

The BLX-SP returns a list of the settings and the status (ON or OFF) of the:

- LOG
- Maximum LINE COUNT
- SYSOUT class
- Time-of-day

To set the log to off, at the command line type

```
F BLX1PROC,TL LOG=OFF
```

and press Enter.

To test the log, at the command line type the query command

```
F BLX1PROC,TL QUERY=LOG
```

and press Enter. The status line shows the log is off.

To set the log to on, with a maximum line count for the SYSOUT data set of 150 lines, at the command line type

```
F BLX1PROC,TL LOG=ON,LOGLINES=150
```

and press Enter.

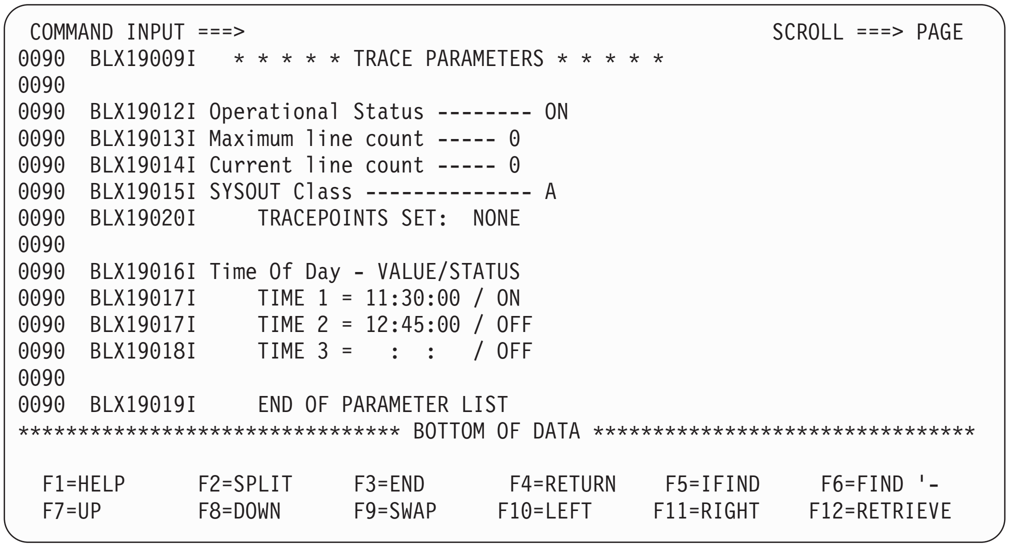
To view the new parameters, at the command line type

```
F BLX1PROC,TL QUERY=LOG
```

and press Enter. You see that the current line count has a number also. When the current line count meets or exceeds the maximum line count, the current data set is closed and freed, and then a new data set is created to store the next series of log messages.

To set a TRACETOD of 11:30:00; set the TRACETOD as shown (see “Syntax” on page 38). Or, if your system administrator has already set 11:30:00 during Tivoli Information Management for z/OS initialization, just specify ON for the time to activate it.

Use the QUERY keyword to view the parameters you have set.



```
 COMMAND INPUT ===>                                           SCROLL ===> PAGE
0090  BLX19009I   * * * * * TRACE PARAMETERS * * * * *
0090
0090  BLX19012I Operational Status -------- ON
0090  BLX19013I Maximum line count ----- 0
0090  BLX19014I Current line count ----- 0
0090  BLX19015I SYSOUT Class -------------- A
0090  BLX19020I     TRACEPOINTS SET:  NONE
0090
0090  BLX19016I Time Of Day - VALUE/STATUS
0090  BLX19017I     TIME 1 = 11:30:00 / ON
0090  BLX19017I     TIME 2 = 12:45:00 / OFF
0090  BLX19018I     TIME 3 =   :  :   / OFF
0090
0090  BLX19019I     END OF PARAMETER LIST
********************************* BOTTOM OF DATA *********************************

  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=FIND '-
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Time-of-day values are positional. To activate the third preset time-of-day, but not the first two, at the command line type

```
TRACETOD=(,,ON)
```

and press Enter. The commas indicate that there is no change to the value already in that position. The status of the first and second time-of-day values is unchanged (if already ON, they remain on; if already OFF, they remain off).

The time-of-day value status is specified independently of the TRACE keyword. TRACETOD can have a value of ON, even if the TRACE value is OFF. In that case, the TRACETOD value is ignored. The TRACETOD values affect only Tivoli Information Management for z/OS operations if the TRACE data set is activated (ON).

The time-of-day value status is specified independently of the LOG keyword. LOGTOD values can have values of ON, even if the LOG value is OFF. In that case, the LOGTOD value is ignored. The LOGTOD values affect only Tivoli Information Management for z/OS operations if the log data set is activated (ON).

9. TL (trace/log) Command

10

Maintaining Data Sets and Databases

Tivoli Information Management for z/OS provides utilities to help you maintain your databases. Database maintenance can include the tasks of expanding your database, as well as preventing or recovering from loss of data.

Tivoli Information Management for z/OS provides the following utilities:

- BLGUT1 to rebuild the SDIDS, using the SDDS as input
- BLGUT1M to copy/migrate SDIDS data to a new SDIDS structure
- BLGUT3 to update an SDDS and SDIDS, using the offloaded SDLDS as input
- BLGUT4 to offload the SDLDS
- BLGUT5 to install and maintain the dictionary data set
- BLGUT5F to offload the dictionary data set
- BLGUT6 to install and maintain a panel data set
- BLGUT6F to offload a panel data set
- BLGUT6M to change panel field lengths and validation patterns
- BLGUT7 to copy the SDDS and
  - Convert Key 7 format to Key 8 format
  - Convert Key 8 record format to Key 7 record format
  - Convert 1-cluster SDDS to a multiple-cluster SDDS
  - Convert a multiple-cluster SDDS to a 1-cluster SDDS
  - Change the SDDS MAXLRECL size
- BLGUT8 to build LLAPI/HLAPI or RDMT tables
- BLGUT9 to set database options
- BLGUT10 to convert the case of data
- BLGUT17 to convert date data
- BLGUT18 to build PIDTs and PIPTs
- BLGUT20 to analyze the SDDS
- BLGUT21 to analyze the SDIDS
- BLGUT22 to analyze a VSAM panel data set
- BLGUT23 (BLGUT23B, BLGUT23P, BLGUT23U, BLGUT23R) to back up the SDDS, prune offloaded log data, update the data, and restore the SDDS
- BLGUTR to initialize the recovery log data set (SDLDS)
- BLGOZUD to create user databases

The BLX-SP must be started before you can use Tivoli Information Management for z/OS in any way, including running utilities. This can present a problem if your users still have access to the data set you need to work on. To keep your users from accessing a Tivoli Information Management for z/OS data set, first use the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about these commands.

Expanding and Analyzing Databases

You can expand your Tivoli Information Management for z/OS database to increase the VSAM space for Tivoli Information Management for z/OS records by adding additional cylinders and/or additional VSAM clusters. You can also create a backup of the current SDDS and SDIDS. To expand a database, you need to remember that there are two types of expansion:

- Physical expansion that does not affect the logical characteristics of the SDDS or SDIDS. This type of expansion is limited to:
  - Increasing (or decreasing) the number of cylinders allocated to the current SDDS or SDIDS cluster(s)
  - Increasing (or decreasing) the FREESPACE used by the current SDDS or SDIDS cluster(s)
  - Increasing the CONTROLINTERVALSIZE of the current SDDS or SDIDS cluster(s)
  - Increasing the MAXLRECL of the current SDDS or SDIDS cluster(s)
- Physical expansion that affects the logical characteristics of the SDDS or SDIDS. This type of expansion includes one or more of the following:
  - Increasing (or decreasing) the number of VSAM clusters that make up the current SDDS or SDIDS
  - Decreasing the MAXLRECL of the current SDDS or SDIDS
  - Changing the VSAM key length of the SDDS or SDIDS
  - Changing the key ranges used by the SDIDS cluster(s)

IDCAMS REPRO can be used to expand a database when the expansion does not affect the logical characteristics of the database. If the expansion affects any of the logical characteristics of the database, then the appropriate Tivoli Information Management for z/OS utility must be used. Use the following steps as a guide:

1. Review the existing cluster definition characteristics for the SDDS and SDIDS to determine whether a revision is necessary. See the sample JCL provided in BLGEXPND in the sample library (SBLMSAMP).

   Run the BLGUT20 and BLGUT21 utilities to obtain statistics on your database. This information helps determine how to redefine your database to more efficiently use space. See “BLGUT20—Analyze the SDDS” on page 151 and “BLGUT21—Analyze the SDIDS” on page 155 for further information.
2. Define alternate SDDS and SDIDS clusters, including new characteristics, if appropriate.
3. Load the current SDDS cluster(s) into the alternate SDDS cluster(s) using the BLGUT7 utility if you are performing one or more of the following tasks:
   - Increasing the number of SDDS clusters used
   - Decreasing the number of SDDS clusters used
   - Decreasing the MAXLRECL size of the SDDS cluster(s)
   - Changing the SDDS key length from 7 to 8
   - Changing the SDDS key length from 8 to 7

   Otherwise, IDCAMS REPRO can be used if desired.

4. Load the current SDIDS cluster(s) into the alternate SDIDS clusters using the BLGUT1M or BLGUT1 utility if you are doing one or more of the following tasks or if one or more of the following are true:

   - Increasing the number of SDIDS clusters used
   - Decreasing the number of SDIDS clusters used
   - Decreasing the MAXLRECL size of the SDIDS cluster(s)
   - The current SDIDS MAXLRECL size is greater than 32 752
   - Changing the KEYRANGES of the SDIDS cluster(s)
   - Changing the SDIDS key length from 18 to 34
   - Changing the SDIDS key length from 34 to 18
   - Changing the SDIDS key length from 16 or 32 to 18 or 34
   - Changing from a spanned SDIDS to a non-spanned SDIDS

   Otherwise, IDCAMS REPRO can be used if desired.
5. Use the IDCAMS ALTER command to rename the original SDDS and SDIDS clusters if you want to keep them as a backup. Otherwise, delete them.
6. Rename the alternate data sets to the original SDDS and SDIDS cluster names, using the IDCAMS ALTER command.

If you are expanding a multiple-cluster SDDS, perform the above described steps for each data set. As an alternative to using IDCAMS for the copy, you can use BLGUT7, which can copy a multiple-cluster SDDS to another multiple-cluster SDDS. Refer to “BLGUT7—Copy an SDDS” on page 105.

Users cannot access a database while you are expanding it. Therefore, expansion provides an opportunity for you to back up that database.

Determining Your Backup/Recovery Approach

Tivoli Information Management for z/OS provides various ways for you to back up and recover your data. Depending on your specific needs, the size of your database, frequency of record updates, and user activity, you may decide that one approach is better than another. The methods available to you include the use of the BLGUT3 utility, the BLGUT4 utility, the BLGUT23 series of utilities, and the Automatic Log Save facility. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for a general discussion of the advantages and disadvantages of the database backup and recovery options that are available to you.

11

BLGOZUD—Create a User Database

The BLGOZUD utility enables loading of data from a sequential data set into a user-defined database. After the data is loaded into the database, you can perform searches for information. You can display and print the records from the database, but you cannot update them.

Two applications where having a user-defined database would be helpful are:

- Process documentation. Your company probably spends a lot of time and money in establishing processes. After these processes are finalized, you want your employees to be able to display and print the processes, but you probably do not want your employees updating them at will. Putting your processes in a user-defined database meets those requirements. A suggested logical file for process documents is P. (See "Creating User Entries" on page 52 for information about logical files.)
- Vacation schedules. Many companies require their employees to schedule their vacations for a given year in January. After all the vacation data is compiled, you can store it in a user-defined database. A probable logical file for vacation schedules is V.

In both examples, putting your data on a user-defined database allows your users to quickly search for specific information. Any data that must be searchable and does not change frequently is a candidate for a user-defined database.

A user-defined database differs from a Tivoli Information Management for z/OS database in the following ways:

Table 2. Differences between User-defined and Tivoli Information Management for z/OS Databases

Database Aspects	User-Defined Database	Tivoli Information Management for z/OS Database
Database number	0, 1, 2, 3	4, 5, 6, 7, 8, 9
Accessibility	Is a read-only database. Records can be created and deleted, but not changed. You must use BLGOZUD to create and delete records.	Can be a read-only (database 4, 6, 7, 8, and 9) or a read/write (database 5) database. Database 6 is reserved for Tivoli Information Management for z/OS Inventory records. In database 5, records can be created, modified, and deleted. Users update records directly in the database.
Database format	Must be defined with an 8-byte key length. Can only be a 1-cluster SDDS and a 1-cluster SDIDS.	Can be defined with either a 7-byte or an 8-byte key length. Can be a 1- or multiple-cluster SDDS, and a 1- or multiple-cluster SDIDS.

Table 2. Differences between User-defined and Tivoli Information Management for z/OS Databases (continued)

Database Aspects	User-Defined Database	Tivoli Information Management for z/OS Database
Record IDs	7 characters in length: First character must be alphabetic, remaining 6 characters must be numeric.	8 characters in length: System-assigned record IDs are all numeric. User-assigned record IDs must begin with an alphabetic character.
Record format	Freeform text does not use s-words or p-words	Structured order depends on s-words and p-words
Logical files	The first character of the record ID determines to which logical file a record belongs (for example, P000513 belongs to logical file P, while record V240002 belongs to logical file V).	Does not use logical files.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The EXEC statement PARM keywords to use with BLGOZUD are as follows:

```
PARM=‘SESS=aa,NAME=n’
```

SESS=aa
Specifies the suffix for the session-parameters member name that BLGOZUD uses. Use 1 or 2 alphanumeric or national characters for *aa*, except when the load module name is 7 characters long.If the load module name is 7 characters long, use one character for *aa*.

NAME=n
Specifies the name of the database—0, 1, 2, or 3.

Input

BLGOZUD accepts a sequential data set as input and converts it to a read-only, user-defined database. The data can be either SBCS or DBCS character strings.

Create the sequential data set that BLGOZUD processes through ISPF/PDF or an equivalent facility. This sequential data set contains control statements and input data consisting of user entries.

BLGOZUD supports a one-cluster SDDS and SDIDS only.

Output

The output from BLGOZUD is a read-only user-defined database.

DD Statements

Note: All control statements, header, title, and keyword character restrictions remain the same. The entry identifier is not DBCS enabled, and it retains the restriction of 1 letter and 6 numbers.

Use control statements to delete previously added entries. The control statements can appear at the beginning of the data set or be interspersed between entries in the data set, such as between a $EOM and an H line.The delete function also erases from the SDIDS the keyword references to the deleted entries.

If you try to add an entry that is already in the database, the entry does not change and the input data set is searched for the next entry.

BLGOZUD reads the sequential data set defined by the SYSIN DD statement and either inserts the entries and associated keywords into the database or deletes them from the database. In addition to the user data entries, the SYSIN data set also contains the control statements required for processing the entries.

Figure 3 on page 56 contains the JCL that initiates BLGOZUD.

BLGOZUD has a delete control statement:

$DEL
Delete a user entry

BLGOZUD control statements begin in column 1. You can code them in uppercase or lowercase characters.

To delete a user entry, use the $DEL statement. The syntax of the $DEL control statement is:

```
$DEL fnnnnnn
```

fnnnnnn
Specifies the file and the identifier. Leading zeros need not be present in the entry identifier. (For example, you can use $DEL U1234 to delete entry U0001234.) Delete an old entry before you replace it with a new entry.

The $DEL control statement deletes the entry and its associated keywords in the keyword index.

To update an entry or reuse its identifier for a different entry, delete the old entry by including a $DEL statement before the new or updated entry occurs in the input sequential data set.

If you attempt to add an entry that is already in the database (in other words, you try to reuse an identifier without deleting the original entry), the original entry is not changed; BLGOZUD ignores the new entry and processes the next user data-entry or control statement in the input data set.

Creating User Entries

To create the data that you want for a user database, use ISPF/PDF or an equivalent product.The data set must have the format of a text file. It must be a sequential data set with LRECL=80 and RECFM=F or FB. You can specify any BLKSIZE. Allocate the sequential data set with a SYSIN DD statement, as shown in Figure 3 on page 56. Ensure that each entry in this data set appears in the following format:

One header line (H line)

The header line must contain an H in the first data column, followed by a blank. The entry identifier, consisting of a letter and six digits must start in the third column. The letter, the name of the logical file, specifies in which logical file you want the entry to appear. Refer to the *Tivoli Information Management for z/OS User's Guide* for information about using logical files.

The H and the entry identifier are the only data required in the header line. An optional parameter is:

D=mmmyy|mmmyyyy

The date on which you place the entry in the database, or the date on which you last modified the entry. A 3-character abbreviation, such as JUN, indicates the month. The last 2 or 4 digits of a year, such as 97 or 1997, indicates the year. BLGOZUD stores the date as a keyword in the format DATE/YYMM or DATE/YYYYMM. Consequently, you can retrieve an entry with the date parameter D=JUN1997 through the keyword DATE/199706. BLGOZUD ignores any other data in the H line but stores it as part of the H line in the database.

One or more title lines (T lines)

The title lines follow the H line. The title lines must contain a T in the first data column followed by a blank. You can enter any number of T lines; however, Tivoli Information Management for z/OS displays a maximum of 201 characters of data. A typical entry has one or two T lines. One T line is required.

One or more keyword lines (K lines or L lines)

The keyword lines follow the T lines. Keyword lines are optional; if present, they must contain a K or an L in the first data column, followed by a blank. Enter any number of K or L lines. These lines contain all the keywords you can use to locate this entry during a keyword search.

L lines are not displayed when you display a record. You can use either a K line or an L line in a record, but not both. If you use an L line after a K line, the L line is ignored. If you use a K line after an L line, you receive an error message.

Keywords that start with any of the following character sequences are not valid:

- RNXX/
- CLXX/
- NAXX/
- DATX/
- FILE*

x Represents any character.

* Represents any one character.

Only Tivoli Information Management for z/OS can use the keywords beginning with these characters; do not use these keywords in user data entries.

One blank line (a separator line)
A blank line must separate the last keyword line from the first text line.

One or more text lines (the text of the entry itself)
You can use as many text lines as you require. A text line cannot contain $EOM starting in the first column, because this indicates an end line.

One $EOM line (an end line)
A line containing $EOM starting in the first column must be the last line of an entry.

The header, title, keyword, and text lines must contain from 1 to 72 characters of input data. You can mix uppercase and lowercase characters for the data in the records. However, the output of the H, K, L, and T lines is uppercase.

The next header record should follow $EOM with no blank lines between them.

Error Processing

If any of the required data sets (SYSIN, SYSPRINT, the SDDS or the SDIDS) is not usable, BLGOZUD ends with an ABEND and an associated reason code.

BLGOZUD bypasses entries such as the following:

- An entry that already exists in the user database.

  The entry is not added to the database, a message identifies the error, and processing continues with the next entry.
- An entry has no valid header line.

  A message identifies the error and processing continues with the next entry.
- An entry whose lines within the entry are not in the required sequence.

  A message identifies the error, and processing continues with the next entry.
- An entry with keywords that are not valid (K line data) and a date on a header line that is not valid.

  The entry is added to the database, messages identify the error, and processing continues with the next entry.
- A $DEL statement that does not contain a valid entry identifier (when attempting to delete an entry from the user database).

  No entry is deleted, warning messages identify the error and processing continues with the next entry.
- A $DEL statement lists a valid entry identifier, but the entry is not in the database.

  A warning message identifies the error, and the processing continues.

Restart BLGOZUD

BLGOZUD processes each entry (or delete control statement) and any associated keywords as a unit.Therefore, you can start BLGOZUD and rerun it, as it is, without changing the input and without restoring any data sets to a previous state. Thus, to restart BLGOZUD, resubmit the JCL shown in Figure 3 on page 56, and the program skips any entries already processed.

Scenarios Using the BLGOZUD Utility Program

The following rules apply when creating input data for BLGOZUD:

- Separate the first text line from the last keyword line or the last title line by a blank line.
- Do not place blank lines between entries.
- Place the $EOM line of one entry immediately before the H line of the next entry or immediately before a control statement, as shown in Figure 2 on page 55.

A sample input to the BLGOZUD utility is as follows:

```
COLUMNS                           *COMMENTS* (for explanation in this
                                              example - not part of
123456789...                                  customer data)

H u000001                         *Header line, start of entry*
T the U file                      *Title line*
K CICS* data staff hints          *Keyword line*
                                  *Blank line separating text*
This is the text for this entry. Use as many lines as required.
Use up to 80 bytes of text per line. This entry can be retrieved
using the keywords CICS, DATA, STAFF, and HINTS.

This is the first U file entry. The U file contains information
that is useful to the Widget Company's CICS application staff.
$EOM
H U000010 D=NOV94                 *Start of entry*
T Widget Company CICS Application Standards - PROGRAM NAMES
K CICS program name names module CSECT standard

CICS application modules must follow this naming convention:
  --naming convention description goes here as text lines--
$EOM
H W000001                         *Start of entry*
T The Widget Company W file
T This file is the Widget Company DP Department Memo File
K memo file

This file contains all department memos issued since January 1st.
$EOM
H W000003 D=SEP91                 *Start of entry*
T Hoffmyer to Blinderschmitz:  Education Schedules
K Hoffmyer Blinderschmitz memo education class schedules
The following education plan is suggested for 1993:
  1st Qtr:
  2nd Qtr:
$EOM

$DEL U25                          *Delete record U000025*

$del u000026                      *Delete record U000026*
H U000025 D=JUN94                 *Start of entry*
T Title line for entry U25
K some keywords here

This is the text for this entry. Use as many lines as required.
Use up to 80 bytes of text per line. This entry is retrievable
using the keywords SOME, KEYWORDS, HERE, DATE/9106, and RNID/U000025.
$EOM
$DEL U000015                      *Delete record U000015*

H U000015 D=JUN94
T This is another title line
T As many title lines as needed
K AAA BBB ccc ddd
This is the text for this entry.
Keywords for this entry are AAA, BBB, CCC, and DDD. All keyword
searches are made in uppercase characters. Keywords can be present in
any mixture of uppercase and lowercase characters. The entry identifier
in the header line must begin in text column 3.
$EOM
```

Figure 2. Input to BLGOZUD Utility

The JCL in Figure 3 is only a skeleton with sample values; some parameters on the DD statements are not included.

```
//UDATA    EXEC PGM=BLGOZUD,PARM='SESS=nn,NAME=n'
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
$del  b000001
 .
 .
 .
 /*
```

Figure 3. Example of JCL

12

BLGUTR—Format a Recovery Log Data Set

Use BLGUTR to format a recovery log data set, SDLDS, before it is used.

Define SDLDS clusters with no secondary extents. If you use secondary extents, all secondary extents are allocated when running BLGUTR, because this utility formats the entire data set.

> **Important!**
>
> The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.
>
> When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.
>
> When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The EXEC statement PARM field keywords to use with BLGUTR are as follows:

```
PARM=‘SESS=aa,NAME=5’
```

SESS=aa

Specifies the suffix for the session-parameters member name that BLGUTR uses. Use 1 or 2 alphanumeric or national characters for `aa`, except when the load module name is 7 characters long. If the load module name is 7 characters long, use only 1 character for `aa`.

If you do not use the `BLGSL` DD statement, the output SDLDS is dynamically allocated, as specified in the session-parameters member. If it is specified, it overrides the data set specified in the session-parameters member.

NAME=5
Specifies the name of the database whose SDLDS is initialized. Use 5 for the Tivoli Information Management for z/OS database. Do not specify databases 0, 1, 2, 3, 4, 6, 7, 8, or 9, because they are read-only databases and do not require a log data set.

Input

Input to BLGUTR is the SDLDS that you initially allocated.

Output

The output from BLGUTR is:

- An initialized SDLDS with Tivoli Information Management for z/OS control information in record 1
- Informational and error messages in a message data set
- A return code of `0` if BLGUTR finishes successfully.
- A return code of `>0` if an error occurs.

After you initially format the SDLDS using BLGUTR, use BLGUT4 to reinitialize that data set thereafter.

Restrictions

The following restrictions apply to formatting the SDLDS:

- BLGUTR fills the SDLDS with preformatted initialization records, including any secondary space allocations for the data set.
- Users cannot access the database while BLGUTR is running.

DD Statements

The DD statements required to run BLGUTR are as follows:

BLGSL (optional)
The Tivoli Information Management for z/OS VSAM SDLDS to format.

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenario Using the BLGUTR Utility Program

The following sample illustrates how you can use BLGUTR to initialize the SDLDS for the Tivoli Information Management for z/OS database. To obtain the sample JCL, see member BLGUTRJ in the SBLMSAMP library.

```
//INIT     JOB
//STEP1    EXEC PGM=BLGUTR,PARM='SESS=00,NAME=5',REGION=4096K
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//BLGSL    DD   DISP=SHR,DSN=BLM.SDLDS
//SYSPRINT DD   SYSOUT=A
```

Figure 4. Using BLGUTR to Initialize the SDLDS for the Tivoli Information Management for z/OS Database

13

BLGUT1—Rebuild the Structured Description Index Data Set

BLGUT1, the database index utility, builds the SDIDS from its corresponding SDDS. BLGUT1 also allows you to set database options at the time you build or rebuild the index. Use BLGUT1 if the SDIDS was destroyed and the SDDS is intact. If you want to change the length of your SDIDS key (from 18 bytes to 34, or vice versa), you can also use this utility to rebuild the SDIDS. However, you may find it more convenient (and less time consuming) to use the BLGUT1M utility if you are changing SDIDS key lengths and your SDIDS is not corrupted. Less DASD space is required for an 18-byte key than for a 34-byte VSAM SDIDS key.

Important

The SDIDS in Tivoli Information Management for z/OS may not be backward compatible, depending on what release of the product you are migrating from:

- It is backward compatible with Tivoli Service Desk for OS/390 Version 1.2. It is not backward compatible with TME 10 Information/Management Version 1.1, Information/Management Version 6.3, or earlier releases because the database architecture associated with those releases is different. Once you migrate to Tivoli Information Management for z/OS Version 7.1, you cannot go back to TME 10 Information/Management Version 1.1 or earlier without rebuilding the SDIDS using BLGUT1.

  Ensure that you use the correct version of BLGUT1 when rebuilding the SDIDS. If you rebuild the SDIDS as part of your Version 7.1 migration and add new records to the SDDS, and, for whatever reason, need to return to the earlier version of the product, you must rebuild the SDIDS using the previous version's BLGUT1 utility in order to reuse the SDDS.

The SDDS is backward compatible with any previous release unless you are using the mixed case support or have selected an external date format that is different from the default used by Tivoli Information Management for z/OS.

The SDIDS is an index to the searchable data items in the SDDS. BLGUT1 rebuilds the SDIDS by reading each record in the SDDS. For each record read, every searchable data item is extracted and given to the SORT routine as a separate record. After the sort, BLGUT1 uses the externally sorted information to rebuild the records contained in the SDIDS. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information about specifying sort characteristics in your session parameters.

The number of records produced for sorting depends on:

- The number of records contained in the SDDS
- The number of searchable data items entered for each record

Each SDDS record relates to many (over 500) SDIDS records. However, the relationships overlap such that there are usually only a few (2 to 4) completely new SDIDS records generated for each SDDS record. The statistics that are produced by SORT/MERGE and BLGUT1 help you arrive at a correct figure for your installation. Refer to the *DFSORT™ Application Programming Guide* for information about size, space, and data set requirements for SORT/MERGE.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Summarized, the steps to run the BLGUT1 utility when sysplex mode is not enabled are:

1. Use the Tivoli Information Management for z/OS FREE operator command with the QUIESCE option on the SDIDS that you want to rebuild.
2. Use the Tivoli Information Management for z/OS FREE operator command with the QUIESCE option on the SDDS that corresponds to the SDIDS.
3. Back up the SDIDS.
4. Delete and redefine the SDIDS.
5. Reallocate (REALLOC) the SDIDS with the UTIL parameter.
6. Reallocate the SDDS with the UTIL parameter.
7. Run the BLGUT1 utility.
8. Reallocate the SDIDS without the UTIL parameter.
9. Reallocate the SDDS without the UTIL parameter.

The steps to run the BLGUT1 utility when sysplex mode is enabled are:

1. Use the Tivoli Information Management for z/OS FREE operator command on the SDIDS that you want to rebuild.
2. Use the Tivoli Information Management for z/OS FREE operator command on the SDDS that corresponds to the SDIDS.

3. Back up the SDIDS.
4. Delete and redefine the SDIDS.
5. Run the BLGUT1 utility.
6. Reallocate the SDIDS.
7. Reallocate the SDDS.

Syntax

This utility builds and writes data access control records which contain SDIDS cluster information. It also stores cognized words in one or more clusters according to the key ranges specified in a session-parameters member.

The EXEC statement PARM field keywords to use with BLGUT1 are as follows:

```
PARM='SESS=aa[,NAME=n][,UNIQUE][,REUSE][,CLUST=(n1,n2,n3...)]'
```

SESS=aa
Specifies the suffix for the session-parameters member name that BLGUT1 uses. Use 1 or 2 alphanumeric or national characters for *aa*. This session-parameters member contains the structure definition of the database to be used and a date conversion routine specification. This keyword is required.

The input SDDS and the output SDIDS structure are automatically derived from the session-parameters member defined in the PARM statement.

NAME=n
Identifies which database (in a session-parameter member containing multiple database definitions) to use. This name identifies the database whose SDIDS BLGUT1 will rebuild from its associated SDDS. You can specify 4, 5, 7, or 8 as the database. If the name is omitted, the default is 5 for the Tivoli Information Management for z/OS database. (Needs checking)

If you are running BLGUT1 to rebuild the SDIDS for a read-only database (database 4, 6, 7, 8, or 9), you do not need a session-parameters member which specifies the read-only database as a database 5 read-write database for use with BLGUT1.

UNIQUE
Indicates that you want the record number ID incremented regardless of what logical database partition is being used. UNIQUE specifies that the RNIDs will be unique across partitions. This keyword is optional, and if omitted, the RNIDs generated will *not* be unique and duplicates will occur (for example, partition #1 will have RNID 00000001, partition #2 will have RNID 00000001, and so on).

REUSE
Specifies that you want to reuse VSAM sequence numbers when creating records on the specified database. This keyword is optional, and if omitted, VSAM sequence numbers are not reused when records are created. For a detailed discussion of the REUSE parameter, see "BLGUT9—Set Database Options" on page 123.

CLUST=(n1,n2,n3...)
Identifies specific clusters of a multiple-cluster SDIDS that should be built. You could use this keyword to rebuild one or more of your multiple SDIDS clusters in the event one or more clusters were damaged. This keyword is optional, and if omitted, all SDIDS clusters for the specified database are built. The value for this keyword can be specified

as either a single number (e.g., `CLUST=15`) or as a parenthesized list of numbers separated by commas (e.g., `CLUST=(1,2,3)`). If you use a list, it does not matter in what order you place the numbers. Each entry must be a value in the range from 1 to the number of clusters in the SDIDS. If you enter a number that is greater than the number of clusters for the SDIDS, an error message is generated and processing stops before any rebuild processing is started.

Multiple BLGUT1 programs can be run simultaneously to rebuild a multiple-cluster SDIDS if each use of BLGUT1 has a different CLUST= value.

Input

The input for BLGUT1 is the SDDS cluster. The SDDS is a key-sequenced VSAM cluster. The data set name of the SDDS is associated with the database (NAME=) named in the specified session-parameters member (SESS=).

Output

The output from BLGUT1 is:

- The SDIDS, an index data set for the SDDS. Delete the existing SDIDS before using BLGUT1. Allocate a new SDIDS with the same data set name, as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.
- Informational and error messages in a message data set.
- A return code of 0 if the utility finishes successfully.
- A return code >0 if an error occurs.

Restrictions

The following restrictions apply to rebuilding the SDIDS:

- Allocate the new SDIDS as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual before you use BLGUT1.
- Users cannot access the database while BLGUT1 is running.
- If you use the UNIQUE keyword, you must run BLGUT1 with this keyword whenever you rebuild the SDIDS. Otherwise, RNID integrity is not maintained.

DD Statements

The DD statements required to run BLGUT1 are as follows.

Note: The BLGSD and BLGSI statements are no longer supported. If you use them, delete them and specify a valid/appropriate session.

BLGSMSG (optional)
Specifies a data set that will receive any Sort/Merge licensed program messages generated during your use of Tivoli Information Management for z/OS.

SORTWKnn
Used by SORT/MERGE.

SYSPRINT
Specifies a sequential message data set that you can write to a system output device or data set.

SYSOUT
Specifies a sequential message data set that you can write to a system output device or data set.

Scenarios Using the BLGUT1 Utility Program

The following scenarios show how to rebuild the SDIDS and improve the run time associated with this utility.

Rebuilding the SDIDS

The first two examples illustrate the use of BLGUT1 to rebuild the SDIDS from the SDDS.

The following example rebuilds the entire SDIDS.

```
//REBUILD   JOB
//STEP1     EXEC PGM=BLGUT1,PARM='SESS=01,NAME=5',REGION=4096K
//STEPLIB   DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//SYSOUT    DD   SYSOUT=A
//SORTWK01  DD   UNIT=SYSDA,SPACE=(CYL,(10))
```

Figure 5. Using BLGUT1 to Rebuild a Complete SDIDS

The following example shows how three SDIDS clusters can be rebuilt for a multiple-cluster SDIDS. A session-parameters member (BLGCLUST macro) provides the overall definition of how many clusters exist in the SDDS or SDIDS database structure.

```
//REBUILD   JOB
//STEP1     EXEC PGM=BLGUT1,PARM='SESS=01,NAME=5,CLUST=(3,9,10)',REGION=4096K
//STEPLIB   DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//SYSOUT    DD   SYSOUT=A
//SORTWK01  DD   UNIT=SYSDA,SPACE=(CYL,(10))
```

Figure 6. Using BLGUT1 to Rebuild 3 Clusters of a Multiple-Cluster SDIDS

Reducing Run Time

The following scenario describes how to reduce the time it takes for BLGUT1 to run, by allocating sort work space that better matches the amount of data to be sorted.

When using a multiple-cluster SDIDS it is possible to reduce the time needed to run BLGUT1 by running multiple BLGUT1 jobs. Each BLGUT1 job must have a different value specified for the CLUST= parameter, so that when they are run together, they can rebuild the entire SDIDS.

The Sort function allocates work space based on the number and size of the records it needs to sort. The size of the records is fixed, but the number of records is not.

To get a good estimate of the number of sort records BLGUT1 will contain, use the output from a prior run of BLGUT1. The “words” the BLGUT1 output refers to are keywords. As BLGUT1 reads the SDDS, it writes each keyword to the sort input data set. Multiply the average number of words per logical record times the number of logical records in the SDDS. The result is the sort estimate. For example:

```
Statistics for SDDS data set
 Number of physical records............................. 9487
 Number of logical records.............................. 8221
 Total number of bytes in data set...................... 1102349
 Maximum number of physical records per logical record... 2
Maximum number of words per logical record............. 48
Minimum number of words per logical record............. 24
Average number of bytes per physical record............ 1144
Average number of physical records per logical record... 1
Average number of words per logical record............. 42
Statistics for SDIDS data set
Number of records written.............................. 9421
Total number of bit references......................... 341092
Total number of bytes in data set...................... 211437
Average number of bits per record...................... 36
Average number of bytes per record..................... 29
The SDIDS utility completed processing with a highest return code of 0
```

Using this BLGUT1 output, the equation would be 42 x 8221 = 345 282 records. You can use your values in this equation, but the numbers will change as you create, update, and delete records from the database.

Total number of bit references shows the exact number of sort records. Each bit refers to a searchable keyword.

The way you pass the estimated number of sort records depends on your sort product. Most sort products enable you to use a DD card statement to override the values that are passed. Include the DD card in the BLGUT1 JCL job stream. For example, to override the estimated number of records for IBM® DFSORT, use the DFSPARM DD card:

```
//DFSPARM DD *
FILSZ=Ennnnnnn
```

In the example, replace nnnnnnn with the number you calculated. Refer to your sort program's documentation for more information about coding overrides.

14

BLGUT1M—Migrate Structured Description Index Data Set

You can use the BLGUT1M utility (as an alternative to the BLGUT1 utility) to copy or migrate data from an existing SDIDS to another SDIDS without requiring the SDDS to be read and sorted. BLGUT1M enables you to perform the following tasks:

- Migrate SDIDS data from all previous releases of the product to Tivoli Information Management for z/OS Version 7.1.
- Copy data from one Version 7.1 SDIDS structure to another structure as defined by a session-parameters member.
- Make changes in the SDIDS key length.

Data can be copied from an SDIDS made up of one or more clusters to an SDIDS made up of one or more clusters. The input SDIDS may contain data from previous versions of the product, but the output SDIDS will always contain data in the Tivoli Information Management for z/OS Version 7.1 format.

If you are currently using Tivoli Service Desk for OS/390 Version 1.2 or TME 10 Information/Management Version 1.1, you are not required to use BLGUT1M to migrate to Tivoli Information Management for z/OS Version 7.1. If you are using Information/Management Version 6.3 or an earlier release, you can use BLGUT1M instead of BLGUT1 to migrate SDIDS data to Version 7.1. If you need to change SDIDS key lengths and/or clusters, you can use BLGUT1M.

Important

The SDIDS in Tivoli Information Management for z/OS may not be backward compatible, depending on what release of the product you are migrating from:

- It is backward compatible with Tivoli Service Desk for OS/390 Version 1.2. It is not backward compatible with TME 10 Information/Management Version 1.1, Information/Management Version 6.3, or earlier releases because the database architecture associated with those releases is different. Once you migrate to Tivoli Information Management for z/OS Version 7.1, you cannot go back to TME 10 Information/Management Version 1.1 or earlier without rebuilding the SDIDS using BLGUT1.

  Ensure that you use the correct version of BLGUT1 when rebuilding the SDIDS. If you rebuild the SDIDS as part of your Version 7.1 migration and add new records to the SDDS, and, for whatever reason, need to return to the earlier version of the product, you must rebuild the SDIDS using the previous version's BLGUT1 utility in order to reuse the SDDS.

If you are copying or migrating from a multiple-cluster environment, this utility will copy all SDIDS records into the new structure, assuming you have supplied the required ITRIG keyword. BLGUT1M cannot copy a subset of your multiple clusters into a new structure.

The SDIDS key length can be changed from 16, 18, 32, or 34 to either 18 or 34. You cannot use this utility to change the key length from 18 or 34 back to either 16 or 32, since the 16 and 32 key lengths are no longer supported. (The key length of the SDIDS was chosen when the SDIDS was defined through the IDCAMS DEFINE CLUSTER command, or it was specified through the Installation Tailoring Facility.)

Note: The BLGUT1M utility does not call the date conversion routine and cannot be used to change the internal date format. If you need to change the internal date format, use the BLGUT1 utility.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See "FREE Command" on page 15 and "REALLOC Command" on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

Use BLGUT1M to migrate or copy data from one SDIDS structure to another, or to change key lengths. The EXEC statement PARM field keywords to use with BLGUT1M are as follows:

PARM=‘SESS=aa[,NAME=n][,ITRIG=c]’

SESS=aa

Specifies the suffix for the session-parameters member name that contains the output SDIDS structure definition. Use 1 or 2 alphanumeric or national characters for *aa*. This session-parameters member contains the structure definition of the database to be generated. This keyword is required.

The output SDIDS structure is automatically derived from the session-parameters member defined in the PARM statement.

NAME=n

Specifies the name of the database whose SDIDS you want BLGUT1M to migrate or copy to (the target). You can specify 0, 1, 2, 3 (a user-defined format database) or 4, 5, 6, 7, 8, or 9 (a Tivoli Information Management for z/OS format database) as the database. If the name is omitted, the default is 5 for the Tivoli Information Management for z/OS database.

If you are running BLGUT1M to rebuild the SDIDS for a read-only database, you do not need a session-parameters member which specifies the read-only database as a database 5 read-write database for use with BLGUT1M.

ITRIG=c

Identifies a trigger character for an input multiple-cluster SDIDS data set that precedes the two-digit count in the data set name. For example, if the data set is `BLM.PROD#03.SDIDS`, you would specify the pound sign character here. This keyword is required when a multiple-cluster SDIDS is specified for input.

As described earlier in this chapter, the SDIDS in Tivoli Information Management for z/OS Version 7.1 may not be backward compatible with previous versions of the product.

Input

The input to the BLGUT1M utility is the VSAM data set (the primary SDIDS cluster) containing the data to be migrated. If you are using a multiple-cluster SDIDS as input, you must specify the ITRIG keyword.

Note: If you are running in sysplex mode, your old SDIDS may not be defined correctly for VSAM RLS. You may need to alter it or redefine it to a new data set with the correct attributes for RLS. For more information on migrating old data sets for use in sysplex mode, see the ″Migrating from Previous Releases″ chapter in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

Output

- An SDIDS containing migrated data, which is defined through the session-parameters member
- A return code of 0 if BLGUT1M finishes successfully
- A return code >0 if an error occurs

Restrictions

The following restrictions apply to copying or migrating SDIDS data using this utility:

- Users cannot access the databases while BLGUT1M is running.
- The new SDIDS cluster(s) must be empty before you run BLGUT1M.

DD Statements

The DD statements required to run BLGUT1M are as follows:

BLGIDSI
The primary SDIDS input data set being copied or migrated.

SYSPRINT
A sequential message data set that can be written to a system output device or data set.

Scenarios Using the BLGUT1M Utility Program

The following scenario shows a migration from either a 16- or 32-byte SDIDS key to an 18- or 34-byte SDIDS key. You can specify the SDIDS key when you define the VSAM data sets for the SDIDS.

```
//STEP1    EXEC  PGM=BLGUT1M,PARM='SESS=01,NAME=5'
//STEPLIB  DD    DISP=SHR,DSNAME=BLM.SBLMMOD1
//BLGIDSI  DD    DISP=SHR,DSNAME=BLM.PROD.SDIDS
//SYSPRINT DD    SYSOUT=A
```

If your original SDIDS contains a single cluster and you want to restructure it into multiple clusters, define the new required SDIDS VSAM clusters, and then run this utility to load the data. In this case, you would omit the ITRIG keyword since your input data set was never associated with trigger characters. The new structure is obtained from the session-parameters member you define with cluster counts and key range values.

The next example illustrates the use of BLGUT1M to copy or migrate SDIDS data from a multiple-cluster environment into a new structure.

```
//STEP1    EXEC  PGM=BLGUT1M,PARM='SESS=01,NAME=5,ITRIG=#'
//STEPLIB  DD    DISP=SHR,DSNAME=BLM.SBLMMOD1
//BLGIDSI  DD    DISP=SHR,DSNAME=BLM.PROD#01.SDIDS
//SYSPRINT DD    SYSOUT=A
```

Figure 7. Using BLGUT1M with the ITRIG Keyword

15

BLGUT3—Restore the Database from the Log Data Set

Use BLGUT3 to update a database (the SDDS and the SDIDS), from recovery log data collected from that database's offloaded SDLDS. BLGUT3 updates the SDDS by reading each record in the pruned backup data set created by BLGUT4. For each record read, every searchable data item is extracted and given to the SORT routine as a separate record. After the sort, BLGUT3 uses the sorted information to update the appropriate records in the SDIDS. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information about specifying an external sort in your session parameters.

For a general discussion of the available methods to back up and restore your database, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*. "BLGUT4—Offload the Recovery Log Data Set" on page 73 also describes how to prepare for and execute recovery actions.

Important!

| The BLX-SP must be started before you can run Tivoli Information Management for
| z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See "FREE Command" on page 15 and "REALLOC Command" on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The EXEC statement PARM field keywords to use with BLGUT3 are as follows:

```
PARM='SESS=aa,NAME=5[,TYPE={RESTORE|ADD|REPLACE}]'
```

SESS=aa

Suffix for the session parameters member name that BLGUT3 uses. Use 1 or 2 alphanumeric or national characters for *aa*.

NAME=5

Name of the database that you must restore. Use 5 for the Tivoli Information Management for z/OS database. Although you can specify databases 0, 1, 2, 3, 4, 6, 7, 8, or 9, do not do so, because these are read-only databases and have no offloaded recovery log data sets.

TYPE=

RESTORE

Indicates that BLGUT3 is to read records from the offloaded SDLDS, file them in the SDDS, and update the SDIDS with the searchable words for those records. The default type is RESTORE, which restores the database from the offloaded recovery log for that database.

RESTORE replaces records using the SDDS position number (VSAM key) rather than by record identifier (RNID). If you are transferring records with system-assigned RNIDs from one database to another, use TYPE=ADD instead of RESTORE. If you are transferring records with user-assigned RNIDs, use TYPE=REPLACE. ADD and REPLACE use the RNID and ignore the SDDS position number.

ADD

Indicates that BLGUT3 is to read records from the offloaded SDLDS and add them to the SDDS. One use of this type of operation would be to transfer SRC and privilege class records from a test database into your production database.

When BLGUT3 detects a record in the database with the same system-assigned record identifier (prefix RNID/), BLGUT3 assigns a new record identifier for the record from the offloaded SDLDS and uses it under the new record identifier. If BLGUT3 detects a record in the database with the same user-assigned record identifier, BLGUT3 skips the record and continues to the next record in the offloaded SDLDS. As records are read from the offloaded SDLDS and filed in the SDDS, the SDIDS is updated with the searchable words from those records.

REPLACE

Indicates that BLGUT3 is to read records from the offloaded SDLDS and replace them in the SDDS.

When BLGUT3 detects a record in the database with the same record identifier, BLGUT3 replaces that record in the SDDS with the record from the offloaded SDLDS. It makes no difference whether the record identifier is a system-assigned or a user-assigned record identifier. As records are read from the offloaded SDLDS and filed in the SDDS, the SDIDS is updated with the searchable words from those records.

Notes:

1. BLGUT3 is used primarily for recovery. If you would like to move or copy records from one Tivoli Information Management for z/OS database to another, consider using the Tivoli Information Management for z/OS Archiver function. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information on the Archiver.

2. You can use the ADD and REPLACE keywords for records other than SRC or privilege class records. However, if the record is part of a Tivoli Information Management for z/OS parent-child relationship, the ADD keyword does not produce the correct connectivity between the records. Tivoli Information Management for z/OS does not check for this condition.
3. If you mistype or misspell the ADD, REPLACE, or RESTORE keywords, then the process halts and identifies the error for you to correct.
4. If you use the TYPE= keyword and omit the value for TYPE, then RESTORE is the default.
5. If you omit the TYPE keyword, then RESTORE is the default.
6. If the BLGBKUP data set being processed contains entries for deleted records, TYPE=RESTORE and TYPE=REPLACE will delete those records from the target database. TYPE=RESTORE deletes the record using the SDDS position number from the entry in the BLGBKUP data set. TYPE=REPLACE deletes the record using the RNID from the BLGBKUP data set.

   TYPE=ADD skips any delete entries in the BLGBKUP data set.
7. The same SDDS position number can exist in both the source and the target database. In such a case, it is possible that the SDDS position number represents different RNIDs on the two databases. During the BLGUT3 RESTORE process, BLGUT3 deletes the record on the target database using the SDDS position number, and then adds the new record from the input database using the same SDDS position number. The original record is then lost. To prevent this problem, use TYPE=REPLACE or TYPE=ADD when running BLGUT3. The ADD and REPLACE processes use the RNID of the source record, not the SDDS position number. Back up your SDDS and SDIDS before performing this type of record processing.

Input

Before running BLGUT3, use Access Method Services (AMS) to restore the most recent backup of the database.

The input to BLGUT3 must contain:

- The offloaded recovery log created since the most recent backup for that database. This is a physical sequential data set created by BLGUT4.

Output

The output from BLGUT3 is:

- The updated SDDS and SDIDS that were restored from the latest backup for that database
- Informational and error messages in a message data set
- A return code of `0` if BLGUT3 finishes successfully
- A return code of `>0` if an error occurs.

Restrictions

The following restrictions apply to restoring a database from the offloaded recovery log:

- You cannot restore data directly from an SDLDS. First use BLGUT4 to offload any data in the SDLDS to the offloaded recovery log (the pruned backup data set). There should be only one backup data set used as the BLGBKUP data set. Then use this backup data set as input to BLGUT3.

- If your organization supports more than one database, make sure that each database has its own SDLDS and backup logs and that the correct backup data sets are used for each database.
- Users cannot access the databases while BLGUT3 is running.

DD Statements

The DD statements required to run BLGUT3 are as follows:

BLGBKUP
The offloaded recovery log.

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

SYSOUT
A sequential message data set that you can write to a system output device or data set. This data set is used by the SORT routine and is required.

Scenarios Using the BLGUT3 Utility Program

The following figures illustrate how you can use BLGUT3 to rebuild the Tivoli Information Management for z/OS database.

```
//RESTORE  JOB
//*       RESTORE THE DATABASE FROM THE PRUNED LOG DATA SET
//STEP1    EXEC PGM=BLGUT3,PARM='SESS=01,NAME=5',REGION=4096K
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//SYSOUT   DD   SYSOUT=A
//BLGBKUP  DD   DISP=SHR,DSNAME=LOG139,UNIT=TAPE,LABEL=(,SL),
//              VOL=SER=001234
```

Figure 8. Restore a database using the offloaded log data on a 9-track tape volume.

```
//RELOAD    JOB
//STEP1     EXEC PGM=BLGUT3,PARM='SESS=01,NAME=5,TYPE=ADD',REGION=4096K
//STEPLIB   DD   DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//SYSOUT    DD   SYSOUT=A
//BLGBKUP   DD   DISP=SHR,DSNAME=LOG139
```

Figure 9. Load SRCs from a previously offloaded (BLGUT4) test system to your production system.

16

BLGUT4—Offload the Recovery Log Data Set

This section describes the steps necessary to recover data with the BLGUT4 utility, and provides details on how to use the BLGUT4 utility. For a general discussion of database recovery options, including the BLGUT23 utilities which also provide a backup/recovery solution, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Preparing to Recover Data

With Tivoli Information Management for z/OS, it is possible to recover your data without ever needing to follow all of the steps described in the following section. The Automatic Log Save facility can be integrated into your data recovery process. To understand how the facility works and how you can use it to your best advantage, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Steps to Take Before Recovery is Necessary

Database recovery requires the following prerequisite tasks:

Defining session-parameters members

Specify an SDLDS cluster for all read/write databases when you construct your session-parameters members. All subsequent Tivoli Information Management for z/OS sessions enable users to update the databases only when a recovery log data set (SDLDS) is available to record the transactions.

Define SDLDS clusters with no secondary extents. If you use secondary extents, all secondary extents are allocated when running BLGUTR, because this utility formats the entire data set.

Initializing the recovery log

Format a recovery log (SDLDS) using BLGUTR when you initialize the databases. During normal system processing, Tivoli Information Management for z/OS writes all database updates to that database's SDLDS. When a database is damaged, use the SDLDS to restore all updates made since the last backup for that database.

Offloading the recovery log

Use BLGUT4 to offload the SDLDS to a sequential backup data set (offloaded log) as the SDLDS becomes full. Delete all offloaded recovery logs whenever you back up the database to which the offloaded log relates.

Migrating Databases

To provide a starting point for potential future recovery, you should keep several generations (backups) migrated to backup tapes or data sets. Only the database backups are needed. You do not need to migrate offloaded logs since the database backups contain the same information as the offloaded logs.

Recovering Data Sets

Depending on whether damage occurs to the SDIDS, the SDDS, or both data sets for a database, perform the following procedure to recover those databases.

When only the SDIDS is damaged, recover it by running the database index rebuild utility, BLGUT1:

- Delete the damaged SDIDS
- Allocate a new SDIDS
- Run BLGUT1 against the undamaged SDDS

When the SDDS is damaged, recovery includes the following procedure, illustrated in Figure 10 on page 75.

1. Restore the SDDS for that database from the most recent backup, using standard AMS (IDCAMS).

   Similarly, restore the matching SDIDS for that database from the same backup using IDCAMS. If the matching SDIDS is not available, use BLGUT1 to rebuild the SDIDS. If you are using multiple SDDS clusters, restore all of these clusters to the same level before you run BLGUT1.
2. Use BLGUT4 to offload the current SDLDS to capture all changes entered for that database since the most recent backup.
3. Use the offloaded recovery log as input for the recovery onload utility, BLGUT3, to update the restored SDDS and SDIDS.

Back up a restored database immediately. BLGUT4 automatically reinitializes the SDLDS after offloading it.

Note: Do not use BLGUTR to perform this reinitialization.

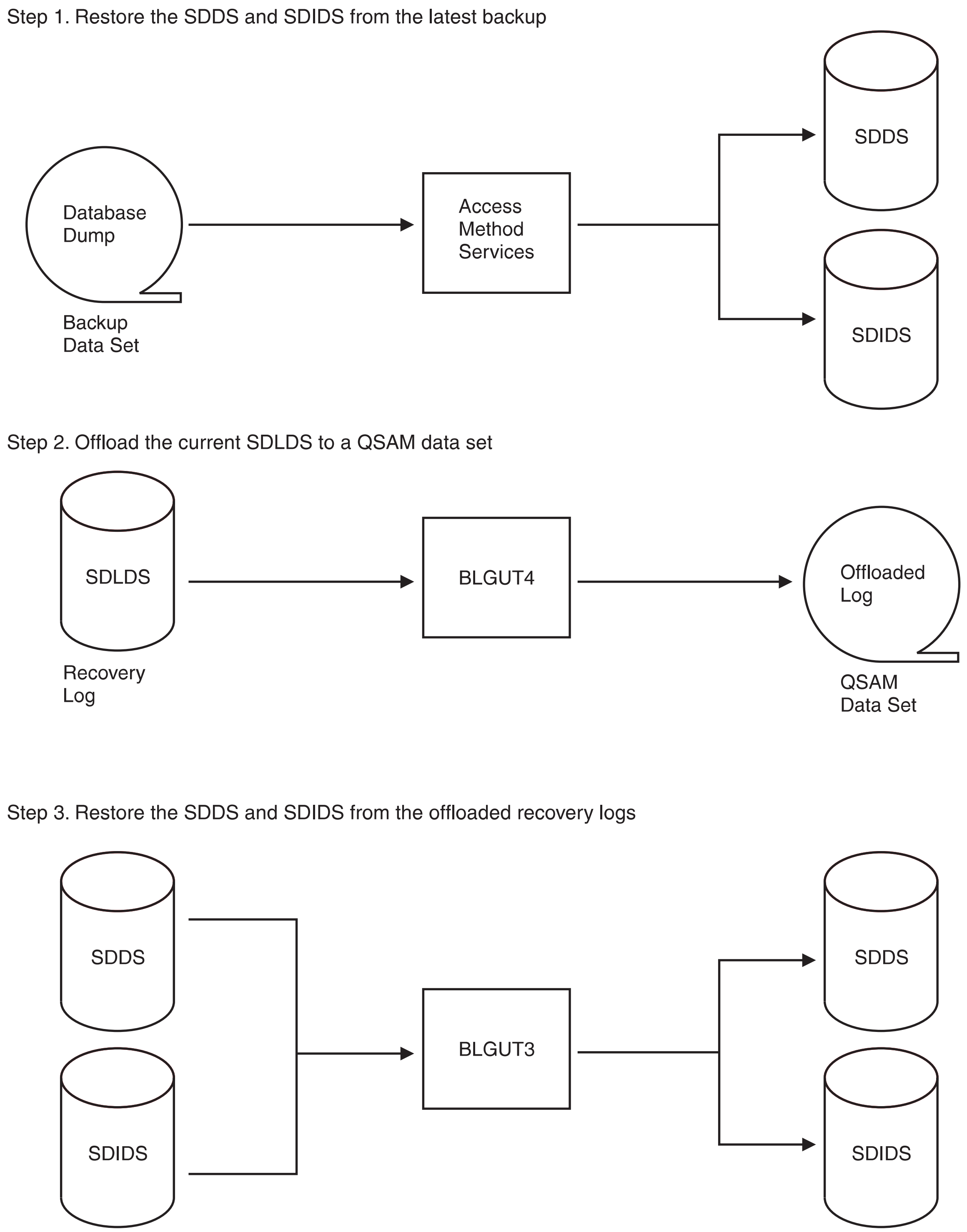

Step 1. Restore the SDDS and SDIDS from the latest backup
Database Dump
Backup Data Set
Access Method Services
SDDS
SDIDS
Step 2. Offload the current SDLDS to a QSAM data set
SDLDS
Recovery Log
BLGUT4
Offloaded Log
QSAM Data Set
Step 3. Restore the SDDS and SDIDS from the offloaded recovery logs
SDDS
SDIDS
BLGUT3
SDDS
SDIDS

Figure 10. Recovering a Damaged Database

BLGUT4 Description

Attention: If you have implemented the Automatic Log Save Facility or the DB2® Extract Facility into your data recovery process, do not run the BLGUT4 utility until you fully understand how it may affect your data. Doing so can result in lost data. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about the Automatic Log Save and DB2 Extract facilities and backup and recovery processes.

Note: If you have a need to back up the SDDS without having to force users off the system and stop the BLX-SP, you should consider using the BLGUT23 utilities instead.

Use BLGUT4 to offload data from a recovery log data set (SDLDS) to a sequential backup data set on disk or tape. If a database must be recovered, the backup data set produced by BLGUT4 is used as input to the recovery utility (BLGUT3). The SDLDS must be large enough to hold all updates made to a database between periodic database backups, or offloading the SDLDS is necessary. A database backup created using IDCAMS REPRO makes any data in the SDLDS obsolete, or any offloads made using BLGUT4, so the SDLDS must be emptied using BLGUT4 when the backup is taken and the output data discarded. If you must recover the SDDS, use BLGUT4 to offload the current SDLDS for the database so that the data is in the required input format for the recovery utility, BLGUT3.

The job that invokes BLGUT4 consists of two steps. The first step, OFFSTEP, offloads data from the SDLDS to a backup data set. This data set is an intermediate data set that can be deleted after the job finishes. The second step, PRNSTEP, reads the ″offloaded″ data set created by OFFSTEP and writes only the last update for a given record to a new backup data set. The “pruned” backup data set created by PRNSTEP is the data set to use as input when BLGUT3 is run. Although the pruned backup data set has fewer records than the SDLDS, the data set contains all the information needed to restore the database. Also, by using the pruned backup data set as input to BLGUT3, the performance of BLGUT3 should improve because BLGUT3 has fewer records to process.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Syntax

The EXEC statement PARM field keywords to use with BLGUT4 are as follows:

```
PARM=‘SESS=aa,NAME=5[,RECS=nn][,%FULL=xxx]’
```

SESS=aa
Suffix for the session-parameters member name that BLGUT4 uses. Use 1 or 2 alphanumeric or national characters for *aa*.

NAME=5
Name of the database whose SDLDS you want to offload. Use 5 for the Tivoli Information Management for z/OS database. Although you can specify databases 0, 1, 2, 3, 4, 6, 7, 8, or 9, do not do so because they are read-only databases.

RECS=nn

Specifies the number of physical records processed between each release of the SDLDS. *nn* must be in the range 0 ≤ nn ≤99 999 999. A value of 0 indicates that the SDLDS is not released. The default value is 0.

By specifying a value for RECS, you can offload the SDLDS while other users continue to update and create records stored in the Tivoli Information Management for z/OS database. There is a performance trade-off when using this parameter. Specifying a value between 1 and the maximum size of the SDLDS increases the amount of time the offload takes, but users waiting to access the SDLDS proceed more quickly. A smaller value of RECS causes a smaller performance impact on users, but it also causes a longer delay to the offload. A larger value gives a quicker offload, but causes a greater delay to users. Values between 15 and 100 provide a reasonable range of benefits. It is suggested that your initial value be set to 20, and adjusted from there.

%FULL=xxx

Specifies that if the SDLDS becomes more than *xxx* percent full during offload processing, then processing continues without releasing the SDLDS until the offload is complete. The range of values for *xxx* is 0 to 100. The default value is 80. If the RECS parameter is omitted or set to 0, the %FULL parameter is ignored.

Specifying this parameter helps to prevent the SDLDS from becoming full before offload processing is completed.

Input

The input to the first step of the BLGUT4 job, OFFSTEP, is the current SDLDS of a database. The SDLDS is a VSAM relative record data set. The input to the second step of the BLGUT4 job, PRNSTEP, is the offloaded data set created by OFFSTEP plus any pruned backup data set created since the most recent backup for the database. The pruned data set must be concatenated in front of the offloaded data set in the BLGBKIN DD statement. After the new pruned data set is created, the old pruned data set used as input should be deleted. The offloaded data set should be deleted also.

Output

The output from BLGUT4 is:

- The SDLDS is reinitialized (emptied) if BLGUT4 completes with return code 0. Any other return code indicates that the SDLDS was not offloaded.
- A new, physical sequential data set containing offloaded log data. This is the pruned backup data set that is used as input for BLGUT3 or for additional BLGUT4LP runs between full database backups.
- The BLGBKTM output data set, which can be saved and used as input to the BLGUT23P utility.
- Informational and error messages in a message data set.
- A return code of `0` if BLGUT4 finishes successfully.
- A return code `>0` if an error occurs.

Restrictions

The following restrictions apply to offloading the SDLDS:

- The record size of the backup data sets (BLGBKTM DD, BLGBKIN DD, and BLGBKUP DD statements) is variable length and equal to the maximum record size of the SDLDS cluster plus 17. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for a discussion of the record size.

- When you specify a DUMMY data set on the BLGBKTM DD, BLGBKIN DD, or BLGBKUP DD statements, you must include the DCB parameter describing a sequential data set with a valid record size and format.
- When sysplex mode is not enabled, the BLXDSN statement for the SDLDS in the VSAM Resource Definition Member must be defined with DSORG=NUM. For instructions on how to define the DSORG=NUM, refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual.
- In the job that invokes BLGUT4, the second step, called PRNSTEP, uses MVS data spaces to create the pruned backup data set. Before running PRNSTEP, you may need to update the MVS installation exit IEFUSI to increase the limit on the size and number of data spaces allowed. Refer to *OS/390 MVS Installation Exits* for more information on IEFUSI.
- Use of this utility might conflict with the function of the Automatic Log Save or DB2 Extract facilities. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information.

DD Statements

The DD statements required to run BLGUT4 are as follows:

BLGBKIN
The offloaded backup data set created by the first step of the BLGUT4 job plus any pruned backup data set created since the most recent backup for the database. The pruned data set must be concatenated in front of the offloaded data set in this DD statement. The pruned data set, if any, included in this DD statement should be deleted after the BLGUT4 job finishes.

BLGBKTM
The offloaded backup data set. This is a new physical sequential data set created by the first step of the BLGUT4 job. This data set can be deleted after each BLGUT4LP job finishes with return code 0. This data set can also be saved and used as input to the BLGUT23P utility.

BLGBKUP
The pruned backup data set. This is a new physical sequential data set created by the second step of the BLGUT4 job. This data set will be the input to BLGUT3 if the database needs to be restored, or used as input (BLGBKIN) to additional BLGUT4LP jobs between backups.

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenarios Using the BLGUT4 Utility Program

The following examples illustrate how you can use BLGUT4 to offload an SDLDS. The current SDLDS is a VSAM data set connected with the Tivoli Information Management for z/OS database.

```
//OFFLOAD  JOB
//*       CREATE THE OFFLOADED LOG DATA SET
//OFFSTEP  EXEC PGM=BLGUT4,PARM='SESS=01,NAME=5',REGION=2048K
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//*       OFFLOADED LOG DATA SET
//BLGBKTM  DD   DISP=(NEW,CATLG),DSN=LOGTEMP,
//     UNIT=SYSDA,SPACE=(TRK,(010,010)),
//     DCB=(RECFM=VB,LRECL=8202,BLKSIZE=8206)
//*       CREATE A NEW PRUNED LOG DATA SET
//PRNSTEP  EXEC PGM=BLGUT4LP,REGION=2048K,COND=(0,NE,OFFSTEP)
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//*       OFFLOADED LOG DATA SET
//*       IF A PRUNED LOG DATA SET EXISTS, CONCATENATE IT
//*       IN FRONT OF *.OFFSTEP.BLGBKTM
//BLGBKIN  DD   DISP=SHR,DSN=*.OFFSTEP.BLGBKTM
//*       NEW PRUNED LOG DATA SET
//BLGBKUP  DD   DISP=(NEW,KEEP),DSN=LOG139,UNIT=TAPE,LABEL=(,SL),
//     VOL=SER=001234,
//     DCB=(RECFM=VB,LRECL=8202,BLKSIZE=8206)
```

Figure 11. Offloading log data to a new data set on a 9-track tape volume.

```
//OFFLOAD  JOB
//*       CREATE THE OFFLOADED LOG DATA SET
//OFFSTEP  EXEC PGM=BLGUT4,PARM='SESS=01,NAME=5',REGION=2048K
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//*       OFFLOADED LOG DATA SET
//BLGBKTM  DD   DISP=OLD,DSN=LOGTEMP
//*       CREATE A NEW PRUNED LOG DATA SET
//PRNSTEP  EXEC PGM=BLGUT4LP,REGION=2048K,COND=(0,NE,OFFSTEP)
//STEPLIB  DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//*       OFFLOADED LOG DATA SET
//*       IF A PRUNED LOG DATA SET EXISTS, CONCATENATE IT
//*       IN FRONT OF *.OFFSTEP.BLGBKTM
//BLGBKIN  DD   DISP=SHR,DSN=LOG139
//         DD   DISP=SHR,DSN=*.OFFSTEP.BLGBKTM
//*       NEW PRUNED LOG DATA SET
//BLGBKUP  DD   DISP=OLD,DSN=LOGNEW
```

Figure 12. Offloading log data into an existing backup data set.

17

BLGUT5—Load and Maintain the Dictionary Data Set

Use BLGUT5 to load and maintain the Tivoli Information Management for z/OS dictionary data set.

Important!

| The BLX-SP must be started before you can run Tivoli Information Management for
| z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See "FREE Command" on page 15 and "REALLOC Command" on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Input

The input to BLGUT5 is a member in a PDS. The input to BLGUT5 cannot be a concatenation of data sets. This data set contains the dictionary s-words and p-words for the product being installed or serviced. For service, it contains a total replacement of the dictionary elements for the product being serviced. Therefore, any modifications you made to Tivoli-supplied s-words or p-words (dictionary entries below X'8000') are overwritten. If you added p-words and s-words to your original dictionary and you defined a new dictionary, use IDCAMS REPRO or BLGUT5F to put the user-defined portions (dictionary entries above X'8000') of your current dictionary into the new dictionary.

Select user-defined p-word records using REPRO keywords:

```
FROMKEY (X'D78000')
TOKEY (X'D7FFFF')
```

Select user-defined s-word records using REPRO keywords:

```
FROMKEY (X'E28000')
TOKEY (X'E2FFFF')
```

Output

The output from BLGUT5 is:

- The dictionary data set, DICTDS. The DICTDS is a key-sequenced VSAM cluster
- Informational and error messages in a message data set
- A return code of `0` when the utility finishes successfully
- A return code of `>0` if an error occurs

Restrictions

The following restrictions apply to installing and maintaining the DICTDS:

- Allocate space on a direct access device for the DICTDS (BLGDICT statement).
- Allocate a new DICTDS before you run BLGUT5 for the first time, refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual for more information.
- Allocate the DICTDS (BLGDICT statement) as DISP=SHR.
- When you add a user entry in the dictionary, then run BLGUT5F to copy the dictionary to a sequential data set, and then change the index to one in the range of Tivoli-reserved entries, your attempt fails. BLGUT5 prevents that entry from being put into the Tivoli part of the dictionary (lower than X’8000’). Your user-defined index entry is in the set higher than X’8000’.

DD Statements

The DD statements required to run BLGUT5 are as follows:

BLGDICT
The output VSAM DICTDS.

BLGSWDS
The input PDS member.

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenario Using the BLGUT5 Utility Program

The following example illustrates the use of the BLGUT5 utility to load or maintain the DICTDS. To get the sample JCL, refer to member BLGLDICT or BLGUT5J in the SBLMSAMP library.

```
//DICT      JOB
//BLG       EXEC PGM=BLGUT5,REGION=2048K
//STEPLIB   DD   DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//BLGSWDS   DD   DISP=SHR,DSN=BLM.SBLMDICT(BLMVDICT)
//BLGDICT   DD   DISP=SHR,DSN=BLM.DICT
```

Figure 13. Using the BLGUT5 Utility.

18

BLGUT5F—Offload the Dictionary Data Set

Use BLGUT5F to create your own enterprise-level copy of a Tivoli Information Management for z/OS dictionary data set. The utility copies a VSAM dictionary to a sequential data set. All dictionary entries are copied—Tivoli s-words or p-words (dictionary entries below X'8000') and user s-words or p-words (dictionary entries above X'7FFF'). You can then use BLGUT5 to copy this sequential data set to a VSAM dictionary data set.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Input

The input for BLGUT5F is a key-sequenced VSAM dictionary data set.

Output

The output from BLGUT5F is:

- A sequential data set containing a copy of the input VSAM dictionary data set. This copy contains IBM and user dictionary entries.
- Information and error messages in a message data set.
- A return code of `0` when BLGUT5F finishes successfully.
- A return code of `>0` if an error occurs.

Restrictions

When you add a user entry in the dictionary, then you run BLGUT5F to copy the dictionary to a sequential data set, and then you change the index to one in the range of Tivoli-reserved entries, your attempt fails. BLGUT5 prevents that entry from being put into the Tivoli part of the dictionary (lower than X'8000'). Your user-defined index entry is in the set higher than X'8000'.

DD Statements

The DD statements required to run BLGUT5F are as follows:

BLGDICT
The input VSAM dictionary.

BLGSWDS
The output sequential data set (LRECL=80,RECFM=FB).

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenario Using the BLGUT5F Utility Program

The following example illustrates how to use BLGUT5F to offload a VSAM dictionary to a sequential data set.

```
//BLG       EXEC PGM=BLGUT5F,REGION=2048K
//STEPLIB   DD DSN=BLM.SBLMMOD1
//SYSPRINT  DD SYSOUT=*
//BLGDICT   DD DISP=SHR,DSN=BLM.DICT
//BLGSWDS   DD DISP=SHR,DSN=BLM.BLGDICT.SEQ
```

19

BLGUT6—Install and Maintain a Panel Data Set

BLGUT6 replaces or adds panels to the VSAM panel data set it is updating. A panel in the input partitioned data set (PDS) replaces any existing panel with the same name in the VSAM panel data set. A panel in the input PDS with a name that is not already in the VSAM panel data set is added to the VSAM data set. The rules of the INCLUDE and EXCLUDE parameters are followed as well.

BLGUT6 resets the **Panel Modified** field to NO for all panels it adds or replaces.

Another utility program, BLGUT6M, is very similar to BLGUT6 but it offers the additional function of allowing you to modify the field lengths and validation patterns used by the panels as they are being loaded to the panel data set. If you want to convert date fields on the shipped panels from 10-character to 8-character or some other date format, see “BLGUT6M—Migrate Panel Fields” on page 95 for more information.

Important!

| The BLX-SP must be started before you can run Tivoli Information Management for
| z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The first record in the input sequential data set contains control keywords. The second through last records contain a list of panel names. The control keywords are not order dependent; however, they must appear within the first 72 bytes of data in the first record because BLGUT6 ignores bytes 73–80. Separate keywords by one or more blanks or commas. If you want a comment in this first record, begin your comment with the characters /*. There is no ending comment delimiter; hence your comment must appear in the first record only.

The control keyword line for BLGUT6 is as follows:

```
[INCLUDE|EXCLUDE][NODBCS|DBCS][AETYPE(M|S|D,BYP|REP)][UPPERCASE]
```

INCLUDE

Indicates that only those panels you identify immediately following will be transferred from the PDS to the VSAM panel data set. Give a list of panel names if you specify the INCLUDE keyword in the first record. The second through the last records in the input data set contain the list of panel names. Panel names must appear within the first 72 bytes of data in the record because BLGUT6 ignores bytes 73–80.

You can specify any number of panel names in each record. Separate panel names by one or more blanks or commas. Ensure that you place each entire panel name within a record. For example, do not split a panel name across two records (that is, lines). Do not include comments in the panel name list (with the exception of bytes 73–80 of each record).

Use uppercase or lowercase characters, or both, when you specify the panel names and control keywords for BLGUT6; the control information and panel names are converted to uppercase before being processed by BLGUT6. Do not confuse this conversion with the one requested by the UPPERCASE keyword, which applies to the words in a panel and not to the name of the panel.

The INCLUDE and EXCLUDE keywords are mutually exclusive. If you do not specify either keyword, all panels that are in the BLGPDS data set are transferred to the VSAM panel data set.

EXCLUDE

Indicates that all panels identified in the PDS's directory, with the exception of those panels identified in the panel list you create, are transferred from the PDS to the VSAM panel data set. Give a list of panel names if you specify the EXCLUDE keyword in the first record. The second through the last records in the input data set contain a list of panel names. Panel names must appear within the first 72 bytes of data in the record because BLGUT6 ignores bytes 73–80.

You can specify any number of panel names in each record. Separate panel names by one or more blanks or commas. Ensure that you place each entire panel name within a record. For example, do not split a panel name across two records (that is, lines). Do not include comments in the panel name list (with the exception of bytes 73–80 of each record).

Use uppercase or lowercase characters, or both, when you specify the panel names and control keywords for BLGUT6; the control information and panel names are converted to uppercase before being processed by BLGUT6. Do not confuse this conversion with the one requested by the UPPERCASE keyword, which applies to the words in a panel and not to the name of the panel.

The INCLUDE and EXCLUDE keywords are mutually exclusive. If you do not specify either keyword, all panels that are in the BLGPDS data set are transferred to the VSAM panel data set.

NODBCS
Indicates you are loading panels that do not contain DBCS data.

Load Tivoli Information Management for z/OS panels with the NODBCS keyword included if you do not use DBCS characters on your panels. Panels with DBCS characters must use the DBCS keyword.

The NODBCS and DBCS keywords are mutually exclusive. If you do not specify a keyword then NODBCS is the default.

DBCS
Indicates you are loading panels that contain DBCS data. X'0F' attribute bytes are converted to X'0B' attribute bytes.

The NODBCS and DBCS keywords are mutually exclusive. If you do not specify a keyword, then NODBCS is the default.

AETYPE
Indicates the data type that you want in your assisted-entry panels.

M Specifies mixed data that includes any combination of SBCS and DBCS strings. M is the default.

S Specifies SBCS-only data.

D Specifies DBCS-only data.

BYP
Specifies that any assisted-entry panels being loaded that already contain a data type are not changed (that is, they retain their current data type setting). BYP is the default.

REP
Specifies that any assisted-entry panel being loaded is assigned the specified data type.

Regardless of the option specified, any assisted-entry panel processed that is not already migrated to use the data type is automatically migrated and assigned the specified data type. If a data type is not specified, the default M is used.

If you do not specify AETYPE the defaults used are M and BYP.

UPPERCASE
Indicates that all panels when written to the VSAM panel data set must have their externals converted to uppercase characters. Thus, the panels are displayed in uppercase at a user’s terminal. If you do not specify this keyword, then BLGUT6 leaves lowercase alphabetic characters unchanged in the loaded panels.

Do not use UPPERCASE with any panels that have DBCS data because double-byte characters can be misinterpreted. If you install the non-Latin tables, then you must run BLGUT6 with the UPPERCASE keyword when you load the Tivoli-supplied panels.

Note: All panels contained in the BLGPDS data set are transferred to the VSAM panel data set in the following cases:

- If you do not specify the optional SYSIN DD statement
- If the SYSIN DD statement points to a null data set

- If the SYSIN DD statement points to a data set that does not contain the INCLUDE or EXCLUDE keywords

Input

The input to BLGUT6 is as follows:

- A PDS that includes the panels for the product being installed or serviced. Each panel is a separate member in the PDS. For service, the input PDS contains just those panels being changed with the service for the product. Input to BLGUT6 cannot be a concatenation of data sets.
- An optional sequential SYSIN data set that contains operating information indicating how BLGUT6 is to run.

Output

The output from BLGUT6 is:

- The panel data set that is a key-sequenced VSAM cluster
- Informational and error messages in a message data set
- A return code of 0 when BLGUT6 finishes successfully
- A return code >0 if an error occurs

Restrictions

The following restrictions apply to installing and maintaining a VSAMpanel data set:

- Space must be allocated on a direct access device for the panel data set (BLGPNLS statement).
- A new VSAM panel data set must be allocated as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual before you run BLGUT6 for the first time.

DD Statements

The DD statements required to run BLGUT6 are as follows:

BLGPDS
The input PDS.

BLGPNLS
The output VSAM panel data set.

SYSIN
Input control statements for BLGUT6 (LRECL=80,RECFM=FB).

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenario Using the BLGUT6 Utility Program

The following sample illustrates use of BLGUT6 to initialize or maintain a panel data set by loading all panels contained in the input panel data set. To get the sample JCL, see member BLGLRPNL in the SBLMSAMP library.

```
//LOADPNL  JOB
//*********************************************************************
//* THE FOLLOWING SAMPLE JCL LOADS THE READ
//* PANEL DATA SET.
//* YOU MUST CHANGE THE DATA SET NAMES FOR DD NAMES
//* STEPLIB, BLGPDS AND BLGPNLS TO CORRESPOND TO THE DATA SET NAMES
//* AT YOUR INSTALLATION.
//*
//*  UPDATE THE FOLLOWING WITH UPPERCASE TEXT:
//*
//*  1) JOB CARD INFORMATION//*
//*  2) DATA SET NAMES FOR DD NAMES STEPLIB, BLGPDS AND BLGPNLS TO
//*     CORRESPOND TO THE DATA SET NAMES AT YOUR INSTALLATION
//*
//*     DD NAME BLGPDS IS THE INPUT PDS - THE TARGET DATA SET THAT
//*        YOU DEFINED FOR THE PANELS DURING SMP/E INSTALLATION.
//*     DD NAME BLGPNLS IS THE VSAM READ
//*        PANEL DATA SET
//*
//*********************************************************************
//LOAD     EXEC PGM=BLGUT6,REGION=2048K
//STEPLIB  DD DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=*
//BLGPDS   DD DISP=SHR,DSNAME=BLM.SBLMPNLS
//BLGPNLS  DD DISP=SHR,DSNAME=BLM.IBMPNLS
//*--------------------------------------------------------------------
//*NOTE: FOR ENGLISH PANELS, THE "SYSIN" DD IS OPTIONAL.
//* THE DEFAULT KEYWORDS - NODBCS AETYPE(M,BYP) - WILL CORRECTLY
//* MIGRATE PANELS TO VERSION 6 FORMAT AND SET ASSISTED-ENTRY PANELS TO
//* ALLOW EITHER SINGLE-BYTE OR DOUBLE-BYTE CHARACTERS.
//* DO NOT SPECIFY THE "DBCS" KEYWORD FOR ENGLISH PANELS.
//*
//*NOTE: FOR JAPANESE PANELS, THE "SYSIN" DD MUST BE CODED
//* AND THE "DBCS" KEYWORD MUST BE SPECIFIED (INSTEAD OF "NODBCS") TO
//* AVOID CORRUPTION OF PANELS.  THE "AETYPE" KEYWORD IS OPTIONAL.  IF
//* NOT SPECIFIED, ASSISTED-ENTRY PANELS WHICH DO NOT HAVE A DATA TYPE
//* WILL BE SET TO ACCEPT MIXED SBCS AND DBCS DATA.
//*--------------------------------------------------------------------
//SYSIN    DD *
  NODBCS AETYPE(M,BYP)
/*
```

Performance measurements for BLGUT6 are given in the tables in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual.

20

BLGUT6F—Offload a Panel Data Set

Use BLGUT6F to copy panels from a Tivoli Information Management for z/OS VSAM panel data set to a partitioned data set (PDS). Because panels are distributed in PDS form, you can create a new or enterprise-level PDS distribution data set. When you want to use an application program interface (API) and you have changed or created panels that refer to fields you want to access using the API, copy these panels to a PDS before creating your API data and pattern tables using the BLGUT8 utility.

After you copy the panels into the PDS, they are not in a displayable form; they are in the distribution format. You must use BLGUT6 to copy the panels from the PDS to a VSAM panel data set to put them in a displayable form.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Input

The input for BLGUT6F is:

- A key-sequenced VSAM panel data set. This data set has the panels you can copy to the PDS.
- An optional SYSIN sequential data set. This data set has a list of 8-character panel names to copy from the VSAM panel data set. It can also contain 1- to 7-character partial panel names. You can copy a group of panels that have names beginning with the same characters by specifying a partial name as the input. The names must begin in column 1, and only one name per record is valid. When you do not specify a SYSIN DD statement, all panels in the VSAM panel data set are written to the specified PDS.

Output

The output from BLGUT6F is:

- A PDS containing its previous contents plus any panels copied from the VSAM panel data set. If the name of the panel to be copied already exists, the original panel is replaced by the newly copied panel.
- Information and error messages in a message data set.
- A return code of 0 when BLGUT6F finishes successfully.
- A return code of 4 when one or more panels listed in the SYSIN data set specified are not in the VSAM panel data set.
- A return code of 8 when data set access errors (open, read, write, or close errors) occur for any of the data sets involved.

Restrictions

If the data set referenced by the BLGPDS statement is allocated to another job or user, the utility fails.

DD Statements

The DD statements required to run BLGUT6F are as follows:

BLGPDS
The output PDS (LRECL=80,RECFM=FB)

BLGPNLS
The input VSAM panel data set

SYSIN
Optional input sequential data set. Do not specify this DD statement if you want to copy all of the panels from the VSAM panel data set.

For information on creating this SYSIN sequential data set, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* manual.

SYSPRINT
The sequential output message data set you can write to a system output device or data set.

Scenario Using the BLGUT6F Utility Program

The following sample illustrates how you can use BLGUT6F to copy panels from a VSAM panel data set to a PDS. To obtain the sample JCL, refer to member BLGUT6FJ in the SBLMSAMP library.

```
//OFFLOAD  JOB
//*-------------------------------------------------------------------*
//*                                                                   *
//*            Licensed Materials - Property of IBM                   *
//*                                                                   *
//*                              5648-142                             *
//*                                                                   *
//*   (C) Copyright Tivoli Systems Inc., an IBM Company, 1997, 1999.  *
//*                                                                   *
//*                      See Copyright Instructions                   *
//*                                                                   *
//*-------------------------------------------------------------------*
//*--------------------------------------------------------------
//* THIS IS A SAMPLE JOBSTREAM FOR RUNNING BLGUT6F TO OFFLOAD PANELS
//* FROM A VSAM PANEL DATA SET TO A PDS.  YOU MUST CHANGE THE
//* DATA SET NAMES FOR DD NAMES STEPLIB, BLGPNLS, AND BLGPDS TO
//* CORRESPOND TO THE DATA SET NAMES AT YOUR INSTALLATION.
//*
//* DD NAME BLGPNLS IS THE NAME OF THE VSAM PANEL DATA SET.
//* DD NAME BLGPDS IS THE NAME OF THE PDS TO CONTAIN THE OFFLOADED
//*    PANELS.
//* DD NAME SYSIN IS OPTIONAL.  IF YOU OMIT IT, ALL OF THE PANELS IN
//*    BLGPNLS WILL BE COPIED TO BLGPDS.
//*
//* IN THIS EXAMPLE, ONLY PANELS BLG0B100 AND BLG0EN20 WILL BE COPIED
//* FROM THE VSAM PANEL DATA SET TO THE PDS.
//*--------------------------------------------------------------
//STEP1    EXEC  PGM=BLGUT6F,REGION=2048K
//STEPLIB  DD   DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT DD   SYSOUT=A
//BLGPNLS  DD   DISP=SHR,DSNAME=BLM.SBLMPNLS
//BLGPDS   DD   DISP=OLD,DSNAME=BLM.PANELS.PDS
//SYSIN    DD   *
BLG0B100
BLG0EN20
/*
//
```

21

BLGUT6M—Migrate Panel Fields

BLGUT6M modifies panels as they are replaced or added to the VSAM panel data set it is updating. It is used to make changes to panels you may have customized or tailored for your own site. Two types of modifications are supported with use of this utility:

- Changing the field lengths on data entry panels and table panels
- Changing the validation patterns on assisted-entry panels to new values

BLGUT6M is very similar to BLGUT6, but it offers the additional function of allowing you to modify the field lengths and validation patterns on your customized panels. You may find BLGUT6M useful to:

- Change date-related field lengths on your customized panels to get ready for the Year 2000 (for example, to change the length of a MM/DD/YY field from 8 to 10 characters to start showing dates with 4-digit years as MM/DD/YYYY)
- Lengthen or shorten other fields that may change from time to time in your organization, such as department number fields (after a reorganization) or perhaps program name fields (after new application systems have been installed)

BLGUT6M can help to reduce the amount of time required to make these types of changes to your panels. You can use the utility with the Panel Modification Facility to make changes, or in many cases, use it as a replacement for PMF.

Note: If you currently use the base product panels and want your users to see date fields expressed in 4-digit years, you can load the Tivoli Information Management for z/OS panel data set to get 10-character date fields and NN</>NN</>NNNN validation patterns without having to run the BLGUT6M migration utility. The BLGUT6M utility is intended for panels you have customized, or the base product panels assuming you need to change something else (such as the date length to be something other than 10 characters, or to set up a different validation pattern). Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for details on loading the BLM.SBLMPNLS data set.

Field lengths on data entry panels and table panels that are associated with a certain set of s-words can be changed with BLGUT6M if the new lengths do not overlay anything other than blank spaces on the panels. Through use of the LENGTH keyword, you can control whether or not a field should be expanded as far as possible or left unchanged if the extension encounters a non-blank space on the panel. For example, if you have asked to expand an 8-character field to 10 characters, but there is room for only 9, Tivoli Information Management for z/OS will expand the field to 9 if the partial expansion option is specified.

A panel in the input partitioned data set (PDS) replaces any existing panel with the same name in the VSAM panel data set. A panel in the input PDS with a name that is not already in the VSAM panel data set is added to the VSAM data set. The rules of the INCLUDE and EXCLUDE parameters are followed as well.

To identify the panels it has changed, BLGUT6M resets the **Panel Modified** field to YES. It also sets the **Userid last altered** to BLGUT6M, and the **Date last altered** and **Time last altered** to the date/time when the panel is migrated. Setting the fields this way enables you to generate PMF reports to print a list (or a content report or some other PMF report) containing only those panels modified by BLGUT6M.

> **Important!**
>
> The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.
>
> When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.
>
> When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The first record in the input sequential data set must be the keyword MODIFY. It is followed by one or more LENGTH and VALIDATION records. The LENGTH and VALIDATION records can be in any order, but each LENGTH or VALIDATION statement must be on a separate line.

After all LENGTH and VALIDATION records, you may optionally specify an options record (INCLUDE, EXCLUDE, and so on, just like BLGUT6) and a list of panel names.

Rules for where keywords need to appear in the data are the same as those for the BLGUT6 utility.

The control keyword lines for BLGUT6M are as follows:

MODIFY [LOAD={MOD | ALL}]

LENGTH index,newlen[,PARTIAL]

VALIDATION Poooo,Pnnnn

[INCLUDE|EXCLUDE][NODBCS|DBCS][AETYPE(M|S|D,BYP|REP)][UPPERCASE]

MODIFY

Indicates that you want to modify either field lengths or validation patterns or both field lengths and validation patterns on the panels you specify. This keyword is required if you specify any LENGTH or VALIDATION records.

You can specify whether or not you want to load only the panels that require changes, or to load all panels into a VSAM data set.

LOAD=MOD

Loads only the panels that need to be modified into the VSAM data set. This includes panels that could not be modified due to insufficient expansion space. You might choose this option if you want to overlay an existing VSAM panel data set with the modified panels, so that only the panels with changes are modified (and the rest are left untouched). Or, if you need to isolate only those panels requiring modification, you can load them into an empty VSAM data set.

LOAD=ALL

Loads all panels into the VSAM data set, even if they contain no fields that need to be modified. You might choose this option if you do not need to isolate changed panels, or if you are migrating panels from a partitioned data set into an emtpy VSAM data set.

LENGTH

Specifies the s-word or p-word index value, new length of the field, and field expansion designation. This keyword is optional.

index

The s-word or p-word index value that identifies the field on the data entry or table panels which should have its length changed. This keyword is required to change a field length.

newlen

The new length of the field specified in digits (0 to 132). This keyword is required to change a field length.

PARTIAL

If specified, this keyword indicates that the field should be expanded as far as possible if expanding the field to the new length would overlay a non-blank character. If PARTIAL is not specified, the field will be left unchanged if it cannot be fully expanded.

VALIDATION

Indicates that a validation pattern should be modified on assisted-entry panels. This keyword is optional.

Poooo

The p-word index value of the current validation pattern. This keyword is required to modify a validation pattern.

Pnnnn

The p-word index value of the new validation pattern which should replace the old pattern in all assisted-entry panels. This keyword is required to modify a validation pattern.

INCLUDE

Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

EXCLUDE
Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

NODBCS
Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

DBCS
Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

AETYPE
Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

UPPERCASE

Same as for BLGUT6. Refer to “Syntax” on page 85 for details.

Input

The input to BLGUT6M is as follows:

- A PDS that includes the panels to be migrated. Input to BLGUT6M cannot be a concatenation of data sets.
- A sequential SYSIN data set that contains operating information indicating how BLGUT6M is to run.
- The VSAM dictionary data set (required) containing the new validation patterns. This data set is required whether or not validation patterns are being changed.

Output

The output from BLGUT6M is:

- The panel data set with revised field lengths and/or validation patterns. The panel data set is a key-sequenced VSAM cluster.
- Informational and error messages in a message data set.
- A return code of 0 when BLGUT6M finishes successfully.
- A return code of 4 when BLGUT6M finishes successfully, but was unable to fully expand one or more field lengths.
- A return code >4 if an error occurs.

Restrictions

The following restrictions apply to installing and maintaining a VSAM panel data set:

- Space must be allocated on a direct access device for the panel data set (BLGPNLS statement).
- The VSAM panel data set must be allocated as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual before you run BLGUT6M.
- There should be sufficient blank space following a field to expand it to the desired length. If the space is not sufficient a message is generated to warn you that the length of the field was not changed. If you specified PARTIAL, the field is expanded as much as possible; otherwise, it is left unchanged.
- This utility will not automatically update any existing report format tables, API applications, TSPs/TSXs, control panels, or help panels associated with the fields you modify. You may need to update these items to reflect the changes that were made by BLGUT6M.

DD Statements

The DD statements required to run BLGUT6M are as follows:

BLGDICT
The input VSAM dictionary data set.

BLGPDS
The input PDS.

BLGPNLS
The output VSAM panel data set.

SYSIN
Input control statements for BLGUT6M (LRECL=80,RECFM=FB).

SYSPRINT
A sequential message data set that you can write to a system output device or data set.

Scenarios Using the BLGUT6M Utility Program

The following scenarios illustrate use of the BLGUT6M utility to migrate field lengths and validation patterns on panels. The first scenario shows use of the BLGDATE8 sample JCL provided with Tivoli Information Management for z/OS to convert panels to use an 8-character date in the form NN/NN/NN (for example, external date format MM/DD/YY or DD/MM/YY). The panels are modified as they are loaded into the VSAM data set. The second scenario shows use of the BLGUT6MJ sample JCL, which you can use if you would like to convert your customized panels to use 10-character date fields in the format NN/NN/NNNN (external date format MM/DD/YYYY or DD/MM/YYYY). To get the sample JCL, see member BLGDATE8 or BLGUT6MJ in the SBLMSAMP library.

In the samples shown, the LENGTH keyword defines, for each unique piece of date data, the s-word for the field undergoing a length change on data entry panels. The VALIDATION keyword defines the new validation pattern that should replace the old validation pattern on assisted-entry panels.

Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for a discussion of how the BLGUT6M utility can be used as part of your overall date migration activities.

Once the BLGUT6M utility changes a panel, the panel is no longer considered a Tivoli panel; it is considered customized.

Convert Panels to Use 8-Character Dates (BLGDATE8)

The prefix index values shown as the second set of values on the VALIDATION keyword (the ″new″ validation patterns, for example P0801, P00CF) are provided in the Tivoli Information Management for z/OS dictionary data set and support the date validation pattern of NN</>NN</>NN (or external date format of MM/DD/YY or DD/MM/YY).

```
//PNLMIG   JOB
//*-------------------------------------------------------------------*
//*                                                                   *
//*            Licensed Materials - Property of Tivoli Systems        *
//*                                                                   *
//*                              5697-SD9                             *
//*                                                                   *
//*   (C) Copyright Tivoli Systems Inc., an IBM Company 1997, 2001.   *
//*                                                                   *
//*                      See Copyright Instructions                   *
//*                                                                   *
//*-------------------------------------------------------------------*
```

```
//*-----------------------------------------------------------------
//*  THIS IS A SAMPLE JOBSTREAM FOR RUNNING BLGUT6M TO MIGRATE FIELD
//*  LENGTHS AND VALIDATION PATTERNS ON PANELS.  YOU MUST CHANGE THE
//*  DATA SET NAMES FOR DD NAMES STEPLIB, BLGPDS, BLGPNLS, AND BLGDICT
//*  TO CORRESPOND TO THE DATA SET NAMES AT YOUR INSTALLATION.
//*
//*  DD NAME BLGPDS IS THE NAME OF THE INPUT PDS
//*  DD NAME BLGPNLS IS THE NAME OF THE OUTPUT VSAM PANEL DATA SET
//*  DD NAME BLGDICT IS THE NAME OF THE VSAM DICTIONARY DATA SET
//*
//*  IN THIS EXAMPLE, SINCE THERE IS NEITHER THE "INCLUDE" NOR
//*  "EXCLUDE" KEYWORD SPECIFIED, ALL PANELS IN THE INPUT PDS
//*  WILL BE PROCESSED.  SINCE THE MODIFY STATEMENT SPECIFIES NEITHER
//*  LOAD=MOD NOR LOAD=ALL, ONLY PANELS WHICH CONTAIN FIELDS SPECIFIED
//*  ON ONE OR MORE OF THE LENGTH AND VALIDATION STATEMENTS WILL BE
//*  LOADED INTO THE VSAM PANEL DATA SET.
//*
//*  THE SYSIN STATEMENTS IN THIS EXAMPLE, WHEN USED WITH THE BASE
//*  PRODUCT DICTIONARY ENTRIES, WILL CONVERT PANELS TO USE AN
//*  8-CHARACTER DATE IN THE FORM NN/NN/NN (E.G. MM/DD/YY, DD/MM/YY).
//*
//*-----------------------------------------------------------------
//MIGRATE  EXEC PGM=BLGUT6M,REGION=2048K
//STEPLIB  DD DSN=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=A
//BLGPDS   DD DISP=SHR,DSN=BLM.SBLMPNLS
//BLGPNLS  DD DISP=SHR,DSN=BLM.IBMPNLS
//BLGDICT  DD DISP=SHR,DSN=BLM.DICT
//*----------------------------------------------------------------------
//*
//*NOTE: FOR PANELS WHICH CONTAIN ONLY SINGLE BYTE CHARACTERS (E.G.
//* ENGLISH, FRENCH, GERMAN, SPANISH, ITALIAN) THE OPTION RECORD AT THE
//* END OF SYSIN IS OPTIONAL.
//* THE DEFAULT KEYWORDS - NODBCS AETYPE(M,BYP) - WILL CORRECTLY
//* MIGRATE PANELS TO THE CURRENT FORMAT AND SET ASSISTED-ENTRY PANELS
//* TO ALLOW EITHER SINGLE-BYTE OR DOUBLE-BYTE CHARACTERS.
//* DO NOT SPECIFY THE "DBCS" KEYWORD FOR ENGLISH PANELS.
//*
//*NOTE: FOR PANELS WHICH CONTAIN DOUBLE-BYTE CHARACTERS (E.G.
//* JAPANESE), THE OPTION RECORD IS REQUIRED
//* AND THE "DBCS" KEYWORD MUST BE SPECIFIED (INSTEAD OF "NODBCS") TO
//* AVOID CORRUPTION OF PANELS.  THE "AETYPE" KEYWORD IS OPTIONAL.  IF
//* NOT SPECIFIED, ASSISTED-ENTRY PANELS WHICH DO NOT HAVE A DATA TYPE
//* WILL BE SET TO ACCEPT MIXED SBCS AND DBCS DATA.
//*
//*----------------------------------------------------------------------
//SYSIN    DD *
  MODIFY
  LENGTH S0C34,8
  LENGTH S0C35,8
  LENGTH S0C36,8
  LENGTH S0C37,8
  LENGTH S0C38,8
  LENGTH S0C3D,8
  LENGTH S0C3E,8
  LENGTH S0C3F,8
  LENGTH S0C40,8
  LENGTH S0C41,8
  LENGTH S0C42,8
  LENGTH S0C43,8
  LENGTH S0C44,8
  LENGTH S0C45,8
  LENGTH S0C46,8
  LENGTH S0C47,8
  LENGTH S0C48,8
  LENGTH S0C49,8
  LENGTH S0C4A,8
  LENGTH S0C5F,8
  LENGTH S0CA2,8
  LENGTH S0DA1,8
  LENGTH S0DE2,8
  LENGTH S0DE3,8
  LENGTH S0DE4,8
  LENGTH S0DE5,8
  LENGTH S0DE6,8
  LENGTH S0DE7,8
  LENGTH S0DEB,8
  LENGTH S0DEC,8
  LENGTH S0DED,8
  LENGTH S0DF0,8
  LENGTH S101F,8
  LENGTH S1282,8
  LENGTH P00D9,8
  LENGTH P00DE,8
VALIDATION P0800,P0337             /* Unprefixed date
  VALIDATION P0801,P00CF             /* DATA/
  VALIDATION P0802,P00D1             /* DATB/
  VALIDATION P0803,P00D3             /* DATC/
  VALIDATION P0804,P00D6             /* DATD/
  VALIDATION P0805,P00D9             /* DATE/
  VALIDATION P0806,P00DC             /* DATF/
  VALIDATION P0807,P7E4A             /* DATG/
  VALIDATION P0808,P01C3             /* DATH/
  VALIDATION P0809,P004F             /* DATI/
  VALIDATION P080A,P01A5             /* DATJ/
  VALIDATION P080B,P7F01             /* DATK/
  VALIDATION P080C,P7E44             /* DATL/
  VALIDATION P080D,P00E0             /* DATM/
  VALIDATION P080E,P00E2             /* DATN/
  VALIDATION P080F,P00E4             /* DATO/
  VALIDATION P0810,P00E6             /* DATP/
  VALIDATION P0811,P7F04             /* DATQ/
  VALIDATION P0812,P00E8             /* DATR/
  VALIDATION P0813,P7F07             /* DATS/
  VALIDATION P0814,P00EA             /* DATT/
  VALIDATION P0815,P7E47             /* DATU/
  VALIDATION P0816,P00EC             /* DATV/
  VALIDATION P0817,P7F0A             /* DATW/
  VALIDATION P0818,P00EE             /* DATX/
  NODBCS AETYPE(M,BYP)
/*
```

Convert Panels to Use 10-Character Dates (BLGUT6MJ)

The prefix index values shown as the second set of values on the VALIDATION keyword support the date validation pattern of NN</>NN</>NNNN (or external date format of MM/DD/YYYY or DD/MM/YYYY).

```
//PNLMIG   JOB
//*--------------------------------------------------------------------*
//*                                                                    *
//*            Licensed Materials - Property of Tivoli Systems         *
//*                                                                    *
//*                              5697-SD9                              *
//*                                                                    *
//*   (C) Copyright Tivoli Systems Inc., an IBM Company 1997, 2001.    *
//*                                                                    *
//*                  See Copyright Instructions                        *
//*                                                                    *
//*--------------------------------------------------------------------*
//*----------------------------------------------------------------
//*  THIS IS A SAMPLE JOBSTREAM FOR RUNNING BLGUT6M TO MIGRATE FIELD
//*  LENGTHS AND VALIDATION PATTERNS ON PANELS.  YOU MUST CHANGE THE
//*  DATA SET NAMES FOR DD NAMES STEPLIB, BLGPDS, BLGPNLS, AND BLGDICT
//*  TO CORRESPOND TO THE DATA SET NAMES AT YOUR INSTALLATION.
//*
//*  DD NAME BLGPDS IS THE NAME OF THE INPUT PDS
//*  DD NAME BLGPNLS IS THE NAME OF THE OUTPUT VSAM PANEL DATA SET
//*  DD NAME BLGDICT IS THE NAME OF THE VSAM DICTIONARY DATA SET
//*
//*  IN THIS EXAMPLE, SINCE THERE IS NEITHER THE "INCLUDE" NOR
//*  "EXCLUDE" KEYWORD SPECIFIED, ALL PANELS IN THE INPUT PDS
//*  WILL BE PROCESSED.  SINCE THE MODIFY STATEMENT SPECIFIES NEITHER
//*  LOAD=MOD NOR LOAD=ALL, ONLY PANELS WHICH CONTAIN FIELDS SPECIFIED
//*  ON ONE OR MORE OF THE LENGTH AND VALIDATION STATEMENTS WILL BE
//*  LOADED INTO THE VSAM PANEL DATA SET.
//*---------------------------------------------------------------
//MIGRATE  EXEC PGM=BLGUT6M,REGION=2048K
//STEPLIB  DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=A
//BLGPDS   DD DISP=SHR,DSN=BLM.SBLMPNLS
//BLGPNLS  DD DISP=SHR,DSN=BLM.IBMPNLS
//BLGDICT  DD DISP=SHR,DSN=BLM.DICT
//*--------------------------------------------------------------------
//*
//*NOTE: FOR PANELS WHICH CONTAIN ONLY SINGLE BYTE CHARACTERS (E.G.
//* ENGLISH, FRENCH, GERMAN, SPANISH, ITALIAN) THE OPTION RECORD AT THE
//* END OF SYSIN IS OPTIONAL.
//* THE DEFAULT KEYWORDS - NODBCS AETYPE(M,BYP) - WILL CORRECTLY
//* MIGRATE PANELS TO THE CURRENT FORMAT AND SET ASSISTED-ENTRY PANELS
//* TO ALLOW EITHER SINGLE-BYTE OR DOUBLE-BYTE CHARACTERS.
//* DO NOT SPECIFY THE "DBCS" KEYWORD FOR ENGLISH PANELS.
//*
//*NOTE: FOR PANELS WHICH CONTAIN DOUBLE-BYTE CHARACTERS (E.G.
//* JAPANESE), THE OPTION RECORD IS REQUIRED
//* AND THE "DBCS" KEYWORD MUST BE SPECIFIED (INSTEAD OF "NODBCS") TO
//* AVOID CORRUPTION OF PANELS.  THE "AETYPE" KEYWORD IS OPTIONAL.  IF
//* NOT SPECIFIED, ASSISTED-ENTRY PANELS WHICH DO NOT HAVE A DATA TYPE
//* WILL BE SET TO ACCEPT MIXED SBCS AND DBCS DATA.
//*
//*--------------------------------------------------------------------
//SYSIN    DD *
  MODIFY
  LENGTH S0C34,10
  LENGTH S0C35,10
  LENGTH S0C36,10
  LENGTH S0C37,10
  LENGTH S0C38,10
  LENGTH S0C3D,10
  LENGTH S0C3E,10
  LENGTH S0C3F,10
  LENGTH S0C40,10
  LENGTH S0C41,10
  LENGTH S0C42,10
  LENGTH S0C43,10
  LENGTH S0C44,10
  LENGTH S0C45,10
  LENGTH S0C46,10
  LENGTH S0C47,10
  LENGTH S0C48,10
  LENGTH S0C49,10
  LENGTH S0C4A,10
  LENGTH S0C5F,10
  LENGTH S0CA2,10
  LENGTH S0DE2,10
  LENGTH S0DE3,10
  LENGTH S0DE4,10
  LENGTH S0DE5,10
  LENGTH S0DE6,10
  LENGTH S0DE7,10
  LENGTH S0DEB,10
  LENGTH S0DEC,10
  LENGTH S0DED,10
  LENGTH S0DF0,10
  LENGTH S101F,10
  LENGTH S1282,10
  VALIDATION P00CF,P0801
  VALIDATION P00D1,P0802
  VALIDATION P00D3,P0803
  VALIDATION P00D6,P0804
  VALIDATION P00D9,P0805
  VALIDATION P00DC,P0806
  VALIDATION P7E4A,P0807
  VALIDATION P01C3,P0808
  VALIDATION P004F,P0809
  VALIDATION P01A5,P080A
  VALIDATION P7F01,P080B
  VALIDATION P7E44,P080C
  VALIDATION P00E0,P080D
  VALIDATION P00E2,P080E
  VALIDATION P00E4,P080F
  VALIDATION P00E6,P0810
  VALIDATION P7F04,P0811
  VALIDATION P00E8,P0812
  VALIDATION P7F07,P0813
  VALIDATION P00EA,P0814
  VALIDATION P7E47,P0815
  VALIDATION P00EC,P0816
  VALIDATION P7F0A,P0817
  VALIDATION P00EE,P0818
  NODBCS AETYPE(M,BYP)
/*
```

22

BLGUT7—Copy an SDDS

Use BLGUT7 to copy an SDDS or to convert an SDDS to a different format, such as:

- A key 7 format database to a key 8 format database
- A key 8 format database to a key 7 format database
- A key 7 format database to a key 7 format database with a different MAXLRECL (RECORDSIZE)
- A key 8 format database to a key 8 format database with a different MAXLRECL

Depending on the OTRG and NTRG parameters that you specify in the PARM field, you can run BLGUT7 to convert:

- A one-SDDS cluster database to a multiple cluster database
- A multiple cluster database to a one-SDDS cluster database
- A multiple cluster database to a different multiple-SDDS cluster
- A one-SDDS cluster database to a different one-SDDS cluster database

BLGUT7 can convert the key format, MAXLRECL, and the number of clusters in one run. Separate runs are not needed. All desired changes can be made simultaneously in a single run of BLGUT7.

Note: For performance reasons, when changing the number of clusters do not change the key format or decrease the MAXLRECL size unless you have a need to do so. If you do not change the key format or decrease the MAXLRECL size, BLGUT7 simply copies the data to the correct cluster. If the key format is changed or the MAXLRECL size is decreased, BLGUT7 does additional processing on every SDDS record before writing the record to the new SDDS.

BLGUT7 does not destroy the SDIDS component for the database. The SDIDS is still valid for the database after BLGUT7 completes its processing.

BLGUT7 copies or converts the SDDS by reading each Tivoli Information Management for z/OS record in the SDDS being copied or converted and writing the record to the newly allocated format SDDS. After BLGUT7 completes, the new SDDS can be used with the existing SDIDS. The old SDDS can be deleted if desired.

Notes:

| 1. BLGUT7 does not update the BLX-SP VSAM resource definitions in a non-sysplex
| environment. If you change the key format and you use LSR, update the appropriate BLDVRP macros in the BLX-SP VSAM resource definition. If you are adding or

removing clusters, add or remove the appropriate BLXDSN macros. Also, verify that the STRNO is adequate to support the number of clusters you are using. If you change the SDDS MAXLRECL size, the CONTROLINTERVALSIZE will have to be changed. Verify that the BUFFER keyword on the BLDVRP macro is large enough to support the new CISIZE.

2. You will need to update the BLGCLDSN macro in your session-parameters members (BLGSESnn) to access the new SDDS. Also, you will need to change the TRIGGER keyword in the BLGCLUST macro if you change the number of SDDS clusters. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for macro details.
3. If you would like to use the BLGUT23 utilities to perform backups and restorations of your Tivoli Information Management for z/OS database, your SDDS clusters **must** use key 7 format. Consider changing to a key 7 format if you need to run BLGUT7 for some reason.

The SDDS key 7 format provides for better packing of data in the physical VSAM records used to store the Tivoli Information Management for z/OS record than does the key 8 format. In addition, a more efficient algorithm is used when filing a record in the database. Consequently, a performance improvement occurs for those databases with high activity. However, it is not necessary to convert an SDDS to the key 7 format when Tivoli Information Management for z/OS Version 7.1 or later is used.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See “FREE Command” on page 15 and “REALLOC Command” on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The EXEC statement PARM field keywords to use with BLGUT7 are as follows:

```
PARM=‘OTRG=(c,mmm),NTRG=(d,nnn)’
```

OTRG=(c,mmm)

where *c* specifies the trigger character of the database from which you are copying. Use this keyword when the database you are copying from is a multiple-SDDS cluster. The *mmm* parameter specifies a value from 1 to 100 for the number of SDDS clusters. The default value for *mmm* is 5.

NTRG=(d,nnn)

where *d* specifies the trigger character of the database to which you are copying. Use this keyword when the database you are copying to is a multiple-SDDS cluster. The *nnn* parameter specifies a value from 1 to 100 for the number of SDDS clusters. The default value for *nnn* is 5.

When neither the database you are copying from, nor the database you are copying to, is a multiple-SDDS cluster, do not code the PARM field on the EXEC statement.

Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual for further information on trigger characters.

Input

The input for BLGUT7 is the SDDS cluster to convert. The SDDS is a key-sequenced VSAM data set. When you are using a multiple-SDDS cluster database, specify only the first cluster name for the old SDDS cluster (OLDSDDS) and new SDDS cluster (NEWSDDS). Be sure to reassemble their session-parameters members after the processing.

Note: If you are running in sysplex mode, your input SDDS must be defined correctly for VSAM RLS. If your input SDDS is from a previous release, you may need to alter it or redefine it to a new data set with the correct attributes for RLS. For more information on migrating old data sets for use in sysplex mode, see the ″Migrating from Previous Releases″ chapter of the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

Output

The output from BLGUT7 is:

- The SDDS copied or converted to key 7 or key 8 format, or a one- or multiple-cluster SDDS, or both. This SDDS is also a key-sequenced VSAM data set. You can delete the old SDDS after BLGUT7 runs successfully, and the new, copied, or converted SDDS is available for use.
- Informational and error messages in a SYSPRINT data set.
- A return code of 0 if the utility finishes successfully.
- A return code of >0 if an error occurs.

Restrictions

The following restrictions apply:

- Define the new SDDS cluster(s) using AMS as described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* manual, before you run BLGUT7.
- Users cannot access either SDDS being processed while BLGUT7 is running.

DD Statements

The DD statements required to run BLGUT7 are as follows:

NEWSDDS

Defines the copied or converted SDDS.

OLDSDDS

Defines the SDDS to copy or convert.

SYSPRINT
Must be a sequential non-VSAM data set that you can write to a system output device or data set.

Scenarios Using the BLGUT7 Utility Program

The following examples illustrate how you can use BLGUT7 to convert formats. In a multiple-cluster SDDS the cluster name associated with OLDSDDS and NEWSDDS must represent the first SDDS cluster in the sequence for that database.

The following example shows how to convert a one-SDDS cluster to the new format for a one-SDDS cluster. BLGUT7 examines the key format and record size of both SDDS clusters and copies the records if the key format and record sizes are the same. If the key format is different or the record size is decreased, BLGUT7 converts the records. The copying of records is faster than the conversion of records.

```
//CONVERT   JOB
//STEP1     EXEC PGM=BLGUT7
//STEPLIB   DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//OLDSDDS   DD   DISP=SHR,DSN=BLM.SDDS.OLD
//NEWSDDS   DD   DISP=SHR,DSN=BLM.SDDS.NEW
```

The following example shows how to convert a 1-cluster SDDS to a new 3-cluster SDDS. BLGUT7 examines the key format and record size of the OLDSDDS and NEWSDDS clusters and copies the records if the key format and record sizes are the same. If the key format is different or the MAXLRECL (RECORDSIZE) is decreased, BLGUT7 converts the records. Actual format changes are controlled by the key length and MAXLRECL as defined when the data set is allocated. The copied or converted records are written to the correct NEWSDDS cluster based on the record’s actual VSAM key. The copying of records is faster than the conversion of records, regardless of the number of clusters used.

```
//CONVERT   JOB
//STEP1     EXEC PGM=BLGUT7,PARM='NTRG=($,3)',REGION=4096K
//STEPLIB   DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//OLDSDDS   DD   DISP=SHR,DSN=BLM.SDDS.OLD
//NEWSDDS   DD   DISP=SHR,DSN=BLM.NEW$01.SDDS
```

The following example shows how to convert a 5-cluster SDDS to a new format for 10 new SDDS clusters. BLGUT7 examines the key format and record size of the OLDSDDS and NEWSDDS clusters and copies the records if the key format and record sizes are the same. If the key format and record sizes are different, BLGUT7 converts the records. The copied or converted records are written to the correct NEWSDDS cluster based on the record’s actual VSAM key. The copying of records is faster than the conversion of records, regardless of the number of OLDSDDS and NEWSDDS clusters used.

```
//CONVERT   JOB
//STEP1     EXEC PGM=BLGUT7,PARM='OTRG=#,NTRG=($,10)',REGION=4096K
//STEPLIB   DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSPRINT  DD   SYSOUT=A
//OLDSDDS   DD   DISP=SHR,DSN=BLM.SDDS#11.OLD
//NEWSDDS   DD   DISP=SHR,DSN=BLM.SDDS$27
```

23

BLGUT8—Build Tables Utility

The Tivoli Information Management for z/OS APIs use various tables to function. BLGUT8 builds program interface data tables (PIDTs), program interface pattern tables (PIPTs), and program interface alias tables (PALTs) with data that it extracts from the input stream, the dictionary, and specified panels. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for information on how the API uses these tables and how to use the table build utility to build these tables.

This chapter tells you how to use the Table Build Utility (BLGUT8) to create data tables (PIDTs), validation pattern tables (PIPTs), and alias tables (PALTs) that the APIs use. The description includes control statements, input statements, and sample output tables. You can find descriptions of messages from the Table Build Utility in the *Tivoli Information Management for z/OS Messages and Codes* manual.

You may choose to store information about the “composition” of records in data model records rather than in panels. If you use this method, the need to use the Table Build Utility (BLGUT8) to create PIDTs is eliminated. Both the*Tivoli Information Management for z/OS Application Program Interface Guide* and the *Tivoli Information Management for z/OS Panel Modification Facility Guide* contain additional information on data model records.

| You can also use the BLGUT18 utility to build program interface data tables (PIDTs) and
| program interface data tables (PIPTs) with data extracted from the data view records and
| data attribute records. All of the tables (create, update, retrieve, inquiry, add record relations)
| associated with a particular data view record are written to a single partitioned data set
| member that has the same name as the data view record. The BLGUT18 utility is described
| in “BLGUT18—Build PIDTs and PIPTs” on page 147.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Using BLGUT8 To Build RDMTs

The DB2 Extract Utility uses relational data mapping tables (RDMTs) to map Tivoli Information Management for z/OS panels and fields to relational database tables and columns. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for details on using BLGUT8 to build RDMTs.

Using BLGUT8 To Create Data Tables (PIDTs) and Validation Pattern Tables (PIPTs)

The BLGUT8 syntax description here applies to building static PIDTs, PIPTs, and PALTs; it does not apply to building relational data mapping tables (RDMTs). Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information on building RDMTs.

You can propagate your production data from Tivoli Information Management for z/OS to DB2 using the DB2 Extract Facility. The Table Build Utility builds Relational Data Mapping Tables (RDMTs) that the DB2 Extract Facility uses. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information on using the DB2 Extract Facility.

The Table Build Utility builds static PIDTs, PIPTs, and PALTs with data that it extracts from the input stream, the dictionary, and specified panels. The PIDT defines the data that passes between an application and the APIs. The PIPT tests record or search argument data against validation patterns. The PALT enables applications to use external symbolic names (alias values) for data fields, prefixes, and PIDT member names instead of using s-word index, p-word index, or PIDT member name values. The *Tivoli Information Management for z/OS Application Program Interface Guide* contains specific information on the PIDT, the PIPT, and the PALT.

Building Static PIDTs and PIPTs

You define the fields that you want to process for each record type. Predefined tables provide access to many of the Tivoli Information Management for z/OS record types, and all fields for these records are included in the tables. If you want to use different views for the data or if you want to process your own user-defined record types with the APIs, you must use this Table Build Utility to build the various tables you want.

Input control statements tell the Table Build Utility which fields to process. After the utility processes the control statements, it stores the tables it built in a partitioned data set (PDS) that has the same characteristics as a Tivoli Information Management for z/OS Report Format Table (RFT) partitioned data set. The APIs retrieve these tables from the PDS associated with Tivoli Information Management for z/OS.

Input statements that use DB2 Extract Facility keywords can be used to build LLAPI PIDTs. DB2 Extract Facility keywords are ignored if you are building a PIDT. Therefore, if you are using both DB2 Extract Facility and an API, you can use the same input statements to build PIDTs and RDMTs.

Each of the specifications to the Table Build Utility defines a view of the Tivoli Information Management for z/OS data. Complete views for a particular record type consist of every field that it is possible to collect for that record type. A more practical view for problem records consists of the following fields:

- Reporter name
- Problem status
- Description
- Initial priority
- User problem number

- Description freeform text

Input/Output

Inputs for the Table Build Utility (BLGUT8) are:

Control statements
Tells the Table Build Utility how to build the tables. The primary function of the control statements is to list, through such things as panel names and dictionary index keys, the fields to be included in the tables.

Dictionary
A VSAM data set that contains the s-words and p-words, indicated in the control statements, that you use to build the PIDT.

Panels
A partitioned data set (PDS) of panels. The panels contain the p-words and s-words necessary to construct the PIDT. The utility extracts the control information from the panels to fill in the corresponding PIDT and PIPT fields. You can use the VSAM to PDS utility BLGUT6F to build this data set. You can concatenate panel sets by successively running BLGUT6F into the same PDS, starting with the lowest panel set in the concatenation.

Outputs from BLGUT8 are:

PDS member or members
Contains table objects (PIDT and PIPT or PALT) built by the Table Build Utility. The utility adds each table object it builds to this partitioned data set as a separate member, replacing like-named members with the newer version.

Message and report output
Contains the input control statements. This output is followed by either informational messages, indicating successful table generation, or error messages. If the utility does not encounter errors, two summary reports, showing the values collected in the data and validation tables, accompany the informational messages.

A return code
A value of 0 indicates the utility completed successfully. A return value of 8 indicates that the utility did not finish successfully.

Errors

The following error types can occur:

- Allocation, open, and I/O errors
- Control statement specification not valid
- Missing dictionary or panel data

Control

Start BLGUT8 as a batch utility. The system administrator or table developer must determine when and against what data sets to run this utility. Panel modifications can necessitate running BLGUT8 again. The following table shows the DD statements required to run BLGUT8.

Table 3. BLGUT8 DD Statements

DD Statement	Description
SYSIN	The input control statement sequential data set
BLGPNLS	The input partitioned panel data sets

Table 3. BLGUT8 DD Statements (continued)

DD Statement	Description
BLGDICT	The input dictionary data set (VSAM)
BLGRFT	The output partitioned table data set
SYSPRINT	The output message data set (sequential)

Input Stream

The input stream for BLGUT8 consists of several control statements. The PIDT and PIPT are constructed in the sequence in which these control statements are specified.

The following representations are used to describe keywords and values:

- Braces { } represent alternatives; you must choose one.
- Brackets [ ] represent optional entries; you can choose one or none.
- In a statement name, keyword, or keyword value, uppercase letters designate the minimum truncation, and the rest of the statement name, keyword, or keyword value is in lowercase.
- Enter a keyword value with at least the first character in uppercase as shown (or in a valid truncated form as indicated); replace a value shown in all lowercase with your own data.

The Table Build Utility control statement syntax is:

```
[ label: ] statement [ keyword (value) ] [ keyword (value) ] ...; /* comment */
```

label
An optional 1- to 8-character alphanumeric label you assign to this statement, followed by a colon. It can begin in any column (1 - 72), but it must be first. Labels are ignored; they are strictly for your convenience.

statement
The name of the statement. It can begin in any column (1 - 72) but must appear before any keywords. Blanks between the colon following a label and a statement name are optional.

keyword(value)
A parameter that consists of a predefined word (up to 15 alphabetic characters) and its associated data (value). If you include the keyword, you must also specify the value in parentheses, immediately following the keyword. Keywords must be separated from the statement name by blanks. Blanks are allowed, but not required, between keywords.

; The statement terminator. A statement can occupy multiple physical lines, but must end with a semicolon (unless it is only a combination of blanks and comments).

/* comment */
An optional explanation added to or interspersed among the statements. Comments can appear wherever blanks are allowed. Comments must be preceded by a slash-asterisk (/*) and ended by an asterisk-slash (*/). Comments can contain any characters except the asterisk-slash combination, which ends the comment.

Follow these rules when you use the utility:

- At least one blank is required between a statement name and the first keyword. When you code multiple keywords on a statement, they can be separated by blanks. A keyword value must begin and end on the same line.
- Comments can appear wherever blanks are allowed.
- The same keyword can appear on more than one statement, but the usage of the keyword can vary from statement to statement.
- The alphabetic characters on a statement can be in uppercase, lowercase, or mixed case.
- Use columns 1 through 72 for statement data. One statement can occupy multiple lines. Continuation is automatic, and no symbol is required.

TABLE/ETABLE Statement

You can specify multiple data validation tables in one input stream by using the TABLE/ETABLE combination. FIELD statements (described in “Field Statement” on page 114) make up a record or search argument specification. Each FIELD statement corresponds to an entry in the data table.

Keywords on the TABLE statement specify table attributes, and the TABLE/ETABLE combination defines the boundaries of a table specification. Each table specification corresponds to a record or search argument definition and generates one data table and one validation table. All other statements must lie within a TABLE/ETABLE combination.

The syntax of the TABLE statement is:

```
[ label: ]     TAble  Name(name)  Use {(Inquiry)  }
                                      {(Retrieve) }
                                      {(Create)   }
                                      {(Update)   }
                                      {(Add)      }
                                      {(ALias)    }
                                      {(Header)   }

                       Code(authorization code)

                       [Separator(xx)]
          .
          .
          .
[label:]       Etable
```

Name

Identifies the name to be given to the table. This name also serves as the PDS member name. You can use 8 or fewer characters for the table name when the value specified with Use is ALias. All other tables are named using 7 or fewer characters. Two member names are determined by this specification. The PIDT is named with the value for this keyword, and the PIPT name is this value with a P appended. This keyword determines PIDTNAME, PIDTPTNM, and PIPTNAME.

Use

Identifies the function for which the table is used. The value specified with this keyword determines how the APIs set PIDTUSEF for all Use keyword values except ALias. U(ALias) causes the Table Build Utility to process an alias table only. Use(Header) causes the Table Build Utility to process header rows only.

Inquiry

For fields to include in a search argument.

Retrieve
For fields to retrieve from an existing record.

Create
For fields to include in a new record.

Update
For fields to update in an existing record.

Add
For record relation fields to add to an existing record.

ALias
For fields to collect for an alias table. You can use only ALIAS statements when you specify USE(ALias).

Header
For fields to create a model PIDT containing only header rows. This PIDT is for use in generating dynamic PIDTs on the retrieve transaction (T100). A FIELD statement specified with Use(Header) is ignored.

Code
Identifies the authorization code that allows the APIs to access the Tivoli Information Management for z/OS database for the specified use. This keyword sets field PIDTAUTH. The utility does not use the Code keyword when creating alias tables.

authorization code
Four hexadecimal characters (0-F) that identify the last four positions of the authorization code XIMRAtxxxx, where t identifies the record type, and xxxx identifies the authorization code. The utility does not use the authorization keyword when creating alias tables.

When creating a table that has only header rows, use an authorization code that applies to retrieve (T100), create (T102), and update (T105) transactions.

Separator
Identifies the separator character the utility stores in the table field PIDTSEPC. This character separates multiple responses for one field in the response buffer. A blank separator is not allowed. The default separator is a comma. You specify a single quote with two single quotes. You specify a right parenthesis by enclosing it in single quotes. Do not use this keyword when creating alias tables. Any SBCS value can be used except for an SO (X'0E'), SI (X'0F'), or blank (X'40').

xx is a single character (subject to the above restrictions), or two hexadecimal characters (0-F).

Field Statement

The FIELD statement defines the Tivoli Information Management for z/OS fields included in the data table. A field is any attribute that can be assigned to a record type and is characterized by an s-word, a p-word, or both. Examples are reporter name, record ID, freeform text description, and date entered. This term applies to search arguments as well as records. At least one FIELD statement must be defined when the TABLE statement declares any value for Use except Header. A FIELD statement specified with Use(Header) is ignored.

The syntax of the FIELD statement is:

```
[label:] Field Panel(panel)  Index(dictionary index key)

            [REquired {(Yes)}]  [List {(Yes)}] [Text {(Yes)}]
            [         {(No) }]  [     {(No) }] [     {(No) }]

            [RCdsword {(Yes)}]     [Notlogic {(Yes)}]
            [         {(No) }]     [         {(No) }]
```

The default for each keyword above is NO if you omit the keyword from your Field statement. However, if you include the keyword in your statement, you must also enter YES or NO.

Panel

Identifies the panel in which the Tivoli Information Management for z/OS field is located.

Assisted-entry panels provide a prefix, validation, and control information for validating and collecting response data. The Table Build Utility sets field PIDTRDEF to R for this panel type.

Selection, option, and data-entry panels provide an s-word, a visible phrase, and control information. The Table Build Utility sets field PIDTRDEF to P when you specify these panel types unless you also specify TEXT(Y). In that case, the Table Build Utility sets field PIDTRDEF to X.

Control panels supply a p-word, an s-word, or both, and control information. The Table Build Utility sets field PIDTRDEF to D (for direct) when you specify a control panel. Data items that are collected by a control panel are called *direct-add data* items. Do not specify a control panel on a retrieve or inquiry transaction. Instead, create an assisted-entry panel and specify it in the PIDT along with the s-word.

The Table Build Utility also sets fields PIDTPNLN, PIDTPNLT, and PIDTMNTF, depending on the information from each type of panel.

panel

An 8-character alphanumeric string. The first character must be alphabetic (A-Z). This string names a member of a panel data set.

Index

Identifies the Tivoli Information Management for z/OS field. The Index keyword sets field PIDTSYMB.

dictionary index key

The character P or S followed by 4 hexadecimal characters identify a p-word or an s-word, respectively, in the dictionary data set or specified panel. This s-word index or p-word index must be in the panel named by the PANEL keyword unless you specify an assisted-entry panel with its Collect From Caller field set to YES. In this case, the s-word is obtained from the dictionary.

For control panels, the utility uses only ADD control lines. Within these control lines, it uses only p-words with literal p-word validation data (pfx/<data> or <data>). For assisted entry-panels, the utility uses only p-words (pfx/).

When you specify an s-word index key and control panel, the utility retrieves corresponding literal prefix validation data, if available, along with the s-word. If you specify an s-word index key and an assisted-entry panel, the utility uses the first p-word that appears in the validation section of the specified assisted-entry panel to define the field.

You can only specify a p-word index key (Pxxxx) for an assisted-entry or control panel. When you specify a p-word index key, the utility does not retrieve the s-word. Specify a p-word key only if you are describing a field that does not have an s-word.

Duplicate index keyword values are allowed only across table specifications.

Unless you specify RCDSWORD(Y), only replaceable fields are allowed for USE(CREATE) and USE(UPDATE) tables, and only nonreplaceable fields are allowed for USE(ADD) tables.

REquired
Identifies required Tivoli Information Management for z/OS fields. This keyword corresponds to field PIDTREQD.

Yes
Indicates a required field.

No (default)
Indicates an optional field.

List
Identifies list processing for the field. This keyword corresponds to field PIDTLIST.

Yes
Indicates a list field. Specify LIST(Y) only when the PANEL keyword value specifies an assisted-entry panel and the Index keyword value is a s-word index (Sxxxx) that refers to an s-word of no more than 8 characters. LIST(Y) is not allowed with TEXT(Y), RCDSWORD(Y), or USE(ADD).

No (default)
Indicates a nonlist field.

Text
Identifies a freeform text field.

Yes
Indicates a freeform text field. The utility sets field PIDTRDEF to X. Specify TEXT(Y) only when the PANEL keyword value names a selection, option, control, or data-entry panel. TEXT(Y) is not valid when the Index keyword value represents a p-word index (Pxxxx). Do not use TEXT(Y) with LIST(Y), RCDSWORD(Y), or NOTLOGIC(Y).

No (default)
Indicates a field that does not identify freeform text.

RCdsword
Identifies the field that defines the record type. This keyword corresponds to field PIDTRTYP.

Yes
Indicates a record type field. Specify RCDSWORD(Y) only when the PANEL keyword value names a selection, option, data-entry, or control panel. RCDSWORD(Y) is not valid when the Index keyword value represents a p-word index (Pxxxx). For a USE(INQUIRY) table specification, you must specify RCDSWORD(Y) for one FIELD statement or none. Otherwise, you must specify RCDSWORD(Y) for one and only one FIELD statement within a table specification.

User exit BLGYAPGP uses the record type s-word to index into the control panels BLG1AACP and BLG1AAUP to determine the appropriate summary panel for a create or update.

To use the panel specified in panel BLG1AAUP as the update summary panel instead of the regular target panel of the record update, specify an authorization code of `0001` for that panel in BLG1AAUP. RCDSWORD(Y) is not allowed with LIST(Y), TEXT(Y), or NOTLOGIC(Y).

The panel that is identified in the statement where RCSDWORD(Y) is specified is not used to determine the create or update flow.

No (default)
Indicates a field that does not define the record type.

Notlogic
Indicates that Boolean not logic is to be used for this field in constructing a search argument for a USE(INQUIRY) table. This keyword sets field PIDTNOTL.

Yes
Indicates that not logic is to be applied. NOTLOGIC(Y) is not allowed with TEXT(Y) or RCDSWORD(Y).

No (default)
Indicates that not logic does not pertain to this field.

PIDT and PIPT Table Build Job Stream Example

This example shows how to use BLGUT8 to build a CREATE data table (PIDT) named PROBPUT and a validation table (PIPT) named PROBPUTP. The example also shows how these tables are stored in a partitioned data set named XXX.SBLMFMT.PDS. Only a portion of the problem record type fields are included.

```
//TABLEO   JOB
//STEP1    EXEC PGM=BLGUT8,REGION=2048K
//STEPLIB  DD DISP=SHR,DSN=BLM.SBLMMOD1
//BLGDICT  DD DISP=SHR,DSN=BLM.DICT
//BLGPNLS  DD DISP=SHR,DSN=BLM.SBLMPNLS
//BLGRFT   DD DISP=SHR,DSN=BLM.TABLES.PDS
//SYSPRINT DD SYSOUT=A
//* 'DLM=%%' CHANGES END-OF-STREAM DELIMITER FROM '/*' to '%%'
//* OTHERWISE, COMMENT STARTED IN FIRST COLUMN TERMINATES INPUT STREAM
//SYSIN    DD *,DLM='%%'
TABLE NAME(PROBPUT) USE(CREATE) CODE(0110) SEPARATOR(,);

FIELD PANEL(BLG00000) INDEX(S0032)  /*  PROBLEM RECORD TYPE        */
      REQUIRED(Y) RCDSWORD(Y);

FIELD PANEL(BLG6REQN) INDEX(S0B59)  /*  REPORTER NAME              */
      REQUIRED(Y);

FIELD PANEL(BLG6PTYP) INDEX(S0C09); /*  PROBLEM TYPE               */

FIELD PANEL(BLG6RQDP) INDEX(S0B9B); /*  REPORTER DEPARTMENT        */

FIELD PANEL(BLG6STAT) INDEX(S0BEE)  /*  REPORTER STATUS            */
      REQUIRED(Y);

FIELD PANEL(BLG6PHON) INDEX(S0B2D); /*  REPORTER PHONE             */

FIELD PANEL(BLG6URN0) INDEX(S0CCF); /*  USER PROBLEM NUMBER        */

FIELD PANEL(BLG6PRII) INDEX(S0BE6); /*  INITIAL PRIORITY           */

FIELD PANEL(BLG6DSAB) INDEX(S0E0F)  /*  DESCRIPTION                */
      REQUIRED(Y);

FIELD PANEL(BLG0B010) INDEX(S0E01)  /*  TEXT, DESCRIPTION          */
      TEXT(Y);
```

```
ETABLE;                              /*  END TABLE                 */
%%
//
```

This example shows the output from a successful run of BLGUT8 in building the PIDT and PIPT tables.

```
 DATE: 05/21/99 TIME:13:57:52 INFORMATION MANAGEMENT FOR Z/OS PROGRAM INTERFACE TABLE BUILD UTILITY PAGE:00001
 LINE  STMT  ---------1---------2---------3--SOURCE-4---------5---------6---------7---------8
    1     1  TABLE NAME(PROBPUT) USE(CREATE) CODE(0110) SEPARATOR(,);
    2     2
    3        FIELD PANEL(BLG00000) INDEX(S0032)  /*  PROBLEM RECORD TYPE       */
    4              REQUIRED(Y) RCDSWORD(Y);
    5     3
    6        FIELD PANEL(BLG6REQN) INDEX(S0B59)  /*  REPORTER NAME             */
    7              REQUIRED(Y);
    8     4
    9        FIELD PANEL(BLG6PTYP) INDEX(S0C09); /*  PROBLEM TYPE              */
   10     5
   11        FIELD PANEL(BLG6RQDP) INDEX(S0B9B); /*  REPORTER DEPARTMENT       */
   12     6
   13        FIELD PANEL(BLG6STAT) INDEX(S0BEE)  /*  REPORTER STATUS           */
   14              REQUIRED(Y);
   15     7
   16        FIELD PANEL(BLG6PHON) INDEX(S0B2D); /*  REPORTER PHONE            */
   17     8
   18        FIELD PANEL(BLG6URN0) INDEX(S0CCF); /*  USER PROBLEM NUMBER       */
   19     9
   20        FIELD PANEL(BLG6PRII) INDEX(S0BE6); /*  INITIAL PRIORITY          */
   21    10
   22        FIELD PANEL(BLG6DSAB) INDEX(S0E0F)  /*  DESCRIPTION               */
   23              REQUIRED(Y);
   24    11
   25        FIELD PANEL(BLG0B010) INDEX(S0E01)  /*  TEXT, DESCRIPTION         */
   26              TEXT(Y);
   27    12
   28        ETABLE;                             /*  END TABLE                 */


 NO ERRORS DETECTED DURING TABLE BUILD PROCESSING FOR DATA TABLE: PROBPUT AND PATTERN TABLE: PROBPUTP


 DATE: 05/21/99 TIME:13:57:52 INFORMATION MANAGEMENT FOR Z/OS PROGRAM INTERFACE TABLE BUILD UTILITY  PAGE:00002

 FIELD SPECIFICATION SUMMARY FOR PROGRAM INTERFACE DATA TABLE (PIDT): PROBPUT
 TABLE USE: CREATE   AUTHORIZATION CODE: 0110   MULTIPLE RESPONSE SEPARATOR: ,(6B)   FIELDS DEFINED: 10
 HEADINGS: F=FIELD TYPE Q=REQUIRED C=SEARCHABLE A=AUTHORIZATION L=TABLE LIST ITEM R=REC TYPE SWORD D=DATE
           T=PANEL TYPE J=JOURNALED S=STRING DATA Z=LEFT ZERO PAD N=APPLY NOT LOGIC M=MAINT FLAGS(HEX)
           G=GROUPED MAXL=MAX RESPONSE LENGTH MNCR=MAXIMUM RESPONSE COUNT PROW=PATTERN TABLE ENTRY ROW
           X=DATA TYPE V=CASE-SENSITIVE VALIDATE M=MIXED CASE COGNIZE K=COLLECTED DATA CASE

 INDEX   SWORD     PANEL   T M  PREFIX X F MAXL MNCR PROW  Q D C J L R A S Z N G V M K   VISIBLE PHRASE
 ____________________________________________________________________________________________________
 S0032 Z//S/TXS   BLG00000 S 00        . P    0    1    0  Y . Y . . Y Y . . . .         RECS=PROBLEM
 S0B59 XIM0I0PS00 BLG6REQN A 00 PERS/  M R   40   20    1  Y . Y . . . . U . . . . . A
 S0C09 XIMGCT0000 BLG6PTYP A 00 TYPE/  M R    8    1    3  . . Y O . . . . . . . . . U
 S0B9B XIM0I0GS00 BLG6RQDP A 00 GROS/  M R   11    1    4  . . Y . . . . . . . . . . U
 S0BEE XIMS0SSC00 BLG6STAT A 00 STAC/  M R    7    1    6  Y . Y O . . . . . . . . . U
 S0B2D XIM00NPS00 BLG6PHON A 00 PH/    M R   13    1    9  . . Y . . . . . . . . . . U
 S0CCF XIM00NR001 BLG6URN0 A 00 RNID/  M R    8    1   11  . . Y . . . . . . . . . . U
 S0BE6 XIMS0SPI00 BLG6PRII A 00 PRII/  M R    2    1   12  . . Y O . . . . Y . . . . U
 S0E0F XIM0TXCA00 BLG6DSAB A 00        M R   45   23    0  Y . Y . . . . Y . . . . . A
 S0E01 XIM0TX0S00 BLG0B010 S 00        . X   44    1    0  . . Y . . . . . . . .         DESCRIPTION
 BLG21800I Data table PROBPUT was successfully written to table data set DEWRIG.TEST.RFT.

   DATE: 05/21/99 TIME: 13:57:52 INFORMATION MANAGEMENT FOR Z/OS PROGRAM INTERFACE TABLE BUILD UTILITY  PAGE:00003
 FIELD VALIDATION PATTERN SUMMARY FOR PROGRAM INTERFACE PATTERN TABLE (PIPT): PROBPUTP
 HEADINGS:   PFXL=PREFIX LENGTH   PFXI=PREFIX INDEX   AUTH=AUTHORIZATION CODE   PATL=PATTERN LENGTH

  ROW  INDEX  PFXL  PREFIX  PFXI AUTH FLAG PATTERN TYPE           PATL    PATTERN DATA
 ____________________________________________________________________________________________________
    1  S0B59    5   PERS/   0347 0000  00  AUTOMATIC                5    =NAME
    2  S0B59    5   PERS/   06DF 0000  00  EXPRESSION               0
    3  S0C09    5   TYPE/   02AF 0000  00  EXPRESSION               4    CCV7
    4  S0B9B    5   GROS/   0151 0000  00  AUTOMATIC                5    =DEPT
    5  S0B9B    5   GROS/   0150 0000  00  EXPRESSION               5    CCV10
    6  S0BEE    5   STAC/   0276 0000  00  EXPRESSION               9
    7  S0BEE    5   STAC/   026D 0000  00  EXPRESSION               6
    8  S0BEE    5   STAC/   0272 0000  00  EXPRESSION               8
    9  S0B2D    3   PH/     0013 0000  00  AUTOMATIC                6    =PHONE
  10  S0B2D    3   PH/     0010 0000  00  EXPRESSION              5    IIV12
  11  S0CCF    5   RNID/   01EC 0000  00  EXPRESSION              4    ACV7
  12  S0BE6    5   PRII/   01B6 0000  00  EXPRESSION              5    NNV1L
BLG21801I Pattern table PROBPUTP was written to the DEWRIG.TEST.RFT table data set.

BLGUT8 - PROGRAM INTERFACE TABLE BUILD UTILITY COMPLETED
```

Using BLGUT8 To Build an Alias Table

You can use the Table Build Utility to build an alias table by specifying the Alias keyword in the TABLE statement Use field.

ALIAS Statement

The ALIAS statement defines the fields that you want collected in a row of an alias table. The field value combinations that you can collect in an alias table row are controlled by the Table Build Utility.

This is the syntax of the ALIAS statement.

```
{label:} Alias  Name(name) {Field(index)}    [Default]
                           {Prefix(pindex)}
                           {Table(tablename)}
```

Name

Identifies a 1- to 32-character external name of the field, prefix, or alias table. The name must contain the uppercase characters A-Z, 0-9, underscore, or period with no imbedded blanks. The first character must be alphabetic.

The external name specified in an alias table row provides a means to identify a Tivoli Information Management for z/OS data field using something other than a s-word or p-word index value. You can also use an external name to identify a Tivoli Information Management for z/OS prefix value to be stored in an alias table row. You use the prefix value in constructing freeform argument inquiries. Duplicate external names are not allowed in a single table.

Field

Specifies a 5-character field index that identifies a Tivoli Information Management for z/OS field. It corresponds to the field value stored in the PIDT field PIDTSYMB, but it is not verified against the dictionary. The HLAPI uses it to navigate the PIDT. Do not specify duplicate FIELD index values.

The FIELD keyword is mutually exclusive from the PREFIX and TABLE keywords.

Prefix

Specifies a 5-character prefix index that identifies a Tivoli Information Management for z/OS prefix that the Table Build Utility stores in an alias table row. The HLAPI uses the prefix value to construct freeform inquiry arguments.

The PREFIX keyword is mutually exclusive with the FIELD, DEFAULT, and TABLE keywords.

Default

Specifies a 1- to 45-character data response value that the API collects as the default response for a Tivoli Information Management for z/OS field. The APIs use default values in processing response fields when the APIs create records. You specify a single quote by specifying two single quotes. You specify one or more right parentheses by enclosing them within single quotes.

The DEFAULT keyword is mutually exclusive from the PREFIX keyword.

Table

Specifies a 1- to 7-character field that identifies a static PIDT or a 1- to 8-character field that identifies a data view record. Your application uses the table alias to specify a static PIDT or data view record ID when performing transactions.

The TABLE keyword is mutually exclusive from the FIELD, DEFAULT, and PREFIX keywords.

Alias Table Build Job Stream Example

This example illustrates the BLGUT8 job stream that builds an alias table (named ALIASONE) in a partitioned data set named XXX.SBLMFMT.PDS.

```
//TABLEO   JOB
//STEP1    EXEC PGM=BLGUT8,REGION=2048K
//STEPLIB  DD DISP=SHR,DSN=BLM.SBLMMOD1
//BLGPNLS  DD DISP=SHR,DSN=BLM.SBLMPNLS
//BLGDICT  DD DISP=SHR,DSN=BLM.DICT                     /*  VSAM */
//BLGRFT   DD DISP=SHR,DSN=BLM.TABLES.PDS
//SYSPRINT DD  SYSOUT=A
//* 'DLM=%%' CHANGES END-OF-STREAM DELIMITER FROM '/*' to '%%'
//* OTHERWISE, COMMENT STARTED IN FIRST COLUMN TERMINATES INPUT
//* STREAM
//SYSIN    DD  *,DLM='%%'
TABLE NAME(ALIASONE) USE(ALIAS);

  ALIAS NAME(PROBLEM_REPORTER_NAME)          /*   PROBLEM REPORTER    */
        FIELD(S0B59)                         /*   NAME                */
        DEFAULT(SYSTEM);

  ALIAS NAME(PROBLEM_REPORTER_PHONE)         /*   PROBLEM REPORTER    */
        FIELD(S0B2D);                        /*   PHONE               */

  ALIAS NAME(PROBLEM_REPORTER_LOCATION)      /*   PROBLEM REPORTER    */
        FIELD(S0C0C);                        /*   LOCATION            */

  ALIAS NAME(PROBLEM_STATUS_CODE)            /*   PROBLEM STATUS      */
        FIELD(S0BEE)                         /*   CODE                */
        DEFAULT(INITIAL);

  ALIAS NAME(PROBLEM_TYPE_CODE)              /*   PROBLEM TYPE        */
        FIELD(S0C09);                        /*   CODE                */

  ALIAS NAME(PROBLEM_ABSTRACT)               /*   PROBLEM ABSTRACT    */
        FIELD(S0E0F)
        DEFAULT(SYSTEM OPENED PROBLEM);

  ALIAS NAME(GENERIC_PHONE_PREFIX)           /*   GENERIC PHONE       */
        PREFIX(P0013);                       /*   PREFIX              */

  ALIAS NAME(DATE_OCCURRED_PREFIX)           /*   DATE OCCURRED       */
        PREFIX(P00E4);                       /*   PREFIX              */

  ALIAS NAME(DESCRIPTION_TEXT_DDNAME)        /*   DESCRIPTION TEXT    */
        FIELD(S0E01);                        /*   DDNAME              */

  ALIAS NAME(SAMPLE_DATA_ITEM)               /*   SAMPLE DATA         */
        FIELD(P1234)                         /*   USING PREFIX INDEX  */
        DEFAULT(SAMPLE);

  ALIAS NAME(PROBLEM_CREATE_TABLE)           /*   PROBLEM CREATE      */
```

```
        TABLE(BLGYPRC);                     /*   PIDT ALIAS NAME     */
ETABLE;                                     /*   END TABLE           */
%%
//
```

The following examples show partial output results from a successful alias table build run.

```
  1 TABLE NAME(ALIASONE) USE(ALIAS);

  2 ALIAS NAME(PROBLEM_REPORTER_NAME)       /*  PROBLEM REPORTER     */
          FIELD(S0B59)                      /*  NAME                 */
          DEFAULT(SYSTEM);

  3 ALIAS NAME(PROBLEM_REPORTER_PHONE)      /*  PROBLEM REPORTER     */
          FIELD(S0B2D);                     /*  PHONE                */

  4 ALIAS NAME(PROBLEM_REPORTER_LOCATION)   /*  PROBLEM REPORTER     */
          FIELD(S0C0C);                     /*  LOCATION             */

  5 ALIAS NAME(PROBLEM_STATUS_CODE)         /*  PROBLEM STATUS       */
          FIELD(S0BEE)                      /*  CODE                 */
          DEFAULT(INITIAL);

  6 ALIAS NAME(PROBLEM_TYPE_CODE)           /*  PROBLEM TYPE         */
          FIELD(S0C09);                     /*  CODE                 */

  7 ALIAS NAME(PROBLEM_ABSTRACT)            /*  PROBLEM ABSTRACT     */
          FIELD(S0E0F)                      /*  DATA                 */
          DEFAULT(SYSTEM OPENED PROBLEM);

  8 ALIAS NAME(GENERIC_PHONE_PREFIX)        /*  GENERIC PHONE PREFIX */
          PREFIX(P0013);                    /*  PREFIX               */

  9 ALIAS NAME(DATE_OCCURRED_PREFIX)        /*  DATE OCCURRED PREFIX */
          PREFIX(P00E4);                    /*  PREFIX               */

 10 ALIAS NAME(DESCRIPTION_TEXT_DDNAME)     /*  DESCRIPTION TEXT     */
          INDEX(S0E01);                     /*  DDNAME               */

 11 ALIAS NAME(SAMPLE_DATA_ITEM)            /*  SAMPLE DATA ITEM     */
          FIELD(P1234)                      /*  USING PREFIX         */
          DEFAULT(SAMPLE);

 12 ALIAS NAME(PROBLEM_CREATE_TABLE)        /*  PROBLEM CREATE       */
          TABLE(BLGYPRC);                   /*  PIDT ALIAS NAME      */

 13 ETABLE;
```

```
NO ERRORS DETECTED DURING TABLE BUILD PROCESSING FOR ALIAS TABLE ALIASONE

ALIAS SPECIFICATION SUMMARY FOR ALIAS TABLE (PALT): ALIASONE
TABLE USE: ALIAS, ALIASES DEFINED: 11
      EXTERNAL                 INTERNAL    PREFIX   TABLE     DEFAULT
        NAME                     NAME      DATA     NAME       DATA

PROBLEM_REPORTER_NAME            S0B59                        SYSTEM
PROBLEM_REPORTER_PHONE           S0B2D
PROBLEM_REPORTER_LOCATION        S0C0C
PROBLEM_STATUS_CODE              S0BEE                        INITIAL
PROBLEM_TYPE_CODE                S0C09
PROBLEM_ABSTRACT                 S0E0F                        SYSTEM OPENED
GENERIC_PHONE_PREFIX             P0013     PH/
DATE_OCCURRED_PREFIX             P00E4     DATO/
DESCRIPTION_TEXT_DDNAME          S0E01
SAMPLE_DATA_ITEM                 P1234                        SAMPLE
PROBLEM_CREATE_TABLE                                BLGYPRC
```

24

BLGUT9—Set Database Options

Use the BLGUT9 utility to set the database options defining the uniqueness of record number IDs created, the p-word used for the logical database partition, and whether you want to reuse SDDS position numbers (also called root VSAM keys or VSAM sequence numbers) for newly created records. The options selected are stored as a flag in the control record of the SDIDS and apply to all new records created.

Prior to TME 10 Information/Management Version 1.1, as records were deleted from the database, the SDDS position numbers for those records were reused for new records created in the database. Starting with Version 1.1, SDDS position numbers are not reused as records are deleted, unless you take specific action through this utility to reuse them. The default option of this utility allows the database to remain in system-assigned RNID order after records are deleted.

If you do not reuse SDDS position numbers, newly created records with system-assigned record number IDs (RNIDs) will appear in the database in RNID order. Because the records are added in chronological order, records with user-assigned RNIDs will be intermingled with the system-assigned RNIDs which will be in order. Records will display in ascending order in a search results list, eliminating the need for many users to sort the search results list by RNID. In some cases, this can help to improve search performance because it reduces contention for the SDIDS.

If you reuse SDDS position numbers, records displayed in a search results list will not necessarily appear sorted by RNID, and users may have to sort the records if they want to see them in RNID order.

You can use the BLGUT9 utility to set the option in either an empty database or one that already contains data. Databases that start off empty can remain in RNID order with system-assigned SDDS position numbers as records are created. Databases that are not empty can be sorted first in RNID order, so that as new records are created, all records in the database will appear in RNID order. If an established database is not sorted first, then, only newly created records will appear in RNID order. User-assigned RNIDs will still remain unsorted.

If your database is set up to reuse SDDS position numbers and you subsequently run the BLGUT1 utility to rebuild the SDIDS, you must either specify the REUSE parameter when running BLGUT1 or you must run the BLGUT9 utility again to set the option to REUSE.

If your existing database is not in RNID order, you can rebuild it in RNID order. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for instructions on creating a database sorted by record number.

The BLGUT9 utility can be run while the database is in use; however, users will notice the new setting only after they log on to Tivoli Information Management for z/OS after the BLGUT9 utility is run.

If you do not reuse SDDS position numbers and are concerned about wasting physical space used by deleted records, be aware that you can reclaim this space by reorganizing the database with IDCAMS REPRO.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Syntax

Use BLGUT9 to set database options. The EXEC statement PARM field keywords to use with BLGUT9 are as follows:

```
PARM=‘SESS=aa[,NAME=n][,UNIQUE][,PIDPWORD=pword][,REUSE]’
```

SESS=aa
Specifies the suffix for the session-parameters member name that BLGUT9 uses. Use 1 or 2 alphanumeric or national characters for *aa*. This session-parameters member contains the structure definition of the database to be used. This keyword is required.

NAME=n
Identifies which database (in a session-parameters member containing multiple database definitions) to use. This keyword is optional, and if omitted, defaults to 5 for the Tivoli Information Management for z/OS database. Valid values are 4, 5, 7, or 8 as the database.

UNIQUE
Indicates that you want the record number ID incremented regardless of what logical database partition is being used. UNIQUE specifies that the RNIDs will be unique across partitions. This keyword is optional, and if omitted, the RNIDs generated will *not* be unique and duplicates will occur (for example, partition #1 will have RNID 00000001, partition #2 will have RNID 00000001, and so on).

If your installation uses a p-word other than the default value of PTID/ (or PTID_) for the logical database partition, you must also specify the PIDPWORD when if you use the UNIQUE keyword.

PIDPWORD=pword
Specifies the p-word used for the logical database partition if the default partition prefix is not PTID/ or PTID_. This keyword is optional. The default value of PIDPWORD is PTID/. Specify a p-word only if you are not using the default partition prefix value of PIDT/ or PIDT_ for your owning logical database partition name. (A default partition prefix is specified through use of the BLG01448 program exit, as described in the *Tivoli Information Management for z/OS Panel Modification Facility Guide*).

If your installation uses PTID/ or PTID_ as the default partition prefix value, you can omit the PIDPWORD keyword. This keyword enables the BLGUT9 utility to use the correct logical database partition when it runs.

When entering the p-word value, include the forward slash (or underscore character) as appropriate. The p-word value specified here must match the p-word used with the BLG01448 program exit when the logical database partition was created. Only one p-word can be specified with the PIDPWORD keyword. If you specify this keyword, you must also specify the UNIQUE keyword.

REUSE
Indicates that you want to reuse SDDS position numbers when creating records on the specified database. This keyword is optional, and if omitted, the SDDS position numbers are not reused when records are created.

Input

The input to BLGUT9 is Database 5 or the database specified by the NAME keyword.

Output

The output from BLGUT9 is:

- The database option is set as requested.
- Summary and error messages in a message data set.
- A return code of 0 if the utility finishes successfully.
- A return code of >0 if an error occurs.

Restrictions

If you use the UNIQUE keyword, you must run BLGUT9 with this keyword whenever you rebuild the SDIDS. Otherwise, RNID integrity is not maintained.

DD Statements

The DD statements required to run BLGUT9 are as follows:

SYSPRINT
A sequential data set (RECFM=VBA) containing summary and error messages that you can write to a system output device or data set.

Scenario Using the BLGUT9 Utility Program

The following example illustrates the use of the BLGUT9 utility to set the option to reuse SDDS position numbers. The session-parameters member suffix is specified to provide the database structure. The default database name of 5 is used.

```
//REUSE      JOB
//*         SET THE DATABASE OPTION TO REUSE SDDS POSITION NUMBERS
//STEP1      EXEC PGM=BLGUT9,PARM='SESS=01,REUSE',REGION=2048K
//STEPLIB    DD DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT   DD SYSOUT=A
```

Figure 14. Using the BLGUT9 Utility.

25

BLGUT10—Convert the Case of Data

The BLGUT10 utility enables you to convert the case of data stored in records in the database. For example, you can change the case of data stored in the Description abstract field of your problem records from:

```
RESPONSE TIME TOO SLOW
```

to:

```
Response time too slow
```

The utility enables you to define the records and fields to update and the type of conversion required for each field.

You can convert existing data to any of these specifications:

- Uppercase
- Lowercase
- According to a validation pattern

Note: The BLGUT10 utility will not convert freeform text or history data.

The BLGUT10 utility can run against a database while other users are using Tivoli Information Management for z/OS on the same database. Because BLGUT10 automatically updates both the SDDS and SDIDS, you do not have to run the BLGUT1 database index utility, or any other Tivoli Information Management for z/OS utility, when using BLGUT10.

The BLGUT10 utility is a low-level API (LLAPI) job that makes use of dynamic program interface data tables (PIDTs), and works in a manner that is similar to the Tivoli Information Management for z/OS Archiver function which is used to manage records on the database. If you are familiar with the Archiver function, you will notice that the user interface is very similar.

The BLGUT10 utility logs the following:

- Parameters that you specified when you invoked BLGUT10
- Messages related to each record processed
- A summary of what the utility has successfully done
- LLAPI activity
- Internal analysis of each record processed, with details of all errors encountered

Before running the BLGUT10 utility, you should understand more about storing data in the SDDS and SDIDS in mixed case format. For additional information on setting up and managing mixed case data, refer to:

- *Tivoli Information Management for z/OS Application Program Interface Guide* for a description of how the LLAPI uses the data attribute fields in the program interface data table to define how mixed case data is handled
- *Tivoli Information Management for z/OS Panel Modification Facility Guide* for a description of how you set the options that govern how mixed case data is validated and stored in Tivoli Information Management for z/OS
- *Tivoli Information Management for z/OS User's Guide* for information about searching mixed case data

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Also, the database must be accessed as Database 5, read/write.

Syntax

The functions of BLGUT10 are controlled by parameters passed to it from an input data set at invocation. The parameters data set is read and processed sequentially. The parameters you specify define such items as the number of records to process and the search argument to be used for inquiry. These parameters enable you to initialize and control the use of the program interface communications area (PICA), which in turn controls the functions available within the LLAPI.

The parameters to use with BLGUT10 are as follows:

```
CLASS=privilege class
SESSION=member name
[TIMEOUT=00000300]
[MODEL=GENERIC]
INQPIDT=name
[DYNMPIDT=BLGYDYN]
[LOGGING=N]
[ARG=argument]
[APPLID=CASECONV]
[HITS=00000001]
[CHECKIO=Y]
```

The input parameters that you can specify with BLGUT10 are listed below. Note the following expected conditions:

- Parameter specifications begin in column 1.
- Comment line follows '/*' which starts in column 1.
- Comments following a parameter specification begin in column 40 (if specified).

- Each line contains a parameter or comment line; blank lines are not allowed.

CLASS=
Defines the privilege class name that must exist on the database. The privilege class will be used for the entire session. The class must have display, update, and database administration authority for the records being processed.

This is a required parameter, and no default value is supplied. It is used to set the value of the PICACLSN field for the API transactions that will be performed. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for additional information on the PICACLSN field used in LLAPI database access transactions.

SESSION=
Specifies the session-parameters member name that BLGUT10 uses to access the database, for example, BLGSES00. The database must be accessed as Database 5, the read/write database. This is a required parameter, and no default value is supplied.

TIMEOUT=
The number of seconds in which the transaction must complete. If you specify a value of 0 or omit this field, the value is set to 00000300 seconds (5 minutes). This value sets the PICATINT parameter, which is defined in the *Tivoli Information Management for z/OS Application Program Interface Guide*. The value must be 8 characters in length, containing leading zeroes as necessary.

MODEL=
Defines the name (the record number ID) of the data view record stored on the database that will control the conversion process. This is an optional parameter. The default value is GENERIC.

INQPIDT=
Defines the name of the inquiry PIDT used to locate the records to be processed. This PIDT is used in association with the search arguments specified in the ARG= parameters. This is a required parameter. There is no default value.

DYNMPIDT=
Defines the name of the dynamic PIDT used by the database retrieve and update transactions. The default value is BLGYDYN. The PIDT BLGYDYN is supplied in the SBLMFMT library for this purpose. The PIDT specified in this parameter should contain only the USE(HEADER) parameter.

LOGGING=
Defines whether to perform API logging. Valid values are:

Y A value of Y causes full logging to the APIPRINT data set.

N A value of N suppresses logging.

The default value is N. This parameter sets the PICAMSGD parameter to P. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for additional information.

ARG=
Defines up to 10 freeform arguments. The maximum length of each argument is 33 characters. The arguments specified are used in addition to the inquiry PIDT. Each argument must be prefixed by an ARG= statement. A period (.) at the end of an argument specifies an abbreviated search. A Boolean operator, if specified, must be in the first position of the argument. An argument cannot contain leading or imbedded

blanks. Examples of arguments are as follows:

`ARG=RELIEF`	Example showing how to find a word on a description abstract line

`ARG=DATE/2000/01/01`
`ARG=-DATE/2000/12/31`
	Example showing how to perform a date range

`ARG=PRIO/1`
`ARG=|PRIO/2`
	Example showing a search for priority 1 or priority 2

`ARG=¬DEPT/T53B`	Example of using the NOT sign
`ARG=COMD/3390.`	Example of using an abbreviated argument
`ARG=DATO/2000/**/**`	Example showing how to look for anything in 2000

Th ARG parameter is optional. No default value is supplied.

APPLID=
Defines the application identifier (APPLID) of the BLGUT10 utility. The application identifier must exist in the privilege class defined by the CLASS= parameter. This parameter is optional. The default value is CASECONV.

HITS=
Defines the number of records to be processed. The value must be between 00000001 and 99999999, and must be specified as an 8-digit decimal number. This parameter controls execution times by limiting the number of records to be processed. This parameter is optional. The default value is 00000001.

CHECKIO=
Defines whether or not to perform check-in/check-out for each record processed. This parameter allows you to check out records on the database before starting the conversion process. The valid values for CHECKIO are as follows:

Y Specifies that records should be checked out before conversion processing starts, and checked in when the process completes.

N Specifies that no check-in or check-out processing is requested.

This parameter is optional. The default value is Y.

Setting CHECKIO=Y prevents a user from updating a record on the database between the retrieve and update transactions. It is recommended that you specify CHECKIO=Y if there is a chance that the records to be updated may be undergoing updates by users at the same time. (In contrast, when you use the Archiver, it is not recommended that you specify CHECKIO=Y because you do not expect records that you intend to archive to still be undergoing updates.)

Defining the Input Parameters

The next example illustrates how you could set up your input parameters data set to convert up to 1000 problem records that were encountered in 2000 with a status of not CLOSED. The SESSION parameter defines the database where the records reside. The INQPIDT parameter specifies BLGYPRI as the PIDT to use so that the inquiry can be performed on problem records.

Refer to member BLGUT10P in the SBLMSAMP data set for a sample parameters data set that you can modify to suit your needs.

```
/*-------------------------------------------------------------------*/
/*                                                                   */
/*                 Licensed Materials - Property of IBM              */
/*                                                                   */
/*                      Program Number:  5697-SD9                    */
/*                                                                   */
/*                 (c) Copyright IBM Corp. 1981, 2001.               */
/*                                                                   */
/*                      See Copyright Instructions                   */
/*                                                                   */
/*-------------------------------------------------------------------*/
CLASS=SYSADMIN
APPLID=CASECONV
SESSION=BLGSES00
CHECKIO=N
LOGGING=N
ARG=¬STAC/CLOSED
ARG=DATE/2000/01/01
ARG=-DATE/2000/12/31
MODEL=GENERIC
HITS=00001000
INQPIDT=BLGYPRI
DYNMPIDT=BLGYDYN
```

Figure 15. Sample Input Data Set (BLGCASEP) to Convert the Case of Data in Closed Problem Records (member BLGUT10P in SBLMSAMP)

As shown in Figure 16 on page 132, sample JCL is provided for you to run the BLGUT10 utility. Refer to member BLGUT10J in the SBLMSAMP library. In this sample, BLGCASEP is the input parameters data set you specify as input to the BLGUT10 utility.

```
//BLGUT10 JOB
//*-------------------------------------------------------------------*
//*                                                                   *
//*           Licensed Materials - Property of Tivoli Systems        *
//*                                                                   *
//*                              5697-SD9                             *
//*                                                                   *
//*   (C) Copyright Tivoli Systems Inc., an IBM Company 1997, 2001.   *
//*                                                                   *
//*                      See Copyright Instructions                   *
//*                                                                   *
//*-------------------------------------------------------------------*
//*                                                                   *
//* This is a sample jobstream for running the Case                   *
//* Converter, BLGUT10.                                               *
//*                                                                   *
//*-------------------------------------------------------------------*
//CONVERT EXEC PGM=BLGUT10,REGION=6M
//*
//* Do not forget to STEPLIB to your session load library if needed.
//*
//STEPLIB DD DISP=SHR,DSN=BLM.SBLMMOD1
//*
//* BLGCASEP DD is the input parameters data set and is required.
//*
//BLGCASEP DD DISP=SHR,DSN=????????.BLGUT10P
//*
//*RFTDD DD HAS YOUR PIDT EXECUTABLE TABLES
//*
//RFTDD DD DISP=SHR,DSN=BLM.SBLMFMT
//APIPRINT DD SYSOUT=*
//*
//* BLGOUT DD is the data set where Case Converter messages are routed.
//*
//BLGOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Figure 16. Sample JCL to Run BLGUT10 (Member BLGUT10J in SBLMSAMP Data Set)

Input

The input to BLGUT10 is as follows:

- A data set containing the parameters necessary to identify and update selected records on the database
- A data set containing the static PIDTs and model PIDTs (those with the USE(HEADER) parameter) used to generate the dynamic PIDTs
- Data model records (existing on the database) that are needed to build a conversion table

Output

The output from BLGUT10 is:

- Converted records in the SDDS and an updated SDIDS.
- Summary and error messages in a message data set.
- LLAPI messages in an APIPRINT data set (if LOGGING=Y is specified as an input parameter).
- A return code of 0 if the utility finishes successfully.
- A return code of 4 if the utility encounters a non-zero PIDTCODE, or PIRTCODE. The record in error is skipped and processing continues until all of the requested records are processed.

Return code 4 can also result if the utility encounters a non-zero return code from the validation module BLGPPFVM. The PIDT row in error is skipped and processing continues with the next PIDT row for that record.

Messages are generated to describe the exact failure. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for additional information on return codes associated with the validation module BLGPPFVM.

- A return code of 8 if the utility encounters an error and cannot continue. Messages are generated to describe the exact failure.

Restrictions

You can use the BLGUT10 utility to convert the case of single-byte character set (SBCS) data only. You cannot use it to convert double-byte character set (DBCS) data.

The following list of data types can be converted:

- Prefixed, non-prefixed, and multiple response prefixed data collected through assisted-entry panels
- String data (including data cognized as unparsed strings)
- Single response list processor data

The following list of data types cannot be converted:

- Direct-add data
- Multiple response list processor data
- History data
- Freeform text

Although the BLGUT10 utility can convert the case of data cognized as unparsed strings, it cannot convert normal cognized string data into cognized unparsed string data.

DD Statements

The DD statements required to run BLGUT10 are as follows:

APIPRINT
The output data set containing LLAPI messages (if LOGGING=Y was specified). The *Tivoli Information Management for z/OS Application Program Interface Guide* contains additional information on the APIPRINT data set.

BLGCASEP
A sequential data set or a member of a partitioned data set (with RECFM=FB and LRECL=80) containing input parameters. Parameters may also be specified in-stream, if desired.

BLGOUT
A sequential data set (RECFM=FB, LRECL=80) containing summary and error messages that you can write to a system output device or data set.

RFTDD
The name of the partitioned data set containing the input PIDTs.

Processing Overview

The BLGUT10 utility uses two program interface data tables (PIDTs) to control the database inquiry and update functions. The sample input parameters data set (BLGUT10P) shown in Figure 15 on page 131 uses the inquiry PIDT for problem records, BLGYPRI, which is a member of the SBLMFMT data set. The source code for this PIDT and other PIDTs can be found in the SBLMSAMP library.

The SBLMFMT and SBLMSAMP libraries are described in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*. You should review the sample PIDTs to ensure they contain the values you need to locate the records you wish to process. If they do not, you should create your own PIDTs, modeled on the samples. For more information on modifying PIDTs, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.

The static PIDT used for inquiry can contain a high-level inquiry only, such as the record type. The specific search argument can be passed to BLGUT10 as a parameter list to form a freeform search.

The dynamic PIDT used to retrieve a record and to update it after the conversion is built dynamically by the LLAPI and automatically represents an exact data map of the retrieved record, excluding freeform text and history data. The update PIDT is based on a generic PIDT which is a PIDT that has only the header statement.

Both the static PIDT and the generic header PIDT used to build the dynamic PIDT are in the SBLMFMT partitioned data set.

The BLGUT10 utility uses a specific search argument to determine what records should be processed. An inquiry is performed using the inquiry PIDT with the freeform argument specified by the ARG= parameters in the parameter data set (BLGCASEP). The values supplied by the ARG= parameters form the freeform search argument for the inquiry. The inquiry PIDT must contain the structured word (s-word) of the record type to be processed.

The BLGUT10 utility uses data model records—a data view record and its associated data attribute records— to build a conversion table. Once you determine what fields you would like to convert, you must create a corresponding data attribute record for each of those fields. (For specific instructions on setting up data model records, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.) You need to specify the type of mixed case conversion to perform on each field selected in the data attribute record. After all of the data attribute records have been created, you must create a data view record to group together the list of data attribute records to be used during the conversion process.

As each record to be converted is retrieved and its PIDT is processed row by row, the BLGUT10 utility compares the s-word associated with the data in the record with all of the s-words in the conversion table. If the s-words match, BLGUT10 calls the validation module, BLGPPFVM, and passes to it the data to be converted. The validation module uses the conversion table to compare the s-word index in the record to the s-word index in the conversion table to ensure the correct item is being converted. The validation module then converts the data in the record based on the following data attribute fields:

- Case sensitive validation — PIDT field PIDTCSVL
- Collected data case — PIDT field PIDTCDCA

- Cognize in mixed case — PIDT field PIDTCGMX

Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for additional information on PIDT fields used in LLAPI transactions.

If the PIDT row or data within the record contains an associated s-word index of ’S0000’ (to indicate, for instance, that the assisted-entry panel for a commonly used field collects the s-word index from the calling data entry panel), substitution of the s-word index value occurs. The BLGUT10 utility automatically substitutes the correct s-word index as defined in the data attribute record, and passes this index to the validation module BLGPPFVM.

The BLGUT10 utility updates the requested records in the database and automatically checks them back in on the database when parameter CHECKIO=Y is specified.

The BLGUT10 utility logs the start-up parameters you specified, messages related to each record processed, and a completion summary. Logging is controlled by the LOGGING= parameter specified in the input parameter data set. Messages are written to the data set defined by the APIPRINT DD statement, assuming that these data sets are allocated and LOGGING=Y is specified.

Record Processing

To use the BLGUT10 utility, you should understand how the LLAPI processes the data in your records. When the LLAPI retrieves a record from the database, a dynamic PIDT is created. Each PIDT consists of a header and multiple data rows that correspond to the data in your records. For the sake of simplicity, you can use the output from the VIEW INTERNALS command to understand how PIDT data rows are loaded. Essentially, the lines of output from the VIEW INTERNALS command are stored line by line from the top to the bottom into the PIDT data rows. The BLGUT10 utility assigns an entry number to each row; this entry number corresponds to the VIEW INTERNALS line number. The SDDSROOT entry in the VIEW INTERNALS output is not processed, and freeform text entries (one per freeform text type) are not retrieved.

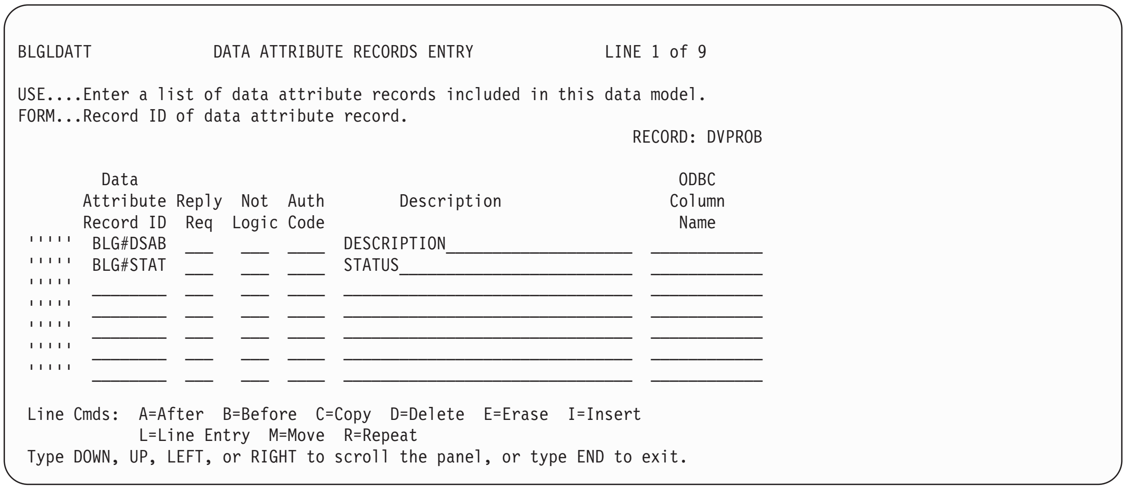
```
BLG1TVID                     VIEW INTERNAL DATA                LINE 1 OF 30

PANEL    PANEL  REL COG-   FLAGS    SWORD  STRUCTURED    PREFIX WORD OR
NAME    TYP/RSP LEV NIZE F  M  D    INDEX     WORD       VISIBLE PHRASE

BLG0EN20  D/ 5  12  B/   19/00/88  S002C   BA //S/TSI   ENTRY
SDDSROOT  C/    00  N/   12/04/10  S0000                00003E44
BLG00000  S/ 1  12  B/   18/04/00  S0032   BA //S/TXS   RECS=PROBLEM
BLG0B001  S/ 1  12  B/   19/00/00  S0CFC   BC IMDIAENT0 REPORTER
BLG0B100  D/ 1  12  N/   41/00/00  S0B59
BLG6REQN  A/ 2  12  B/U* 0D/04/00  S0B59   BC IM0I0PS00 PERS/ANY/ONE
BLG0B100  D/25  12  N/   41/00/00  S0E0F
BLG6DSAB  A/ 2  12  B/   0D/04/00  S0E0F   BC IM0TXCA00 TEST
BLG0B100  D/14  12  N/   41/00/00  S0BEE
BLG6STAT  A/ 2  12  B/U  0D/04/00  S0BEE   BC IMS0SSC00 STAC/CLOSED
BLG0B100  D/13  12  N/   41/00/00  S0C09
BLG6PTYP  A/ 2  12  B/U  0D/04/00  S0000   BC IMGCT0000 TYPE/USER
BLG0B100  D/ 4  12  N/   41/00/08  S0C3D
BLG6OCCD  A/ 2  12  B/U  0D/04/00  S0C3D   BC IM00SD001 DATO/10/21/96
```

Figure 17. VIEW INTERNALS Output

Scenario Using the BLGUT10 Utility Program

Setting up the Data Model Records

As mentioned previously, the BLGUT10 utility uses a data view record and associated data attribute records to build the conversion table that is used to convert your data. Figure 18 shows an example of a data view record. In the following scenario, two data attribute records are created: one to convert the case of data in the problem Description (abstract) field, and the other to convert the case of data in the Problem Status field. These two records contain the information needed by the utility to successfully build the conversion table and to perform the conversion process. These data attribute records must exist on the database for BLGUT10 to convert the data for these two fields successfully. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for additional information on setting up data model records.

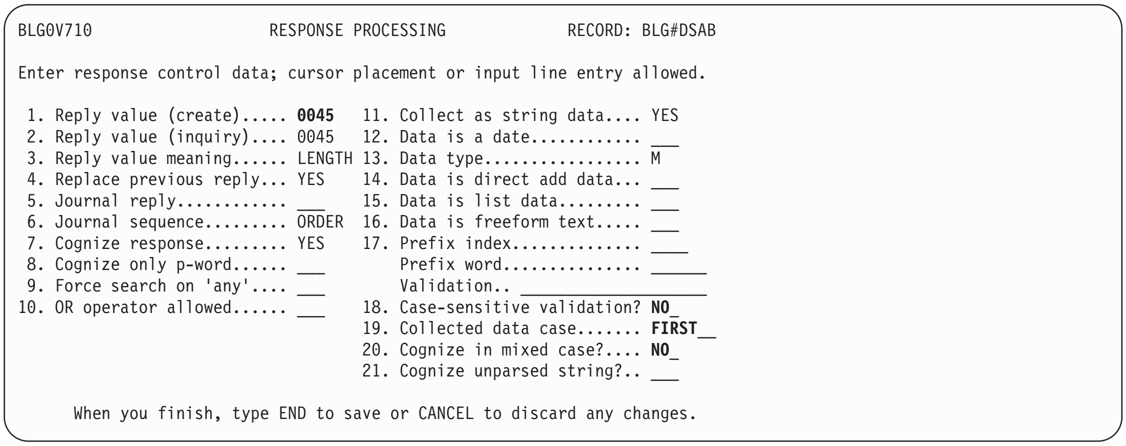


```
BLGLDATT             DATA ATTRIBUTE RECORDS ENTRY             LINE 1 of 9

USE....Enter a list of data attribute records included in this data model.
FORM...Record ID of data attribute record.
                                                                 RECORD: DVPROB

         Data                                                     ODBC
       Attribute Reply  Not  Auth        Description             Column
       Record ID  Req  Logic Code                                 Name
 '''''  BLG#DSAB  ___   ___  ____  DESCRIPTION__________________  ____________
 '''''  BLG#STAT  ___   ___  ____  STATUS_______________________  ____________
 '''''  ________  ___   ___  ____  _____________________________  ____________
 '''''  ________  ___   ___  ____  _____________________________  ____________
 '''''  ________  ___   ___  ____  _____________________________  ____________
 '''''  ________  ___   ___  ____  _____________________________  ____________
 '''''  ________  ___   ___  ____  _____________________________  ____________

 Line Cmds:  A=After  B=Before  C=Copy  D=Delete  E=Erase  I=Insert
             L=Line Entry  M=Move  R=Repeat
 Type DOWN, UP, LEFT, or RIGHT to scroll the panel, or type END to exit.
```

Figure 18. Example of a Data View Record

The BLGUT10 utility uses the following six fields from the data attribute records. All other data attribute fields are ignored by BLGUT10.

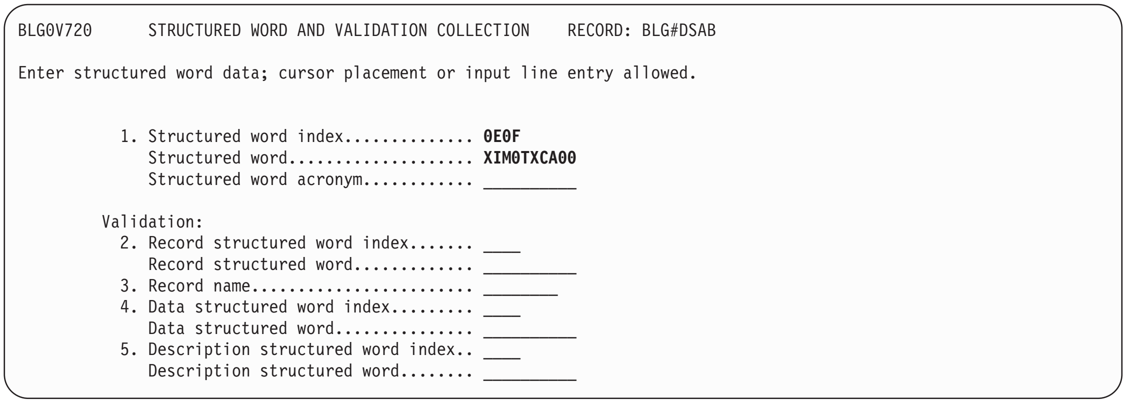
- S-word
- S-word index
- Case sensitive validation flag
- Collected data case flag
- Cognized in mixed case flag
- Reply value (create) — with non-string data



```
BLG0V710                 RESPONSE PROCESSING            RECORD: BLG#DSAB

Enter response control data; cursor placement or input line entry allowed.

 1. Reply value (create)..... 0045    11. Collect as string data.... YES
 2. Reply value (inquiry).... 0045    12. Data is a date........... ___
 3. Reply value meaning...... LENGTH  13. Data type................ M
 4. Replace previous reply... YES     14. Data is direct add data... ___
 5. Journal reply........... ___      15. Data is list data......... ___
 6. Journal sequence......... ORDER   16. Data is freeform text..... ___
 7. Cognize response......... YES     17. Prefix index............. ____
 8. Cognize only p-word...... ___         Prefix word.............. ______
 9. Force search on 'any'.... ___         Validation.. ___________________
10. OR operator allowed...... ___     18. Case-sensitive validation? NO_
                                      19. Collected data case....... FIRST__
                                      20. Cognize in mixed case?.... NO_
                                      21. Cognize unparsed string?.. ___

      When you finish, type END to save or CANCEL to discard any changes.
```

Figure 19. Data Attribute Record for Description Field (BLG#DSAB) — Part 1

Figure 19 illustrates how the response processing is defined for the Description abstract field. As part of setting up the data attribute record for the Description abstract field, you would also fill in the structured word index field on the following panel:



```
BLG0V720      STRUCTURED WORD AND VALIDATION COLLECTION    RECORD: BLG#DSAB

Enter structured word data; cursor placement or input line entry allowed.


           1. Structured word index............. 0E0F
              Structured word................... XIM0TXCA00
              Structured word acronym........... _________

         Validation:
           2. Record structured word index....... ____
              Record structured word............ _________
           3. Record name....................... _______
           4. Data structured word index......... ____
              Data structured word.............. _________
           5. Description structured word index.. ____
              Description structured word........ _________
```

Figure 20. Data Attribute Record for Description Field (BLG#DSAB) — Part 2

As a result, in this example, the Description abstract field (a 45-character field) of the problem record will be converted to a data case of FIRST (first character uppercase, all others lowercase. The data will not be cognized (stored in the SDIDS for searching) in mixed case; it will be cognized in uppercase (the default). When the BLGUT10 utility runs, it will convert a description abstract of:

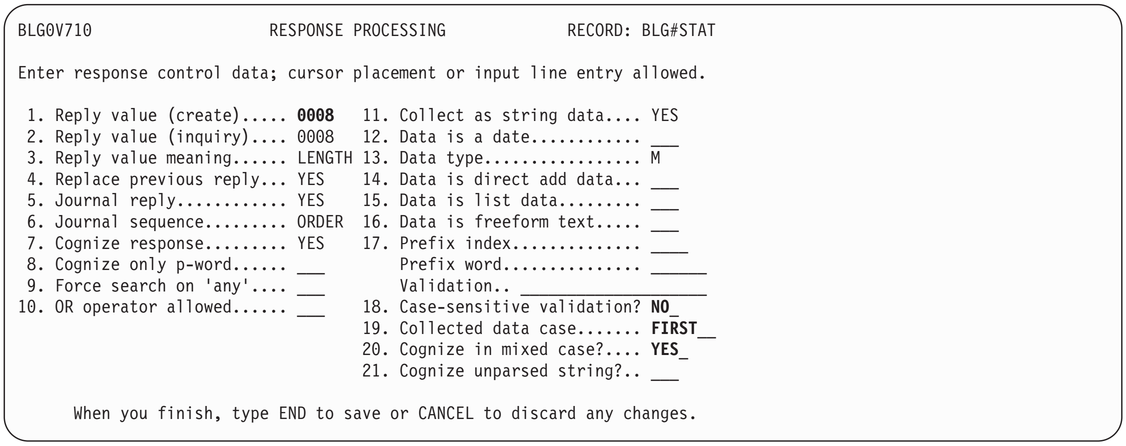
```
UNABLE TO REBOOT SERVER AFTER UPGRADE
```

to:

```
Unable to reboot server after upgrade
```

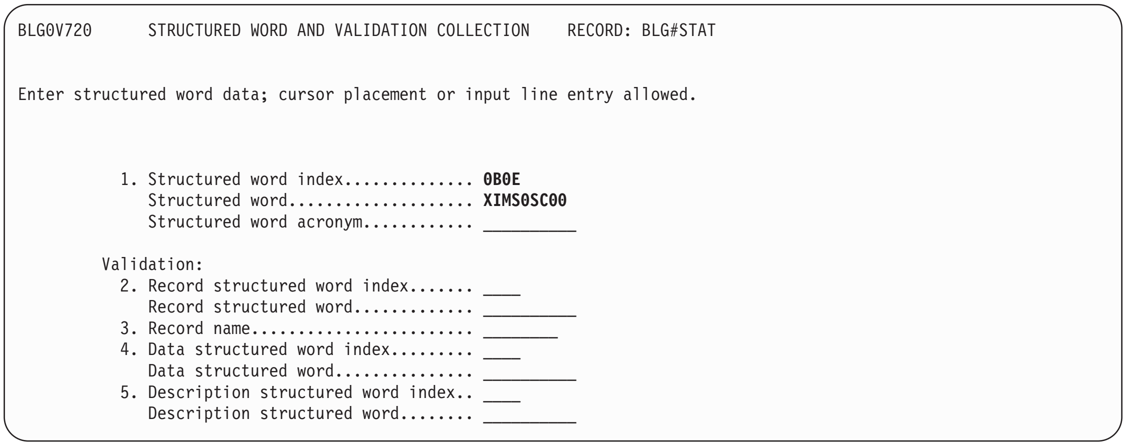
and store it in the SDDS. The data will be stored in the SDIDS (for searching) in all uppercase.

Now that you have defined a data attribute record for the Description field, you can define a data attribute record for the Problem Status field in a similar manner, as shown Figure 21 and Figure 22.



```
BLG0V710                  RESPONSE PROCESSING            RECORD: BLG#STAT

Enter response control data; cursor placement or input line entry allowed.

 1. Reply value (create)..... 0008   11. Collect as string data.... YES
 2. Reply value (inquiry).... 0008   12. Data is a date........... ___
 3. Reply value meaning...... LENGTH 13. Data type................ M
 4. Replace previous reply... YES    14. Data is direct add data... ___
 5. Journal reply............ YES    15. Data is list data......... ___
 6. Journal sequence......... ORDER  16. Data is freeform text..... ___
 7. Cognize response......... YES    17. Prefix index............. ____
 8. Cognize only p-word...... ___        Prefix word.............. ______
 9. Force search on 'any'.... ___        Validation.. ___________________
10. OR operator allowed...... ___    18. Case-sensitive validation? NO_
                                     19. Collected data case....... FIRST__
                                     20. Cognize in mixed case?.... YES_
                                     21. Cognize unparsed string?.. ___

      When you finish, type END to save or CANCEL to discard any changes.
```

Figure 21. Data Attribute Record for Problem Status Field (BLG#STAT) — Part 1



```
BLG0V720      STRUCTURED WORD AND VALIDATION COLLECTION    RECORD: BLG#STAT


Enter structured word data; cursor placement or input line entry allowed.



           1. Structured word index............. 0B0E
              Structured word................... XIMS0SC00
              Structured word acronym........... _________

         Validation:
           2. Record structured word index....... ____
              Record structured word............ _________
           3. Record name....................... ________
           4. Data structured word index......... ____
              Data structured word.............. _________
           5. Description structured word index.. ____
              Description structured word........ _________
```

Figure 22. Data Attribute Record for Problem Status Field (BLG#STAT) — Part 2

The 8-character Problem Status field data will be converted to a data case of FIRST and cognized in mixed case. As a result, a status of:

```
CLOSED
```

would be converted and stored in the SDDS as:

```
Closed
```

and stored in the SDIDS (for searching) in mixed case.

Performance Considerations

You can improve the performance of BLGUT10 by following this list of recommendations:

- Do not code STEPLIB DD statements in your JCL unless they are required. Place all Tivoli Information Management for z/OS modules and session members in the linklist or linkpack area. (Refer to the *Tivoli Information Management for z/OS Planning and*

Installation Guide and Reference for additional information.) A STEPLIB DD statement will cause all calls to load modules to access the STEPLIB DDNAME in an attempt to locate the modules, regardless of whether or not they reside in the data set(s).

- Disable CHECKIO processing unless you specifically require records to be checked out before retrieval and processing.
- If you do not require LLAPI logging, specify LOGGING=N in the input parameters data set.
- Consider defining a separate session-parameters member for the BLGUT10 utility with different characteristics than those used by interactive users. Refer to the description of the COGENQ parameter in the BLGCLUST macro in the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

Panel Processing Considerations

If you are planning to use BLGUT10, you should also consider the following:

- You need to modify the API TSP BLGAPI05 to enable special processing. Delete line 17 which says `BRANCH DISABLED`.
- Ensure notification processing is removed. If you use TSPs and TSO Send for notification, ensure that user exits are in place for the problem and/or change record file panels to prevent TSPs from running during API processing. If you use TSXs for e-mail notification, disable the calls to the TSXs in copies of the record file panels used by the API.
  - BLG1A111 — Problem Record File
  - BLG1A121 — Change Record File
- Only one record type can be converted at a time. Use one of the following values for INQPIDT in the input parameters list:
  - BLGYACI — change activity records
  - BLGYCHI — change records
  - BLGYDCI — data center records
  - BLGYHCI — configuration hardware component records
  - BLGYHFI — configuration hardware component feature records
  - BLGYHNI — configuration financial hardware records
  - BLGYHSI — configuration hardware sub-component records
  - BLGYHXI — configuration hardware component connection
  - BLGYPRI — problem records
  - BLGYSCI — configuration software component records
  - BLGYSFI — configuration software component feature records
  - BLGYSNI — configuration financial software records
  - BLGYSVI — configuration service records
  - BLGYSXI — configuration software component connection records
  - BLGYSYI — configuration system records

If you created your own record types, you will need to create a unique inquiry PIDT to accommodate your records. Use the samples in the SBLMSAMP library as a model.

- Since the BLGUT10 utility is a LLAPI job, it will be affected by the setting of the APISECURITY keyword in the BLX-SP parameters member. The APISECURITY keyword specifies whether to activate the BLX-SP security checking for APIs to validate that the MVS application user ID is allowed to access a Tivoli Information Management for z/OS database. For more information on this keyword, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* and the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

26

BLGUT17—Convert Date Data

Use the BLGUT17 utility to convert date data in the input SDDS to internal date format. The converted records are written back to the same database. You can run the BLGUT17 utility while users are actively using Tivoli Information Management for z/OS to create, update, or delete records on the database. **You cannot use the BLGUT17 utility to convert your dates into universal time format.** For information about enabling universal time processing, see Appendix A of the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*.

The BLGUT17 utility converts data identified as date data and data associated with p-words that begin with the letters 'DAT' (for example, DATE/ or DATO/). SDDS records are processed, one at a time. As date data is identified, it is passed to the default date conversion routine, BLGCDATS. The session-parameters member specified in the SESS= parameter is used to determine the external date formats to be converted. Date data appearing in the default or old external date format is converted to the internal date format. Therefore, both the default and old external date formats should be specified on the DATEFMT and ODATEFMT keywords for the session-parameters member.

Important!

If your organization meets *both* of the following criteria:

- you have old date records in which all three variables are of the same length, such as MM/DD/YY, DD/MM/YY, or YY/MM/DD
- you are using a date format in which all three variables are of the same length (such as MM/DD/YY, DD/MM/YY, or YY/MM/DD) as the ODATEFMT value for the session-parameters member

then the BLGUT17 utility may not determine correctly whether the date it is converting is of the format MM/DD/YY, DD/MM/YY, or YY/MM/DD; the result may be that the BLGUT17 conversion is inaccurate.

For this reason, if your organization does meet these two criteria, it is suggested that you use BLGUT23B to create a backup copy of your SDDS before you run the BLGUT17 utility. (For information on running the BLGUT23B utility, see "BLGUT23B—Back Up the Database" on page 173.) You can then return to this backup in the event that running the BLGUT17 utility produces inaccurate date conversions.

The BLGUT17 utility also converts data in date fields that are journalized—tracked by Tivoli Information Management for z/OS in a history journal. BLGUT17 assumes that the journalized data is associated with a p-word that begins with the letters ‘DAT’. If you have customized the p-word of date fields that are journalized, BLGUT17 does not convert that data. (For more information on journalizing fields, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.)

Use BLGUT17 to convert the dates in an SDDS from the old and default external date formats to the internal date format, such as:

- All dates with MM/DD/YY format and MM/DD/YYYY format to YYYY/MM/DD
- All dates with MM/DD/YY format and DD/MM/YYYY format to YYYY/MM/DD
- All dates with MM/DD/YY format and YYYY-MM-DD format to YYYY/MM/DD
- All dates with DD/MM/YY format and DD/MM/YYYY format to YYYY/MM/DD

If you do not code the LIST keyword on the PARM field, the BLGUT17 utility converts a complete single SDDS cluster.

If you code the LIST keyword on the PARM field, the BLGUT17 utility converts selected records in an SDDS cluster.

When you run BLGUT17, an error log is created to inform you of any records that could not be converted. Records are not converted if they are in a busy or checked-out status, or if, for some reason, they could not be converted (for example, a date field may not actually contain a date). A message in the error log displays the p-word and associated data that failed to be converted so that you can correct the error in your database. You can use the information supplied in the error log to run BLGUT17 again (with the LIST keyword specified) to convert records when they are no longer busy or checked out.

The BLGUT17 utility does not destroy the SDIDS component of the database. The SDIDS is still valid for the database after the SDDS is converted to the internal date format.

The BLGUT17 utility must be run for each cluster of the SDDS. For a multiple-cluster SDDS, you can start the batch jobs at the same time since they are independent of one another. Specify the SDDS cluster to be converted on the BLGSDDS DD statement in the JCL for each job.

The BLGUT17 utility does not require exclusive use of the database and can run while users are updating the database. However, to reduce the possible number of records that are busy or checked out, it is recommended that you run this utility during “off-peak” hours.

Note: If you are using the BLGUT17 utility to convert dates as part of your overall date migration activities, you should run it as the last step in those activities. Ensure that your date format parameters or options are specified on the DATEFMT and ODATEFMT keyword of the BLGPARMS macro, and that you have performed any required panel migration/field expansion first before running the BLGUT17 utility. The BLGUT17 utility is intended to convert the database at a particular point in time and is not meant to be used periodically to convert date records as they are updated or created.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Before running this utility, ensure that any partially filed records in the database have completed filing (use DBCLEANUP), and that any records that are checked out are checked back in as appropriate. Procedures on how to perform DBCLEANUP and CHECK IN are described in the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Syntax

The EXEC statement PARM field keywords to use with BLGUT17 are as follows:

```
PARM=‘SESS=aa[,NAME=n][,RECS=nn][,LIST]’
```

SESS=aa
Specifies the suffix for the session-parameters member name that BLGUT17 uses. Use 1 or 2 alphanumeric or national characters for *aa*. This session-parameters member contains the default and old external date formats. This keyword is required.

NAME=n
Specifies the name of the database, in a session-parameters member with multiple database definitions, to use. This keyword is optional, and, if omitted, defaults to 5. Valid values are 4, 5, 6, 7, 8, or 9.

RECS=nn
Specifies the number of logical records processed between each release of the SDDS cluster. The value for *nn* must be within the range 1<= nn<= 2 147 483 647.

The default is 100 logical records processed between releases.

LIST
Indicates that the BLGUT17 utility should convert the records listed in the data set specified by the BLGLIST DD statement. If LIST is not specified, the entire SDDS cluster is converted.

Input

The input to BLGUT17 is the SDDS cluster to be converted and optionally a list of records in the SDDS to convert.

Output

The output from BLGUT17 is:

- The SDDS with dates in the internal date format. Dates that previously existed in external date format are converted to the internal date format.
- Error messages in an error data set.
- A return code of 0 if the utility finishes successfully.
- A return code >0 if an error occurs.

Restrictions

- The BLGUT17 utility cannot process an SDDS with a key length of 8. The SDDS must have a key length of 7.
- The BLGUT17 utility does not process uncognized records that may exist in an SDDS. These are records that are deleted when the filing process is completed.
- The BLGUT17 utility does not process busy or checked-out records.

DD Statements

The DD statements required to run BLGUT17 are as follows:

BLGELIST
A sequential data set (RECFM=VBA , LRECL=137) containing one line of output for each logical record that is NOT converted in the SDDS that you write to a system output device or data set. This data set can be used to create the BLGLIST DD data set.

BLGLIST
An optional sequential data set (RECFM=VBA) containing one line of input for each logical record in the SDDS to be converted. Only the logical records listed in the BLGLIST DD data set are converted. The first time you run BLGUT17, you cannot specify the BLGLIST DD data set as input; you must first convert an SDDS cluster or the entire database. After the conversion, you can run BLGUT17 again and include the BLGLIST DD statement. Use the error log (BLGELIST DD) from the previous run of BLGUT17 as the input BLGLIST DD data set. The data set specified by the BLGLIST DD is the data set specified by the BLGELIST DD on a prior run of BLGUT17.

BLGSDDS
Defines the SDDS VSAM data set.

SYSPRINT
A sequential data set (RECFM=VBA) containing summary and error messages that you can write to a system output device or data set.

Scenarios Using the BLGUT17 Utility Program

The following examples illustrate how you can use the BLGUT17 utility to convert date formats in an SDDS.

The following example shows how to convert all of the dates in an SDDS to the internal date format. In this example of the first run of BLGUT17, BLGUT17 processes 5 logical records between each release of the SDDS.

```
//STEP1    EXEC PGM=BLGUT17,REGION=2048K,
//         PARM='SESS=17,RECS=5'
//STEPLIB  DD DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=A
//BLGSDDS  DD DISP=SHR,DSNAME=BLM.#01.SDDS
//BLGELIST DD DISP=(NEW,CATLG),DSNAME=ERROR01.LIST1
//            UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//            DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
```

Figure 23. Specifying a Number of Records to Process Between Releases with BLGUT17.

The following example shows, in the second run of BLGUT17, how to convert the dates in selected records in an SDDS to the internal date format. The information in the previous error log is used as input to BLGUT17.

```
//STEP1    EXEC PGM=BLGUT17,REGION=2048K,
//         PARM='SESS=17,RECS=5,LIST'
//STEPLIB  DD DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT DD SYSOUT=A
//BLGSDDS  DD DISP=SHR,DSNAME=BLM.#01.SDDS
//BLGLIST  DD DISP=SHR,DSNAME=ERROR01.LIST1
//BLGELIST DD DISP=(NEW,CATLG),DSNAME=ERROR01.LIST2,
//             UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
```

Figure 24. Converting Selected Records using BLGUT17.

27

BLGUT18—Build PIDTs and PIPTs

BLGUT18 is a utility that you can use to build program interface data tables (PIDTs) and program interface pattern tables (PIPTs) with data that it extracts from the data view records and data attribute records. The tables built by BLGUT18 are written to a partitioned data set. All of the tables (create, update, retrieve, inquiry, add record relations) associated with a particular data view record are written to a single PDS member that has the same name as the data view record. BLGUT18 uses the LLAPI and is consistent with the format used by other Tivoli Information Management for z/OS utilities.

API processing is changed to use static data view PIDTs and PIPTs. If the DATA_VIEW_NAME control PDB is specified, the API tries first to read the static data view tables from the partitioned data set. If it is unsuccessful, the data views and attributes are read from the database.

The database administrator must run the BLGUT18 batch job to create the static PIDTs and PIPTs from the data view and data attribute records. The BLGUT18 batch job must be run whenever the data view and data attribute records are changed or new views and attributes are created. Static PIDTs and PIPTs are provided for all of the data view and data attribute records shipped with the base product. These tables are shipped in data set SBLMFMT.

BLGUT18 runs as a batch job on MVS. BLGUT18J, described in "Sample JCL" on page 148, is shipped by Tivoli Information Management for z/OS as a sample.

Syntax

This is the command syntax for BLGUT18:

```
PARM='SESS=aa,CLASS=class,APPLID=applid(,NAME=n)'
```

where

SESS=aa
Specifies the suffix for the session parameters member name that BLGUT18 uses with the LLAPI. You must specify a value for aa using either 1 or 2 alphanumeric or national characters.

CLASS=class
Specifies the privilege class that BLGUT18 uses with the LLAPI.

APPLID=applid
Specifies the name by which Tivoli Information Management for z/OS recognizes the BLGUT18 application. This value must be an eligible user of the privilege class specified on the CLASS= parameter.

NAME=n
Identifies the database (in a session parameter member containing multiple

database definitions) that contains the data model records. Valid values are 4, 5, 6, 7, 8, and 9. This keyword is optional; if it is omitted, NAME=5 is the default.

Input Requirements

The input for BLGUT18, the Data Model Table Build utility, is:

Data view records A SYSIN data set that contains the names of the data view records to process.

Output Requirements

The outputs for BLGUT18 are:

PDS member or members Contains table objects (PIDTs and PIPTs) built by the Data Model Table Build Utility.

Message and report output Contains the input control statements. This output is followed by either informational messages, indicating successful table creation, or error messages.

A return code A value of 0 indicates that BLGUT18 completed successfully. Any other value indicates that BLGUT18 did not finish successfully.

Control

Start BLGUT18 as a batch utility. The system administrator or table developer must determine when and against which data sets to run this utility. If changes are made to data views or to data attribute records, BLGUT18 should be run. These are the DD statements needed to run BLGUT18:

Table 4. BLGUT18 DD Statements

DD Statement	Description
SYSIN	Input control statements (LRECL=80,RECFM=RB)
BLGRFT	The output partitioned table data set
SYSPRINT	The sequential message data set
APIPRINT	The LLAPI sequential message data set

Sample JCL

BLGUT18J is shipped in the BLM.SBLMSAMP data set. You can use the following sample jobstream to run BLGUT18 to build the PIDT and PIPT tables from data model records to run with the application program interface. If you use this sample, you must update the PARM field and change the data set names for DD names STEPLIB, BLGRFT, SYSPRINT, and APIPRINT to correspond to the data set names that you use.

```
//STEP1    EXEC PGM=BLGUT18,REGION=2048K,
//   PARM='SESS=00,CLASS=MASTER,APPLID=applid,NAME=5'
//STEPLIB  DD  DISP=SHR,DSN=BLM.SBLMMOD1
//BLGRFT   DD  DISP=SHR,DSN=BLM.TABLES.PDS
//SYSPRINT DD  SYSOUT=*
//APIPRINT DD  SYSOUT=*
//SYSIN    DD  *
```

```
|                  BLMCALL
|                  BLMPROB
|                  /*
|                  //CHECKCC EXEC PGM=IEFBR14
|                  //
```

28

BLGUT20—Analyze the SDDS

You can use BLGUT20 to read an SDDS and generate statistics about its contents. This information can help you analyze the SDDS in a sysplex or non-sysplex environment. However, in a non-sysplex environment, the information generated can also help you tune the VSAM definition for the SDDS for optimum performance. BLGUT20 provides a line of data for each Tivoli Information Management for z/OS record including the record number identifier and summary information. A logical Tivoli Information Management for z/OS record might be one or more VSAM physical records. Refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* for information on determining the proper size for the SDIDS and SDDS.

To provide meaningful results, run BLGUT20 only after creating a representative number of records for all record types. Use the summary information to estimate the best control interval (CI) size, average record length, and maximum record length for the data set. The primary goal is to minimize the number of SDDS records that do not fit into a single CI without having a larger CI size than needed.

If you change any VSAM definitions in a non-sysplex environment for your SDDS, you can create a new SDDS cluster with different CI sizes or record lengths using the IDCAMS DEFINE command. Use the BLGUT7 utility to convert the data from the old SDDS to the new SDDS cluster if you are changing the key length or creating a new SDDS cluster with a smaller MAXLRECL (RECORDSIZE) or CI size. Otherwise, you can use IDCAMS REPRO to copy data from the old SDDS to the new SDDS cluster.

You can run BLGUT20 while the SDDS data set is in use. It sequentially processes the data set and calculates results. While running, BLGUT20 competes with other Tivoli Information Management for z/OS users for access to the SDDS.

In a non-sysplex environment BLGUT20 periodically releases the data set so users of Tivoli Information Management for z/OS can have write access to it. Then BLGUT20 continues processing from the point of interruption. You can change the number of logical records BLGUT20 processes between each release of the data set with a parameter in your JCL (see the RECS parameter on page 152). When you decrease the number of records processed between each release of the data set, the performance impact on Tivoli Information Management for z/OS users decreases, but the utility runs slower. When you increase the number, the performance impact on Tivoli Information Management for z/OS users increases, but the utility runs faster. When you do not specify the number of logical records, the default is 100 records.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Syntax

The EXEC statement PARM field keywords to use with BLGUT20 are as follows:

```
PARM='[RECS=nn]'
```

RECS=nn
Specifies the number of logical records processed between each release of the SDDS cluster. *nn* must be within the range 1≤nn≤2 147 483 647. The RECS keyword is ignored if sysplex mode is enabled.

When you do not want to supply this option, do not code the PARM field on the EXEC statement. The default is 100 logical records processed between releases.

Input

The input for BLGUT20 is one SDDS cluster. The SDDS is a key-sequenced VSAM data set. If you are using a 5-SDDS cluster database you must run BLGUT20 five times to see the statistics from all five data sets, once for each cluster, to obtain all the statistics. Usually all five would yield similar results.

Output

The output from BLGUT20 is:

- A sequential data set containing a line of data for each VSAM logical record in the SDDS. Each line of data contains the following information:
  - Columns 1 to 8: record length in bytes
  - Columns 11 to 23: X followed by the hexadecimal key of the logical record enclosed in single quotes
  - Columns 25 to 32: record number identifier (RNID)
  - Columns 34 to 41: number of physical records in the logical record
- Summary and error messages in a message data set. The summary information includes:
  - Name of the data set analyzed
  - Control interval size of the data component, key length, and maximum record size of the data set
  - Number of physical and logical records processed
  - Largest and smallest logical record sizes
  - Total size of the SDDS cluster
  - Average logical record size
  - A summary of CI sizes required by logical records in the data set
- A return code of 0 if the utility finishes successfully.
- A return code >0 if an error occurs.

Restrictions

Because you can update the SDDS data set when BLGUT20 releases the SDDS, the calculated results do not include any changes made to the part of the data set processed by the utility before the interruption. Therefore, when the utility completes processing, the results may not exactly reflect the current contents of the SDDS.

BLGUT20 does not process uncognized records that may exist in an SDDS. These are records that are removed when the filing process is completed.

DD Statements

The DD statements required to run BLGUT20 are as follows:

ANALYZE
Sequential data set that contains one line of output per VSAM logical record in the SDDS. DCB defaults are: LRECL=42,RECFM=FB,BLKSIZE=8400. If you omit this DD statement, only summary and error messages are generated. Also, the utility will complete with return code 4 and message BLG21287W.

BLGSDDS
The SDDS VSAM data set.

SYSPRINT
Sequential data set (RECFM=VBA) containing summary and error messages that you write to a system output device or data set.

Scenarios Using the BLGUT20 Utility Program

The following example illustrates the use of BLGUT20 to get statistical data about the contents of an SDDS. In this example, BLGUT20 processes 50 logical records between each release of the SDDS.

```
//SDDSR      JOB
//STEP1      EXEC PGM=BLGUT20,PARM='RECS=50',REGION=2048K
//STEPLIB    DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//BLGSDDS    DD  DISP=SHR,DSNAME=BLM.SDDS
//ANALYZE    DD  DISP=SHR,DSNAME=BLM.SDDS.ANALYZE
//SYSPRINT   DD  SYSOUT=A
//
```

Summary information messages are illustrated in the following example.

```
SDDS DATA SET ANALYZED:  'KNUDSEN.SUSAN.SDDS'     TIME: 14:10:19 DATE: 11/13/00
CI SIZE: 8192     KEY LENGTH: 7     MAXIMUM RECORD LENGTH: 8185
TOTAL DATA SET SIZE: 178760
    TOTAL SDDS STRUCTURED DATA SIZE: 143836
    TOTAL SDDS STRUCTURED TEXT SIZE: 3810
    TOTAL SDDS STRUCTURED HISTORY SIZE: 31114
LOGICAL RECORDS PROCESSED: 38
PHYSICAL RECORDS PROCESSED: 38
MINIMUM SDDS LOGICAL RECORD SIZE: 3437
AVERAGE SDDS LOGICAL RECORD SIZE: 4704
MAXIMUM SDDS LOGICAL RECORD SIZE: 6572
SDDS RECORDS REQUIRING A  512 CI SIZE: 0
SDDS RECORDS REQUIRING A 1024 CI SIZE: 0
SDDS RECORDS REQUIRING A 2048 CI SIZE: 0
SDDS RECORDS REQUIRING A 3072 CI SIZE: 0
SDDS RECORDS REQUIRING A 4096 CI SIZE: 19
SDDS RECORDS REQUIRING A 6144 CI SIZE: 17
SDDS RECORDS REQUIRING A 8192 CI SIZE: 2
SDDS RECORDS REQUIRING A GREATER THAN 8192 CI SIZE: 0
SDDS RECORDS REQUIRING MULTIPLE CONTROL INTERVALS: 0
SDDS ANALYZE UTILITY COMPLETE
```

Figure 25. Example Summary Message Display

The following example partially illustrates the output provided in the analysis data set.

```
DATE: 11/13/00   TIME: 14:10:19
RECORD     KEY (1:5)      RECORD  PHYSICAL
LENGTH      SEQ#          ID      RECORDS
--------   ------------- -------- --------
00006831   X'0000003733' 00000001 00000001
              .
              .
              .
  (one line of output for each logical record in the data set)
```

Figure 26. Example of Part of the Analysis Data Set

29

BLGUT21—Analyze the SDIDS

You can use BLGUT21 to generate statistics about the contents of an SDIDS. This information assists in estimating the VSAM attributes of this data set, so you can define your SDIDS for optimum performance. BLGUT21 provides a line of data for each VSAM record in the SDIDS plus summary information.

To provide meaningful results, run BLGUT21 only after creating a representative number of records for all record types. Use the summary information to estimate the best CI size, average record length, and maximum record length for the data set. For more information refer to the *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*. If you decide to change any of these characteristics, you can create a new SDIDS data set with different CI size or record lengths using the IDCAMS DEFINE command. Use the BLGUT1 utility to rebuild the SDIDS from an existing SDDS, or the BLGUT1M utility to copy or migrate SDIDS data if you do not want to rebuild the entire SDIDS.

In a non-sysplex environment, you can use BLGUT21 while the SDIDS data set is in use, but response times will be increased because of competition for the SDIDS. It sequentially processes the data set and calculates results. Periodically, BLGUT21 releases the data set so users of Tivoli Information Management for z/OS can access it. Then BLGUT21 continues processing from the point of interruption. You can change the number of records that BLGUT21 processes between each release of the data set (see the RECS parameter on page 156). If you decrease the number of records processed between each release of the data set, the performance impact on Tivoli Information Management for z/OS users decreases, but the utility runs slower. If you increase the number, the performance impact on Tivoli Information Management for z/OS users increases, but the utility runs faster. When you do not specify the number of records, the default is 100 records.

BLGUT21 includes the bit count in its statistical output for each SDIDS record. This count is not required to analyze the performance of the SDIDS. Therefore, you can run the utility without generating this count. When you choose option NOCNT, the utility runs faster and requires a smaller amount of storage. When you do not choose this option, the bit count is generated.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Syntax

The EXEC statement PARM field keywords to use with BLGUT21 are as follows:

```
PARM=‘[RECS=nn][,NOCNT][,FROMKEY=xx]
    [{,TOKEY=yy|,COUNT=nnnnn}][,OTRG=c]’
```

RECS=nn

Specifies the number of records processed between each release of the SDIDS data set. `nn` must be within the following range: 1≤nn≤2 147 483 647. If you omit this keyword, 100 records are processed between each release of the SDIDS. The RECS keyword is ignored if sysplex mode is enabled.

NOCNT

Indicates that you do not want to generate the bit count. Omit this keyword to generate the bit count.

FROMKEY=xx

Specifies a character string consisting of 1 to 32 characters that identify the key of the first record to process. The value is treated as an approximate key value. The FROMKEY keyword is optional and defaults to the key of the first SDIDS record.

Some SDIDS records begin with a watermark character (a hexadecimal value) that cannot be entered using a keyboard. In these cases, you must use the ISPF line command HEX ON or HEX DATA to enter the hexadecimal value for the character.

TOKEY=yy

Specifies a character string consisting of 1 to 32 characters that identify the key of the last record to process. The value is treated as an approximate key value. The TOKEY keyword is optional and defaults to the key of the last SDIDS record. The value you specify must be greater than or equal to the value of FROMKEY.

Some SDIDS records begin with a watermark character (a hexadecimal value) that cannot be entered using a keyboard. In these cases, you must use the ISPF line command HEX ON or HEX DATA to enter the hexadecimal value for the character.

COUNT=nnnnn

Specifies the number of records to process. The COUNT keyword is optional and defaults to the number of records in the SDIDS. The COUNT and TOKEY keywords are mutually exclusive.

OTRG=c

Identifies the trigger character of the SDIDS to be analyzed. When this keyword is specified, all of the clusters associated with the SDIDS cluster named by the BLGSDIDS DD statement are analyzed. If you omit this keyword, only the BLGSDIDS cluster (the primary cluster) is analyzed. This keyword is optional.

Examples showing how to specify both the FROMKEY and TOKEY keywords follow:

```
PARM='FROMKEY='from-key-value',TOKEY='to-key-value'
PARM='FROMKEY='']MASTER_BIT_LIST'',TOKEY='']MASTER_BIT_LISU'''
PARM='FROMKEY=RNID/,TOKEY=RNOR/'
```

In these examples, the watermark character] is entered as X'BA'.

Examples showing how to specify the FROMKEY keyword and COUNT keyword follow:

```
PARM='FROMKEY='from-key-value',COUNT=5'
PARM='FROMKEY='']MASTER_BIT_LIST'',COUNT=1'
PARM='FROMKEY=RNID/,COUNT=200'
```

When you want to specify only one of the above optional keywords, code it in the PARM field on the EXEC statement. For example, when you want to process 150 records between each release of the SDIDS and you also want to generate the bit count, code the RECS keyword and omit the NOCNT keyword, as follows:

```
PARM='RECS=150'
```

When you do not want to change the processing interval and you do not want to generate the bit count, code the NOCNT parameter and omit the RECS parameter, as follows:

```
PARM='NOCNT'
```

When you do not want to change any option or specify a trigger character, do not code the PARM field on the EXEC statement.

Input

The input for BLGUT21 is the SDIDS data set. The SDIDS is a key-sequenced VSAM data set.

Output

The output from BLGUT21 is an output file as follows:

- A sequential data set containing a line of data for each record in the SDIDS. Each line of data contains information as shown in the following example:

```
DATE: 06/05/2000   TIME: 14:27:20
 RECORD        SDIDS                      SPAN/ R   SDDS     TOTAL  UPDTE
-LENGTH- -------KEY---------------------- -SEG- T -RECORDS ---BITS- COUNT

llllllll kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk sssss t rrrrrrrr bbbbbbbb uuuuu
```

llllllll Specifies the length of a whole SDIDS record in bytes.

kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk
Specifies a cognized word.

sssss Segment indicator:
’ ’ - The record is not segmented.
sssss - Specifies the segment index value of a segmented record.

t Identifies the type of data contained in the record:
B - The record contains an uncompressed bit string.
C - The record contains a compressed bit string.
S - The record contains a 4-byte binary number (such as a last entry number).

rrrrrrrr
Specifies the number of one bits (SDDS record references) in a bit string record.

bbbbbbbb
Specifies the total number of bits in a bit string record.

uuuuu
Indicates the approximate number of times a bit string record has been updated (1 to 250; starts over with 1 after 250).

- Summary and error messages in a message data set. The summary information includes:
  - Name of the SDIDS analyzed
  - Control interval size of data component
  - Key length
  - Maximum physical record size of the data set
  - REUSE – If specified, VSAM sequence number reuse indicator
  - UNIQUE – If specified, unique record number ID indicator
  - Current number of database records (generated only when none of the FROMKEY, TOKEY, or COUNT keywords are specified)
  - Maximum number of database records this SDIDS can support
  - Total size of the SDIDS data set
  - Number of records processed
  - Smallest logical record size
  - Average record size
  - Cluster identification and key range for each cluster
  - Largest logical record size
  - Maximum segment index value
- A return code of `0` if the utility finishes successfully.
- A return code >0 if an error occurs.

Restrictions

- Because you can update the SDIDS data set when BLGUT21 releases the SDIDS, the calculated results do not include any changes made to the part of the data set processed by the utility before the interruption. Therefore, when the utility completes processing, the results may not exactly reflect the current contents of the SDIDS.
- You can use the BLGUT21 utility to analyze a TME 10 Information/Management Version 1.1 SDIDS if desired, but you cannot use it to analyze an SDIDS from a release prior to Version 1.1. In this case, use BLGUT1M to migrate the SDIDS to the Version 7.1 level and then run BLGUT21 to analyze it.

DD Statements

The DD statements required to run BLGUT21 are as follows:

ANALYZE
Sequential data set that contains one line of output per index record in the SDIDS. DCB defaults are: LRECL=80,RECFM=FB. No value is specified for BLKSIZE, to allow the operating system to derive the optimum block size value based upon the track capacity of the DASD device to contain the analyze output. If you omit this DD statement, only summary and error messages are generated. Also, the utility will complete with return code 4 and message BLG21287W.

BLGSDIDS
The SDIDS VSAM data set.

SYSPRINT
Sequential data set (RECFM=VBA) containing summary and error messages that you can write to a system output device or data set.

Scenario Using the BLGUT21 Utility Program

The following example illustrates how you can run BLGUT21 to obtain statistical data about the contents of an SDIDS. In this example, the trigger character parameter is specified to analyze all clusters of a multiple-cluster SDIDS.

```
//SDIDSR     JOB
//STEP1      EXEC PGM=BLGUT21,PARM='OTRG=#',REGION=2048K
//STEPLIB    DD  DISP=SHR,DSN=BLM.SBLMMOD1
//BLGSDIDS   DD  DISP=SHR,DSN=BLM.PROD#01.SDIDS
//ANALYZE    DD  DISP=SHR,DSN=BLM.SDIDS.ANALYZE
//SYSPRINT   DD  SYSOUT=A
//
```

The following example shows summary information messages generated by BLGUT21. The example shows use of an 18-byte SDIDS key and four clusters.

```
SDIDS DATA SET ANALYZED:  'BLM.PROD#01.SDIDS'     TIME: 10:30 DATE: 03/28/2000
OTRG=#
CI SIZE: 2048     KEY LENGTH: 18     MAXIMUM RECORD LENGTH: 2041   REUSE   UNIQUE

SDIDS CLUSTER 1 OF 4 WITH KEY RANGE: <000000000000> - <BA 99>

SDIDS RECORDS PROCESSED:  27
TOTAL DATA SET SIZE: 2357
MINIMUM SDIDS RECORD SIZE: 38
AVERAGE SDIDS RECORD SIZE: 87
MAXIMUM SDIDS RECORD SIZE: 227
MAXIMUM SEGMENT INDEX VALUE: 1

SDIDS CLUSTER 2 OF 4 WITH KEY RANGE: <BA 99> - <A>
SDIDS RECORDS PROCESSED: 363
TOTAL DATA SET SIZE: 22390
MINIMUM SDIDS RECORD SIZE: 38
AVERAGE SDIDS RECORD SIZE: 62
MAXIMUM SDIDS RECORD SIZE: 230
MAXIMUM SEGMENT INDEX VALUE: 1

SDIDS CLUSTER 3 OF 4 WITH KEY RANGE: <A> - <J>
SDIDS RECORDS PROCESSED: 1129
TOTAL DATA SET SIZE: 54328
MINIMUM SDIDS RECORD SIZE: 35
AVERAGE SDIDS RECORD SIZE: 48
MAXIMUM SDIDS RECORD SIZE: 221
MAXIMUM SEGMENT INDEX VALUE: 1

SDIDS CLUSTER 4 OF 4 WITH KEY RANGE: <J> - <FFFFFFFFFFFF>
SDIDS RECORDS PROCESSED: 3918
TOTAL DATA SET SIZE: 172249
MINIMUM SDIDS RECORD SIZE: 35
AVERAGE SDIDS RECORD SIZE: 44
MAXIMUM SDIDS RECORD SIZE: 221
MAXIMUM SEGMENT INDEX VALUE: 1

------------------------------------------------------

CURRENT NUMBER OF DATABASE RECORDS: 1475

SDIDS RECORDS PROCESSED: 5437
TOTAL DATA SET SIZE: 251324
MINIMUM SDIDS RECORD SIZE: 35
AVERAGE SDIDS RECORD SIZE: 46
MAXIMUM SDIDS RECORD SIZE: 230
MAXIMUM SEGMENT INDEX VALUE: 1

SDIDS ANALYZE UTILITY COMPLETE
```

Figure 27. Summary Message Display

Figure 28 on page 161 partially illustrates the output provided in the analysis data set, using an 18-byte SDIDS key.

```
 DATE: 07/30/2000   TIME: 10:47:10
 RECORD        SDIDS                      SPAN/ R   SDDS      TOTAL UPDTE
-LENGTH- -------KEY---------------------- -SEG- T -RECORDS ---BITS- COUNT

00000028 #                                      C 00000001 00000208 00001
00000053 -//S/MSI                               B 00000030 00000227 00001
00000045 -//S/TSI                               C 00000197 00000225 00001
00000028 -LASTENTRYNUMBER                       S 00000089          00000
00000029 -MASTER_BIT_LIST                       C 00000227 00000227 00001
00000031 ADDED                                  C 00000006 00000204 00001
00000053 AUTH/YES                               B 00000016 00000226 00001
00000028 BASIC                                  C 00000001 00000207 00001
00000028 BLG6STAT                               C 00000001 00000224 00001
00000028 BUILDING                               C 00000001 00000205 00001
00000028 CHANGE                                 C 00000001 00000177 00001
00000028 CLAE/NEWPART                           C 00000001 00000192 00001
00000028 COPIED                                 C 00000001 00000038 00001
00000028 COPY                                   B 00000002 00000030 00001
00000040 CREATE                                 C 00000009 00000204 00001
00000031 DADD/YES                               C 00000006 00000203 00001
00000029 DARC/BLG#CRTD                          C 00000002 00000204 00001
00000028 DATA                                   C 00000001 00000201 00001
00000028 DATD/00/04/23                          C 00000001 00000180 00001
00000026 DATE/00/02/09                          B 00000001 00000009 00001
00000028 GROS/TA2                               C 00000001 00000195 00001
00000028 GROS/TEST                              C 00000001 00000180 00001
00000028 LOCC/BLG062                            C 00000001 00000180 00001
00000029 MASTER                                 B 00000003 00000035 00001
00000028 NEWPART                                C 00000001 00000192 00001
00000028 NUMX/Z00001                            C 00000001 00000180 00001
00000028 PERC/BENNETT                           C 00000001 00000177 00001
00000029 PERS/SMITH                             C 00000003 00000222 00001
00000028 PTID/ABC                               C 00000001 00000178 00001
00000028 PWIN/0C20                              C 00000001 00000204 00001
00000028 RNID/ABCD                              C 00000001 00000209 00001
00000046 STAC/OPEN                              C 00000007 00000208 00001
00000028 TYPE/TEST                              C 00000001 00000180 00001
00000028 000                                    B 00000003 00000032 00001
00000038 001                                    C 00000004 00000205 00001
              .
              .
  (one line of output for each record in the SDIDS)
```

Figure 28. Example of Output Provided in Analysis Data Set

30

BLGUT22—Analyze a Panel Data Set

You can use the BLGUT22 utility to display statistics about the contents of a panel data set. This information assists you in determining the VSAM parameters of this data set so you can define it for optimum performance. BLGUT22 produces one line of output for each panel plus summary information.

To obtain meaningful results, run BLGUT22 only after loading the panel data set with the BLGUT6 utility. You can use the summary information to estimate the best control interval size, average record length, and maximum record length for the data set. The primary goal is to choose a CI size that permits most panels to fit in a single CI and still have the smallest CI size permitting reasonable growth in panel size. You can use BLGUT6 to reload the new panel data set.

If you make changes in your definition for the panel data set, then you can define a new panel data set and use BLGUT6 to copy the panels into it.

You can run BLGUT22 while the panel data set is in use. BLGUT22 sequentially processes the data set and calculates results. While it is running, the utility competes with other Tivoli Information Management for z/OS users for access to the SDDS.

In a non-sysplex environment BLGUT22 releases the data set periodically so active PMF users can have write access to it. BLGUT22 continues processing from the point of interruption. You can change the number of panels BLGUT22 processes between each release of the data set (see RECS parameter on page 164). When you decrease the number of panels processed between each release of the data set, the performance impact on active PMF users decreases, but the utility runs slower. When you increase the number, the performance impact on active PMF users increases, but the utility runs faster. When you do not specify the number of panels, the default is 100 records.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Syntax

The EXEC statement PARM field keywords to use with BLGUT22 are as follows:

```
PARM=‘[RECS=nn]’
```

RECS=nn
Specifies the number of records processed between each release of the panel data set. The nn must be within the range 1≤nn≤2 147 483 647. If you omit the keyword, 100 records are processed between each release of the panel data set. The RECS keyword is ignored if sysplex mode is enabled.

Input

The input to BLGUT22 is a panel data set.

Output

The output from BLGUT22 is:

- A sequential data set containing a line of output for each panel in the data set. Each line of output contains the following information:
  - Columns 1–8: Panel name
  - Columns 13–20: Panel length in bytes
  - Columns 25–32: Number of physical records in the panel

  A panel can consist of multiple physical VSAM records.
- Summary and error messages in an output data set. The summary information includes:
  - Name of the panel data set analyzed
  - CI size of data component
  - Key length
  - Maximum physical record size of the data set
  - Number of physical records and panels processed
  - Total size of the panel data set
  - Minimum, maximum, and average panel size
  - A summary of CI sizes required by panels in the data set
  - Number of panels requiring multiple control intervals
- A return code of 0 if the utility finishes successfully.
- A return code >0 if an error occurs.

Restrictions

Because an active PMF user can update the panel data set when BLGUT22 releases it, the calculated results do not include changes made to the part of the data set processed by the utility before the interruption. Therefore, when the utility completes processing, the results may not exactly reflect the current contents of the panel data set.

DD Statements

The DD statements required to run BLGUT22 are as follows:

ANALYZE
Sequential data set that contains one line of output for each panel in the data set. DCB defaults are: LRECL=32,RECFM=FB,BLKSIZE=11200. If you omit this DD statement, only summary and error messages are generated. Also, the utility will complete with return code 4 and message BLG21287W.

BLGPANS
The VSAM panel data set.

SYSPRINT
Sequential data set containing summary and error messages that you can write to a system output device or data set.

Scenarios Using the BLGUT22 Utility Program

The following example illustrates how to run BLGUT22 to get statistical data about the contents of a panel data set. In this example, BLGUT22 processes 500 panels between each release of the panel data set.

```
//PANELS     JOB
//STEP1      EXEC PGM=BLGUT22,PARM='RECS=500',REGION=4096K
//STEPLIB    DD  DISP=SHR,DSN=BLM.SBLMMOD1
//BLGPANS    DD  DISP=SHR,DSN=BLM.IBMPNLS
//ANALYZE    DD  DISP=SHR,DSN=BLM.PANEL.ANALYZE
//SYSPRINT   DD  SYSOUT=A
//
```

The following example illustrates the summary information messages generated by BLGUT22.

```
1PANEL DATA SET ANALYZED: 'BLM.PANELS' TIME: 09:49:04 DATE: 12/12/00
CI SIZE: 2048     KEY LENGTH: 10            MAXIMUM RECORD LENGTH: 2038

 TOTAL DATA SET SIZE: 3177631
 LOGICAL RECORDS PROCESSED: 3730
 PHYSICAL RECORDS PROCESSED: 3919

 MINIMUM PANEL SIZE: 174
 AVERAGE PANEL SIZE: 852
 MAXIMUM PANEL SIZE: 8570

 PANELS REQUIRING A  512 CI SIZE: 1190
 PANELS REQUIRING A 1024 CI SIZE: 1611
 PANELS REQUIRING A 2048 CI SIZE: 772
 PANELS REQUIRING A 3072 CI SIZE: 107
 PANELS REQUIRING A 4096 CI SIZE: 27
 PANELS REQUIRING A 6144 CI SIZE: 16
 PANELS REQUIRING A 8192 CI SIZE: 5
 PANELS REQUIRING A GREATER THAN 8192 CI SIZE: 2
 PANELS REQUIRING MULTIPLE CONTROL INTERVALS: 157

 PANEL ANALYZE UTILITY COMPLETE
```

Figure 29. Summary Message Display

The following example partially illustrates the output provided in the analysis data set.

```
DATE: 08/17/00   TIME: 09:49:04
PANEL      PANEL         VSAM
 KEY        SIZE        RECORDS
--------   --------     --------
BLGAPI00   00003733     00000001
BLGAPI02   00002935     00000001
BLGAPI05   00003558     00000001
BLGAPI09   00003485     00000001
BLGAPI10   00001339     00000001
BLGESC01   00004540     00000002
BLGESC02   00007575     00000002
BLGESC03   00004387     00000002
BLGESC04   00001868     00000001
BLGESC05   00001780     00000001
BLGESC06   00004296     00000002
BLGESC07   00001654     00000001
BLGLABND   00000541     00000001
BLGLABNU   00000503     00000001
BLGLABN1   00000503     00000001
              .
              .
              .
  (one line of output for each panel in the data set)
```

31

BLGUT23 Utilities Overview

In Tivoli Information Management for z/OS, there are various ways you can back up data in the SDDS. For example:

- You can use the BLGUT3 utility to restore the database from the log data set created using the BLGUT4 utility and pruned with the BLGUT4LP utility. This method involves using IDCAMS REPRO to create a backup of the SDDS and SDIDS. This method is commonly used and has been available for a number of years. The advantage of this method is that it is simple. The disadvantage is that the database (SDDS and SDIDS) is unavailable to users while the IDCAMS REPRO copy is being created. See “BLGUT4—Offload the Recovery Log Data Set” on page 73 for more information on this method of backing up data.
- You can use the BLGUT23B utility in a simplified fashion to create a master backup data set each time a backup is desired. You can combine the master backup data set with an SDLDS large enough to hold all your changes between runs of BLGUT23B to form a complete backup copy of your database. If it is necessary to restore the database, you can use the BLGUT23R utility after the SDLDS is offloaded with BLGUT4. “Scenario 1” on page 168 describes this approach.
- Or, you can use the complete series of BLGUT23 utilities to create and merge data into a master backup data set. You can run BLGUT23B once, and then keep the master backup data set current through use of the BLGUT23P and BLGUT23U utilities after the SDLDS is offloaded periodically through the BLGUT4 utility. This approach is described in “Scenario 2” on page 169.

With either BLGUT23 method, your database is available to users 24 hours a day, 7 days a week. For more information on the advantages and disadvantages of the various methods of database backup that are available, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Note: BLGUT23B and BLGUT23R can be used only with key 7 format SDDS clusters. If you currently use key 8 format SDDS clusters and would like to use BLGUT23B and BLGUT23R, use the BLGUT7 utility to convert from a key 8 to a key 7 SDDS format.

Attention: If you have implemented the Automatic Log Save Facility or the DB2 Extract Facility into your data recovery process, do not run the BLGUT4 utility until you fully understand how it may affect your data. Doing so can result in lost data. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about the Automatic Log Save, DB2 Extract facilities, and the various backup methods available in Tivoli Information Management for z/OS.

The BLGUT23 series of utilities consists of the following programs which are depicted in Figure 30:

- BLGUT23B — *Backs up* the SDDS records on a single cluster to a sequential data set.
- BLGUT23P — Accepts offloaded log data from the BLGUT4 utility and *prunes* and sorts the data. This pruned log data set is used as input for additional BLGUT23P runs until it is used to update the master backup data (with either the BLGUT23U or BLGUT23R utility).
- BLGUT23U — *Updates* the master backup data produced by BLGUT23B with the pruned log data created by BLGUT23P.
- BLGUT23R — *Restores* the SDDS from data output from either the BLGUT23U or BLGUT23B utility.

Once the BLGUT23R utility is run to restore the SDDS, you can, if the SDIDS no longer exists, run the BLGUT1 utility to rebuild the SDIDS.

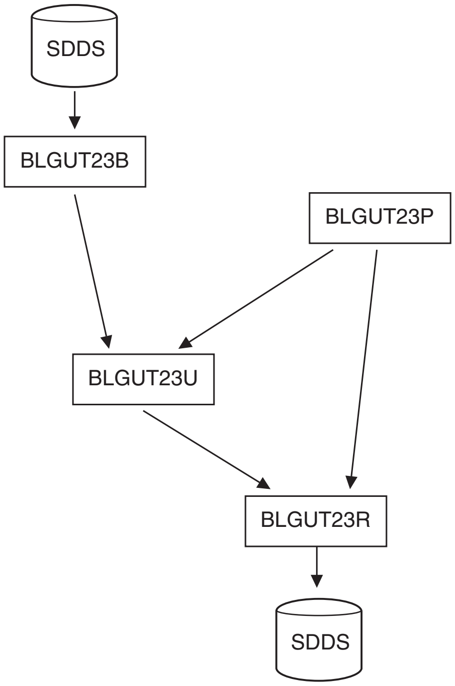


Figure 30. Overview of the BLGUT23 Series of Utilities

Scenario 1

The following scenario depicts how you can run the BLGUT23 series of utilities in a simplified fashion to back up data in the SDDS.

1. When you want to run a database backup, run the BLGUT4 utility to offload the current log data set (SDLDS).

   Note: This offloaded data set can be deleted after the BLGUT23B utility completes in step 2 since the master backup data set has all of these updates.

2. Next, run the BLGUT23B utility for each cluster of the SDDS. For a multiple-cluster SDDS, the batch jobs can be started at the same time, since they are independent of one another. Specify the SDDS cluster to be backed up on the `BLGSDDS DD` statement in the

JCL for each job. BLGUT23B reads each record in order by SDDS position number and writes it to a sequential data set in the same format used by the BLGUT4 utility.

3. After all of the BLGUT23B jobs are complete, discard any previous master backup data set and the log created in step 1 on page 168. Use the DFSORT utility to merge the multiple SDDS backup data sets into a single, master backup data set. Several runs may be required to merge all of the data output by BLGUT23B. Keep this merged master backup for use by BLGUT23R if it becomes necessary to restore your database. This master backup data set can be on tape if the database is large.
4. If the database becomes unusable or must be restored, run IDCAMS to define new SDDS clusters.
5. You will have SDLDS data which has not been offloaded or pruned. Run the BLGUT4 and BLGUT23P utilities.

   Note: The BLGUT23U utility is not needed.
6. Use the BLGUT23R utility to load the newly created (empty) SDDS clusters. The output from the BLGUT23P job and output from step 3 are input to the BLGUT23R utility. The BLGUT23R utility reads the latest master backup tape, updates the records with any log data, and then loads the data onto the new SDDS.

   If some SDDS clusters are bad and others are good, you can use BLGUT23R to restore just the bad clusters. If no SDDS clusters are bad, but you want to resequence your database, you must load all clusters and run the BLGUT1 utility to create a new SDIDS.
7. If the old SDIDS is not usable, run the BLGUT1 utility to rebuild the SDIDS after the SDDS is loaded. Upon completion, the database is ready to be used.

Scenario 2

The following scenario depicts how you can run the BLGUT23 series of utilities to back up, prune, update, and restore your database.

1. When you want to run a database backup, run the BLGUT4 utility to offload the current log data set (SDLDS). This optional step results in a smaller log data set for the prune later in Step 5 on page 170.

   Note: This offloaded data set can be deleted after the BLGUT23B utility completes in step 2 since the master backup data set has all of these updates.
2. Next, run the BLGUT23B utility for each cluster of the SDDS. For a multiple-cluster SDDS, the batch jobs can be started at the same time, since they are independent of one another. Specify the SDDS cluster to be backed up on the `BLGSDDS DD` statement in the JCL for each job. BLGUT23B reads each record in order by SDDS position number and writes it to a sequential data set in the same format used by the BLGUT4 utility.
3. After all of the BLGUT23B jobs are complete, use the DFSORT utility to merge the multiple SDDS backup data sets into a single, master backup data set. Several runs may be required to merge all of the data output by BLGUT23B. Keep the merged master backup for later use. This data set can be on tape if the database is large.
4. As updates are made to the database, a copy of each changed record is written to the SDLDS. Periodically, unload this data set by running BLGUT4. Keep multiple offloaded logs in chronological order.

5. After one or more runs of BLGUT4, use the BLGUT23P utility to prune the extra records (only the last update or purge record need be kept). The output data from one run of BLGUT23P is input into the next run of BLGUT23P to maintain a complete database change log.

   Note: Do not mix the output files from BLGUT23P with the output files of the BLGUT4LP utility. The pruning rules and sort order are different.

6. After some time, as the pruned log data set grows, you should update the master backup tape with these changes. Do this by running the BLGUT23U utility. The output of this utility is a new master backup tape with all the updated records from the log. If a record was purged from the database, it will also be removed from the master backup. After you run the BLGUT23U utility, you can discard the pruned log data used as input since it is no longer needed.

7. If the database becomes unusable or must be restored, run IDCAMS to define new SDDS clusters, and use the BLGUT23R utility to load the newly created (empty) SDDS clusters.

   If some SDDS clusters are bad and others are good, you can use BLGUT23R to restore just the bad clusters. If no SDDS clusters are bad, but you want to resequence your database, you must load all clusters and run the BLGUT1 utility to create a new SDIDS.

   If you have SDLDS data which has not been offloaded or pruned, be sure to run the BLGUT4 and BLGUT23P utilities first. The BLGUT23U utility is not needed since the sorted and pruned log data can be input directly into the BLGUT23R utility, with the latest master backup tape. The BLGUT23R utility reads the latest master backup tape, updates the records with any log data, and then loads the data onto the new SDDS.

8. If the old SDIDS is not usable, run the BLGUT1 utility to rebuild the SDIDS after the SDDS is loaded. Upon completion, the database is ready to be used.

For a detailed illustration of how data flows between the BLGUT23 series of utilities, refer to the following diagram.

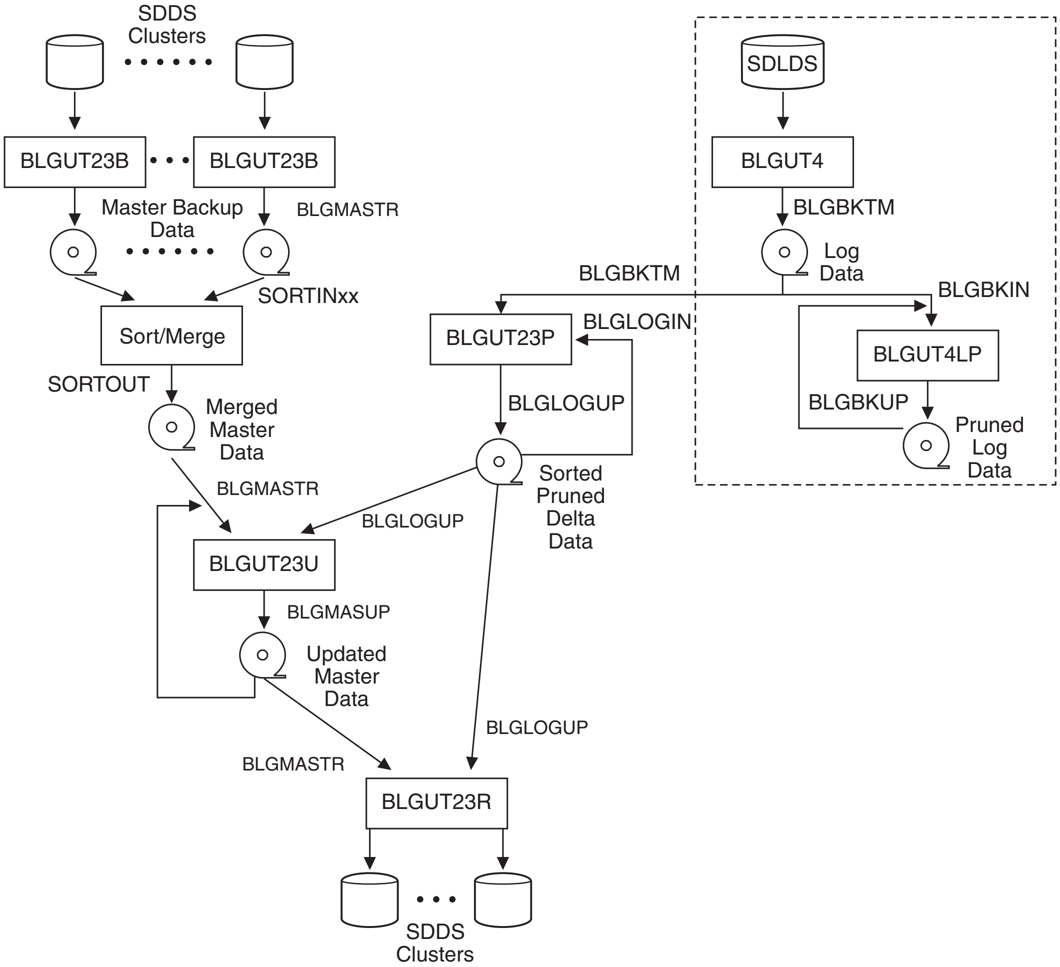


Figure 31. Detailed View of the BLGUT23 Utility Data Flow

32

BLGUT23B—Back Up the Database

The BLGUT23B utility is one in a series of BLGUT23 utilities that enable you to back up, prune, update, and restore your SDDS data. For an overview of these utilities, see “BLGUT23 Utilities Overview” on page 167. Use the BLGUT23B utility to offload an SDDS cluster to a sequential data set while the BLX-SP is active and the database is available to users.

> **Important!**
>
> The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

The BLGUT23B utility reads all records on a single SDDS cluster and writes them to a sequential data set in the same format used by the BLGUT4 utility. If the database consists of multiple clusters, you will need to run the BLGUT23B utility for each cluster. You will need to merge the multiple backup data sets with the DFSORT utility to produce a single master backup data set. Note that this data set can be quite large, so you may want to consider keeping it on tape or cartridge.

Syntax

The EXEC statement PARM field keywords to use with BLGUT23B are as follows:

```
PARM=‘[RECS=nn]’
```

RECS=nn
Specifies the number of logical records processed between each release of the SDDS clusters. *nn* must be in the range 1 ≤ nn ≤1 147 483 647. The default value is 100 logical records processed between releases. If you do not want to supply this option, do not code the PARM field on the EXEC statement.

The RECS keyword is ignored if sysplex mode is enabled.

Input

The input to BLGUT23B is one SDDS cluster to be backed up.

Output

The output from BLGUT23B is:

- A new, physical sequential data set
- A list of offloaded RNIDs in a list data set
- Summary and error messages in a message data set

- A return code of 0 if BLGUT23B finishes successfully
- A return code >0 if an error occurs

Restrictions

- The BLGUT23B utility cannot process a structured description data set (SDDS) with a key length of 8.
- The BLGUT23B utility does not process records that have not yet completed the file process. Updates to the SDDS made by those records are captured in the SDLDS and are merged by other BLGUT23 utility programs.

DD Statements

The DD statements required to run BLGUT23B are as follows:

BLGBLIST
A sequential data set (RECFM=VBA) containing one line of output for each logical record that is offloaded from the SDDS that you write to a system output device or data set.

BLGMASTR
A new physical sequential data set (RECFM=VB) that contains the backed-up SDDS records in BLGUT4 offload format.

BLGSDDS
The input SDDS VSAM data set to be backed up.

SYSPRINT
A sequential message data set (RECFM=VBA) that you can write to a system output device or data set.

Scenario Using the BLGUT23B Utility Program

The following example illustrates how you can use BLGUT23B to back up the records on both clusters of a two-cluster SDDS. The example shows the backup process as three steps in a single batch job. Performance would be enhanced if the steps were run as separate jobs, where STEP3 is started only after STEP1 and STEP2 have successfully completed. In this example, the BLGUT23B utility processes 50 logical records between each release of the SDDS.

To get the sample JCL to run this utility, refer to member BLGU23BJ in the SBLMSAMP library.

```
//BACKUP    JOB
//*        BACK UP THE FIRST SDDS CLUSTER
//STEP1     EXEC PGM=BLGUT23B,PARM='RECS=50',REGION=2048K
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGBLIST  DD  DISP=(NEW,CATLG),DSNAME=BACKUP01.LIST,
//             UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGSDDS   DD  DISP=SHR,DSNAME=BLM.#01.SDDS
//BLGMASTR  DD  DISP=(NEW,KEEP),DSNAME=BACKUP01,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=001111,
//             DCB=(RECFM=VB,LRECL=8202,BLKSIZE=0)
//*        BACK UP THE SECOND SDDS CLUSTER
//STEP2     EXEC PGM=BLGUT23B,PARM='RECS=50',REGION=2048K
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGBLIST  DD  DISP=(NEW,CATLG),DSNAME=BACKUP02.LIST,
//             UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGSDDS   DD  DISP=SHR,DSNAME=BLM.#02.SDDS
//BLGMASTR  DD  DISP=(NEW,KEEP),DSNAME=BACKUP02,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=002222,
//             DCB=(RECFM=VB,LRECL=8202,BLKSIZE=0)
//*        MERGE THE TWO BACKUP DATA SETS
//STEP3     EXEC PGM=SORT,REGION=2048K,COND=(0,NE)
//SYSOUT    DD  SYSOUT=A
//SORTIN01  DD  DISP=SHR,DSNAME=*.STEP1.BLGMASTR
//SORTIN02  DD  DISP=SHR,DSNAME=*.STEP2.BLGMASTR
//SORTOUT   DD  DISP=(NEW,KEEP),DSNAME=MASTER,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=003333,
//             DCB=(RECFM=VB,LRECL=8202,BLKSIZE=8206)
//SYSIN     DD  *
   MERGE  FIELDS=(5,4,A),FORMAT=CH
//
```

Figure 32. Backing up records on a two-cluster SDDS to a new data set on a tape.

In Step 3, DFSORT is used to merge the two data sets. In the MERGE FIELDS statement, specify the sort control values as shown (do not change the values). Be sure to specify the data sets to merge in the SORTINxx DD statements.

If you are not using DFSORT, refer to your sort product's documentation for the correct control statements and JCL to perform the merge. The key field begins in the first character of the logical record (the fifth position of the physical record). The length of the key field is 4 characters.

33

BLGUT23P—Prune the Offloaded Recovery Log Data Set

Attention: If you have implemented the Automatic Log Save Facility or the DB2 Extract Facility into your data recovery process, do not run the BLGUT4 utility until you fully understand how it may affect your data. Doing so can result in lost data. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about the Automatic Log Save and DB2 Extract facilities.

The BLGUT23P utility is one in a series of BLGUT23 utilities that enable you to back up, prune, update, and restore your SDDS data. For an overview of these utilities, see "BLGUT23 Utilities Overview" on page 167.

Use the BLGUT23P utility to prune offloaded data from a recovery log data set, to save only the last entry for a given record. The BLGUT4 utility offloads data from the SDLDS and saves it in chronological order. BLGUT23P prunes this data and writes it out in SDDS position number order. It can then be read into the next run of BLGUT23P (where additional offloaded data will be merged in), or it can be used to update the master backup data by the BLGUT23U or BLGUT23R utilities.

Important!
The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Input

The input for BLGUT23P is:

- The offloaded log data from the first step of BLGUT4
- The pruned and sorted log data from the last run of BLGUT23P

Output

The output from BLGUT23P is:

- A new, physical sequential data set. This is the pruned log data set that is used as input for additional BLGUT23P runs until it is used to update the master backup data with either BLGUT23U or BLGUT23R utility.
- A list of pruned records in a list data set.
- Summary and error messages in a message data set.
- A return code of 0 if BLGUT23P finishes successfully.
- A return code >0 if an error occurs.

Restrictions

- Use of this utility might conflict with the function of the Automatic Log Save or DB2 Extract facilities. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information.
- Do not mix the output files from BLGUT23P with the output files of the BLGUT4LP utility. The pruning rules and sort order are different.
- The record length (LRECL) of the BLGLOGUP data set must be at least as large as that of the BLGBKTM and BLGLOGIN data sets.

DD Statements

The DD statements required to run BLGUT23P are as follows:

BLGBKTM
The offloaded log save data set created by the first step of the BLGUT4 job. The pruned data set can be deleted after the BLGUT23P job completes unless it is required for the Automatic Log Save Facility or the DB2 Extract Facility.

BLGLOGIN
The pruned log data set from the last run of BLGUT23P. This data set, if any, should be deleted after the BLGUT23P job completes. If no pruned log data set is input, specify this statement as DD DUMMY.

BLGLOGUP
The updated pruned log data set (RECFM=VB). This is a new, physical sequential data set created by BLGUT23P. This data set will be one of the inputs to the BLGUT23U or BLGUT23R utility if the master backup data is to be updated. It is also used as input (BLGLOGIN) to the next run of the BLGUT23P job.

BLGPLIST
A sequential data set (RECFM=VBA, LRECL=137) containing one line of output for each record that is pruned from the BLGLOGIN and BLGBKTM data set. You can write this data set to a system output device or data set.

SYSPRINT
A sequential data set (RECFM=VBA) containing summary and error messages that you can write to a system output device or data set.

Scenarios Using the BLGUT23P Utility Program

The following example illustrates how you can use BLGUT23P to prune the offloaded log data. In STEP1, the BLGUT4 utility is included to show its relationship with BLGUT23P; however, BLGUT4 could be run in a separate job. If it is run in a separate job, do not run it again here, since the log data was offloaded and the SDLDS is reinitialized and ready to hold more log data.

To get the sample JCL to run this utility, refer to member BLGU23PJ in the SBLMSAMP library.

```
//PRUNE     JOB
//*        CREATE THE OFFLOADED LOG DATA SET
//STEP1     EXEC PGM=BLGUT4,PARM='SESS=01,NAME=5',REGION=2048K
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGBKTM   DD  DISP=(NEW,CATLG),DSNAME=LOGTEMP,
//             UNIT=SYSDA,SPACE=(TRK,(010,010)),
//             DCB=(RECFM=VB,LRECL=8202,BLKSIZE=0)
//*        CREATE THE PRUNED LOG DATA SET
//STEP2     EXEC PGM=BLGUT23P,REGION=2048K,COND=(0,NE)
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGPLIST  DD  DISP=(NEW,CATLG),DSNAME=PRUNE02.LIST,
//             UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGBKTM   DD  DISP=SHR,DSNAME=*.STEP1.BLGBKTM
//BLGLOGIN  DD  DISP=SHR,DSNAME=LOG001,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=000111
//BLGLOGUP  DD  DISP=(NEW,KEEP),DSNAME=LOG002,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=000222,
//             DCB=(RECFM=VB,LRECL=8202,BLKSIZE=0)
```

Figure 33. Creating the pruned log data set with input provided by BLGUT4

34

BLGUT23U—Update the Master Backup Data

The BLGUT23U utility is one in a series of BLGUT23 utilities that enable you to back up, prune, update, and restore your SDDS data. For an overview of these utilities, see “BLGUT23 Utilities Overview” on page 167.

Use the BLGUT23U utility to update the master backup data created by the BLGUT23B with the pruned log data created by the BLGUT23P utility. The master record is replaced with the log record if it is an update. It is deleted without replacement if it is a purged record. Once the master backup data is updated, you should delete the pruned log data set, since it is no longer needed.

Important!

The BLX-SP must be started before you can run Tivoli Information Management for z/OS utilities.

Input

The input to the BLGUT23U utility is:

- The master backup data from the BLGUT23B utility or the DFSORT utility, if run
- The pruned and sorted log data from the last run of BLGUT23P

Output

The output from BLGUT23U is:

- A new, physical sequential data set. This is the updated master backup data set.
- A list of updated records in a list data set.
- Summary and error messages in a message data set.
- A return code of `0` if BLGUT23U finishes successfully.
- A return code `>0` if an error occurs.

Restrictions

The record length (LRECL) of the BLGMASUP data set must be at least as large as that of the BLGMASTR and BLGLOGUP data sets.

DD Statements

The DD statements required to run BLGUT23U are as follows:

BLGLOGUP
The pruned log data set. This data set is created by the BLGUT23P utility and is used to update the master backup data set. This data set should be deleted after the BLGUT23U job finishes.

BLGMASTR
The master backup data set. This is initially created by either the BLGUT23B utility or the DFSORT utility, if run. After BLGUT23U is run at least once, the updated master from the previous run is used here. This data set should be deleted after the BLGUT23U job finishes.

BLGMASUP
The updated master backup data set (RECFM=VB). This is a new, physical sequential data set created by BLGUT23U. This data set is one of the inputs to the BLGUT23R utility if the database needs to be restored. It is also used as input (BLGMASTR) to the next BLGUT23U job.

BLGULIST
A sequential data set (RECFM=VBA) containing one line of output for each record that is updated in the BLGMASTR data set that you write to a system output device or data set.

SYSPRINT
A sequential message data set (RECFM=VBA) that you can write to a system output device or data set.

Scenarios Using the BLGUT23U Utility Program

The following example illustrates how you can use BLGUT23U to update the master backup data set with the pruned log data.

To get the sample JCL to run this utility, refer to member BLGU23UJ in the SBLMSAMP library.

```
//UPDATE    JOB
//*        UPDATE THE MASTER BACKUP DATA SET
//STEP1     EXEC PGM=BLGUT23U,REGION=2048K,COND=(0,NE)
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGULIST  DD  DISP=(NEW,CATLG),DSNAME=UPDATE01.LIST,
//              UNIT=3390,VOL=SER=223344,SPACE=(TRK,(10,10)),
//              DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGMASTR  DD  DISP=SHR,DSNAME=MASTER,
//              UNIT=TAPE,LABEL=(,SL),VOL=SER=003333
//BLGLOGUP  DD  DISP=SHR,DSNAME=LOG002,
//              UNIT=TAPE,LABEL=(,SL),VOL=SER=000222
//BLGMASUP  DD  DISP=(NEW,KEEP),DSNAME=MASTER,
//              UNIT=TAPE,LABEL=(,SL),VOL=SER=004444,
//              DCB=(RECFM=VB,LRECL=8202,BLKSIZE=8206)
```

Figure 34. Updating the master backup data set to create input for the BLGUT23R utility or for further runs of BLGUT23U

35

BLGUT23R—Restore the SDDS

The BLGUT23R utility is one in a series of BLGUT23 utilities that enable you to back up, prune, update, and restore your SDDS data. For an overview of these utilities, see "BLGUT23 Utilities Overview" on page 167.

Use BLGUT23R to restore the SDDS from the latest master backup data set created by either the BLGUT23B or BLGUT23U utility. If the master backup data does not contain the latest database activity, offload the log data with the BLGUT4 utility, prune the data with BLGUT23P, and input it directly to this utility.

BLGUT23R can optionally resequence the SDDS position numbers to remove the gaps caused by record deletion. If you choose to renumber the SDDS you **must** run BLGUT1 to rebuild the SDIDS. If you do not resequence the SDDS and the SDIDS is not corrupted, you do not need to run the BLGUT1 utility.

Note: Unlike the BLGUT4 and BLGUT3 utilities which manipulate the SDDS and SDIDS, the BLGUT23B and BLGUT23R utilities only back up and restore the SDDS. You must use BLGUT1 (not the BLGUT1M utility) to rebuild the SDIDS under the following circumstances:

- The SDDS was resequenced.
- The SDIDS was corrupted.
- The SDIDS is not synchronized with the SDDS built or restored by BLGUT23R.

Important!

| The BLX-SP must be started before you can run Tivoli Information Management for
| z/OS utilities.

When sysplex mode is not enabled, you can prevent users from accessing the data set you need to work on by first issuing the FREE command to free the data set. Then use the REALLOC command with the UTIL keyword. See "FREE Command" on page 15 and "REALLOC Command" on page 33 for complete information about the FREE and REALLOC commands.

When sysplex mode is enabled, run the utility while the data set is freed. The utility can still access the data set even though users cannot. Then, run the REALLOC command after the utility completes to enable users to access the data set again. You do not need to use the UTIL keyword to prevent users from accessing the data set while this utility runs.

Syntax

The EXEC statement PARM field keywords to use with BLGUT23R are as follows:

```
PARM='SESS=aa[,NAME=n][,RESEQ][,CLUST=(n1,n2,n3...)]'
```

SESS=aa
Specifies the suffix for the session-parameters member name that BLGUT23R uses. Use 1 or 2 alphanumeric or national characters for *aa*. This session-parameters member contains the structure definition of the database to be used. This keyword is required.

NAME=n
Specifies the name of the database, in a session-parameters member with multiple database definitions, to use. This keyword is optional, and, if omitted, defaults to 5. Valid values are 4, 5, 6, 7, 8, or 9.

RESEQ
Indicates that you want to resequence the records on the database. This keyword is optional, and, if omitted, the records are restored with the original VSAM sequence numbers. If RESEQ is specified, the BLGUT1 utility must be used to rebuild the SDIDS.

The RESEQ and CLUST keywords are mutually exclusive.

CLUST=(n1,n2,n3...)
Identifies the specific SDDS clusters in a multiple-cluster SDDS database that are to be built. This keyword is optional, and if omitted, all SDDS clusters for the specified database are built. The value for this keyword can be specified as either a single number (e.g., `CLUST=15`) or a parenthesized list of numbers separated by commas (e.g., `CLUST=(1,3,6)`). If you enter a list, it does not matter in what order you place the numbers.

The RESEQ and CLUST keywords are mutually exclusive.

Input

The input to the BLGUT23R utility is:

- The latest master backup data from the BLGUT23U, BLGUT23B, or the DFSORT utility.
- The pruned and sorted log data from the last run of the BLGUT23P utility. This data set is optional.

Output

The output from BLGUT23R is:

- The restored SDDS cluster or clusters
- A list of processed RNIDs in a list data set
- A list of updated records in a list data set
- A list of record segments that were not restored to the SDDS in a list data set
- Summary and error messages in a message data set
- A return code of `0` if BLGUT23R finishes successfully
- A return code `>0` if an error occurs

Restrictions

The BLGUT23R utility cannot restore a structured description data set (SDDS) with a key length of 8.

DD Statements

The DD statements required to run BLGUT23R are as follows:

BLGELIST
A sequential data set (RECFM=VBA) containing one line of output for each logical record that was not restored to the SDDS that you can write to a data set. An error message with the RNID for each record that is not restored is written to DD BLGRLIST.

BLGLOGUP
The pruned log data set. This data set is created by the BLGUT23P utility and used to update the master backup data set. If no pruned log data set is input, this should be specified as DD DUMMY.

BLGMASTR
The master backup data set. This data set is initially created by either BLGUT23B or the DFSORT utility, if run. After BLGUT23U runs at least once, use the updated master from the last run here.

BLGRLIST
A sequential data set (RECFM=VBA) containing one line of output for each logical record that is processed that you can write to a system output device or data set.

BLGULIST
A sequential data set (RECFM=VBA) containing one line of output for each record that is updated in the BLGMASTR data set that you can write to a system output device or data set.

SYSPRINT
A sequential message data set (RECFM=VBA) that you can write to a system output device or data set.

Scenarios Using the BLGUT23R Utility Program

The following example illustrates how you can use BLGUT23R to restore the SDDS from the master backup data set. Pruned log data is also provided. The session-parameters member suffix is specified to provide the new database structure. The default database name of 5 is used. All clusters are to be restored and the database is to be resequenced.

To get the sample JCL to run this utility, refer to member BLGU23RJ in the SBLMSAMP library.

```
//RESTORE   JOB
//*        RESTORE THE SDDS
//STEP1     EXEC PGM=BLGUT23R,PARM='SESS=01,RESEQ',REGION=2048K
//STEPLIB   DD  DISP=SHR,DSNAME=BLM.SBLMMOD1
//SYSPRINT  DD  SYSOUT=A
//BLGELIST  DD  DISP=(NEW,CATLG),DSNAME=ERROR.LIST,
//             UNIT=3390,VOLUME=223344,SPACE=(TRK,(1,1)),
//             DCB=(RECFM=VBA,LRECL=8202,BLKSIZE=8206)
//BLGRLIST  DD  DISP=(NEW,CATLG),DSNAME=RESTORE.LIST,
//             UNIT=3390,VOLUME=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGULIST  DD  DISP=(NEW,CATLG),DSNAME=UPDATE02.LIST,
//             UNIT=3390,VOLUME=223344,SPACE=(TRK,(10,10)),
//             DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3200)
//BLGMASTR  DD  DISP=SHR,DSNAME=MASTER,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=004444
//BLGLOGUP  DD  DISP=SHR,DSNAME=LOG003,
//             UNIT=TAPE,LABEL=(,SL),VOL=SER=000333
```

Figure 35. Restoring all clusters in the SDDS with resequenced records

A

Relating Publications to Specific Tasks

Your data processing organization can have many different users performing many different tasks. The books in the Tivoli Information Management for z/OS library contain task-oriented scenarios to teach users how to perform the duties specific to their jobs.

The following table describes the typical tasks in a data processing organization and identifies the Tivoli Information Management for z/OS publication that supports those tasks. See "The Tivoli Information Management for z/OS Library" on page 193 for more information about each book.

Typical Tasks

Table 5. Relating Publications to Specific Tasks

If You Are:	And You Do This:	Read This:
Planning to Use Tivoli Information Management for z/OS	Identify the hardware and software requirements of Tivoli Information Management for z/OS. Identify the prerequisite and corequisite products. Plan and implement a test system.	*Tivoli Information Management for z/OS Planning and Installation Guide and Reference*
Installing Tivoli Information Management for z/OS	Install Tivoli Information Management for z/OS. Define and initialize data sets. Create session-parameters members.	*Tivoli Information Management for z/OS Planning and Installation Guide and Reference* *Tivoli Information Management for z/OS Integration Facility Guide*
	Define and create multiple Tivoli Information Management for z/OS BLX-SPs.	*Tivoli Information Management for z/OS Planning and Installation Guide and Reference*
	Define and create APPC transaction programs for clients.	*Tivoli Information Management for z/OS Client Installation and User's Guide*
	Define coupling facility structures for sysplex data sharing.	*Tivoli Information Management for z/OS Planning and Installation Guide and Reference*
Diagnosing problems	Diagnose problems encountered while using Tivoli Information Management for z/OS	*Tivoli Information Management for z/OS Diagnosis Guide*

Table 5. Relating Publications to Specific Tasks (continued)

If You Are:	And You Do This:	Read This:
Administering Tivoli Information Management for z/OS	Manage user profiles and passwords. Define and maintain privilege class records. Define and maintain rules records.	*Tivoli Information Management for z/OS Program Administration Guide and Reference* *Tivoli Information Management for z/OS Integration Facility Guide*
	Define and maintain USERS record. Define and maintain ALIAS record. Implement GUI interface. Define and maintain command aliases and authorizations.	*Tivoli Information Management for z/OS Program Administration Guide and Reference*
	Implement and administer Notification Management. Create user-defined line commands. Define logical database partitioning.	*Tivoli Information Management for z/OS Program Administration Guide and Reference*
	Create or modify GUI workstation applications that can interact with Tivoli Information Management for z/OS. Install the Tivoli Information Management for z/OS Desktop on user workstations.	*Tivoli Information Management for z/OS Desktop User's Guide*
Maintaining Tivoli Information Management for z/OS	Set up access to the data sets. Maintain the databases. Define and maintain privilege class records.	*Tivoli Information Management for z/OS Planning and Installation Guide and Reference* *Tivoli Information Management for z/OS Program Administration Guide and Reference*
	Define and maintain the BLX-SP. Run the utility programs.	*Tivoli Information Management for z/OS Operation and Maintenance Reference*
Programming applications	Use the application program interfaces.	*Tivoli Information Management for z/OS Application Program Interface Guide*
	Use the application program interfaces for Tivoli Information Management for z/OS clients.	*Tivoli Information Management for z/OS Client Installation and User's Guide*
	Create Web applications using or accessing Tivoli Information Management for z/OS data.	*Tivoli Information Management for z/OS World Wide Web Interface Guide*

Table 5. Relating Publications to Specific Tasks (continued)

If You Are:	And You Do This:	Read This:
Customizing Tivoli Information Management for z/OS	Design and implement a Change Management system. Design and implement a Configuration Management system. Design and implement a Problem Management system.	*Tivoli Information Management for z/OS Problem, Change, and Configuration Management*
	Design, create, and test terminal simulator panels or terminal simulator EXECs. Customize panels and panel flow.	*Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* *Tivoli Information Management for z/OS Panel Modification Facility Guide*
	Design, create, and test Tivoli Information Management for z/OS formatted reports.	*Tivoli Information Management for z/OS Data Reporting User's Guide*
	Create a bridge between NetView® and Tivoli Information Management for z/OS applications. Integrate Tivoli Information Management for z/OS with Tivoli distributed products.	*Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*
Assisting Users	Create, search, update, and close change, configuration, or problem records. Browse or print Change, Configuration, or Problem Management reports.	*Tivoli Information Management for z/OS Problem, Change, and Configuration Management*
	Use the Tivoli Information Management for z/OS Integration Facility.	*Tivoli Information Management for z/OS Integration Facility Guide*
Using Tivoli Information Management for z/OS	Learn about the Tivoli Information Management for z/OS panel types, record types, and commands. Change a user profile.	*Tivoli Information Management for z/OS User's Guide*
	Learn about Problem, Change, and Configuration Management records.	*Tivoli Information Management for z/OS Problem, Change, and Configuration Management*
	Receive and respond to Tivoli Information Management for z/OS messages.	*Tivoli Information Management for z/OS Messages and Codes*
	Design and create reports.	*Tivoli Information Management for z/OS Data Reporting User's Guide*

B

Tivoli Information Management for z/OS Courses

Education Offerings

Tivoli Information Management for z/OS classes are available in the United States and in the United Kingdom. For information about classes outside the U.S. and U.K., contact your local IBM representative or visit **http://www.training.ibm.com** on the World Wide Web.

United States

IBM Education classes can help your users and administrators learn how to get the most out of Tivoli Information Management for z/OS. IBM Education classes are offered in many locations in the United States and at your own company location.

For a current schedule of available classes or to enroll, call 1-800-IBM TEACh (1-800-426-8322). On the World Wide Web, visit:

http://www.training.ibm.com

to see the latest course offerings.

United Kingdom

In Europe, the following public courses are held in IBM's central London education centre at the South Bank at regular intervals. On-site courses can also be arranged.

For course schedules and to enroll, call Enrollments Administration on 0345 581329, or send an e-mail note to:

contact_educ_uk@vnet.ibm.com

On the World Wide Web, visit:

http://www.europe.ibm.com/education-uk

to see the latest course offerings.

C

Where to Find More Information

The Tivoli Information Management for z/OS library is an integral part of Tivoli Information Management for z/OS. The books are written with particular audiences in mind. Each book covers specific tasks.

The Tivoli Information Management for z/OS Library

The publications shipped automatically with each Tivoli Information Management for z/OS Version 7.1 licensed program are:

- *Tivoli Information Management for z/OS Application Program Interface Guide*
- *Tivoli Information Management for z/OS Client Installation and User's Guide **
- *Tivoli Information Management for z/OS Data Reporting User's Guide **
- *Tivoli Information Management for z/OS Desktop User's Guide*
- *Tivoli Information Management for z/OS Diagnosis Guide **
- *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications **
- *Tivoli Information Management for z/OS Integration Facility Guide **
- *Tivoli Information Management for z/OS Licensed Program Specification*
- *Tivoli Information Management for z/OS Master Index, Glossary, and Bibliography*
- *Tivoli Information Management for z/OS Messages and Codes*
- *Tivoli Information Management for z/OS Operation and Maintenance Reference*
- *Tivoli Information Management for z/OS Panel Modification Facility Guide*
- *Tivoli Information Management for z/OS Planning and Installation Guide and Reference*
- *Tivoli Information Management for z/OS Program Administration Guide and Reference*
- *Tivoli Information Management for z/OS Problem, Change, and Configuration Management**
- *Tivoli Information Management for z/OS Reference Summary*
- *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*
- *Tivoli Information Management for z/OS User's Guide*
- *Tivoli Information Management for z/OS World Wide Web Interface Guide*

| **Note:** Publications marked with an asterisk (*) are shipped in softcopy format only.

Also included is the Product Kit, which includes the complete online library on CD-ROM.

To order a set of publications, specify order number SBOF-7028-00.

Additional copies of these items are available for a fee.

Publications can be requested from your Tivoli or IBM representative or the branch office serving your location. Or, in the U.S., you can call the IBM Publications order line directly by dialing 1-800-879-2755.

The following descriptions summarize all the books in the Tivoli Information Management for z/OS library.

Tivoli Information Management for z/OS Application Program Interface Guide, SC31-8737-00, explains how to use the low-level API, the high-level API, and the REXX interface to the high-level API. This book is written for application and system programmers who write applications that use these program interfaces.

Tivoli Information Management for z/OS Client Installation and User's Guide, SC31-8738-00, describes and illustrates the setup and use of Tivoli Information Management for z/OS's remote clients. This book shows you how to use Tivoli Information Management for z/OS functions in the AIX®, CICS®, HP-UX, OS/2®, Sun Solaris, Windows NT®, and OS/390 UNIX® System Services environments. Also included in this book is complete information about using the Tivoli Information Management for z/OS servers.

Tivoli Information Management for z/OS Data Reporting User's Guide, SC31-8739-00, describes various methods available to produce reports using Tivoli Information Management for z/OS data. It describes Tivoli Decision Support for Information Management (a Discovery Guide for Tivoli Decision Support), the Open Database Connectivity (ODBC) Driver for Tivoli Information Management for z/OS, and the Report Format Facility. A description of how to use the Report Format Facility to modify the standard reports provided with Tivoli Information Management for z/OS is provided. The book also illustrates the syntax of report format tables (RFTs) used to define the output from the Tivoli Information Management for z/OS REPORT and PRINT commands. It also includes several examples of modified RFTs.

Tivoli Information Management for z/OS Desktop User's Guide, SC31-8740-00, describes how to install and use the sample application provided with the Tivoli Information Management for z/OS Desktop. The Tivoli Information Management for z/OS Desktop is a Java-based graphical user interface for Tivoli Information Management for z/OS. Information on how to set up data model records to support the interface and instructions on using the Desktop Toolkit to develop your own Desktop application are also provided.

Tivoli Information Management for z/OS Diagnosis Guide, GC31-8741-00, explains how to identify a problem, analyze its symptoms, and resolve it. This book includes tools and information that are helpful in solving problems you might encounter when you use Tivoli Information Management for z/OS.

Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications, SC31-8744-00, describes the steps to follow to make an automatic connection between NetView and Tivoli Information Management for z/OS applications. It also explains how to customize the application interface which serves as an application enabler for the NetView Bridge and discusses the Tivoli Information Management for z/OS NetView AutoBridge. Information on interfacing Tivoli Information Management for z/OS with other Tivoli management software products or components is provided for Tivoli Enterprise Console, Tivoli Global Enterprise Manager, Tivoli Inventory, Tivoli Problem Management, Tivoli Software Distribution, and Problem Service.

Tivoli Information Management for z/OS Integration Facility Guide, SC31-8745-00, explains the concepts and structure of the Integration Facility. The Integration Facility provides a task-oriented interface to Tivoli Information Management for z/OS that makes the

Tivoli Information Management for z/OS applications easier to use. This book also explains how to use the panels and panel flows in your change and problem management system.

Tivoli Information Management for z/OS Master Index, Glossary, and Bibliography, SC31-8747-00, combines the indexes from each hardcopy book in the Tivoli Information Management for z/OS library for Version 7.1. Also included is a complete glossary and bibliography for the product.

Tivoli Information Management for z/OS Messages and Codes, GC31-8748-00, contains the messages and completion codes issued by the various Tivoli Information Management for z/OS applications. Each entry includes an explanation of the message or code and recommends actions for users and system programmers.

Tivoli Information Management for z/OS Operation and Maintenance Reference, SC31-8749-00, describes and illustrates the BLX-SP commands for use by the operator. It describes the utilities for defining and maintaining data sets required for using the Tivoli Information Management for z/OS licensed program, Version 7.1.

Tivoli Information Management for z/OS Panel Modification Facility Guide, SC31-8750-00, gives detailed instructions for creating and modifying Tivoli Information Management for z/OS panels. It provides detailed checklists for the common panel modification tasks, and it provides reference information useful to those who design and modify panels.

Tivoli Information Management for z/OS Planning and Installation Guide and Reference, GC31-8751-00, describes the tasks required for installing Tivoli Information Management for z/OS. This book provides an overview of the functions and optional features of Tivoli Information Management for z/OS to help you plan for installation. It also describes the tasks necessary to install, migrate, tailor, and start Tivoli Information Management for z/OS.

Tivoli Information Management for z/OS Problem, Change, and Configuration Management, SC31-8752-00, helps you learn how to use Problem, Change, and Configuration Management through a series of training exercises. After you finish the exercises in this book, you should be ready to use other books in the library that apply more directly to the programs you use and the tasks you perform every day.

Tivoli Information Management for z/OS Program Administration Guide and Reference, SC31-8753-00, provides detailed information about Tivoli Information Management for z/OS program administration tasks, such as defining user profiles and privilege classes and enabling the GUI user interface.

Tivoli Information Management for z/OS Reference Summary, SC31-8754-00, is a reference booklet containing Tivoli Information Management for z/OS commands, a list of p-words and s-words, summary information for PMF, and other information you need when you use Tivoli Information Management for z/OS.

Tivoli Information Management for z/OS Terminal Simulator Guide and Reference, SC31-8755-00, explains how to use terminal simulator panels (TSPs) and EXECs (TSXs) that let you simulate an entire interactive session with a Tivoli Information Management for z/OS program. This book gives instructions for designing, building, and testing TSPs and TSXs, followed by information on the different ways you can use TSPs and TSXs.

Tivoli Information Management for z/OS User's Guide, SC31-8756-00, provides a general introduction to Tivoli Information Management for z/OS and databases. This book has a series of step-by-step exercises to show beginning users how to copy, update, print, create, and delete records, and how to search a database. It also contains Tivoli Information Management for z/OS command syntax and descriptions and other reference information.

Tivoli Information Management for z/OS World Wide Web Interface Guide, SC31-8757-00, explains how to install and operate the features available with Tivoli Information Management for z/OS that enable you to access a Tivoli Information Management for z/OS database using a Web browser as a client.

Other related publications include the following:

Tivoli Decision Support: Using the Information Management Guide is an online book (in portable document format) that can be viewed with the Adobe Acrobat Reader. This book is provided with Tivoli Decision Support for Information Management (5697-IMG), which is a product that enables you to use Tivoli Information Management for z/OS data with Tivoli Decision Support. This book describes the views and reports provided with the Information Management Guide.

IBM Redbooks™ published by IBM's International Technical Support Organization are also available. For a list of redbooks related to Tivoli Information Management for z/OS and access to online redbooks, visit Web site **http://www.redbooks.ibm.com** or **http://www.support.tivoli.com**

Index

Tivoli

File Number: S370/30xx/4300
Program Number: 5697-SD9

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8749-00