Tivoli

Information Management for z/OS

Planning and Installation Guide and Reference

Version 7.1

GC31-8751-00

Information Management for z/OS

Planning and Installation Guide and Reference

Version 7.1

GC31-8751-00

Tivoli Information Management for z/OS Planning and Installation Guide and Reference

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.

Programming Interfaces

This book is intended to help the customer install and tailor Tivoli Information Management for z/OS. It contains information needed to install Tivoli Information Management for z/OS.

This *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* primarily documents information that is NOT intended to be used as Programming Interfaces of Tivoli Information Management for z/OS.

This *Tivoli Information Management for z/OS Planning and Installation Guide and Reference* also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Tivoli Information Management for z/OS. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking: Programming Interface information.

Contents

Preface

This book contains information to help you plan for the installation of Tivoli® Information Management for z/OS and provides instructions to help you install and tailor it. It describes the migration and conversion procedures necessary to move to Tivoli Information Management for z/OS Version 7.1 from previous versions of the product.

Information is provided to assist you with migrating from the following:

- Tivoli Service Desk for OS/390® Version 1.2
- TME 10™ Information/Management Version 1.1
- Information/Management Version 6.3, Version 6.2, Version 6.1
- Information/Management Version 5.1
- Information/Management Version 4.2.2, Version 4.2, Version 4.1

As shown in this list of previous versions of the predecessor product, the version number of Information/Management was reset to Version 1.1 (1997).

Who Should Read This Document

This book is intended for:

- Product planners, who assign personnel to manage Tivoli Information Management for z/OS
- System analysts or programmers, who install Tivoli Information Management for z/OS and provide procedures for using Tivoli Information Management for z/OS
- Database administrators, who maintain databases.

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 407.

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 Document Contains

- Chapter 1, "Overview for Planning and Installing," contains a step-by-step procedure and general checklist that guides you through planning and installing Tivoli Information Management for z/OS.
- Chapter 2, "The Base Tivoli Information Management for z/OS Product," and Chapter 3, "Tivoli Information Management for z/OS Features and Options," help you to understand Tivoli Information Management for z/OS by describing the many Tivoli Information Management for z/OS functions and features that you can consider for your installation.
- Chapter 4, "Evaluating Performance," through Chapter 8, "Assigning and Scheduling Personnel and Systems," help you plan the resources Tivoli Information Management for z/OS requires.
- Chapter 9, "Migrating from Previous Versions," describes migration tasks that must be performed if you are currently using an earlier version of Tivoli Information Management for z/OS or Information/Management, the predecessor product.
- Chapters 10, "Setting Up Your BLX-SP," through 16, "Starting Tivoli Information Management for z/OS," explain the tasks that you must perform to install, tailor and start Tivoli Information Management for z/OS. Examples are included.
- Chapter 17, "Loading Records Provided with Tivoli Information Management for z/OS," describes how to load the data model records and other types of records that are shipped with Tivoli Information Management for z/OS to support particular applications or interfaces.

- The appendixes contain reference material that help you to install Tivoli Information Management for z/OS.

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 host 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.

Commands, such as END, CONTROL, RESUME, or DOWN, appear in all capital letters in text. Although not commands, the user responses YES and NO also appear in capital letters.

Fields designated with <R> are required fields. You must enter information in these fields on the current panel before you can continue to the next panel.

A plus (+) sign appearing to the *right* of the message indicates that more message information is available. If the plus sign appears to the *left* of the message, there are more messages for you to view. Type **;help** on the command line and then press Enter to view these messages.

The Use of Panel Style in This Book

Two panel styles are available: the standard panel style and the enhanced panel style. The style of panel does not affect the data that must be entered from it.

Except where noted, this book uses the Tivoli Information Management for z/OS standard panel style when showing you how a panel looks.

For more information about the enhanced panel style, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* and the *Tivoli*

Information Management for z/OS User's Guide. Also see "Enhanced Panel Style" on page 50 for a description of the enhanced panel style and "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199 for information on how to select a panel style.

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**

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

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.

What's New in Tivoli Information Management for z/OS?

The enhancements made in Tivoli Information Management for z/OS Version 7.1 are listed in "New in Tivoli Information Management for z/OS Version 7.1".

New in Tivoli Information Management for z/OS Version 7.1

Base Product Enhancements	
Support for Parallel Sysplex®	The cross-memory VSAM I/O design and the Multisystem Database Access (MSDA) feature is replaced with a new design that takes advantage of VSAM record-level sharing (RLS) in a parallel sysplex. The primary benefits of the new design are improved performance, reliability, and ease of installation for VSAM access, especially for shared data sets (although users of nonshared data sets will benefit as well). In support of this, the FREE, REALLOC, QUERY, and BRDCST operator commands are also redesigned and enhanced for the parallel sysplex environment. A non-sysplex mode using the previous cross-memory design is still supported, but only for nonshared data sets. *If you are currently using MSDA to share data sets in a Tivoli Information Management for z/OS database, you must now use sysplex data sharing to continue sharing data sets.* If you do not share data sets, the use of parallel sysplex is optional. You can run without sysplex support, or use RLS and other enhancements if you have a parallel sysplex. For more information about setting up for parallel sysplex, see "Setting Up for Sysplex Data Sharing" on page 151. For a description of the changes to the BRDCST, FREE, REALLOC, and QUERY commands, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

Base Product Enhancements	
Enhanced date/time support	Enhancements in this release include the following: ■ Date/time format enhancements – Date and time values in a database can now be processed by different users using different external formats. Each user can select a preferred date format in the user profile. In addition, if an installation has written multiple time conversion routines, each session member can specify the desired time conversion routine for users of that session member. All dates and times in records are now stored in the database (in the SDDS) in internal format. Dates and times from records created in previous versions (in external format) will be automatically converted to internal format when the record is accessed (for example, displayed, updated, or printed on a report). If you have existing records in your database, and you file the records under this new version, the dates and times in the records will be stored in internal format. For more information about date processing, see "Enabling Alternative Date and Time-of-Day Formats" on page 227. ■ Time zone support – The system administrator can optionally define relationships between date fields and their related time fields. This relationship enables users to work with date and time values in their own local time zone while still storing data in a common time zone to enable accurate reporting of durations. When this feature (called universal time processing) is enabled, date and time values are stored in the database in universal time. For more information about using the universal time processing function, see "Implementing Universal Time Processing" on page 251. ■ Session-parameters member changes – As a result of the date/time format and time zone enhancements, the BLGPARMS macro, which governs the operation of a user's session, is changed. The DATECNV keyword is modified and the following keywords are added: DATEFMT, ODATEFMT, TIMEZONE, and OTIMEZON. For more information about these keywords, refer to "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318. ■ User profile enhancements – Users can now specify the desired date format and, if universal time processing is enabled, the desired local time zone through the ″User and database defaults″ option in the user profile. The use of this option is described in "Specifying Preferences in the User Profile" on page 230. More information about user profile options is available in the *Tivoli Information Management for z/OS User's Guide*.
Ability to choose whether field numbers for data entry fields are displayed	Users can hide the entry field numbers displayed on Tivoli Information Management for z/OS panels by setting the ″Display entry field numbers″ option to NO in the screen control defaults section of their user profile. **Note:** This change is not compatible with use of the ISPF graphical user interface. If you use the ISPF GUI, you should not use this option. More information about user profile options is available in the *Tivoli Information Management for z/OS User's Guide*.

Base Product Enhancements	
Enhanced entry of change approver/reviewer data	List processor panels are now displayed to accept the entry of a list of privilege classes that should approve or review a new change request. The new panels are displayed when new change records are created, or when existing change records not containing change approver or change reviewer data are updated. If an existing change request record already contains change approver or change reviewer data, the new list processor panels are not invoked and the former approver and reviewer entry and display panels are shown instead. If you have modified or customized the following panels, you must modify them as described on page 107 so that they will work with this enhancement: BLG0CU01 (Change Request Summary), BLG0CU00 (Change Request Summary), BLG0S020 (Change Summary Display). The new list processor panels include: BLGLAPVR (Change Approver Entry), BLGLAPST (Change Approver Display), BLGLREVR (Change Reviewer Entry), and BLGLREVD (Change Reviewer Display). For illustrations, refer to the *Tivoli Information Management for z/OS Problem, Change, and Configuration Management*. Also, a new program exit, BLG02041, is provided to set approval status for the current change record. The program exit can be used when approver data is entered using the BLGLAPVR list processor panel. For a description of this program exit, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*. If you approve or reject change requests through an API application, you can optionally specify a new API parameter to specify an approver name. If an approver name is specified, it is used in place of the privilege class name specified in the transaction. Applications using the low-level API can specify an approver by setting the PICAAPVR field on Change Record Approval (T112) transactions. Applications using the high-level API can specify the name of an approver through the APPROVER parameter in Change Approval (HL10) transactions. For more information about these transactions, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*. This enhancement does not apply to the Integration Facility.
Enhanced entry of privilege class users	A list processor panel (BLGLJ300) is now displayed to accept the entry of up to 19 274 eligible users of a privilege class (previously, the limit was 24). The new list processor panel is displayed when entering eligible user IDs for new privilege classes. It is also displayed if you update existing privilege class records that do not contain at least one eligible user. If an existing privilege class record is updated and it contains eligible user data, the former data-entry panel is displayed instead of the new list processor panel. To migrate all existing privilege class records in your database to use the new list processor panels (regardless of their contents), you can run the BLGTPRIV TSX. By running this TSX, you can ensure that a consistent set of panels is used to enter, update, or display privilege class records. For more information about entering users of privilege classes, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*. For information about migrating privilege class records with the BLGTPRIV TSX, see 111.

<table>
<tr><th colspan="2">Base Product Enhancements</th></tr>
<tr><td>E-mail support for notification processing</td><td>With the improvements made to notification processing, you now have more choices in how you can notify people in your organization who are responsible for resolving problems. You can send e-mail messages quickly to support personnel to help ensure your service level agreements are not compromised.

A new set of terminal simulator EXECs (TSXs) is provided for you to use or customize. You can send e-mail messages immediately or place them on a queue. You can also send escalation notices. Advantages of this new method of notification over former methods include: being able to use an SMTP server on a platform other than MVS™, flexibility in processing messages, support for ″hot″ queues and normal queues, and ease of defining message content.

The existing notification process is still available if you choose not to use the new TSXs. The new TSXs can be invoked interactively or through the APIs.

For more information about using sending e-mail messages with TSXs, refer to the Tivoli Information Management for z/OS Program Administration Guide and Reference.</td></tr>
<tr><td>Improved problem resolution</td><td>Your help desk analysts can start building a more intelligent database by marking a problem resolution as a ″solution.″ The problem resolution text can be added to a solution knowledge base, and the data can be indexed and queried through use of the Text Search Engine, a component of OS/390 Text Search. Highlights include the following:
Problem resolution data can be marked as solutions through a 3270, Desktop, or API (Web) interface.
Administrators can create indexes for the Text Search Engine through a Tivoli Information Management for z/OS panel interface.
Batch utilities are provided to enable you to add to and update the data in the Text Search indexes and to maintain and get status information for the indexes.
Users can enter text search arguments, including fuzzy searches, and receive results in a search results list. Searches can be done through the 3270, Desktop, and Web connector interfaces.
The ability to search the text data has been added to the API search transaction.
For more information about using OS/390 Text Search with Tivoli Information Management for z/OS to create a knowledge base, refer to the Tivoli Information Management for z/OS Program Administration Guide and Reference. Information on how to search for data stored in a knowledge base is available in the Tivoli Information Management for z/OS User's Guide.</td></tr>
</table>

Base Product Enhancements	
Easier way to create lists of validation data for data model records	You now have an easier way to enter lists of validation data for data attribute records and validation records. A new panel, BLGLVLSB, is available by selecting the "validation data basic" option on panel BLG0VU70 (Data Attribute Summary) or BLG0VU50 (Validation Summary). The new entry panel lets you enter longer fields of validation entry and description text so that you no longer have to use the L line command or scroll the columns to enter longer data. For more information about this new panel refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*. The BLG0VU70 panel is also enhanced to enable you to enter help text for a particular data attribute record. This help text is displayed when the user enters the ;HELP command on BLGLVSEL (Validation data).
Data model record panels enhanced	Additional fields are added to certain panels related to data view records and data attribute records. The data entry, display, and inquiry panels for data view records now include a field to specify the name of a child data view attribute record (panels BLG0V600, BLG0VE61, BLG0VQ61). The data entry, display, and inquiry panels for the response processing of data attribute records are enhanced to include a field to specify that a reply is always data (panels BLG0VE710, BLG0VE72, BLG0VQ72). You can now define groups of related list data fields to be managed as tables by the Desktop. A new panel, BLGLTBDL (Desktop Table Definition), is available by selecting the Desktop tables option on panel BLG0VU60 (Data View Summary). This allows you to define table names and associated fields (list processor data) for the Desktop.
Parenthetical freeform searches	Parentheses can be included around freeform search arguments to specify the order in which arguments should be evaluated. Nesting is supported.
HELP STATUS command improved	The output from the HELP STATUS command is improved to display information about optional features or functions you can implement. The output now includes identification of the time zone selected for universal time processing, the trigger character used by data attribute records, and the database containing data model records. Information about the HELP STATUS command is available in the *Tivoli Information Management for z/OS User's Guide*.

Publication Changes	
Tivoli Information Management for z/OS Diagnosis Guide	The *Tivoli Information Management for z/OS Diagnosis Guide* is updated with more current information about diagnosing problems related to the use of the most recent functions or interfaces provided in the last few releases of Tivoli Information Management for z/OS.

API Enhancements	
API transaction for problem resolution searches	Support for using Tivoli Information Management for z/OS with OS/390 Text Search is added to the HLAPI, HLAPI/REXX, and all client HLAPI platforms. Application writers can specify the name of a text search index and search arguments as input parameters on HLAPI transaction HL11 (Record Inquiry) or the HLAPI/REXX transaction SEARCH.

API Enhancements	
Specify privilege class name for change request approver	The HLAPI transaction HL10 (Change Approval), and LLAPI transaction T112 (Change Record Approval) are enhanced to enable you to optionally specify the privilege class name of a change request approver in place of the privilege class name used in the transaction. In HL10 transactions, you can specify a name with the APPROVER parameter. In T112 transactions, you can specify a name through the PICAAPVR field.
New BLGUT18 utility to build static tables from data model records	If your API applications use data view and data attribute records when interacting with the data base, you can now take advantage of the BLGUT18 utility to reduce the time required to start your application. The BLGUT18 utility creates static program interface data tables (PIDTs) and pattern tables (PIPTs) with data extracted from data view and data attribute records. The utility writes the tables to a partitioned data set which can be read by the API application. For more information about the BLGUT18 utility, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
For more information, refer to the Tivoli Information Management for z/OS Application Program Interface Guide. Information about the Tivoli Information Management for z/OS clients is provided in the Tivoli Information Management for z/OS Client Installation and User's Guide.	

<table>
<tr><th colspan="2">Desktop Enhancements</th></tr>
<tr><td>Enhanced Desktop application</td><td>The Java-based Desktop is enhanced with many new features and improvements. The enhancements include changes such as the following:

Configuration editor improvements and support for user preferences
Better error handling and help facilities, including field and menu help as well as the ability to customize application help using the Toolkit
Ability to attach various types of files to a record (such as image, graph, audio, or video)
Improved security
Improved sample application help and ability to create customized application help
Support for parent/child records such as change/activity records
Enhanced record locking to automatically lock a record when it is retrieved for update. Records can also be manually locked and unlocked.
Hierarchy names can be different from the user's privilege class name. Authorization codes from the user's privilege class are used to restrict access to Desktop fields as needed.
Support for list processor data (display, insert, and delete)
Ability to view record history data
Enhanced search support includes the ability to:

Define canned searches and execute those searches from the Desktop
Display multiple columns of data in a search results list (SRL)
Drag/drop and view record details from an SRL
Print one or multiple record details from an SRL. The record details can be printed on any workstation-defined printer associated with a user's machine (including LAN or Ethernet printers).

For a complete list of the enhancements, refer to the Tivoli Information Management for z/OS Desktop User's Guide.</td></tr>
</table>

<table>
<tr><th colspan="2">Panel Modification Facility (PMF) Enhancements</th></tr>
<tr><td>Report enhancements</td><td>The p-word list report produced through the Panel Modification Facility now sorts data by prefix and data validation pattern, making it easier to see how data is defined in the dictionary.

Also, PMF reports now include a new header that shows the date, time, data set name (or label). The header makes it easier to identify what a report is for and how current the output is.</td></tr>
</table>

Panel Modification Facility (PMF) Enhancements	
Option available to process fields last	An option is added to PMF to allow panel designers to specify that one or more fields on a data entry panel should be processed last after all other fields on the panel are processed in top-to-bottom, left-to-right order. This change can help to improve the usability of applications which have entry fields that have an assisted-entry panel or data attribute record which flows to another panel. This processing order option is available on BLM8CU7B, the Data Collection Processing panel. The option applies to data typed directly in the entry field. It does not apply for immediate response chains (IRCs), terminal simulator panels (TSPs), or TSXs.
For more information, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.	

TSX and TSP Enhancements	
Enhanced TSX to view Desktop data model records	If you build data model records for use with the Tivoli Information Management for z/OS Desktop application, you may be interested in the enhanced BLGDVLAY terminal simulator EXEC (TSX). The EXEC provides useful print output of Desktop panel layout information. You can see, at a glance, the fields, groups, and tables that make up a panel layout in a data view record, and route the report output to the screen or to a pre-allocated data set of your choice. For information on how to use this TSX, see *Tivoli Information Management for z/OS Desktop User's Guide*.
New BLGTDARL TSX to list data attribute records	A new TSX, BLGTDARL, is provided to enable administrators to generate a list of the data attribute records in their database. The list includes the data attribute record names, description, s-words, s-word index, p-words, p-word index, and other information about the records, such as the panels that use the data attribute records. Using the output, you can, for example, find all instances where a certain attribute is used so you can identify the panels that need updating. The BLGTDARL TSX makes it easier to tailor or change your applications when necessary because it provides a current reference of the data attribute records contained in your database. For more information about using this TSX, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

TSX and TSP Enhancements	
New REXX variable and TSX to support character substitution for national languages	If you write TSXs and have a need for national language support, you can now parse the REXX variable BLGSYMB to get the proper values to use for characters which are not universal across all languages. With the BLGSYMB variable you can obtain the proper characters to include in your data for the not sign, or bar, exclamation point, and at sign. BLGSYMB eliminates the need to hardcode characters in your TSXs and helps to ensure that your TSXs are more transportable for processing across languages. The use of the BLGSYMB variable is recommended if you write TSXs and use the Graphic Character Substitution feature. Even if you are not using the Graphic Character Substitution feature, you can use BLGSYMB to avoid having to type in other characters to artificially achieve the proper mapping of graphic characters to code points. For more information about BLGSYMB, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*. Also provided is a new HLAPI extension TSX called BLGTSPCH. If you write applications that use the high-level API (HLAPI) and need to know what character to use as the not sign or the or bar, you can call this extension through the HL14 (Start User TSP or TSX) transaction. BLGTSPCH parses the not sign and the or bar from BLGSYMB and returns the appropriate characters in the output for use by your application on subsequent transactions. For more information about the BLGTSPCH HLAPI extension, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
TSX ADDSDATA control line allows null options	The TSX ADDSDATA control line is enhanced to allow null values to be specified in the option list. If null values are specified, they are ignored. This change makes it easier for TSX writers to code the ADDSDATA control line to handle various types of options. An option can be omitted simply by setting the variable for the option to null. For more information about the ADDSDATA control line, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.
New TSX user exit BLMXSPRM to return session parameter values	The new TSX user exit BLMXSPRM provides the ability to retrieve session parameter values. While the HELP STATUS screen retrieves many session parameter values, BLMXSPRM retrieves values not accessible through the HELP STATUS screen, such as non-panel VSAM data set names and search and sort limits and routine names. For more information about the TSX user exit BLMXSPRM, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

Utility Enhancements	
Support for unique RNIDs across logical database partitions	If your Tivoli Information Management for z/OS database is logically partitioned, you can now decide how system-assigned record number IDs (RNIDs) are incremented in the partitions. You can have unique RNIDs across partitions or have duplicate RNIDs. The BLGUT1 (Rebuild the SDIDS) and BLGUT9 (Set database options) utilities are enhanced with a new keyword, UNIQUE, which enables you to have unique RNIDs across logical partitions. In addition, a REUSE keyword is added to BLGUT1 to enable you to reuse VSAM sequence numbers.
For more information, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.	

Performance Enhancements	
Parallel sysplex offers improved performance	If you run Tivoli Information Management for z/OS in a parallel sysplex, you can take advantage of VSAM RLS. The use of RLS should significantly improve the performance of Tivoli Information Management for z/OS, especially if you are sharing databases (VSAM data sets). Because RLS enables a more precise lock on data in a data set (data is locked at the record level rather than as a data set enqueue), you can achieve a greater interleaving of user transactions with the database than if you used VSAM NSR/LSR. This locking granularity provides advantages in both shared and nonshared database environments. RLS also provides a common cache for the sysplex in the coupling facility, which provides a performance boost over the cross-system buffer invalidation used by Multisystem Database Access in previous releases of the product. For more information about planning for and installing Tivoli Information Management for z/OS in a sysplex, refer to "Setting Up for Sysplex Data Sharing" on page 151.

1 Overview for Planning and Installing

This chapter contains a step-by-step procedure for planning and installing Tivoli Information Management for z/OS. Tivoli Information Management for z/OS is installed in two phases. The first phase involves installing Tivoli Information Management for z/OS from tape using the System Modification Program/Extended (SMP/E) program. If you have not completed this step of the installation process, refer to the Tivoli Information Management for z/OS *Program Directory* for instructions on how to install the product from tape. For a list of the Tivoli Information Management for z/OS product data sets that are installed, see "Tivoli Information Management for z/OS Program Data Sets and Sample Members" on page 359.

The second phase of installation begins after the product data sets are loaded on your system. This phase of the installation involves planning and allocating the resources that Tivoli Information Management for z/OS uses and tailoring the environment that Tivoli Information Management for z/OS runs in. The steps involved in this phase of installation are given in Table 1 on page 2. Perform these steps in the order given in Table 1 on page 2 for the best results.

General Checklist

Table 1. Planning and Installation Steps

Step	Action	See
General Planning		
1	Select the features and options that you want for your Tivoli Information Management for z/OS installation.	“The Base Tivoli Information Management for z/OS Product” on page 19 “Tivoli Information Management for z/OS Features and Options” on page 27
2	Calculate the resources you need for installing and using Tivoli Information Management for z/OS. The resources you need to consider include: ■ I/O demand, processor usage, memory ■ VSAM resources (data sets, placeholders, connection, buffer pools) ■ Virtual storage for the user’s address space and BLX-SP(s) ■ Hardware and software for the base product and any optional features ■ Sysplex environment and a coupling facility ■ Personnel, training, IPL schedule, test system	“Evaluating Performance” on page 57 “Using VSAM Resources in a Non-Sysplex Environment” on page 75 “Determining Storage Requirements” on page 85 “Ordering the Necessary Hardware and Software” on page 89 “Setting Up for Sysplex Data Sharing” on page 151 “Assigning and Scheduling Personnel and Systems” on page 99
3	Tivoli Information Management for z/OS 7.1 has implemented a number of date and time-of-day enhancements. If you are currently using two or more external date formats, it is recommended that you run BLGUT17 before or during the installation process in order to standardize the date formats your organization uses. Information on running BLGUT17 can be found in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.	“Enabling Alternative Date and Time-of-Day Formats” on page 227
Migration Planning		
3	If you are migrating from a previous version of Tivoli Information Management for z/OS, review the material to understand what needs to be changed.	“Migrating from Previous Versions” on page 103
Installing		

Table 1. Planning and Installation Steps (continued)

Step	Action	See
4	Set up one or more BLX-SPs. If you want to share the Tivoli Information Management for z/OS VSAM data sets, set up for data sharing. To share data sets, you must do the following: ■ If you are setting up VSAM record-level sharing (RLS) for the first time, define RLS sharing control data sets. Run the BLXRLSCD sample provided in the SBLMSAMP library to do this. BLXRLSCD is a sample IEFBR14 job. ■ Set up the coupling facility structures that will enable you to implement data sharing. Run the BLXCFSTR sample to do this. BLXCFSTR is a sample IXCMIAPU job. ■ Define the coupling facility cache structures in your SMS base configuration using the ISMF panel interface.	"Setting Up Your BLX-SP" on page 143 "Setting Up for Sysplex Data Sharing" on page 151
5	If you are running Tivoli Information Management for z/OS from the link pack area (LPA), update the ISPF ISPTCM table to add an entry for BLGINIT1.	"The Base Tivoli Information Management for z/OS Product" on page 19

Table 1. Planning and Installation Steps (continued)

Step	Action	See
6	Consult “Evaluating Base Product Tailoring Considerations” on page 163 for base product considerations. Create and tailor the following items. (Inexperienced installers can use the Installation Tailoring Facility to define these items.) ■ Session-parameters members (described in “Defining Tivoli Information Management for z/OS Session-Parameters Members” on page 317) If you are migrating from a previous release, you *must* change your existing session-parameters members before reassembling them. New BLGPARMS macro keywords were added and others were changed in this release. For most installations, the new DATEFMT keyword is now required. Refer to the appendix on session-parameters members for information on new or changed keywords. ■ VSAM data sets (described in “Defining Tivoli Information Management for z/OS Data Sets” on page 277): SDDS, SDIDS, SDLDS, DICTDS, RPANLDS, WPANLDS. **Note about SDIDS architecture: If you are migrating from Information/Management Version 6.3 or earlier releases and you have an existing SDIDS, you MUST rebuild the SDIDS with the BLGUT1 utility, or migrate it with the BLGUT1M utility to ensure database compatibility. The internal architecture of the SDIDS is different from your existing SDIDS. Tivoli Information Management for z/OS format databases (database 4, 5, 6, 7, 8, and 9) can be rebuilt using the BLGUT1 utility or migrated using the BLGUT1M utility. User-defined format databases (database 0, 1, 2, and 3) can be rebuilt using the BLGOZUD utility or migrated using the BLGUT1M utility.** If you are migrating from TME® 10 Information/Management Version 1.1 or Tivoli Service Desk for OS/390 Version 1.2, the architecture of the SDIDS is not changed, so you need not rebuild or migrate the SDIDS. Note about Parallel Sysplex: To use Tivoli Information Management for z/OS in a sysplex and exploit sysplex services, you must migrate your existing VSAM data sets to use VSAM RLS. Also, if you are using a previous version of the product and are using Multisystem Database Access (you are sharing databases), you must use Tivoli Information Management for z/OS in a sysplex to continue sharing databases, and your existing VSAM data sets must be enabled for RLS. See “Setting Up for Sysplex Data Sharing” on page 151 for more information about setting up for sysplex. Create or modify the following if you did not do so in step 4. ■ BLX-SP parameters member ■ BLX-SP procedure ■ VSAM resource definition member (this is not required if you will use Tivoli Information Management for z/OS in a sysplex environment and exploit sysplex services) Inexperienced installers of Tivoli Information Management for z/OS can use the Installation Tailoring Facility to define these items.	“Evaluating Base Product Tailoring Considerations” on page 163 “Using the Installation Tailoring Facility” on page 185
7	Use Resource Access Control Facility (RACF®) to protect your Tivoli Information Management for z/OS data sets. Also use RACF and PMF to ensure the proper usage of sensitive Tivoli Information Management for z/OS Terminal Simulator Panel (TSP) or EXEC (TSX) control lines.	“Securing Tivoli Information Management for z/OS Information” on page 191

Table 1. Planning and Installation Steps (continued)

Step	Action	See
Starting the System		
8	If you are currently using a subsystem name which is different from the default subsystem name for the new release (BLX1), add the subsystem name. The MVS SETSSI system command can be used to add the subsystem name dynamically. The default subsystem name for Information/Management Version 5.1 was BLX0. Subsequent releases have since used a default subsystem name of BLX1. You can use any four characters as the subsystem name. See "Defining Subsystems for BLX-SPs" on page 144 for more information about subsystem names. If you are performing a new installation you must define a new subsystem name. If you are currently using BLX1 as your subsystem name, you will not need to add a subsystem name.	
9	Start the BLX-SP procedure. Use the MVS START command.	"Starting and Stopping the BLX-SP" on page 148
10	Load the read panel data set.	"Loading the Read Panel Data Set" on page 197
11	If you have customized data-entry, table, or assisted-entry panels for date or time fields, you should change them so that they have a 10-character wide entry and display fields and a validation pattern of IIV63 (or IIV9, at a minimum). This is necessary because users can now use different date formats and your panels need to support the longest possible format. If you have date fields that are 8 characters wide on data-entry or table panels, users will be unable to work with any of the 9- or 10-character date formats. If the validation patterns are not changed on your assisted-entry panels, users will be unable to use any of the formats that do not match the validation pattern in use. For example, if your validation pattern is NNNNNN, users can use only the MM/DD/YY, DD/MM/YY, or YY/MM/DD formats. If you do not change your customized panels, you should make a copy of data attribute record BLG&DFMT and remove all the formats you are unable to support. Then, change panel BLG0P700 to refer to the modified data attribute record. This will prevent users from choosing a format that the customized panels cannot support.	See page 135 for details.
12	Load the dictionary data set.	"Loading the Dictionary Data Set" on page 199
13	If you are upgrading from a release prior to TME 10 Information/Management Version 1.1 and are using an existing database, run the BLGUT1 or BLGUT1M utility to convert your SDIDS.	*Tivoli Information Management for z/OS Operation and Maintenance Reference*
14	Format the SDLDS.	"Formatting the SDLDS" on page 199
15	Select an ISPF panel style. Be sure the ISR@PRIM primary options panel is in the ISPPLIB concatenation.	"Selecting the Tivoli Information Management for z/OS Panel Style" on page 199
16	(Optional) Install the graphical user interface.	"Installing the Graphical User Interface" on page 200

Table 1. Planning and Installation Steps (continued)

Step	Action	See
17	Define your report format table data sets.	"Defining Report Format Table Data Sets" on page 200
18	(Optional) Set up a remote printer.	"Setting Up a Remote Printer" on page 201
19	Copy the BLXABMSG member from the SBLMSAMP library into a data set in the ISPPLIB concatenation.	"Sample Members" on page 360
20	Copy the BLGPVARS member from the SBLMSAMP library into a data set in the ISPPLIB concatenation.	*Tivoli Information Management for z/OS Program Administration Guide and Reference* (for customization if necessary)
21	(Optional) Perform this step only if you use the PDF editor. 1. Copy the BLGISMAC member from the SBLMSAMP library into a data set in the SYSPROC concatenation. 2. Copy the BLM@EDIT member from the SBLMSAMP library into a data set in the ISPPLIB concatenation.	*Tivoli Information Management for z/OS Program Administration Guide and Reference* (for customization if necessary)
22	Define a method for starting Tivoli Information Management for z/OS (e.g., interactive or batch mode).	"Starting Tivoli Information Management for z/OS" on page 203
	Allocate the BLGTSX DD in your startup procedure (to specify the SBLMTSX data set) regardless of whether or not you intend to use your own TSXs. (Tivoli Information Management for z/OS uses the TSXs in the SBLMTSX data set.)	Figure 17 on page 208
23	Perform the following procedure to verify that installation was successful. 1. Verify that the SBLMMOD1 load library is allocated to the ddname ISPLLIB or STEPLIB concatenation. 2. From the TSO READY prompt, issue ISPSTART PGM(BLGINIT) or ISPSTART PGM(BLGINIT) PARM(SESS(aa)). *aa* represents the suffix that was chosen for the session-parameters member. **Note:** If you are not using session-parameters member BLGSES00, then to avoid errors, you must specify the SESS(aa) parameter in the ISPSTART command as previously indicated. The first panel to appear is the proprietary product panel, BLG00002. If you are migrating from earlier versions of the product and you changed your profile to bypass the proprietary statement, you will not see the Version 7.1 proprietary product panel. You will go directly to the Tivoli Information Management for z/OS Primary Options Menu. 3. Press Enter to view the Tivoli Information Management for z/OS Primary Options Menu.	"Starting Tivoli Information Management for z/OS" on page 203
24	(Optional) Complete customization tasks to enable universal time processing, if you want to use that feature: ■ If you have not already done so, ensure that the TIMEZONE keyword is added to the session members through the BLGPARMS macro. See on page 332 for more information about this keyword. ■ Determine which date and time fields on your panels are related, and create a DATETIME record to define the relationships.	"Implementing Universal Time Processing" on page 251
Other Tasks		

Table 1. Planning and Installation Steps (continued)

Step	Action	See
25	Enable or disable the Notification Management facility for problem, change, and activity records. Tivoli Information Management for z/OS is shipped with the Notification Management facility partially disabled. **Notes:** 1. If you have a previous version of the product and you are currently using notification, you can continue to use your existing USERS record. 2. If you have applied all Tivoli-supplied maintenance to your modified notification TSPs, you can continue to use those modified TSPs for problem, change, and activity records. You must still enable or disable notification for activity records if you are migrating from a release earlier than Version 5.1. As part of enabling the facility, you also need to decide on whether or not to use TSPs for processing, or make use of the TSXs provided which provide support for TCP/IP SMTP, e-mail, and BLX-SP queueing. If you have not previously used the notification management feature of Tivoli Information Management for z/OS, consider using the message notification function provided with Tivoli Information Management for z/OS to send e-mail messages through an SMTP TCP/IP server on a host or distributed platform.	*Tivoli Information Management for z/OS Program Administration Guide and Reference*

Table 1. Planning and Installation Steps (continued)

Step	Action	See
26	(Optional) Install the Integration Facility interfaces: ■ Install the Integration Facility interface to the NetView® Hardware Monitor. See member BTNNPDA in the SBLMSAMP library. ■ Install the Integration Facility interface to OPC/A or OPC/ESA. See member BTNX6JOB in the SBLMSAMP library when using OPC user exit EQQUX006. See member BTNX7JOB in SBLMSAMP when using OPC user exit EQQUX007. ■ Install the Integration Facility interface to SAM. (The following information is provided as a convenience to those customers who may still be using the SAM function in earlier versions of the Resource Management Facility.) 1. See member BTNAMS in SBLMSAMP. 2. Modify panels BTN0BU00 and BTN0S010. When you installed SAM, the directions required you to tailor two Tivoli Information Management for z/OS problem summary panels by adding selections that take you to two AMS panels where you can access SAM data. If you are using the Integration Facility panels, you must make similar modifications to the BTN0BU00 and BTN0S010 panels. You can flow to the same AMS panels (AMS0S101 and AMS0S100, respectively). When you add these selections, choose a selection number different from those you already used for your panels. Refer to the *MVS/ESA™ Resource Measurement Facility (RMF) Program Directory* for details on how to modify these panels for your installation. 3. See members BLGTOAMS and BTNTOAMS in SBLMSAMP. **Note:** The Integration Facility interface to SLR is automatically installed when you install Tivoli Information Management for z/OS.	Sample members in SBLMSAMP library
27	(Optional) If you want to use one or more Tivoli Information Management for z/OS clients: ■ Decide which clients you want to use. ■ Decide how many clients you want to use on each server. ■ Set up the appropriate servers on z/OS ■ Install and configure each client.	*Tivoli Information Management for z/OS Client Installation and User's Guide* for information on planning, installing, and configuring clients and servers.
28	(Optional) If you want to use any of the Tivoli Information Management for z/OS Web connectors, install them.	*Tivoli Information Management for z/OS World Wide Web Interface Guide*
29	(Optional) If you want to use the Open Database Connectivity (ODBC) driver to generate reports from an ODBC-enabled workstation application, install the ODBC driver on the workstation. The ODBC driver requires installation of the HLAPI/NT. You must also define the data model records that will be used with the ODBC driver.	*Tivoli Information Management for z/OS Data Reporting User's Guide*

Table 1. Planning and Installation Steps (continued)

Step	Action	See
30	(Optional) If you intend to use the NetView AutoBridge feature of Tivoli Information Management for z/OS, verify that the FMIDs for NetView AutoBridge have been applied. The *Program Directory* has more information about installing this feature. For more information about installing the NetView Bridge Adapter and AutoBridge setup and administration, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.	*Program Directory for Tivoli Information Management for z/OS Version 7.1* *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*
31	Load the base records shipped with Tivoli Information Management for z/OS. **Note:** The base records *must* be loaded because certain base Tivoli Information Management for z/OS functions in this release are dependent on the existence of these records. Additionally, if you are using optional features or functions that require data model records or other types of records to be loaded, load the records required for those functions. For example, data model records are provided to support the use of Tivoli Information Management for z/OS with Tivoli Decision Support, Tivoli Inventory, Tivoli Problem Management, and the Tivoli Information Management for z/OS Desktop.	"Loading Records Provided with Tivoli Information Management for z/OS" on page 219
32	(Optional) If you are integrating with other Tivoli products or components, such as those listed below, install any other necessary components. ■ Problem Service ■ Tivoli Enterprise Console integration facility ■ Tivoli Inventory ■ Tivoli Problem Management (the Tivoli Service Desk Bridge) ■ Tivoli Software Distribution **Note:** If you are integrating with Tivoli Decision Support, you must have Tivoli Decision Support for Information Management (a separate complementary product). For an overview, see the *Tivoli Information Management for z/OS Data Reporting User's Guide*.	*Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*

Procedures for New Installations

If you need to install Tivoli Information Management for z/OS in a new environment (for example, in a test or staging environment) and are not interested in customizing or using many of its optional features right away, you can use the following procedures to get Tivoli Information Management for z/OS set up as quickly as possible. It is assumed that you have already used the *Program Directory for Tivoli Information Management for z/OS* (or the appropriate installation documentation provided with your offering) to install the basic program materials and optional features from program tapes. Therefore, these procedures do not address how to install the product from tape. They do contain information about the second phase of installation, which begins after the product data sets are loaded on your system.

Tivoli Information Management for z/OS contains many features and is highly customizable. The previous General Installation checklist reflects that fact and contains many more steps than are shown in the following procedures. Do not panic if the order of the steps shown here does not match the order of the General Installation checklist. The order of steps in these procedures is a recommended sequence for a new installation where migration considerations or immediate selection and use of optional features are not factors in your environment. Although the order is not critical, you must have the BLX-SP started before you can run any of the Tivoli Information Management for z/OS utility programs as mentioned in these procedures.

Also, because the use of sysplex support is optional if you are not sharing databases, the setup of Tivoli Information Management for z/OS in a sysplex environment is not addressed in this section. Refer to “Setting Up for Sysplex Data Sharing” on page 151 for more information about exploiting sysplex services.

Note: As an alternative, you can use the Installation Tailoring Facility provided with Tivoli Information Management for z/OS to perform many of these tasks. For example, you can define a BLX-SP procedure, VSAM data sets, a VSAM resource definition member, and a session-parameters member using the facility.

By performing these tasks, you can create the following:

- A single BLX-SP (central address space database server)
- A database to hold records (SDDS), a database index (SDIDS), and a log data set to hold database changes (SDLDS)
- Panel data sets (RPANLDS, WPANLDS)
- A dictionary data set (DICTDS) containing control information that is required to modify panels or process reports
- A session-parameters member that identifies the processing options for a Tivoli Information Management for z/OS session, such as which data sets, database, or BLX-SP to use

If you are not familiar with these components, you should review the material provided in this manual in “The Base Tivoli Information Management for z/OS Product” on page 19 for more information.

Additionally, these procedures show you how to create a master privilege class which can be used by an administrator to perform administrative duties such as defining user profiles and other privilege classes.

To install Tivoli Information Management for z/OS in a new environment, follow these procedures:

1. Set up the BLX-SP (the started task):

   a. Modify the Program Properties Table by adding an entry to make the BLX-SP program entry nonswappable. To do this, add the following to the appropriate SCHEDaa members of SYS1.PARMLIB:

   ```
   PPT PGMNAME(BLXSSP00)  /* BLX SERVICE PROVIDER                    */
            NOSWAP        /* PROGRAM IS NOT SWAPPABLE                */
            KEY(8)        /* PROGRAM IS A KEY(8) TASK                */
   ```

   To implement the changes to the table, you must IPL the system or change the SYS1.PARMLIB SCHEDAA member dynamically using the SET command.

   Reference: "Modifying the Program Properties Table" on page 143.

   b. Add the Tivoli Information Management for z/OS load library to the APF list. This requires changing the appropriate IEAAPFaa or PROGaa member. You may have one or more libraries that contain Tivoli Information Management for z/OS load modules. The SBLMMOD1 library is the one that should be APF-authorized.

   To implement changes to the PROGaa member, you can use the SET command. To implement changes to the IEAAPFaa member, you must IPL the system.

   Reference: "Adding a Tivoli Information Management for z/OS Load Library to the APF List" on page 144.

   c. Define the subsystem for the BLX-SP.

   During Tivoli Information Management for z/OS installation, you must identify each of your BLX-SPs as an MVS subsystem. You must give your BLX-SP a subsystem name. Define the subsystem to MVS and define each subsystem to its respective BLX-SP. The subsystem name must have 4 characters. The first character must be an alphabetic or national (#, $, @) character. The remaining characters can be alphabetic, numeric, or national characters.

   Reference: "Defining Subsystems for BLX-SPs" on page 144.

   1) Define a subsystem to MVS (for example: BLX1, which is the subsystem name for BLX1PROC.)

      Add this subsystem name to the appropriate IEFSSNaa member of SYS1.PARMLIB. For example:

      ```
      BLX1

      - or -

      SUBSYS SUBNAME(BLX1)
      ```

      To implement this change you will need to perform an IPL or use the SETSSI command.

      Reference: "Defining a Subsystem to MVS" on page 144.

   2) Define a subsystem to the BLX-SP.

All Tivoli Information Management for z/OS initialization code uses a load module named BLXSSINM that defines the subsystem to be used by the BLX-SP and all user sessions. A sample of this module is provided in the SBLMMOD1 library. This sample uses the subsystem name BLX1. The sample is already assembled and link-edited for you.

Reference: "Defining a Subsystem to the BLX-SP" on page 145.

If you decide to create a new BLXSSINM with a different subsystem name, you can use the following source code to rebuild BLXSSINM:

```
BLXSSINM  CSECT
          DC        CL4'BLX1'      Define the BLX subsystem name.
          END
```

The source code is also available in the SBLMSRC1 library (member BLXSSINM)

Then, reassemble the source with code similar to the following:

```
//MBACONT JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=R
//ASM       EXEC PGM=ASMA90,REGION=1024K,PARM='NODECK'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIB    DD DSN=BLM.SBLMMACS,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//SYSLIN    DD DSN=&LOADSET,UNIT=SYSDA,DISP=(MOD,PASS),
//             SPACE=(80,(200,50))
//SYSIN     DD DSN=BLM.SBLMSRC1(SOURCEcode),DISP=SHR
//LINK1     EXEC PGM=IEWL,PARM='LIST,XREF,LET,NORENT'
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DSN=BLM.SBLMMOD1,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DSN=&LOADSET,DISP=(OLD,DELETE)
//          DD *
NAME BLXSSINM(R)
/*
```

Notes:

a) The SYSLMOD DD card must specify the desired load library for the new BLXSSINM load module.

d. Define a procedure for each BLX-SP in SYS1.PROCLIB.

```
//BLX1PROC PROC PRM=00
//BLXSPCAS EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB DD DISP=SHR,DSN=BLM.SBLMMOD1 APF AUTHORIZED
//BLXPRM DD DISP=SHR,DSN=BLM.SBLMSAMP  BLX-SP PARMS (blx100)
```

If your BLX-SP subsystem is named BLX1, you can use the sample procedure BLX1PROC contained in your SBLMSAMP library. If your BLX-SP subsystem is named something else, use the Installation Tailoring Facility to create a procedure. Or, insert a DD statement for the load library containing your copy of BLXSSINM into the STEPLIB concatenation ahead of the DD statement for SBLMMOD1.

Reference: "Defining a BLX-SP Procedure" on page 146.

e. Define a BLX-SP parameters member. See "Defining BLX-SP Parameters Members" on page 343 for parameters if creating a new BLX-SP such as BLX200.

If you are using BLX1PROC you can use the BLX100 sample that is provided in the SBLMSAMP sample library. The BLXPRM DD card in the BLX1PROC sample points to SBLMSAMP.

2. Define the VSAM data sets

   a. Define data sets for the database (SDDS, SDIDS, SDLDS). A sample (BLGDATAB) is provided in the SBLMSAMP library to help you do this:

```
//BLGDATAB JOB
//DEFINE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DELETE (BLM.SDDS) CLUSTER PURGE
    DELETE (BLM.SDIDS) CLUSTER PURGE
    DELETE (BLM.SDLDS) CLUSTER PURGE
    DEFINE -
CLUSTER(NAME(BLM.SDDS)
      INDEXED  -
          KEYS(7 0)  -
      SHAREOPTIONS(1)  -
          LOG(NONE) -
      VOLUMES(vvvvvv) -
      UNIQUE) -
   DATA(NAME(BLM.SDDS.DATA) -
      CYLINDERS(2 1) -
      CISZ(4096) -
      RECORDSIZE(4000 4089) -
      SPEED) -
   INDEX(NAME(BLM.SDDS.INDEX) -
      NOIMBED -
      NOREPLICATE )
      DEFINE -
    CLUSTER(NAME(BLM.SDIDS)    -
       INDEXED  -
       KEYS(34 0)  -
       SHAREOPTIONS(1)  -
             LOG(NONE) -
       VOLUMES(vvvvvv) -
       UNIQUE) -
    DATA(NAME(BLM.SDIDS.DATA) -
       CYLINDERS(1 1) -
       CISZ(26624) -
       RECORDSIZE(54 26617) -
       FREESPACE(20 20) -
       SPEED) -
    INDEX(NAME(BLM.SDIDS.INDEX) -
       NOIMBED -
       NOREPLICATE )
       DEFINE -
    CLUSTER(NAME(BLM.SDLDS)           -
       NUMBERED -
       SHAREOPTIONS(1)  -
             LOG(NONE) -
       VOLUMES(vvvvvv) -
       UNIQUE) -
    DATA(NAME(BLM.SDLDS.DATA) -
       CYLINDERS(15 0) -
       RECORDSIZE(4089 4089) -
       CISZ(4096))
```

   Reference: "Defining Tivoli Information Management for z/OS Data Sets" on page 277

   b. Define the panel data sets.

      1) Write panel data set (WPANLDS) – Use the BLGWPNL sample JCL provided in the SBLMSAMP sample library to define and allocate an empty panel data set.

```
//BLGWPNL JOB
//DELDEF1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DELETE (BLM.WPANELS) CLUSTER PURGE
    DEFINE -
      CLUSTER(NAME(BLM.WPANELS) -
         KEYS(10 0) -
        SHAREOPTIONS(1) -
         LOG(NONE) -
      VOLUMES(vvvvvv) -
      INDEXED -
      UNIQUE) -
   DATA(NAME(BLM.WPANELS.DATA) -
      CYLINDERS(2 1) -
      CISZ(4096) -
      FREESPACE(0 0) -
      RECORDSIZE(900 4089) -
      SPEED) -
   INDEX(NAME(BLM.WPANELS.INDEX) -
      NOIMBED -
         NOREPLICATE)
/*
```

Reference: “Defining the WPANLDS” on page 297

2) For Tivoli-supplied panels (IBM® panel data set) – Use the BLGRPNL sample JCL provided in the SBLMSAMP sample library.

```
//BLGRPNL  JOB
//DELDEF1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DELETE (BLM.IBMPNLS) CLUSTER PURGE
    DEFINE -
    CLUSTER(NAME(BLM.IBMPNLS) -
         KEYS(10 0) -
     SHAREOPTIONS(1) -
         LOG(NONE) -
     VOLUMES(vvvvvv) -
     INDEXED -
     UNIQUE) -
      DATA(NAME(BLM.IBMPNLS.DATA) -
       CYLINDERS(12 1) -
     CISZ(4096) -
     FREESPACE(0 0) -
     RECORDSIZE(900 4089) -
     SPEED) -
      INDEX(NAME(BLM.IBMPNLS.INDEX) -
     NOIMBED -
     NOREPLICATE )
 /*
```

Reference: “Defining the RPANLDS” on page 293

3) For your modified panel data set. This data set will contain all your modified panels as you customize them.

```
     DELETE (BLM.MPANELS) CLUSTER PURGE
     DEFINE -
     CLUSTER(NAME(BLM.MPANELS) -
    KEYS(10 0) -
          SHAREOPTIONS(1) -
          LOG(NONE) -
          VOLUMES(vvvvvv) -
          INDEXED -
          UNIQUE) -
```

```
            DATA(NAME(BLM.MPANELS.DATA) -
               CYLINDERS(1 1) -
               CISZ(4096) -
               FREESPACE(0 0) -
               RECORDSIZE(900 4089) -
               SPEED) -
            INDEX(NAME(BLM.MPANELS.INDEX) -
               NOIMBED -
               NOREPLICATE )
   /*
```

c. Define the dictionary data set (DICTDS). Use the BLGDICT sample provided in the SBLMSAMP sample library.

```
     DELETE (BLM.DICT) CLUSTER PURGE
     DEFINE -
    CLUSTER(NAME(BLM.DICT) -
      KEYS(3 0)  -
          SHAREOPTIONS(1)  -
          LOG(NONE) -
     VOLUMES(vvvvvv) -
     UNIQUE) -
   DATA(NAME(BLM.DICT.DATA) -
     CYLINDERS(3 1) -
     CISZ(4096) -
     RECORDSIZE(115 115) -
     FREESPACE(0 0) -
     SPEED) -
   INDEX(NAME(BLM.DICT.INDEX) -
     NOIMBED -
     NOREPLICATE )
/*
```

Reference: "Defining the DICTDS" on page 289

3. Define a VSAM resource definition member (BLXVDEF) – required only if sysplex support will *not* be used.

   - You can modify and compile the BLXVDEF sample member in the SBLMSAMP sample library. The VSAM resource definition member is a load module that defines the following:
     - The VSAM local shared resource (LSR) pools used by the VSAM data sets.
     - The number of nonshared resource (NSR) placeholders needed by the VSAM data sets.
     - The connection of the VSAM data sets to VSAM resources.
     - The type of VSAM data set (key-sequenced or relative record).
   - Define separate resource pools for the following Tivoli Information Management for z/OS VSAM datasets: RPANLDS, SDDS, SDIDS, SDLDS (if used).

     Use LISTCAT output from these data sets to help you enter the correct key lengths. Also, the buffer size must be equal to or greater than the CISIZE of the data set you are defining.

     Note: If you do not properly build the BLXVDEF, you may encounter slow responsiveness and various abends.

Reference: "Defining a VSAM Resource Definition Member" on page 301

4. Select a panel style. Copy BLGISPFD from the SBLMSAMP library to a new BLGISPFM member that is concatenated to the user's ISPPLIB ddname. (Copy BLGISPFE instead of BLGISPFD if you want users to default to the enhanced windows style.)

   If you are using the enhanced windows style, copy BLG0CMDS and BLG0KEYS from the SBLMSAMP library to a data set concatenated to the user's ISPTLIB data set.

   Reference: "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199

5. Copy the BLXABMSG member from the SBLMSAMP library to a data set in the ISPPLIB concatenation.

6. Protect the data sets using z/OS RACF or an equivalent product. Ensure that the BLX-SP procedure (BLX1PROC) has the correct RACF authority.

   Reference: "Securing Tivoli Information Management for z/OS Information" on page 191

7. Start the BLX-SP (BLX1PROC) using the MVS START command.

   ```
   S BLX1PROC
   ```

   Reference: "Starting and Stopping the BLX-SP" on page 148

8. After the task is running, you can run the utility programs.

   - Use the BLGUT6J sample to run the BLGUT6 utility to load the panels from the Tivoli-supplied partitioned data set to the VSAM data set you defined in step 2b2 on page 14. The input is the SBLMPNLS data set.

     ```
     //LOAD     EXEC PGM=BLGUT6,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
     //SYSIN   DD*
        NODBCS AETYPE(M,BYP)
     /*
     ```

   - Use the BLGUT5J sample to run the BLGUT5 utility to load the dictionary data set you defined in step 2c on page 15. The input is the SBLMDICT(BLMVDICT) data set.

     ```
     //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
     ```

   Reference: Tivoli Information Management for z/OS Operation and Maintenance Reference

9. Create a session-parameters member that points to your panel data sets, dictionary, SDDS, SDIDS, and SDLDS names. Then, assemble the session-parameters member. There are two samples that can be modified for use to accomplish these tasks.

   You can use the BLGSES00 sample (in this example, the modified panel data set is also included):

   ```
   ***********************************************************************
   *   SAMPLE SESSION-PARAMETERS MEMBER WITH ONE READ PANEL DATA SET,
   *   ONE WRITE PANEL DATA SET, AND A SINGLE-CLUSTER SDDS.
   ************************************************************************
            TITLE 'BLGSES00 - SESSION PARAMETERS'
   BLGSES00 CSECT
   ```

```
         BLGPARMS  DICTDS=DICTDS,       NAME THE DICTIONARY            X
               RFTDS=RFTS,              NAME THE REPORT FORMAT TABLES  X
               DATEFMT=MM/DD/YYYY       EXT DATE FORMAT                X
               RPANLDS=(MPANELS,RPANEL1) NAME THE READ PANEL DATA SET  X
               WPANLDS=WPANELS          NAME THE WRITE PANEL DATA SET
********       CAS=BLX1,                SPECIFY TARGET BLX-SP SERVER
* UNCOMMENT THE ABOVE LINE AND ADD IT TO THE BLGPARMS MACRO INVOCATION
* IF YOU WANT TO USE THIS MEMBER WITH A BLX-SP SERVER OTHER THAN THE
* DEFAULT BLX-SP SERVER
*
MGMT     BLGCLUST NAME=5,               READ/WRITE CLUSTER             X
               SDDS=MGTSDDS,            NAME THE SDDS                  X
               SDIDS=MGTSDIDS,          NAME THE SDIDS                 X
               SDLDS=MGTSDLDS           NAME THE SDLDS
MGTSDDS  BLGCLDSN DSN=BLM.SDDS
MGTSDIDS BLGCLDSN DSN=BLM.SDIDS
MGTSDLDS BLGCLDSN DSN=BLM.SDLDS
DICTDS   BLGCLDSN DSN=BLM.DICT
RFTS     BLGCLDSN DSN=BLM.SBLMFMT,FILE=RFTDD
RPANEL1  BLGCLDSN DSN=BLM.IBMPNLS,RDONLY=YES
WPANELS  BLGCLDSN DSN=BLM.WPANELS
MPANELS  BLGCLDSN DSN=BLM.MPANELS
         BLGGEN
         END
```

Reference: "Sample Session-Parameters CSECT" on page 340

The BLGALSPM sample is as shown:

```
//BLGALSPM JOB
//*********************************************************************
//*
//* SAMPLE JCL TO ASSEMBLE AND LINK-EDIT SESSION MODULE USING SESSION
//* MODEL BLGSES00.  IN THIS EXAMPLE, THE SOURCE IS LOCATED IN
//* BLM.SBLMSAMP(BLGSES00), THE INFORMATION MANAGEMENT for z/OS MACROS
//* ARE LOCATED IN BLM.SBLMMACS AND THE MODULE IS PLACED IN
//* BLM.SBLMMOD1
//*
//*********************************************************************
//ASM       EXEC PGM=ASMA90,REGION=1024K,PARM='NODECK'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIB    DD DISP=SHR,DSN=BLM.SBLMMACS
//          DD DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN    DD DISP=(MOD,PASS),DSN=&LOADSET,UNIT=SYSDA,
//             SPACE=(80,(200,50))
//SYSIN     DD DISP=SHR,DSN=BLM.SBLMSAMP(BLGSES00)
//LINK1     EXEC PGM=IEWL,PARM='LIST,XREF,LET,NORENT'
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DISP=(OLD,DELETE),DSN=&LOADSET
//          DD *
            ENTRY BLGSES00
            NAME BLGSES00(R)
/*
```

Reference: "Sample JCL for a Session-Parameters Member" on page 340

10. Use the BLGUTRJ sample to run the BLGUTR utility to format a recovery log data set (SDLDS). You will not need to run this again. It is used only once to initialize the log data set to make it ready for use.

```
//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
```

Reference: Tivoli Information Management for z/OS Operation and Maintenance Reference

11. Define a startup CLIST or logon procedure that invokes the session for users.

    Reference: “Sample CLISTs to Start Tivoli Information Management for z/OS” on page 207

12. Log on to Tivoli Information Management for z/OS and enter a master privilege class record. To do this, follow these steps:

    a. Log on to your session through the CLIST.

    b. On the BLG0EN20 panel, select option **3. Application**.

    c. On the BLG00030 panel, select option **1. System**.

    d. On the BLG00030 panel, select option **5. Entry**.

    e. On the BLG00010 panel, select option **1. Class**.

    You should see the following panel displayed:

```
BLG0J100                CLASS DESCRIPTION ENTRY              CLASS: _______

Enter privilege class data; cursor placement or input line entry allowed.


1. Privilege class name..<R> _______

3. Transfer-to class........ _______
4. Contact name............. _____________
5. Contact phone............ ___________
6. Contact department....... _________
7. Location code............ _______
8. Description...........<R> ________________________________________
9. Primary partition id..... ________


    When you finish, type END to save or CANCEL to discard any changes.
```

    On this panel:

    1) Tab to field 1 and type **MASTER**.

    2) Type your user ID in field 2.

    3) Type a description in field 8.

    4) Press PF3 until the record is filed.

    5) Quit out of the session by typing Q in the command line.

    Go back into the session using your user ID. It should automatically bring you up in the MASTER privilege class. You will need to create new classes later and add users to these classes. The MASTER privilege class is for administrators only and has all authority available to the session. You will mostly likely want to create classes that have more limited authority for users. For more information about privilege classes, see the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

2

The Base Tivoli Information Management for z/OS Product

2. The Base Product

This chapter provides an overview of the following:

- The base product components
- The BLX-Service Provider (BLX-SP) and BLX-SP parameters member
- The data sets
  - The databases
  - The panels
  - The dictionary
  - The report format table data set
- The VSAM resource definition member
- The session-parameters member

This chapter is designed to give you a better understanding of the components of Tivoli Information Management for z/OS that must be defined and tailored during installation of the base product.

The Base Product Components

Tivoli Information Management for z/OS consists of various components:

- BLG
- BLM
- BLX
- BLH
- BTN
- EYL
- EYM

The BLG and BLM components provide all the functional capability of Tivoli Information Management for z/OS. They are the parts of the product that logically connect the VSAM data sets (structured description data set, structured description index data set, and structured description log data set) into an entity called the Tivoli Information Management for z/OS database. The BLG and BLM portions of the product deal with the meaning and content of the data in each separate data set and how they interconnect with each other. They also contain functional capabilities of searching the database, creating or deleting records, and all the other functions that make up Tivoli Information Management for z/OS.

The BLX component has nothing to do with the system functional capabilities. It provides supervisor and data management services to the other components. It provides functions such as virtual storage allocation and deallocation, timing services, enqueue and dequeue services, data set allocation and deallocation, data set open and close functions, and record retrieval. It does not manipulate the content of a physical record. The BLX component simply provides the capability to get the requested function done.

The BLG and BLM components make all requests for VSAM input/output (I/O) through the BLX component. These requests are processed through the BLX-Service Provider (or in a sysplex, through the SMSVSAM server and a coupling facility).

The BLH component was formerly associated with a product called Information/Access. Now, it supports the integration of Tivoli Information Management for z/OS with other products including Tivoli Inventory, OS/390 Text Search, and the Tivoli Information Management for z/OS Desktop.

The BTN component provides the functional capability for the Integration Facility, which is an application provided with Tivoli Information Management for z/OS that models a typical problem and change management system to help you initiate your system management tasks.

The EYL and EYM components provide the function for the NetView AutoBridge and NetView AutoBridge Postprocessor. The NetView AutoBridge is a NetView application interface to the NetView Bridge Adapter and is used to automate network monitoring. The NetView AutoBridge receives data from specific NetView alerts, messages, and other applications and uses the data to build and perform Tivoli Information Management for z/OS transactions.

The BLX Service Provider

Every user of Tivoli Information Management for z/OS has a self-contained user address space. A *user* can be an interactive TSO user, a batch job or utility program, or a user interfacing with Tivoli Information Management for z/OS through NetView or an API.

Before Information/Management Version 5.1, each of these self-contained address spaces did all of its own VSAM I/O operations (for the database data sets, panels, and so on) through its own VSAM buffers. Currently, in a non-sysplex environment, Tivoli Information Management for z/OS uses a central address space server to perform the actual VSAM I/O operations for all users' address spaces. This central address space server, known as the BLX-Service Provider (BLX-SP), controls the actual physical resources needed to perform VSAM I/O functions.

A user's address space is connected to the BLX-SP during the initialization of Tivoli Information Management for z/OS in the user's address space. You can connect many users' address spaces to the same BLX-SP. The BLX-SP receives requests from the users' address spaces, performs the appropriate functions, and manages resources it obtains for each user.

In a sysplex, the BLX-SP is not used to control resources needed for VSAM I/O functions. Users open and access VSAM data sets directly instead of sending requests through the BLX-SP. VSAM processing is done through use of an SMSVSAM server which handles communications with a coupling facility to maintain VSAM buffers. The role of the BLX-SP is therefore limited to performing other functions, such as handling operator commands or sending notifications to users.

Figure 1 on page 21 shows the approaches used before and since Information/Management Version 5.1.

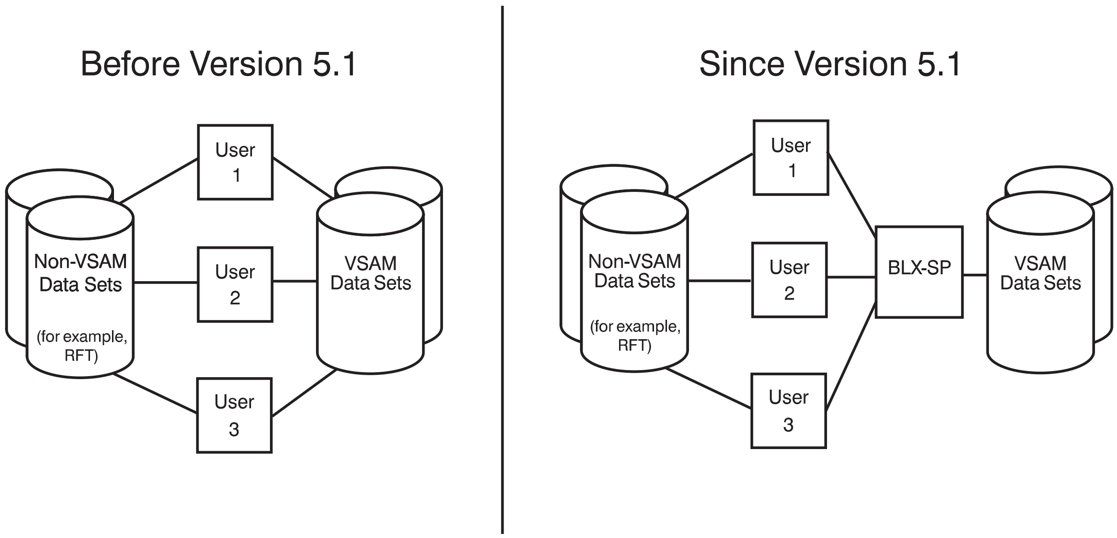

Figure 1. The BLX-SP Approach. The illustration shows how the BLX-SP is used to perform VSAM I/O processing since Information/Management Version 5.1 in a non-sysplex environment.

With Tivoli Information Management for z/OS, you can have multiple BLX-SPs on a single z/OS system (for more information on using multiple BLX-SPs, see “Multiple BLX Service Providers” on page 39).

BLX-SP operator commands enable you to change a BLX-SP’s operating characteristics. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for complete information about these commands. The operator commands work with the MVS MODIFY command to perform functions. With these commands, you can include or remove data sets, query statistical information, and more.

BLX-SP Parameters Member

The operation of the BLX-SP is controlled by a BLX-SP parameters member. The BLX-SP parameters member is a standard text file that contains the operating parameters you want your BLX-SP to run with. You must define at least one BLX-SP parameters member for your BLX-SP. If your system supports multiple BLX-SPs, then you must define at least one BLX-SP parameters member for each BLX-SP.

For an explanation of all the parameters that are specified in a BLX-SP parameters member, you can view the online introduction that the Installation Tailoring Facility provides (see “Using Online Help Information” on page 187), or read about the parameters, in “Defining BLX-SP Parameters Members” on page 343.

Special Notes about the BLX Service Provider

Users *must* be connected to the BLX-SP before they can use the Tivoli Information Management for z/OS database.

Modifying the ISPTCM Table

LIBDEF support enables you to dynamically change which load libraries are allocated to ISPLLIB. If you want to run Tivoli Information Management for z/OS from the link pack area (LPA), you must modify the ISPF TSO command table (ISPTCM). The ISPF sample library contains an example of this table. Its member name is ISPTCMA.

Add the following entry to the table used by your installation:

```
ISPMTCM FLAG=42,ENTNAME=BLGINIT1 INFO COMMAND PROCESSOR
```

Once you update the table, you must assemble and link-edit the ISPTCMA member.

If you use the Program Control Facility II (PCF II), 5798-CLW, you must also add BLGINIT1 to the PCF II command table.

For more information about changing the ISPF command table, refer to *ISPF Planning and Customizing*.

The Data Sets

Tivoli Information Management for z/OS uses the following types of data sets for its operations:

- The structured description data set (SDDS)
- The structured description index data set (SDIDS)
- The structured description log data set (SDLDS)
- The dictionary data set (DICTDS)
- The read panel data set (RPANLDS)
- The write panel data set (WPANLDS)
- The report format table data set (RFTDS)
- The TSX data set (SBLMTSX)

The SDDS, SDIDS, and SDLDS are VSAM data sets that make up the database that Tivoli Information Management for z/OS uses. The DICTDS is a VSAM data set that serves as a dictionary for Tivoli Information Management for z/OS, and the RPANLDS and WPANLDS are VSAM data sets that contain the Tivoli Information Management for z/OS panels.

The RFTDS is a partitioned data set that holds report format tables, application program interface (API) tables, and relational data mapping tables (RDMTs) used by Tivoli Information Management for z/OS. The SBLMTSX data set is a partitioned data set containing terminal simulator EXECs (TSXs) written in REXX. Some functions provided in Tivoli Information Management for z/OS use TSXs.

You can share VSAM data sets among BLX-SPs. For more information on sharing data sets, see “Data Sharing” on page 40.

For detailed information on the VSAM data sets that are used by Tivoli Information Management for z/OS, you can view the online introduction provided with the Installation Tailoring Facility (see “Using Online Help Information” on page 187). To read about the parameters that are used in defining the data sets, see “Defining Tivoli Information Management for z/OS Data Sets” on page 277.

The Databases

Tivoli Information Management for z/OS uses two types of databases: Tivoli Information Management for z/OS format databases and user-defined format databases (known as the Information/MVS format in previous versions). Databases 0, 1, 2, and 3 are user-defined format databases; databases 4, 5, 6, 7, 8, and 9 are Tivoli Information Management for z/OS format databases. Database 5 is a read/write database. The other databases are read-only databases. For information about the different formats, see “Database Formats” on page 45. The databases contain the following data sets:

SDDS The SDDS is made up of VSAM key-sequenced data sets (clusters) that contain the data records. For a Tivoli Information Management for z/OS format database, the SDDS contains the information supplied by the user (either interactively through panels or using API programs) or collected by Tivoli Information Management for z/OS. Each Tivoli Information Management for z/OS logical record can be one of several types and can consist of one or more VSAM records. For a user-defined format database, the SDDS contains the text entries that you load from a sequential file or from a tape you create in the Information/MVS format.

Tivoli Information Management for z/OS requires that you define an SDDS. The SDDS of a Tivoli Information Management for z/OS format database can consist of from 1 to 100 VSAM clusters. A 100-cluster SDDS enables users to store up to 400GB (GB equals 1 073 741 824 bytes) of information in a database. For information about using a multiple-cluster SDDS, see “Multiple-Cluster SDDS” on page 41.

The SDDS of a user-defined format database can consist of only 1 VSAM cluster.

SDIDS

The SDIDS is a special high-performance index that accelerates the search for records in the SDDS. It contains the words that you use as search keywords for records in the SDDS. The SDIDS is sometimes called the glossary. Each record in the SDIDS contains a searchable word and position markers to indicate which records in the SDDS contain that word. This is true for both Tivoli Information Management for z/OS and user-defined format databases.

The SDIDS is made up of from 1 to 100 VSAM key-sequenced data sets (clusters). You define one or more clusters, depending on your needs. The number of SDIDS clusters and SDDS clusters do not have to be the same, since they are independent of one another. For information on using a multiple-cluster SDIDS, see “Multiple-Cluster SDIDS” on page 41.

The information in the SDIDS is created and managed by Tivoli Information Management for z/OS and is in addition to the index component that VSAM defines for any key-sequenced data set.

SDLDS

The SDLDS is an optional data set that stores copies of the records written to the SDDS. The SDLDS provides a backup for changes made to the SDDS. If the SDDS is damaged, you can use the records in the SDLDS to reconstruct the changes made to the SDDS since its last backup.

The SDLDS is a VSAM relative record data set that consists of a single VSAM cluster and contains fixed-length records. Only the Tivoli Information Management for z/OS read/write database (database 5) has an SDLDS.

For a more detailed discussion of the Tivoli Information Management for z/OS databases and your database options, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

The Panels

Tivoli Information Management for z/OS provides a set of panels that you can use for navigation and data-entry purposes. Tivoli Information Management for z/OS accesses the panels it uses from VSAM key-sequenced data sets. The two types of panel data sets are:

RPANLDS

The RPANLDS is a VSAM key-sequenced data set that contains the Tivoli

Information Management for z/OS panels that are displayed to the user. You must define at least one RPANLDS to hold the Tivoli Information Management for z/OS panels. Use the BLGUT6 utility to load the panels from the Tivoli Information Management for z/OS product data setinto the RPANLDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information about the BLGUT6 utility.

To read about using multiple read panel data sets, see "Using One or More Read Panel Data Sets" on page 198.If you have more than one RPANLDS, then Tivoli Information Management for z/OS logically concatenates multiple RPANLDSs in the order specified in the session-parameters member (see "Multiple Read Panel Data Sets" on page 43).

WPANLDS

The WPANLDS is an optional data set that contains the Tivoli Information Management for z/OS panels that you create or modify using the Tivoli Information Management for z/OS Panel Modification Facility (PMF). Tivoli Information Management for z/OS reads and displays panels from the WPANLDS during PMF panel modification and panel testing (see "The Panel Modification Facility (PMF)" on page 55). The WPANLDS is a VSAM key-sequenced data set that consists of a single VSAM cluster.

The Dictionary

The Tivoli Information Management for z/OS dictionary contains the following control information:

- Structured words (s-words) and prefix words (p-words) that are used to search the SDIDS and to control the display of data on Tivoli Information Management for z/OS panels for all databases
- Validation patterns that are used to control the syntax of values entered into Tivoli Information Management for z/OS panels.

The dictionary data set (DICTDS) is a VSAM key-sequenced data set that consists of a single VSAM cluster.Use the BLGUT5 utility to load the dictionary from the Tivoli Information Management for z/OS product data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information about the BLGUT5 utility program.You must define a DICTDS for Tivoli Information Management for z/OS if you want to use PMF or if you want to process reports.

The Report Format Table Data Set

The RFTDS is a partitioned data set (PDS) that contains:

- The report format tables (RFTs) used to define the content and format of the print record and report output listings for the Tivoli Information Management for z/OS and user-defined format databases.
- Data tables used by the Tivoli Information Management for z/OS APIs. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for details about the API tables.
- Relational data mapping tables (RDMTs) used by the DB2 Extract Facility. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for details about the DB2 Extract Facility.

For more information about the report format data sets, see "Defining Report Format Table Data Sets" on page 200.

The VSAM Resource Definition Member – Non-Sysplex

Note

The following information applies to a non-sysplex environment. The VSAM resource definition member is not used in a sysplex to define VSAM resources. In a sysplex, resources are defined through coupling facility structures to support record-level sharing, rather than through VSAM resource definitions to support nonshared resources or local shared resources. For more information about how VSAM is used in a sysplex, see "Setting Up for Sysplex Data Sharing" on page 151.

In a non-sysplex BLX-SP environment, the VSAM resources that are used by the Tivoli Information Management for z/OS VSAM data sets are allocated in the BLX-SP and are used by all Tivoli Information Management for z/OS users' address spaces. The VSAM resources are defined to the BLX-SP by the VSAM resource definition member.

A VSAM resource definition member is a load module that defines the following:
- The VSAM local shared resource (LSR) pools used by the VSAM data sets
- The number of nonshared resource (NSR) placeholders needed by the VSAM data sets
- The connection of the VSAM data sets to VSAM resources
- The type of VSAM data set (key-sequenced or relative record)

For more information about defining VSAM resources, see "Using VSAM Resources in a Non-Sysplex Environment" on page 75. For information about defining a VSAM resource definition member, you can view the online introduction that is provided with the Installation Tailoring Facility (see "Using Online Help Information" on page 187). To read about the macros that are used to create a VSAM resource definition member, see "Defining VSAM Resources to the BLX-SP in a Non-Sysplex Environment" on page 301.

The Session-Parameters Member

A session-parameters member is a load module that specifies the parameters that define various processing options for a Tivoli Information Management for z/OS session. It also specifies the data sets that the session accesses for its databases, panels, and dictionary.

You can define as many session-parameters members as you need. Users who have common session needs can share the same members. Or each user can have a unique member. Or users can have several members that they are authorized to use. Each user specifies a single session-parameters member when starting a Tivoli Information Management for z/OS session.

For an explanation of all the parameters that are specified in a session-parameters member, you can view the online introduction that is provided with the Installation Tailoring Facility (see "Using Online Help Information" on page 187). To read about these parameters and the Tivoli Information Management for z/OS macros that are used to create a session-parameters member, see "Defining Tivoli Information Management for z/OS Session-Parameters Members" on page 317. For session-parameters member installation considerations, see "Working with Session-Parameters Members" on page 164. For performance considerations, see “Evaluating Performance” on page 57. Some session parameters can have a significant effect on the performance of a user session and on the performance of Tivoli Information Management for z/OS as a whole.

3

Tivoli Information Management for z/OS Features and Options

This chapter assumes that you understand the base Tivoli Information Management for z/OS product, and it provides an overview of the following features and options. Unless otherwise noted, these features and options are available with Tivoli Information Management for z/OS and do not need to be ordered separately.

This chapter is designed to help you determine which features and options of Tivoli Information Management for z/OS you want to include in your installation of Tivoli Information Management for z/OS.

Installation

- Installation Tailoring Facility

Interfaces

- Integration with Tivoli management software
- TEC Event Adapter
- OS/390 Text Search
- Remote Environment Servers and clients
- Web connector features
- NetView Bridge Adapter
- NetView AutoBridge and the AutoBridge PostProcessor facility
- NetView Hardware Monitor Interface
- Open Database Connectivity (ODBC) driver

Applications

- Desktop
- Integration Facility

System Performance

- Multiple BLX Service Providers
- Sysplex support
- Multiple-cluster SDDS
- Multiple-cluster SDIDS

- 18- or 34-byte SDIDS keys
- Multiple read panel data sets

Data access

- RACF protection
- Logical database partitioning

Data backup, propagation, or archival

- Automatic Log Save Facility
- DB2® Extract Facility
- Archiver

Preferences

- Database formats
- National Language Support (NLS)
- Alternate date and time-of-day formats
- Enhanced panel style
- Graphical user interface
- Notification Management facility

Customization

- Panel Modification Facility (PMF)

The Installation Tailoring Facility

The Installation Tailoring Facility is designed to help an inexperienced installer of Tivoli Information Management for z/OS tailor Tivoli Information Management for z/OS. The Installation Tailoring Facility takes you through a series of interactive ISPF dialogs that step you through the installation tailoring process.

Note: The Installation Tailoring Facility has not been updated to support the new setup tasks for Sysplex Data Sharing. Information about these setup tasks can be found in “Setting Up for Sysplex Data Sharing” on page 151.

Using the Installation Tailoring Facility, you can set system defaults and create or modify customized JCL and text members to:

- Tailor the base Tivoli Information Management for z/OS product. This includes working with:
  - VSAM data sets (SDDS, SDIDS, SDLDS, DICTDS, RPANLDS, WPANLDS)
  - Session-parameters members
  - BLX-SP parameters members
  - BLX-SP procedures
  - BLX-SP VSAM resource definition members (for use in a non-sysplex environment only)

- Tailor optional Tivoli Information Management for z/OS product features. This includes working with:
  - MRES parameters members
  - MRES procedures
  - National language support

For information on using the Installation Tailoring Facility, see “Using the Installation Tailoring Facility” on page 185.

Note: The online introduction to the Installation Tailoring Facility can be useful in providing the experienced installer with detailed information about new or changed Tivoli Information Management for z/OS parameters. See “Using Online Help Information” on page 187 for more information about accessing the online introduction.

Integration with Tivoli Management Software

Tivoli Information Management for z/OS is a Tivoli management software product in the Tivoli Enterprise™ application segment. Tivoli products provide highly scalable, open and cross-platform solutions for enterprise systems management environments that extend from the data center to the desktop. The object-oriented Tivoli Management Framework provides a foundation for integrating these disciplines into an overall network computing solution and integrating Tivoli applications with third-party solutions. Tivoli Information Management for z/OS provides help desk, problem management, change management, and configuration management services to z/OS customers using Tivoli management software products.

Tivoli Information Management for z/OS integrates with Tivoli products in these key areas:

- Tivoli NetView for z/OS

  Tivoli Information Management for z/OS has a built-in interface to NetView for z/OS—the NetView Bridge Adapter. The NetView Bridge Adapter connects the “bridge” in NetView for z/OS to the Tivoli Information Management for z/OS high-level application program interface. This interface enables NetView alerts and messages to be collected so that network problems can be logged and updated in the Tivoli Information Management for z/OS database.

  For centralized network management within a distributed environment, you can also connect Tivoli Information Management for z/OS with NetView running on an AIX® platform through the NetView for z/OS program. For a description of the Tivoli Information Management for z/OS interfaces to NetView, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

  In addition to communicating with NetView for z/OS through the NetView Bridge Adapter, Tivoli Information Management for z/OS can receive NetView alert and message processing through a program-to-program interface provided through use of the Tivoli Global Enterprise Manager. Commands can be sent to NetView for z/OS by clicking on a Tivoli Information Management for z/OS icon displayed in a window in the Application Policy Manager program of Tivoli Global Enterprise Manager.

- Tivoli Operations Planning and Control

  Tivoli Operations Planning and Control (OPC) plans, controls, and automates z/OS batch production workloads. The Event Manager subsystem of OPC tracks and logs events. A customization tool is provided by OPC to identify the types of problems that can be automatically entered into the Tivoli Information Management for z/OS database through the Tivoli Information Management for z/OS Integration Facility. Problems that are detected by OPC that match the identification criteria specified in the OPC database are forwarded to Tivoli Information Management for z/OS for recording and storage. For information on using the OPC user exits (EQQUX006, EQQUX007) that enable this integration, refer to *OPC/ESA Customization and Tuning*.

- Tivoli Decision Support

  Tivoli Decision Support is a workstation-based solution that helps managers, executives, and analysts gain insight into their enterprise's data to facilitate their decision-making process. Analysis tools and business intelligence models are provided to enable you to quickly find and use the data stored in your enterprise's database. Tivoli Decision Support for Information Management (a separately orderable product) enables you to use Tivoli Decision Support with Tivoli Information Management for z/OS data. For an overview of the Tivoli Decision Support for Information Management, refer to the *Tivoli Information Management for z/OS Data Reporting User's Guide*. Installation and usage instructions are provided in the softcopy documentation provided on the Tivoli Decision Support for Information Management CD-ROM.

- Tivoli Business Systems Manager for OS/390

  TBSM is a business systems management tool which enables you to perform distributed management, OS/390 management, or both. Even when a business system spans multiple platforms, TBSM enables you to graphically monitor and control interconnected business components and operating system resources. A business component and its resources are referred to as a Line of Business (LOB). Using the LOB concept, TBSM helps you plan, define, and control your business system. TBSM, together with other Tivoli management components, helps you manage the dependencies between business components and their underlying infrastructure.

- Tivoli Enterprise Console

  The Tivoli Enterprise Console acts as a central resource that receives information from many sources, such as systems, databases, and other applications. It integrates with major network management platforms and collects, processes, and automatically initiates corrective actions to system, application, network and database events.

  Tivoli Enterprise Console events can automatically open problem records in the Tivoli Information Management for z/OS database. The creation, update, or deletion of problem records can be initiated from the Tivoli Enterprise Console or automatically initiated through Tivoli Enterprise Console rules and tasks.

  Tivoli Information Management for z/OS provides an integration facility which integrates the Tivoli Enterprise Console into the Tivoli Information Management for z/OS problem management application. For more information on using the TEC integration facility, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

  In addition, through the use of terminal simulator EXEC control lines (such as OPENRRES, CLOSERRES, and GETRDATA), you can create a *remote data resource* in the Tivoli Information Management for z/OS BLX-SP address space. The remote data resource enables you to temporarily store and retrieve data used to send requests to the Tivoli Enterprise Console for processing. Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* for more information on the TSX control lines used to set up and manage remote data resources.

- Problem Service

Problem Service is a component of Tivoli Information Management for z/OS that provides applications with an interface to Tivoli Information Management for z/OS in a distributed networking environment. It enables Tivoli Enterprise Console events to be sent to Tivoli Information Management for z/OS as problem records. For more information on Problem Service, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

- Tivoli Inventory

  Tivoli Inventory is a hardware and software inventory-gathering application designed to help system administrators and accounting personnel manage the complexity of PC and UNIX® systems in a distributed client/server enterprise. You can use Tivoli Inventory to scan target machines to receive information on the hardware or software used by those machines, system configurations, and physical inventory. You can receive an extract of Tivoli Inventory data on your Tivoli Information Management for z/OS database, copy data to your customized Tivoli Information Management for z/OS panels, and use the data in your problem and change management applications.

  Special data model records are provided with the base Tivoli Information Management for z/OS product to define the Tivoli Inventory views and associated data. An interface program (i2i program) is available to enable you to map the data extracted from Tivoli Inventory to Tivoli Information Management for z/OS fields, and to load the data into the Tivoli Information Management for z/OS host database. For more information on setting up the host database to use Tivoli Inventory data and installing the necessary components, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

- Tivoli Problem Management (Tivoli Service Desk)

  As a core application in the Tivoli Service Desk suite of applications, Tivoli Problem Management is a network help desk system that enables help desk analysts to register calls and resolve problems. With Tivoli Problem Management, help desk analysts can simultaneously access a large database of problems and solutions to provide a high level of service to help desk customers. Online problem resolution tools called diagnostic aids are available to enable help desk agents find solutions quickly for a wide range of problems. Help desk agents can use Tivoli Problem Management to track customer calls and problems and transfer calls and problems to other help desk analysts. Users can submit problems through e-mail or on the Web without calling the help desk, and can even resolve many problems themselves.

  Tivoli Information Management for z/OS users can exchange problem records stored in the Tivoli Information Management for z/OS database with records stored in a Tivoli Problem Management database through the Tivoli Service Desk Bridge. Whether a problem record resides in Tivoli Information Management for z/OS or in Tivoli Problem Management, a help desk analyst working either from the Tivoli Information Management for z/OS host (or API application) or Tivoli Problem Management workstation can request transfer of the record in order to work on it.

  From Tivoli Information Management for z/OS, you can transfer problem records to users in Tivoli Problem Management, and update records sent from Tivoli Problem Management. Likewise, you can transfer problem records from Tivoli Problem Management to users in Tivoli Information Management for z/OS, and update records sent from Tivoli Information Management for z/OS. In addition, you can send solution data from Tivoli Information Management for z/OS to Tivoli Problem Management. From Tivoli Problem Management, you can create change records in Tivoli Information Management for z/OS. Refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications* for more information about integrating with Tivoli

Service Desk applications and setup instructions for Tivoli Information Management for z/OS. Refer to the *Tivoli Service Desk Networking Guide* for information about setting up Tivoli Service Desk to use the Tivoli Service Desk Bridge.

- Tivoli Software Distribution

  Tivoli Software Distribution automates the process of distributing software to clients and servers throughout an enterprise. It allows you to install and update applications and software in a coordinated, consistent manner across platforms, for timely client/server application deployment.

  When a change request is approved in Tivoli Information Management for z/OS, you can generate a Tivoli Enterprise Console event to trigger the distribution of workstation software packages or upgrades to target machines in a Tivoli management software environment. The target machines can be on any workstation platform supported by Tivoli Software Distribution.

  For example, you can define the type of software to install in a Tivoli Software Distribution package file. It could be an in-house application, shrink-wrapped software, or an upgrade to existing workstation software. You can define the target machines in a named Tivoli Inventory query. Functions provided by Tivoli Information Management for z/OS enable you to associate the package file name and query name with a given change request record and initiate a Tivoli Enterprise Console event that will trigger the distribution of software to the target machines when the change request is approved in Tivoli Information Management for z/OS.

  Using Tivoli Software Distribution, you can also remotely install the Tivoli Information Management for z/OS HLAPI client software.

  For more information on using Tivoli Information Management for z/OS with Tivoli Software Distribution, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

TEC Event Adapter

The Tivoli Information Management for z/OS TEC Event Adapter is a C program that runs under OS/390 UNIX System Services and provides an interface to the Tivoli Enterprise Console. The TEC Event Adapter can be installed as an optional component during SMP/E installation, and is required if you are integrating with the following Tivoli products as indicated:

- Tivoli Software Distribution – The TEC Event Adapter is required to enable Tivoli Enterprise Console to distribute software after a change request is approved in Tivoli Information Management for z/OS.
- Tivoli Service Desk – If you are using the Tivoli Service Desk Bridge to exchange problem records with Tivoli Problem Management, the TEC Event Adapter is required only if you want to create TEC events for Service Desk Bridge errors.

The Tivoli Information Management for z/OS TEC Event Adapter receives data sent by a TSX, builds a Tivoli Enterprise Console map, and uses TCP/IP to send data to the Tivoli Enterprise Console. For more information about the TEC Event Adapter (BLGTECAD) program, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

OS/390 Text Search

If you use Tivoli Information Management for z/OS for problem management, you can take advantage of the problem resolution data collected in your Tivoli Information Management for z/OS database by building a knowledge base. Your help desk analyst can search the problem resolution text to find solutions that are likely to apply to future incoming problems. Tivoli Information Management for z/OS can be set up to use the Text Search Engine component of OS/390 Text Search. The Text Search Engine is used to index and query the solution text, and to create the solution knowledge base. For example, you can do the following:

- A system administrator can use a Tivoli Information Management for z/OS panel to create and maintain the text search index that is used to build the solution knowledge base.
- A batch utility is run to identify existing problem records as solutions and add these to the solution knowledge base.
- A help desk analyst handles and resolves a new problem, and marks a problem record as having a valid solution. The solution text is added to the solution knowledge base where it can be searched by other help desk analysts.
- A help desk analyst can search the solution knowledge base through a Tivoli Information Management for z/OS panel or through some other interface such as a Tivoli Information Management for z/OS Desktop or Web connector application. The analyst can enter fuzzy searches or searches using AND, OR, or NOT operators to find possible relevant solutions to a problem. The Text Search Engine, with its powerful linguistic capabilities, retrieves the information from the solution knowledge base.

By taking advantage of your existing data and providing your help desk agents with a robust text searching tool, you can build a knowledge base that can help to improve the productivity of your help desk analysts.

An API search transaction is also provided for the Tivoli Information Management for z/OS HLAPI, HLAPI/REXX, and all client HLAPI platforms.

For more information about building a solution knowledge base, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*. For instructions on how to search for data stored in a knowledge base, refer to the *Tivoli Information Management for z/OS User's Guide*. Information about the API search transaction is provided in the *Tivoli Information Management for z/OS Application Program Interface Guide*.

The Remote Environment Servers and Clients

You can access Tivoli Information Management for z/OS functions from remote environments using the High-Level Application Program Interface (HLAPI) and one of the remote environment servers.

You can access Tivoli Information Management for z/OS functions from the following clients using the protocols listed:

Client	Protocol
OS/2® clients	APPC or TCP/IP

Client	Protocol
CICS/ESA® clients	APPC
OS/390 UNIX System Services clients	TCP/IP
UNIX clients	AIX – APPC or TCP/IP HP-UX – TCP/IP Sun Solaris – TCP/IP
Windows NT® clients	APPC or TCP/IP

The client for CICS® (HLAPI/CICS) and OS/390 UNIX System Services (HLAPI/USS) can be used in either a local or remote environment.

To use a remote environment, you must first install the appropriate client feature on the client. The OS/2, UNIX (AIX, HP-UX, Sun Solaris), and Windows NT client features are available on a CD-ROM that you receive when you order the base Tivoli Information Management for z/OS product. The OS/390 UNIX System Services and the CICS/ESA client features are installed from host tape using SMP/E. Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for client installation instructions.

The servers come with the base Tivoli Information Management for z/OS product. A server must be set up for each BLX-SP that the client needs to access. Either APPC or TCP/IP must be set up on both the client and the server machines. Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for details on how to install and set up your client/server environment.

Client application programs use the client feature to communicate with the Tivoli Information Management for z/OS server through APPC or TCP/IP. The server uses the HLAPI to access and use Tivoli Information Management for z/OS functions.

Java™ programs can interface with Tivoli Information Management for z/OS clients on the following operating systems, which must also support the Java run-time environment:
- AIX
- HP-UX
- Sun Solaris
- OS/2
- Windows NT

Java wrappers and a sample Java program are provided with the HLAPI clients for these operating systems to simplify the task of writing a HLAPI program.

Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for a description of the Tivoli Information Management for z/OS clients.

Web Connector Features

The Tivoli Information Management for z/OS Web connector features enable you to access a Tivoli Information Management for z/OS database using a Web browser as a client. Transactions are received by TCP/IP and queued for processing by the Web connector feature server. The server interprets requests from clients, retrieves data from Tivoli Information Management for z/OS, and returns response data to the client machine in Hypertext Markup Language (HTML) code or plain text.

The design of the Web connector features assume there are multiple client machines communicating asynchronously with a Web connector server. The client and server machines are part of the same network and communicate using TCP/IP protocol. The network could be the Internet itself, or a private network (intranet) that has no external connections or is connected to the Internet through a firewall.

The three types of Web connector features available with Tivoli Information Management for z/OS are described below and are available with the base Tivoli Information Management for z/OS product.

- REXX Web connector for OS/2

  Allows client Web browsers to access Tivoli Information Management for z/OS databases through a connection that runs on an OS/2 platform. As a stand-alone OS/2 application, it implements the IBM Internet Connection Secure Server (ICSS) as well as the REXX HLAPI/2 client feature.

- REXX Web connector for MVS

  Allows client Web browsers to access Tivoli Information Management for z/OS databases through a connection that runs as a started task on an MVS system.

- REXX Web connector for OS/390

  Allows client Web browsers to access Tivoli Information Management for z/OS databases through a connection that runs on an OS/390 system. This connector provides additional capabilities such as multithreading and increased security and runs on the IBM HTTP Server. This connector does not require any of the Tivoli Information Management for z/OS HLAPI client programs to function.

For more information on installing or using these Web connector features, refer to the *Tivoli Information Management for z/OS World Wide Web Interface Guide*.

The NetView Bridge Adapter

The Tivoli Information Management for z/OS NetView Bridge Adapter enables the NetView and Tivoli Information Management for z/OS products to work together. Together with the NetView Bridge, NetView enables automated message-handling functions. These functions consist of message routing and transmission within the NetView address space (the NetView Bridge) and message processing and submission to the Tivoli Information Management for z/OS HLAPI (the NetView Bridge Adapter). The HLAPI, in turn, interfaces with the Tivoli Information Management for z/OS Low-Level API (LLAPI), which accesses the Tivoli Information Management for z/OS database.

The Adapter provides the connection between the NetView Bridge and the HLAPI. It transforms user-written NetView automation command procedures or requests into HLAPI transactions and responses. Through the NetView Bridge Adapter, you can use NetView to:

- Create records in a centralized Tivoli Information Management for z/OS database
- Update records in a centralized Tivoli Information Management for z/OS database in a manner that protects the integrity of the records
- Retrieve a list of records or a single record from a centralized Tivoli Information Management for z/OS database based upon a set of search criteria
- Perform user-defined tasks on records in a centralized Tivoli Information Management for z/OS database.

If you decide to use the NetView Bridge Adapter, you must have the prerequisite software listed in “Ordering the Necessary Hardware and Software” on page 89. You can use the NetView Bridge Adapter with or without the NetView AutoBridge.

For more information on the NetView Bridge Adapter, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

The NetView AutoBridge and PostProcessor

The Tivoli Information Management for z/OS NetView AutoBridge is a set of routines, panels, and tables that serve as an application enabler for the Tivoli Information Management for z/OS NetView Bridge Adapter. The AutoBridge requires use of the NetView Bridge Adapter. AutoBridge receives data from specific alerts, messages, and other applications through its application programming interface and uses this data to build and perform Tivoli Information Management for z/OS transactions.

Available as an optional feature you can install with the base Tivoli Information Management for z/OS product, the AutoBridge can help automate network management tasks such as monitoring the network for specific events, creating and updating Tivoli Information Management for z/OS records, searching for duplicate records, and notifying vendors of the status of their products.

AutoBridge uses the Tivoli Information Management for z/OS application programming interface which bypasses panels that allow operators to add record data. If your Problem Management panels invoke control panels, program exits, or terminal simulator panels or EXECs to modify Tivoli Information Management for z/OS records, you should use the NetView AutoBridge PostProcessor facility to supplement the records created by AutoBridge with additional data to complete each record as though it had been entered on a terminal by an operator. The NetView AutoBridge PostProcessor facility is a component of the base Tivoli Information Management for z/OS product.

For more information on planning for AutoBridge, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*. For instructions on installing the AutoBridge feature, refer to the Tivoli Information Management for z/OS *Program Directory*.

NetView Hardware Monitor Interface

You can use the NetView Hardware Monitor with some problem management functions. A user of the NetView Hardware Monitor Interface can create or update a Tivoli Information Management for z/OS problem record representing a NetView Hardware Monitor Interface event and file it in the Tivoli Information Management for z/OS database. You can make minor or major changes to the interface. Although the NetView Hardware Monitor Interface is retained for compatibility reasons, the NetView Bridge Adapter provides a more current interface and provides greater function through use of the Tivoli Information Management for z/OS HLAPI.

For more information, refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.

ODBC Driver

The Tivoli Information Management for z/OS Open Database Connectivity (ODBC) driver enables the Tivoli Information Management for z/OS host database to serve as a source of data for workstation applications that are enabled to use ODBC. From an ODBC-enabled database, spreadsheet, or text processing application running on a Windows NT workstation, you can click on the Tivoli Information Management for z/OS data fields you want to search on. You can specify a filter or query to limit the search to the Tivoli Information Management for z/OS data you need for reporting purposes. The ODBC driver uses standardized structured query language (SQL) to initiate search transactions of the data on the host, and enables you to retrieve the data at your workstation where you can use the power of your application to format the data into various types of reports, including graphical reports with pie charts and bar charts.

To use the ODBC driver, you must have the Tivoli Information Management for z/OS High-Level Application Program Interface Client for Windows NT installed at the workstation, and data model records created in Tivoli Information Management for z/OS to support ODBC access. Refer to the *Tivoli Information Management for z/OS Data Reporting User's Guide* for more information.

The ODBC driver is also a prerequisite if you are using Tivoli Decision Support for Information Management.

The Desktop

The Tivoli Information Management for z/OS Desktop is an optional feature that enables users to create or interact with records in the Tivoli Information Management for z/OS database through a customizable starter application that uses a Java framework to present data in a graphical user interface. The Desktop provides a set of icons users can click on to enter problem or call record information, or to search for data in Tivoli Information Management for z/OS. Using the Desktop, users can use graphical task panels to submit call or problem records without having to use traditional 3270 host panels.

The sample GUI application provided with the Desktop can be customized to fit your business needs through the use of enhanced data model records. The data model records, residing on the host database, provide a centralized location for administrators to use to make changes or updates to the design of the application. Once the design is updated, users can automatically pick up the changes when they start up and log on to their Desktop application. Validation of data is performed at the workstation using validation information obtained from Tivoli Information Management for z/OS when the Desktop application is started.

A Toolkit is provided with the Desktop to enable administrators to customize the appearance and function of the application. Administrators can use the Toolkit to add or delete processes or task panels, define the transactions associated with tasks, and specify the name and function of buttons visible to users. A configuration editor is also provided to enable both users and administrators to identify themselves for login purposes and transactions with the database.

The Integration Facility

The Integration Facility is an optional facility of Tivoli Information Management for z/OS. It includes a base set of panels for problem, change, and configuration management as well as CLISTs and jobs to interface to:

- NetView Hardware Monitor
- System Availability Management (SAM) function of the Resource Measurement Facility (RMF)
- Service Level Reporter (SLR)
- Operations Planning and Control/ESA (OPC/ESA)

The Integration Facility base panels are installed automatically when you load Tivoli Information Management for z/OS panels. The interface functions are contained in the SBLMSAMP library. For more information about the Integration Facility, refer to the *Tivoli Information Management for z/OS Integration Facility Guide*.

Integration Facility Base

The Integration Facility provides a formal process to manage systems management data. To implement this process, some of the basic Tivoli Information Management for z/OS panels are modified and others are added. Your organization must fully understand the systems management process as described in the *Tivoli Information Management for z/OS Integration Facility Guide* and decide whether to use it before attempting to use the base Integration Facility panels. If your organization's systems management process matches the one implemented for the Integration Facility, or if you do not have a formal systems management process, this facility may be of use to your organization. It has panels that provide additional functions and new reports for additional analysis.

You can use an existing Tivoli Information Management for z/OS database with the Integration Facility. However, stored response chains (SRCs) and terminal simulator panels (TSPs) created with the standard Tivoli Information Management for z/OS panels do not work with the Integration Facility panels because the panels are structured differently.

In addition, the data model and flow control information contained in the records built by the Integration Facility differs from the data model and flow control information contained in the records built by the standard Tivoli Information Management for z/OS panels. Therefore, if you have both kinds of data in a database (that is, data from standard Tivoli Information Management for z/OS panels and data from the Integration Facility), users may be taken to panels they are unfamiliar with and may get into a flow that they may not know how to recover from.

The problem dialog assisted-entry panels for Integration Facility are modified to accept an external date format of DDMMMYY (for example, 05JUN00). For more information, see "Alternate Date Format for the Integration Facility" on page 236.

Integration with Other Products

The Integration Facility provides jobs and CLISTs in the SBLMSAMP library. These jobs and CLISTs are used to interface to other products and build problem records that can be used with the Integration Facility panels.

Interface to NetView Hardware Monitor

The Integration Facility provides a ready-to-use interface to pass NetView Hardware Monitor alert data to build incident records with the Integration Facility. The following summarizes what the Integration Facility can do when you select this option:

- Provides Hardware Monitor access to the Integration Facility load modules
- Defines Hardware Monitor users to the Integration Facility
- Converts the format of any dates sent to the Integration Facility, if necessary

Interface to OPC/ESA

The Integration Facility provides a ready-to-use interface to record selected OPC/ESA incidents in your Tivoli Information Management for z/OS database. It provides OPC/ESA with JCL:

- To initialize Tivoli Information Management for z/OS
- To submit an immediate response chain (IRC) to create and store a record in a format needed by the Integration Facility panels

Interface to Service Level Reporter

The Integration Facility provides a ready-to-use interface to produce Service Level Reporter (SLR) reports from your Tivoli Information Management for z/OS database. It extracts data from the Tivoli Information Management for z/OS database using RFTs and passes it to SLR for processing.

Note: Placing the Integration Facility SLR modules in a data set concatenated before your current table definitions is the same as changing the table definitions; thus, SLR cannot empty your data tables. If necessary, you can use the DBMAINT facility to correct any problems, but you may not be able to recover data.

Multiple BLX Service Providers

Tivoli Information Management for z/OS supports the use of multiple BLX-SPs on a single z/OS system. You can run a test system and production system simultaneously or run different versions of Tivoli Information Management for z/OS simultaneously. Running multiple BLX-SPs also makes maintenance easier. For example, you can support user groups that have different maintenance schedules by connecting each group to a different BLX-SP.

For information on setting up multiple BLX-SPs, see “Defining Multiple BLX-SPs” on page 148.

Sysplex Support

You can run Tivoli Information Management for z/OS in a non-sysplex or z/OS Parallel Sysplex environment. A *sysplex* is a set of MVS systems that communicate and cooperate with each other through certain hardware and software components and software services to process workloads. A *parallel sysplex* has one or more coupling facilities that enable multiple central processor complexes to simultaneously process a workload. By allowing two or more processors to share the same data you can maximize performance while minimizing cost, improve system availability and concurrency, expand system capacity, and configure your system environment more flexibly.

Tivoli Information Management for z/OS takes advantage of parallel sysplex, with its superior processing capabilities.

Note: Throughout this document, references to sysplex are meant to imply parallel sysplex rather than base sysplex. Tivoli Information Management for z/OS provides support for parallel sysplex only.

Sysplex Advantages

Tivoli Information Management for z/OS exploits the benefits of a parallel sysplex environment in the following ways:

- It takes advantage of VSAM record-level sharing (RLS).
- It offers performance benefits, especially if you are sharing Tivoli Information Management for z/OS databases. VSAM data sets are allocated and opened directly by the Tivoli Information Management for z/OS user rather than by the BLX-SP.
- It eliminates the need for setup of APPC/MVS when using shared databases. With record-level sharing, applications running on more than one Tivoli Information Management for z/OS system can read from and write to the same set of data concurrently. VSAM handles all cross-system buffer invalidation.
- It enables you to execute operator commands on one or more BLX-SPs.

For more information about Tivoli Information Management for z/OS sysplex support, see “Setting Up for Sysplex Data Sharing” on page 151.

Data Sharing

Tivoli Information Management for z/OS enables users on different BLX-SPs to concurrently share Tivoli Information Management for z/OS VSAM data sets without compromising data integrity. The BLX-SPs that share the data sets can reside either on the same z/OS system or on multiple z/OS systems in the same sysplex.

Data sharing in Tivoli Information Management for z/OS helps you satisfy the requirement of needing very high levels of availability. With data sharing, you can run applications on many Tivoli Information Management for z/OS systems and access the same shared data. If one system must come down, either for planned maintenance or because of a failure, the database can be accessed from another Tivoli Information Management for z/OS system with no perceived outage to end users.

Data sharing in Tivoli Information Management for z/OS is an integrated hardware and software solution. It requires the z/OS parallel sysplex, which provides lower-cost, scalable computing power, and the use of VSAM RLS.

Note: In previous releases of Tivoli Information Management for z/OS, the facility that enabled data sharing to take place in Tivoli Information Management for z/OS was called *Multisystem Database Access (MSDA)*. MSDA is being replaced with the term *sysplex data sharing* in this manual. MSDA required advanced program-to-program communication (APPC/MVS) on all z/OS systems that shared data sets, even if all the BLX-SPs were on a single z/OS system. With the use of sysplex support, Tivoli Information Management for z/OS no longer requires APPC for data sharing. The data and index buffers are maintained in the SMSVSAM data space and in the coupling facility. VSAM handles any necessary communication with the coupling facility to maintain integrity of the VSAM buffers.

No other programs or non-Tivoli Information Management for z/OS utilities can share the data sets with Tivoli Information Management for z/OS unless they access the data sets through the Tivoli Information Management for z/OS APIs. As an exception, under RLS, if read integrity is not required, a SHAREOPTIONS(2) data set can be opened in read-only mode by a nonshared resource application (such as IDCAMS REPRO).

For information on setting up data sharing, see “Setting Up for Sysplex Data Sharing” on page 151.

Multiple-Cluster SDDS

The data component of a VSAM key-sequenced data set (KSDS) can hold a maximum of 4GB (GB equals 1 073 741 824 bytes) of information. In Tivoli Information Management for z/OS, the SDDS component of your database is usually the largest data set. If you expect the SDDS to exceed 4GB, you can use up to 100 VSAM clusters for your SDDS, giving you enough space for 400GB of information. Unless you have more than a million logical records in your SDDS, you probably need not worry about exceeding the single-cluster limit. You may want to use multiple SDDS clusters for performance reasons.

Using a multiple-cluster SDDS does not change the way you use Tivoli Information Management for z/OS. The only exceptions are backup procedure changes, some session-parameters member changes, some VSAM resource definition changes, and some syntax changes needed for activating multiple SDDS clusters when running the BLGUT7 utility program. For information on the BLGUT7 utility program, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

You can run BLGUT20 to obtain statistics that can help you determine the total SDDS size and the number of logical records it contains. You can use these statistics to determine how close your SDDS is to 4GB and the number of logical records your database holds. For information on the BLGUT20 utility program, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* .

For information on setting up a multiple-cluster SDDS, see “Working with Multiple-Cluster SDDSs” on page 172. For information regarding the performance of a multiple-cluster SDDS, see “Multiple-Cluster SDDS” on page 67.

Multiple-Cluster SDIDS

The data created by Tivoli Information Management for z/OS is stored in the SDDS. The SDIDS is a cross-reference to the searchable words stored in the SDDS. It contains index keywords that are used to locate records in the SDDS. Like the SDDS, the SDIDS is a key-sequenced VSAM data set that can consist of either a single or multiple clusters. Each SDIDS can contain up to 4GB of data. The maximum number of records you can store in the SDIDS is 400GB.

A multiple-cluster SDIDS can be useful to help improve your overall database performance by reducing the amount of data that is locked when the SDIDS is accessed, since the clusters can be accessed in parallel. If you store significant amounts of searchable data, you may benefit from setting up a multiple-cluster SDIDS to divide your records based on key contents. For example, you might choose to separate s-words from p-words, or some other arrangement.

For information on setting up a multiple-cluster SDIDS, see "Working with SDIDSs" on page 176 .

SDIDS Keys

Starting with TME 10 Information/Management Version 1.1, the 16- and 32-byte SDIDS keys that were provided in earlier releases of Information/Management, which supported the use of VSAM spanned and non-spanned records, are no longer supported. Only the use of 18- and 34-byte keys and non-spanned data sets are supported.

If you are migrating from Information/Management Version 6.3 or earlier releases and you have an existing SDIDS, you must use either the BLGUT1 to rebuild the SDIDS or the BLGUT1M utility to convert the SDIDS to the new SDIDS data structure and key format. The internal architecture of the SDIDS is different, and therefore a rebuild or conversion is required. If you are migrating from TME 10 Information/Management Version 1.1 or later releases, you do not have to rebuild or migrate the SDIDS because the architecture of the SDIDS is not changed.

The data structure and key format allows Tivoli Information Management for z/OS to:

- Completely eliminate the need for spanned records
- Have an unlimited number of database records from a practical point of view
- Provide information that can help you tune database performance
- Improve overall performance
- Allow multiple SDIDS clusters to be used.

The data structure of the SDIDS is not backward compatible; however, the structure of the SDDS is backward compatible. If necessary, your back-level BLGUT1 utility can be used to rebuild the SDIDS.

You need to determine what SDIDS key size to use—18 or 34 bytes. The key size is specified when you define the SDIDS, and detected when you start Tivoli Information Management for z/OS.

For search purposes, both the 18- and 34-byte key can be used with single-byte character set (SBCS) or double-byte character set (DBCS) data. The 34-byte key is highly recommended for DBCS users. The 18-byte key is not recommended for DBCS because it supports only 7 DBCS characters when doing a DBCS search. The 34-byte key supports up to 15 characters on a DBCS search. For more information on searching and the use of SDIDS keys, see "Effects of SDIDS Key Length Settings on Searches" on page 177.

If you use SBCS characters, you can use either an 18-byte or a 34-byte key. With the 18-byte, the keyword size can be up to 16 characters. The keyword can be up to 32 characters with the 34-byte key.

If you decide to change key sizes, you **must** run the BLGUT1 utility to create the SDIDS using the new key size, or use the BLGUT1M utility to copy the data from the old SDIDS to the new SDIDS.

Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on the BLGUT1 or BLGUT1M utility programs.

Multiple Read Panel Data Sets

You can use one or more read panel data sets, depending on the compromises you make between performance, ease of maintenance, and DASD space. The fewer read panel data sets you have, the better Tivoli Information Management for z/OS performs, because Tivoli Information Management for z/OS must search each read panel data set in a specified order to find a panel.

One read panel data set results in the best performance; multiple read panel data sets reduce that performance, yet having more than one read panel data set simplifies maintenance for the panels because you can separate panels used for different purposes into different panel data sets. For maintenance reasons, it is best to have two read panel data sets: one for Tivoli-supplied panels and one for user-modified panels.

For more information, see “Using One or More Read Panel Data Sets” on page 198.

Security

If you use a security product, such as RACF, to protect or restrict your database and VSAM data sets from unauthorized access, evaluate the effect of using RACF with Tivoli Information Management for z/OS.

For further details about using RACF, see “Using RACF to Protect Tivoli Information Management for z/OS VSAM Data Sets” on page 191 or refer to the RACF product publications.

Logical Database Partitioning

You can organize data in the Tivoli Information Management for z/OS database into “logical partitions”. These partitions can be completely isolated from each other when viewed by a user in a given partition, but accessible as a single database to selected authorized personnel. The ability to partition the Tivoli Information Management for z/OS database provides an additional measure of administrative control. For example, if you provide service to different divisions, you can separate the records of one division from the records of another division. This is done by assigning logical partitions to records.

The database administrator controls which users have access to which databases by assigning partitions to privilege classes. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information.

Automatic Log Save Facility

The Tivoli Information Management for z/OS Automatic Log Save Facility provides the capability for:

- Automatic, near real-time maintenance of a copy (local or remote) of a Tivoli Information Management for z/OS database
- Regular, automatic capture of SDLDS data and management of the SDLDS

These capabilities improve reliability, availability, and performance for Tivoli Information Management for z/OS users in the following ways:

- Users can use the Automatic Log Save Facility to automatically back up their Tivoli Information Management for z/OS production database. The backup database can be used for reports or recovery.
- Users can specify an automatic, time-driven offload of the SDLDS.
- Users can easily tailor the Automatic Log Save Facility for their installation including:
  - Scheduling of the SDLDS offload from the source (send) database
  - Scheduling of the upload to the destination (receive) database
  - Selection criteria for records that are to be uploaded.

If you are currently using the Automatic Log Save Facility, or if you are planning to use it, you should also evaluate the BLGUT23 series of utilities that are available in Tivoli Information Management for z/OS to back up the database while users are updating the database. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details on these utilities.

Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about the Automatic Log Save Facility.

DB2 Extract Facility

The DB2 Extract Facility extends the capability of the Automatic Log Save Facility to DB2. Some of these capabilities include:

- Automatic, near real-time maintenance of a copy (local or remote) of Tivoli Information Management for z/OS data in a DB2 database
- Regular, automatic capture of SDLDS data and management of the SDLDS

These capabilities provide for improved reliability, availability, and performance for Tivoli Information Management for z/OS users in the following ways:

- A user can issue SQL queries against Tivoli Information Management for z/OS data that is stored in a DB2 database.
- The production Tivoli Information Management for z/OS database does not suffer the performance degradation that is commonly associated with random user queries.
- A user can specify an automatic, time-driven offload of the SDLDS.
- Users can easily tailor the DB2 Extract Facility for their installation including:
  - Customizing the mapping of Tivoli Information Management for z/OS record constructs, such as s-words and p-words
  - Scheduling of the SDLDS offload from the source (send) database

For more information about the DB2 Extract Facility, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

Archiver

The Archiver function enables you to migrate (archive or copy) records from one Tivoli Information Management for z/OS database to another. In the process of migrating records, you can specify whether you want to delete the records from the ″live″ database or leave the records intact. Other functions of the Archiver include the ability to perform the following tasks:

- Copy records back to the same database but with the next system-assigned record number identifier
- Transfer records between two databases in the same session-parameters member
- Compress records, or uncognize or delete specific data items before the archival
- Keep the user-defined relationships intact on the archived records

Additionally, the Archiver provides ample logging functions, including:

- The parameters that you specified when you invoked the Archiver
- Messages related to each record processed
- A completion summary of what the Archiver did
- Full low-level application programming interface (LLAPI) logging

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

Database Formats

Tivoli Information Management for z/OS databases can be in one of two formats:

- Tivoli Information Management for z/OS format
- User-defined format (previously known as the Information/MVS format).

Both types of databases are made up of SDDS and SDIDS clusters. Your Tivoli Information Management for z/OS read/write database can also contain an SDLDS cluster. The two formats differ in the way that you enter data into the databases. You enter data into a Tivoli Information Management for z/OS format read/write database interactively through Tivoli Information Management for z/OS panels and dialogs or through an API. You use the BLGOZUD utility to load data from a sequential data set into a user-defined format database. Any data that must be searchable and does not change frequently is a candidate for a user-defined database.

For more information on creating a user-defined format database, see “Working with Databases” on page 168. For information on using the BLGOZUD utility, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

National Language Support

Tivoli Information Management for z/OS supports national languages other than English by providing translate tables, DBCS support, graphic character substitutions, and the ability to sort search results lists with an external sort routine.

Latin and Non-Latin Translate Tables

The English language and other Germanic languages, as well as the Romance languages, use the *Latin* alphabet. Languages that are not Romance or Germanic-based (for example, the Japanese Katakana alphabet) use *non-Latin* alphabetic characters. Tivoli Information Management for z/OS provides translate tables to enable for the use of Latin or non-Latin alphabetic characters.

Latin translate table: Enables you to store all upper and lowercase Latin alphabet characters as searchable uppercase, mixed case, or lowercase Latin alphabet characters in your database.

The Latin translate table is required for use if you elect to collect, display, or store Tivoli Information Management for z/OS data in mixed case.

Non-Latin translate table: Enables you to store searchable non-Latin in your database.

Note: You must select the table you want to use when you install Tivoli Information Management for z/OS; you cannot change tables at a later time. For more information on using translate tables, see “Working with Translate Tables” on page 163. To look at the translate tables used for display and for blank substitution in string data fields, see “Translate Tables” on page 353.

Uppercase and Lowercase Translate Tables

Tivoli Information Management for z/OS enables you to tailor the uppercase and lowercase translate tables for Latin alphabets; you can convert your country unique alphabet from lowercase to uppercase, or uppercase to lowercase. The non-Latin translate table does not have a corresponding uppercase or lowercase translate table because non-Latin alphabets do not have case.

Update your uppercase or lowercase translate table only if you require special national language support. You tailor your translate tables by using the Installation Tailoring Facility. See “Using the Installation Tailoring Facility” on page 185 for information on using the Installation Tailoring Facility.

DBCS Operating Parameter

A parameter in the BLX-SP parameters member indicates whether the BLX-SP and all users connected to the BLX-SP support DBCS data.This parameter enables customers who do not require DBCS support to enjoy better performance than those customers who do require DBCS support. Those Tivoli Information Management for z/OS installations supporting only SBCS data need not perform any additional processing in support of DBCS data. Because of this performance consideration, it is recommended that you not run the BLX-SP with DBCS support if you only require SBCS support.

For more information about defining a BLX-SP parameters member, see the online introduction of the Installation Tailoring Facility or “Defining BLX-SP Parameters Members” on page 343.

Graphic Character Substitutions

Tivoli Information Management for z/OS enables you to use substitute characters for four of the graphic characters that Tivoli Information Management for z/OS uses. The code points for these four characters are X'5F', X'4F', X'5A', and X'7C' (the ¬, |, !, and @ characters on code page 37). In earlier releases of the product, the characters that you used were the ones that were at those code points on your code page. Now, you can specify other characters as substitutes. For example, you can substitute the ¬, |, !, and @ characters from your code page for the characters that are at code points X'5F', X'4F', X'5A', and X'7C'.

Use graphic character substitutions only if you require special national language support. The substitutions apply to all of your Tivoli Information Management for z/OS users. If you decide to use graphic character substitutions, you must review and modify your RFTs (including Tivoli-supplied RFTs), SRCs, and TSPs so that they reflect the substitutions. You must also modify your TSXs and API programs. For TSXs, you can obtain the graphic characters that should be used from the REXX variable BLGSYMB. For API programs, you can obtain the graphic characters that the programs should use from the HLAPI extension TSX BLGTSPCH. For more information about using the REXX variable BLGSYMB, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*. For more information about the HLAPI extension TSX BLGTSPCH, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.

Note: This enhancement applies to graphic characters that are entered by a user, an API, or a report. Information that is displayed to the user (such as text on Tivoli Information Management for z/OS panels) may not reflect the substitution.

You can substitute any or all of these characters. You specify substitution characters through the Installation Tailoring Facility. For information on using the Installation Tailoring Facility see “Using the Installation Tailoring Facility” on page 185.

If you write TSXs to process data with the Tivoli Information Management for z/OS database, you can use a special REXX variable provided with Tivoli Information Management for z/OS to ensure your TSXs process the correct substitution character. For example, instead of hardcoding the ″not″ sign and the ″or bar″ characters in a TSX, you can parse the BLGSYMB REXX variable to obtain the proper substitution characters when the TSX is initialized. For more information on using the BLGSYMB REXX variable, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

Additionally, an HLAPI extension TSX BLGTSPCH is provided if you use high-level API applications with Tivoli Information Management for z/OS and need to know what character to use for the ″not sign″ or the ″or″ bar. For more information about this HLAPI extension, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.

Sorting with an External Sort Routine

With Tivoli Information Management for z/OS, you can sort search results lists with the sort routine that you specify in your session-parameters member. You specify this external sort routine by specifying a value in the **Sort load module** field and YES in the **Use for Search Results Lists?** field when you define a session-parameters member using the Installation Tailoring Facility. See “Using the Installation Tailoring Facility” on page 185 for information on using the Installation Tailoring Facility. You can also specify an external sort routine by specifying values in the SORT and EXTSORT parameters of the BLGPARMS macro. See “BLGPARMS Macro — Defining Tivoli Information Management for z/OS’s Operating Characteristics” on page 318 for more information on the BLGPARMS macro.

Note: Using this external sort routine for sorting list data and searching the search results list is not as efficient as using the Tivoli Information Management for z/OS internal sort routine. Consider specifying an external sort routine only if you find that the Tivoli Information Management for z/OS internal sort routines do not sort the list data and search results list in the right order for your language.

Date and Time-of-Day Formats

The *internal* date format used by Tivoli Information Management for z/OS is YYYY/MM/DD.

Tivoli Information Management for z/OS also provides various *external* date formats to suit preferences for many regions of the world. A description of date formats is provided in “Enabling Alternative Date and Time-of-Day Formats” on page 227. The external date formats provided by Tivoli Information Management for z/OS can help to eliminate the need to write your own external date routine to support special date format requirements. Users sharing a single database can each select a preferred format from a list of supported formats by making a selection in their user profile. If users want to use a format that is not provided, you can write your own external date conversion routine; however, this is not recommended unless absolutely necessary because it usually degrades performance slightly and also requires that all users sharing a session member use the same external date format.

Tivoli Information Management for z/OS ships product panels that support any of the product-supplied date formats. To support all of the product-supplied formats, your customized panels must follow these rules:

1. All entry and display fields for date values on data-entry and table panels must be at least 10 characters wide (the longest product-supplied date format is 10 characters).
2. The validation patterns on data attribute records and assisted-entry panels for date fields must allow any of the formats. The product panels use a pattern of IIV63, which will support all of the product-supplied formats, plus any valid user-defined format. It is recommended that you use a validation pattern of IIV63. You can use IIV9, which will support any of the current product-supplied formats, but you may need to change this pattern later if you want to write you own format or if subsequent releases of Tivoli Information Management for z/OS support longer formats.

If you choose to use panels that do not support all of the product-supplied formats, you should make a copy of data attribute record BLG&DFMT and remove the formats that your panels cannot support. Then, modify data-entry panel BLG0P700 to use your modified attribute record. This action will prevent users from choosing a format that your panels cannot handle.

Note: Whether or not you use panels that support all of the product-supplied formats, it is suggested that you modify the text or help information for all assisted-entry panels so as to make them independent of any one date format. See the following panel BLG60CCD for an example.

```
+ BLG60CCD ------------ PROBLEM OCCURANCE DATE ------------- DATO/-+
|                                                                  |
| USE....Enter the date that the problem occurred or was detected. |
|                                                                  |
| FORM...Date in your external format (e.g. MM/DD/YYYY) -or-       |
|        = for today's date or an offset from today's date.        |
| NOTE...Enter ;HELP STATUS to find your external date format.     |
|                                                                  |
| EXAMPLES: May 27, 2001..Reply...05/27/2001, etc.                 |
|           Today.........Reply...=     1 month ago...Reply...=-1M |
|           Yesterday.....Reply...=-1   2 years ago...Reply...=-2Y |
|           2 weeks ago...Reply...=-2W                             |
|                                                                  |
|                                                                  |
+----------------------- REPLY AS ILLUSTRATED----------------------+





===>
```

A migration utility, BLGUT6M, is available to help you change panels containing date fields regardless of the external date format you use. For example, if your external date format is DDMMMYY (e.g., 30NOV00) and you would like to change it to DDMMMYYYY for your panels, you can use this migration utility to expand all the date fields on your panels. Or, if you have customized the Tivoli Information Management for z/OS product panels, you can use BLGUT6M to make the changes to your copy of the panels. You can use this migration utility to expand date fields to 10 characters, if necessary, and change validation patterns on your panels.

If you do not want to use the available Tivoli Information Management for z/OS date or time-of-day formats, you can use your own user exit routine to select another date or time-of-day format.

For more information on using date formats, see “Enabling Date Formats” on page 228. For more information on using an alternate time-of-day format, see “Enabling an Alternate Time-of-Day Format” on page 245.

Universal Time Support

If your company has locations that are spread geographically, such as in different countries or time zones, you can enable universal time processing in Tivoli Information Management for z/OS. The support for universal time (also known as Greenwich Mean Time) enables users at different locations to enter and view dates and times in records in their own time zone. This option eliminates the need for users to remember that a certain company location is ″ahead″ or ″behind″ their own location by a certain number of hours when reviewing records in the database. For example, a user in Boston can view a problem record and the dates and times in the record reflect Boston time, while a user in Frankfurt can view the same record with the dates and times reflecting Frankfurt time.

To enable universal time processing, the TIMEZONE keyword must be specified on the BLGPARMS macro in the session member. In addition, date and time fields must be defined as related pairs so that they have a relationship to one another. This relationship between date and time is important if you want users to see the data in their own local time.

A set of time zone definitions is provided in a TIMEZONE record with Tivoli Information Management for z/OS. These time zone definitions cover many geographic locations around the world. You can modify these definitions, if necessary, or create your own. In addition, you can define the rules you would like to use for daylight savings time. Users can also specify in what time zone they would like to enter and view times on panels by specifying a user and database default option.

For more information about enabling universal time processing, see "Implementing Universal Time Processing" on page 251.

Enhanced Panel Style

The enhanced panel style provides a graphically oriented method of selecting Tivoli Information Management for z/OS functions. To make Tivoli Information Management for z/OS more familiar to users of CUA® compliant interfaces, the enhanced panels can appear with an action bar at the top, pull-down menus, and context-specific function keys. These Tivoli Information Management for z/OS user interface controls provide a user the opportunity to perform an action by moving the cursor to the desired selection and pressing Enter. It is recommended that you select the enhanced panel style if you plan to enable the Tivoli Information Management for z/OS graphical user interface that is available through ISPF (see "Graphical User Interface (GUI)" on page 53).

An enhanced panel on a display with more than 26 lines and with the function keys showing appears similar to Figure 2 on page 51:

```
  Environment   Dialog   Record   Window   ISPF   Help
------------------------------------------------------------------------------
===>

BLG0EN20              --- PRIMARY OPTIONS MENU ---     APPLICATION: MANAGEMENT

OPTIONS:

      1. OVERVIEW.......Display general information and product enhancements.
      2. PROFILE........Display or alter invocation or session defaults.
      3. APPLICATION....Change application, list available applications.
      4. CLASS..........Change current class, list available classes.
      5. ENTRY..........Create a record.
      6. INQUIRY........Search for records.
      7. UTILITY........Copy, display, print, delete, and update records.
      8. GLOSSARY.......Display a list of searchable words in the database.
      9. PMF............Modify or create panels.


           Select an option, enter a command, or type QUIT to exit.


          Tivoli Information Management for z/OS Version 7 Release 1
                5697-SD9 (C) Copyright IBM Corp., 1981, 2001.




F1=Help      F2=Set 2     F3=End       F4=Split     F5=Back      F6=Suspend
F7=Up        F8=Down      F9=Recall   F10=Actions  F11=Init     F12=Cancel
```

Figure 2. Tivoli Information Management for z/OS Primary Options Menu with 27 Lines

If your screen displays fewer than 26 lines, the enhanced panel is similar to Figure 3 on page 52 . The MORE: + designation at the end of the line separating the action bar from the rest of the panel means that the panel has more lines than appear on your screen. The + indicates that you must scroll down to see the hidden lines.

```
  Environment   Dialog   Record   Window   ISPF   Help
------------------------------------------------------------------ MORE: +
===>

BLG0EN20              --- PRIMARY OPTIONS MENU ---     APPLICATION: MANAGEMENT

OPTIONS:

      1. OVERVIEW.......Display general information and product enhancements.
      2. PROFILE........Display or alter invocation or session defaults.
      3. APPLICATION....Change application, list available applications.
      4. CLASS..........Change current class, list available classes.
      5. ENTRY..........Create a record.
      6. INQUIRY........Search for records.
      7. UTILITY........Copy, display, print, delete, and update records.
      8. GLOSSARY.......Display a list of searchable words in the database.
      9. PMF............Modify or create panels.


           Select an option, enter a command, or type QUIT to exit.

F1=Help      F2=Set 2     F3=End       F4=Split     F5=Back      F6=Suspend
F7=Up        F8=Down      F9=Recall   F10=Actions  F11=Init     F12=Cancel
```

Figure 3. Tivoli Information Management for z/OS Primary Options Menu with 24 Lines

After scrolling down to see the hidden lines, the MORE: + notation changes to MORE: - (see Figure 4). This still means the panel has more lines that do not appear on your screen. The - indicates that you must scroll up to see the hidden lines.

```
  Environment   Dialog   Record   Window   ISPF   Help
------------------------------------------------------------------ MORE:  -
===>

OPTIONS:

      1. OVERVIEW.......Display general information and product enhancements.
      2. PROFILE........Display or alter invocation or session defaults.
      3. APPLICATION....Change application, list available applications.
      4. CLASS..........Change current class, list available classes.
      5. ENTRY..........Create a record.
      6. INQUIRY........Search for records.
      7. UTILITY........Copy, display, print, delete, and update records.
      8. GLOSSARY.......Display a list of searchable words in the database.
      9. PMF............Modify or create panels.


           Select an option, enter a command, or type QUIT to exit.


          Tivoli Information Management for z/OS Version 7 Release 1
                 5697-SD9 (C) Copyright IBM Corp., 1981, 2001.

F1=Help      F2=Set 2     F3=End       F4=Split     F5=Back      F6=Suspend
F7=Up        F8=Down      F9=Recall   F10=Actions  F11=Init     F12=Cancel
```

Figure 4. Tivoli Information Management for z/OS Primary Options Menu after Scrolling

To select a panel style, see “Selecting the Tivoli Information Management for z/OS Panel Style” on page 199. For information on tailoring the enhanced panel style panels, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*. The *Tivoli Information Management for z/OS User’s Guide* and the *Tivoli Information Management for z/OS Program Administration Guide and Reference* contain additional information on the enhanced panel style panels.

Graphical User Interface (GUI)

ISPF enables you to display Tivoli Information Management for z/OS panels in a workstation window. ISPF uses TCP/IP or APPC protocols to present ISPF panels in a remote graphical workstation environment. Tivoli Information Management for z/OS users reap the benefits of graphical windows without having to learn a new application. ISPF automatically transforms the Tivoli Information Management for z/OS panels into graphical form; no panel translation or rewriting is required. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information about using the ISPF GUI.

Figure 5 shows the Tivoli Information Management for z/OS Primary Options Menu displayed using the Tivoli Information Management for z/OS enhanced panel style in GUI mode.

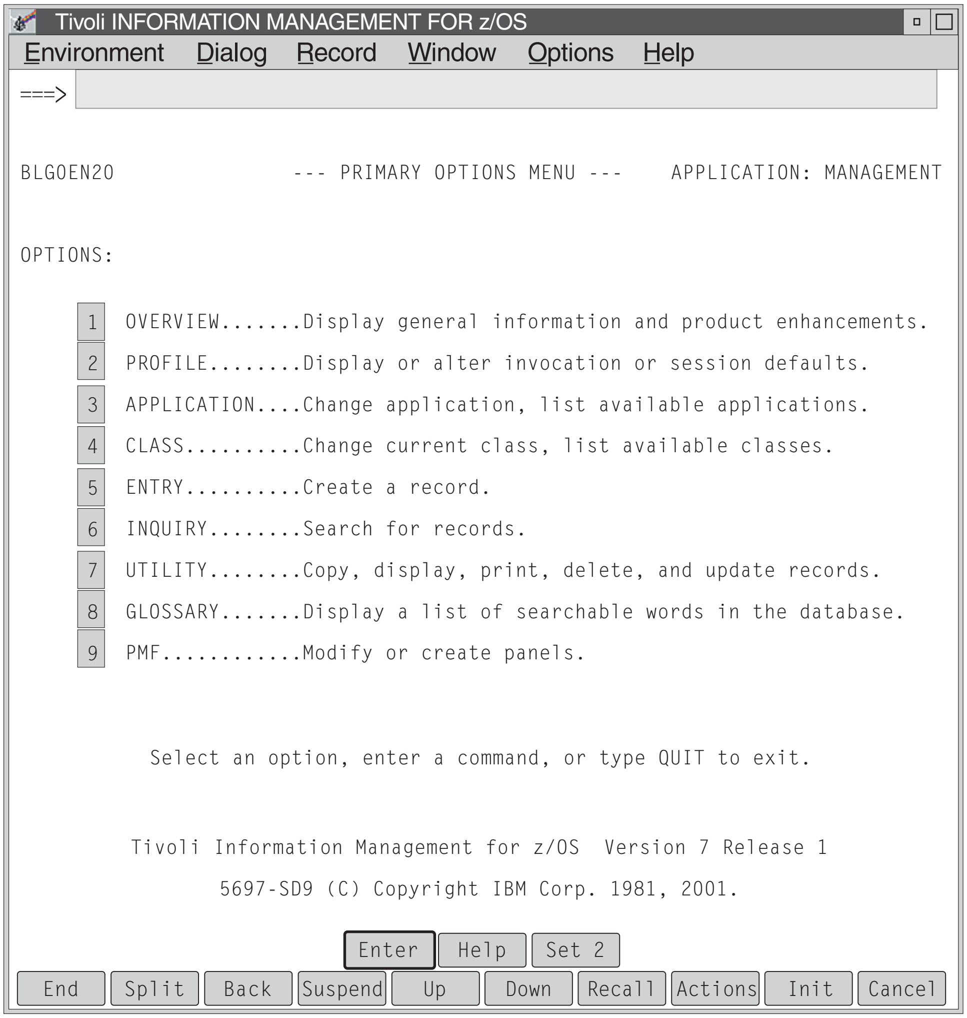

Figure 5. Tivoli Information Management for z/OS Primary Options Menu Displayed in ISPF GUI Mode

Graphical user interface controls, such as sizable, movable, and scrollable windows, push button function keys, action bars, and pull-down menus, are automatically available when the user runs in ISPF's graphical user interface (GUI) mode. Push buttons, mnemonic choice selections, accelerator keys, unavailable pull-down choices, and separator bars are also available. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about the ISPF GUI mode. See "Installing the Graphical User Interface" on page 200 for information on installing GUI.

Note: It is recommended that you select the Tivoli Information Management for z/OS enhanced panel style if you decide to run Tivoli Information Management for z/OS in GUI mode.

The Notification Management Facility

With the Notification Management facility of Tivoli Information Management for z/OS, you can be notified whenever a record is changed. Notification processing can be done through the use of TSPs, as with previous releases of Tivoli Information Management for z/OS, or with terminal simulator EXECs (TSXs), which provide similar but enhanced capabilities.

Using TSXs, you can send electronic mail (e-mail) messages to your users whenever any record is created or updated, regardless of whether the record was created interactively by a user or through an API.

You can send mail messages through MVS TCP/IP simple mail transfer protocol (SMTP) from a TSX running in a user's address space, or you can queue the mail to the BLX-SP, where it can be extracted and sent at a later time. Mail sent through the TCP/IP SMTP can be routed to a workstation e-mail software package, such as Lotus® Notes™, through gateway software that links the product with SMTP mail.

A new set of TSXs is provided with Tivoli Information Management for z/OS Version 7.1 for you to use or customize. These sample TSXs are provided for notification of problem, change, change activity, and Integration Facility problem and change records. You can modify the TSXs that come with Tivoli Information Management for z/OS to invoke mail processing on your system.

Using the new set of TSXs, you can send e-mail messages immediately or place them on a queue. You can also send escalation notices. Advantages of this new method of notification over former methods include the ability to use an SMTP server on a platform other than MVS, increased flexibility in processing messages, support for both ″hot″ queues and normal queues, and greater ease of defining message content.

Tivoli Information Management for z/OS is shipped with the Notification Management facility partially disabled. You must decide whether you want to enable or totally disable this facility for problem, change, and activity records. You must either enable or totally disable this facility during installation, regardless of whether you elect to use TSPs or TSXs for notification processing. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for further details.

The Panel Modification Facility (PMF)

The Panel Modification Facility (PMF) enables you to tailor panels to meet the needs of your organization.You can use PMF to improve the performance of the system and consequently the productivity of your end users. Consider the following:

Panel Consolidation
: You can use PMF to consolidate groups of Tivoli Information Management for z/OS panels into unique organization-tailored panels. Thus, by satisfying your requirements through consolidation, end users' productivity is improved. This can result in generating a higher system workload for each end user.

Mixed case support
: If you need to collect, store, or display data in mixed case, you can use PMF to specify whether or not data entered by users should match a validation pattern, how the data should be collected or stored in the SDDS, and whether or not the data is cognized in mixed case (stored in the SDIDS) for searching.

Selective Indexing
: You can also use PMF to reduce the number of fields of a record that are cognized during a file/update operation. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*. Fields that you do not want searchable need not be cognized, thereby reducing the EXCP activity to the SDIDS during a file, update, or delete operation. The reduction in SDIDS I/O activity can result in improved response time.

Optional S-Word Cognizing
When you collect both s-word and p-word data, cognizing s-words is optional.This enables you to eliminate unnecessary input/output (I/O) to your SDIDS, resulting in shorter response times, especially when filing new or changed records and copying or deleting records. Eliminating unnecessary s-word cognizing also makes your SDIDS smaller and improves the effectiveness of VSAM buffering. For more information on optional s-word cognizing, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

Automatic Panel Migration
When you use PMF, panels are automatically migrated from previous versions to this version. Assisted-entry panels now contain a **Data type** field.This field enables you to specify whether the data-entry fields accept SBCS, DBCS, or mixed data.The **Data type** field is displayed on the Assisted-Entry Summary Panel (BLM8CU53) and can be specified on the Response Processing panel (BLM8CU5D).

Installing PMF does not require any special tasks or selections. For more information about PMF, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

4

Evaluating Performance

This chapter discusses performance considerations for Tivoli Information Management for z/OS. It provides:

- An introduction to performance in Tivoli Information Management for z/OS
- An overview of recent performance changes
- A summary of important performance recommendations

This chapter assumes that you have a general knowledge of:

- The z/OS system environment
- VSAM resources
- Tivoli Information Management for z/OS functional capabilities

This chapter is designed to:

- Help you plan a system environment that will optimize Tivoli Information Management for z/OS performance
- Familiarize you with the types of system resources that Tivoli Information Management for z/OS requires to maintain optimal performance

Note: Estimates and measurements provided in this chapter do not predict the performance of your system.

An Overview of Tivoli Information Management for z/OS Performance

As with any application, resources are used by Tivoli Information Management for z/OS to accomplish a task. This section introduces the key resources and discusses the performance ramifications of each related to interactive workloads in the Tivoli Information Management for z/OS product. The key resources are I/O time, processor, and memory.

I/O Time

Three types of VSAM data sets can have significant I/O processing performed on them during an interactive workload: the SDIDS, the SDDS, and one or more read panel data sets. Your goal in setting up your Tivoli Information Management for z/OS system for the best performance is to reduce or speed up the I/O to these data sets.

SDDS and SDIDS I/O by Type of Transaction

The SDDS and the SDIDS are usually the most important sources of I/O delays. These two data sets contain the data (problem, change, configuration) you are managing. The SDDS actually contains the information, and the SDIDS contains an inverted index pointing to all the cognized data in the SDDS.

Table 2 shows the approximate number of GETs and PUTs required for each of these data sets for several sample types of transactions. Of course, as your workload varies, the pattern of I/O required to these data sets changes. Normally, for interactive mixes, the SDIDS is the busier data set. The SDDS values in the table are for a 7-byte key. An 8-byte key produces higher counts for the SDDS, but in most cases, the net activity for the SDDS in a mixed workload does not exceed the activity of the SDIDS.

Table 2. Estimated SDDS and SDIDS GETs and PUTs

Transaction Type	SDDS Gets	SDDS Puts	SDIDS Gets	SDIDS Puts	Notes
CREATE (SMALL)	5	5	30	30	
CREATE (LARGE)	5	5	80	80	
UPDATE (SMALL)	5	5	8	8	
UPDATE (LARGE)	5	5	50	50	
SEARCH (SIMPLE)	See Note	0	2	0	Must read a full screen of SDDS search results - usually 14 to 44 records, but it depends on screen size.
SEARCH (RANGE)	See Note	0	See Note	0	Must read a full screen of SDDS search results, and must read each cognized item within range from the SDIDS. See LINECNT parameter description, page 323, for more information.
SEARCH (SORTED)	See Note	0	See Note	0	Same as for simple or range searches, plus must read all of the SDDS search results or up to all of the SDIDS sort range. See SORTPFX parameter description, page 329, for more information.
SEARCH (API)	See Note	0	See Note	0	Same as for simple, range, or sorted searches except that, before Information/Management Version 6.1, must read all of the SDDS search results (not just the displayed ones). Now, an option exists to limit search result reads from API search results list.
REPORT (SORTED)	See Note	0	See Note	0	SDIDS same as simple or range searches; must also read all the SDDS search results twice (once for sorting and once for inclusion in the report).

Table 2. Estimated SDDS and SDIDS GETs and PUTs (continued)

Transaction Type	SDDS Gets	SDDS Puts	SDIDS Gets	SDIDS Puts	Notes
SCROLL	See Note	0	0	0	Normally includes one screen of records; see LINECNT parameter description, page 323, for more information.
DISPLAY (RNID)	1	0	2	0	
DISPLAY (search results list)	1	0	0	0	

SDDS and SDIDS I/O Demand by Data Set Component

Most workloads tend to cause a lot of activity to the SDIDS. Workloads that are predominantly changes to the database can cause nearly as many PUTs as they cause GETs to the SDIDS. Because most workloads consist of a large percentage of transactions that do not change the database, they tend to cause many more reads than writes to the SDIDS.

Activity to the SDDS usually consists of many more reads than writes. Usually the SDIDS is busier than the SDDS. However, in organizations that have numerous sorted searches or reports, the SDDS can be the busier data set.

The SDDS and SDIDS are both VSAM key-sequenced data sets, so they each have a data component and an index component. This is important because substantial numbers of physical DASD reads can be avoided by optimizing VSAM buffers.

SDDS records tend to be relatively large; they probably average between 3000 and 4000 bytes each, but can in some cases be *much* larger. Most SDIDS records are very small (many are less than 50 bytes), but a few of the SDIDS records become very large for databases containing many SDDS records. The number of (and time required for) the reads and writes of the SDDS and SDIDS records is perhaps the most important performance factor for Tivoli Information Management for z/OS.

It generally takes longer to write or read larger control intervals to and from DASD (rather than cache or DASD fast write). Some transfer delay is associated with large control intervals. Smaller control intervals take less I/O time and any delays in doing GETs or PUTs are typically seek or rotational in nature. Faster I/O time is achieved by reading control intervals from cache, where delays are mostly related to transfer. The same holds true when writing control intervals to DASD (DASD fast write). The I/O time for GETs and PUTs is decreased further when reading control intervals from LSR buffers, where there are no delays and only CPU time is involved.

In a sysplex, where VSAM record-level sharing (RLS) is the method of access, the cache consists of whole control intervals. RLS caching occurs in the SMSVSAM dataspace and in coupling facility (CF) cache structures. RLS caching provides advantages similar to the use of LSR buffer pools, and it offers the additional advantage of sharing data sets across BLX-SPs in one or more systems.

Avoiding physical I/O is important, and the best way to avoid I/O delay is to not cause the I/O. You have some control over how many I/Os your Tivoli Information Management for z/OS transactions cause. Of course, some I/O is necessary, so the next best thing is to make it faster. You can do this by applying cache, DASD fast write (DFW), and, in a non-sysplex environment, LSR buffering to your data sets to your best advantage.

Figure 6 on page 61 graphs a typical pattern of I/O for the SDDS and the SDIDS. The graph shows the number of GETs or PUTs that are required for the data components of the data sets, and it shows the approximate amount of physical I/O that would be required to the index components under NSR processing.

Caching reduces the amount of actual I/O needed to satisfy GET and PUT requests. For example:

- By using an LSR buffer pool sufficient to contain the entire index component, you eliminate most of the physical I/O because most of the I/O is reads rather than writes.
- By using an LSR buffer pool that can contain many of the control intervals (CIs) in the SDIDS data component, you can eliminate a large percentage of the physical reads. Because the SDIDS is normally the first resource to saturate, reducing its utilization not only reduces response times, but it increases maximum throughput.
- Using an LSR buffer pool that contains a small number of SDDS CIs generally provides good performance. When a record is read from the SDDS, it is typically retrieved more than once while it is still in the buffer. For example, when you perform a search, the search results list is built by reading records from the SDDS. When you update one or more of the records in the search results list, the records are apt to still be in the buffer because they were so recently read.
- Using cache can cut the I/O time of some reads that are not satisfied in the LSR buffer pool.
- DFW can cut the I/O time for many writes.

In general, these recommendations also apply if you are using RLS because your goal is the same—to buffer some amount of data to achieve good performance. If you are using RLS, however, the setup is less granular. You can set up one or several large cache structures in the coupling facility instead of individual buffer pools. Follow the recommendations in the *DFSMS/MVS DFSMSdfp Storage Administration Reference* to determine the total size of your coupling facility cache structures. The cache structures should be equal to or larger than the total size of all your LSR buffer pools.

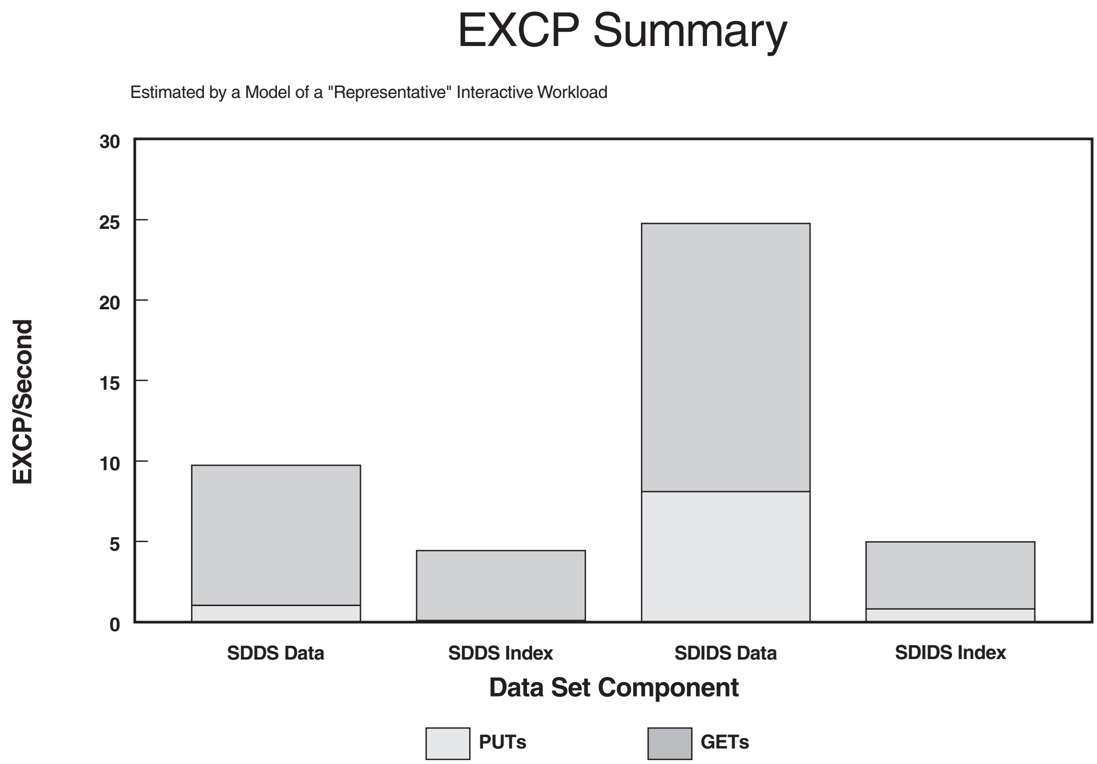

Figure 6. Typical Pattern of I/O for the SDDS and SDIDS

Spanning Records

The use of VSAM spanned records for any VSAM data set used by Tivoli Information Management for z/OS is not supported, regardless of the environment (sysplex or not). Only the 18- and 34-byte SDIDS key lengths are provided. Tivoli Information Management for z/OS creates only nonspanned records through use of the 18- or 34-byte SDIDS keys. The 16- and 32-byte keys which previously could be used with spanned records are no longer available. You may be using spanned SDIDS clusters if you are currently using a release of the Tivoli Information Management for z/OS product that existed before TME 10 Information/Management Version 1.1. If your SDIDS cluster is defined as spanned, you must remove the SPANNED keyword definition from your data sets by redefining them with the AMS (IDCAMS) DEFINE CLUSTER command. The DEFINE CLUSTER command is described in "Understanding the AMS DEFINE CLUSTER Command Syntax Description" on page 278.

Sorting Search Results Lists

If desired, the database can be kept in system-assigned record number identifier (RNID) order. Keeping the database in RNID order may reduce or eliminate the need for users to perform sorting of search results lists by RNID. By reducing the need for sorting, you can also reduce the number of I/O transactions with the SDIDS. Refer to the description of the BLGUT9 utility in the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for more information on setting up your database to keep records in system-assigned RNID order.

Sorting search results lists or reports can significantly change the SDDS and SDIDS I/O patterns. Sorting can be performed in several different ways in Tivoli Information Management for z/OS.

Reports

Reports are sorted by an external sort routine that you specify. The sort is performed by reading each SDDS record in the report, extracting the sort field data, and building a sorted table. The records are then read from the SDDS (based on the sorted table) and the report is constructed. Because sorting doubles the number of SDDS records read, sorting can significantly increase the activity of the SDDS depending on the number (and size) of the reports you submit.

Search Results Lists

Search results lists can be sorted either internally within Tivoli Information Management for z/OS or with the external sort routine that you use to sort reports. For more information on using an external sort routine with search results lists, see "Sorting with an External Sort Routine" on page 47. If you do not specify that you want to sort with an external sort routine, the following methods are used for sorting the search results lists:

- Search results lists with fewer records than defined in the SORTPFX n3 parameter of your BLGPARMS macro are sorted very much like reports. (See "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318 for information about the BLGPARMS macro and its parameters.) However, they are sorted internally by Tivoli Information Management for z/OS. If you perform many searches that result in large search results lists (but fewer than identified in SORTPFX n3, which defaults to 500), a significant number of SDDS reads may be required and can alter the typical pattern shown in Figure 6 on page 61.
- Search results lists with more records than defined in the SORTPFX n3 parameter of your BLGPARMS macro are sorted internally by Tivoli Information Management for z/OS by using the SDIDS. (See "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318 for information about the BLGPARMS macro and its parameters.) Actual sorting is not performed; the sort range in the SDIDS is read to identify the matching SDDS records. The required SDDS records are then read in sorted order and put in the search results list. If only a few thousand SDIDS reads are required, this method of sorting can be very efficient. However, if more than a few thousand reads are required, performance can be adversely impacted. See "Performance Recommendations" on page 70 for ways to avoid large range sorts. It is important for you to consider how you can avoid or minimize the impact of sorting large search results lists.

If you do need to sort search results list, be sure to read "Performance Recommendations" on page 70. Consider limiting search processing using SORTPFX parameters to end the sorting of a search results list, if it has more than a certain number of matches. Implementing training or procedural guidelines on how to limit the search criteria can help speed up sorting and prevent high read activity to the SDDS.

Panel Data Sets

The panel data sets are usually easy to keep from becoming a source of significant delays. When a user enters a transaction in Tivoli Information Management for z/OS, panels are required for control, flow, and display. These panels reside in read panel data sets and are read as follows:

- In a non-sysplex environment, the panels needed are read by the BLX-SP during normal interactive workloads and presented to the user. Because the panels themselves are not usually changed, they are accessed in read mode. When panels are accessed this way, they require a shared enqueue on the read panel data set.
- In a sysplex environment, the panels are read directly by the user. VSAM RLS gets a shared lock on an individual record when doing a read, so an enqueue is not required.

Regardless, multiple users can proceed simultaneously using the same panels, limited only by your DASD configuration.

Tivoli Information Management for z/OS also has special user buffers to store the most frequently used panels for each user, so when a panel is needed, it can be obtained without I/O. The default number of panel buffers is 50, but the actual number for each user can be changed in the session-parameters member (see description of the PNLBCNT keyword, page 327). Users who do several different types of things, but who tend to repeat those activities can benefit from additional panel buffers.

If you have sufficient panel buffers for your users, then panel I/O should not be a serious problem. You can verify that this is true by monitoring the activity to the panel data sets using measurement tools such as RMF™ or by using the Access Method Services (AMS) LISTCAT command. You can apply other techniques to supplement the benefits of the user panel buffers. For example, in a non-sysplex environment you can use larger VSAM LSR buffer pools, a controller cache, and combine data sets to reduce the overhead of concatenated data sets. In a sysplex, when sysplex mode is enabled, it is possible to use dedicated and/or larger RLS cache structures in the coupling facility. Either situation may help, but the user panel buffer is your first line of defense against panel read activity.

Putting TSPs and commonly used message panels into a separate panel data set, and placing this TSP/message panel data set as the first panel data set in the RPANELS concatenation, can improve performance. In a non-sysplex environment especially, performance can be improved if the panel data set has its own unique LSR buffer pool with enough LSR buffers to contain the entire data set. When running in sysplex mode, performance can be improved if the panel data set has a dedicated RLS cache structure for the sysplex.

Tivoli Information Management for z/OS produces a message in the BLX-SP log of the panel activity for each user that QUITs out of Tivoli Information Management for z/OS. The message shows how many panels were needed by each user and how many panels were obtained from the user panel buffers without having to make the call to VSAM. The record includes the number of buffers allocated for each user and the maximum number used. This information helps you tailor your user panel buffer sizes.

The Processor

Normally, the amount of processor power is not the most significant factor in Tivoli Information Management for z/OS performance. For example, a representative mix of interactive transactions may require only about 30 percent of an otherwise empty processor, even when the SDIDS is saturated because of enqueue delays. Tivoli Information Management for z/OS (without sysplex data sharing) can run more than double that workload (with sufficient VSAM buffering) before the processor approaches saturation. Of course, if the availability of that processor is reduced by other work, then performance is affected earlier.

On a larger processor, with an average Tivoli Information Management for z/OS workload, more processor power is available than Tivoli Information Management for z/OS can use by itself, even when the SDIDS enqueues are totally saturated. For example, the processor may be utilized only 20 percent (or more if you are using cache and DFW) even with SDIDS saturation. This leaves up to 80 percent of that processor available for other work before the processor becomes a serious bottleneck. However, even with a large or dedicated processor, the processor utilization can be much higher if the environment consists of hundreds of users and significantly large databases. A heavy use environment can include 1000 or more users of the system at any given time.

In a sysplex environment with sysplex mode enabled, even higher throughput and processor usage is possible because enqueue delays are reduced.

Of course, no matter what the power of the processor, if most of the processor is normally busy, then the processor can be the most critical resource for any application, including Tivoli Information Management for z/OS. In such a case, you can help your Tivoli Information Management for z/OS users access the processor resource by assigning them a higher priority. This action may not seriously impact lower priority users because, in most cases, Tivoli Information Management for z/OS uses the processor frequently but does not use it for long. Typically, Tivoli Information Management for z/OS uses only a small amount of the processor to initiate I/O, and then the processor is interrupted for the I/O to occur, thus freeing the processor for lower priority classes.

Estimating Processor Requirements

You can estimate the Tivoli Information Management for z/OS processor requirements for a non-DBCS, interactive workload in a sysplex or non-sysplex environment. In general, based on the sample measurements specified in Table 3 on page 65, for the workload specified, Tivoli Information Management for z/OS uses about 0.3 to 0.6 processor seconds per significant transaction assuming a single processor. This number represents total processor time. This number also includes the cost of related trivial transactions that do not access the SDDS or SDIDS. This number is not valid for very small Tivoli Information Management for z/OS workloads where the latent MVS overhead is much larger than the cost of the actual work.

To estimate the processor cost per significant transaction for a projected workload:

1. Choose a starting number between 0.3 and 0.6; use the lower number for busy workloads and the higher number for medium workloads.
2. Adjust your starting number for the relative power of the processor that you are using. If the processor is extremely powerful, the starting number is most likely lower.
3. Adjust the number again after considering how much your workload differs (in SDDS and SDIDS I/O content) from the measured workload.
4. Consider whether your workload contains relatively more or fewer trivial transactions than the measured workload. Lower the estimated cost a little if you use fewer trivial transactions per significant transaction, and increase it a little if you use more. About 25 percent of the measured workload are significant transactions.

After you estimate your processor cost per significant transaction, you can estimate the processor utilization for your workload by multiplying the cost per significant transaction by the projected throughput in significant transactions per minute and dividing by 60. If you have `n` processors, you have up to `n x 100 percent` of processor power available.

Table 3. Sample Measurements Data

Hardware	Small processor (VM first-level machine) Two second-level MVS systems with one processor and one second-level MVS system with one processor, using shared DASD
Software	MVS
Primary Users	20 - 30 TSO users
Transactions	75% trivial (panel changes during data collection, construction of search transactions, other panel flow activities resulting from INIT or BACK commands or pressing Enter or PF3) 25% nontrivial (read/writes to the SDDS or SDIDS, such as filing new or updated problem records, searches, scrolling, displaying RNID, displaying from search results list)
Conditions	No sorting, notification/escalation used, no SDLDS, no nonstandard user exits, no TSPs or SRCs, no batch reports or utilities run, no multisystem data set sharing or DBCS
SDIDS contents	About 32 000 logical records
Remote client workloads	OS/2 workstations with simple C program transactions and a Remote Environment Server (RES)

Memory

If you are migrating from Information/Management Version 4, bear in mind that additional storage is required to support the BLX-SP. Unless memory is your most precious resource, you can probably use that saved memory (and more) in a non-sysplex environment on relatively large LSR buffer pools.

When sysplex support is enabled, the storage required to support the BLX-SP will be less than in a non-sysplex environment. In a sysplex, there are no LSR buffers and less cross-memory storage is required for users.

Using the Tivoli Information Management for z/OS APIs from MVS requires a Tivoli Information Management for z/OS user address space. If the Multiclient Remote Environment Server (MRES) is used, you do not need to have a separate user address space for each client connection. A single address space can receive and process transactions from multiple Tivoli Information Management for z/OS clients concurrently.

If you are using RES instead of MRES, each RES is started by APPC in a separate address space. The Tivoli Information Management for z/OS APIs from each HLAPI client environment start at least one APPC subtask. The clients that support APPC are the OS/2, Windows NT, AIX, and CICS clients.

Some amount of paging and swapping is normal for systems that serve a large number of users. The impact on your system performance from this activity is usually not severe. The few more I/Os per transaction for paging and swapping are not usually a significant increase to the I/O that is required without paging and swapping. See “Performance Recommendations” on page 70 for advice on limiting storage use.

Other Performance Factors

The following section discusses other important performance factors you should consider.

Cognize Enqueue Prioritization (COGENQ) – Non-Sysplex

COGENQ is a parameter used in a non-sysplex environment that provides major performance benefits for file response times in busy situations. It is especially helpful in avoiding some of the performance delays related to global resource sharing with multiple-system data sets.

The COGENQ is not needed if you are running Tivoli Information Management for z/OS in a sysplex with sysplex mode enabled; it is ignored if specified in a sysplex environment. In a sysplex, individual records are serialized rather than the entire data set.

The COGENQ keyword parameter enables you to specify in the session-parameters member the number of fields to be cognized when filing a record while holding an enqueue. With the COGENQ parameter, you can prioritize your transactions by type and by user.

When a record is filed, the SDIDS is accessed many times: from 10 to 100 times for update transactions and 50 to 200 times for create, copy, and delete transactions. In an earlier version of Information/Management (Version 5.1), a file transaction held an exclusive enqueue on the SDIDS while each field was processed. This meant that a create transaction might require 50 to 200 SDIDS enqueue opportunities to complete processing.

With the COGENQ parameter, you can hold the enqueue while a file transaction cognizes multiple SDIDS fields. This is beneficial for users, such as help desk operators, who must process transactions quickly. For example, if a create transaction needs to cognize 100 fields in a record and COGENQ is set to 10, it would only require 10 enqueue opportunities instead of the 100 required without the COGENQ parameter. During busy periods, this enhancement gives this create transaction a powerful advantage over other transactions. Other users who occasionally file a record do not need this prioritization. For example, it may not matter that an API that is updating or deleting large blocks of records has a longer response time. The COGENQ parameter gives users the ability to control the SDIDS update speed on a session-parameters member basis.

With COGENQ set to 0, a file transaction holds the enqueue for the duration of all cognize and uncognize activity. Any file activity conducted had maximum priority, but at the expense of competing users. With COGENQ set to 1, a file transaction processes as it did with Information/Management Version 5.1, releasing the enqueue after one field is cognized. This provides maximum interleaving and is ideal for batch processing because of the low priority.

With COGENQ set to n, the priority of the file activity increases as the value of n increases. However, values greater than 50 are essentially equivalent to a setting of 0. With COGENQ set to the default value of 10, most of the interleaving benefits are preserved, even if all users stay at the default, and file transactions are prioritized without a serious delay to other transactions.

The net effect on response times varies with workload and volume. During light loads, increasing the value of COGENQ for some users does not make much difference in performance. It is only in situations where enqueue contention is an important part of the response time that COGENQ values other than 1 help. However, setting COGENQ to 0 or a number larger than 1 can dramatically shorten the response time for the file transactions in situations where the SDIDS enqueue is very busy and the workload includes a mixture of transaction types (some short and some long).

You must manage the priorities, using COGENQ (and SRCHLIMIT), to your organization's best advantage. As a general rule, a SRCHLIMIT value of 100 and a COGENQ value of 10 (the default) work well. You can then modify values for specific users to change their relative priorities for file transactions. Special rules apply if you are sharing data sets (see "Performance Recommendations" on page 70).

Multiple-Cluster SDDS

Tivoli Information Management for z/OS enables the SDDS for a single Tivoli Information Management for z/OS format database to optionally consist of 1 to 100 VSAM clusters.

Another benefit to using a multiple-cluster SDDS is that, while the multiple clusters are treated as a single logical SDDS, they can cut I/O contention for access in either a sysplex or non-sysplex environment, especially if the multiple clusters reside on different DASD.

Usually, because the SDDS is not as busy as the SDIDS, this does not result in a large reduction in response times. In some workloads (for example, a very high percentage of searches, scrolls, or reports compared to creates and updates), the SDDS is the most active data set. In these cases, the reduction in response time can be significant, and maximum throughput can be improved. Whether it significantly improves your performance depends on your particular workload.

Multiple-Cluster SDIDS

Like the SDDS, the SDIDS, which contains an index to the records stored in the SDDS, can consist of 1 to 100 VSAM clusters. The primary benefit of using a multiple-cluster SDIDS is that you can divide your records based on index key contents (for instance, you can separate s-words from p-words). In a non-sysplex environment, you can reduce the amount of data that is locked when the SDIDS is accessed, since the clusters can be accessed in parallel. When sysplex mode is enabled, however, enqueue contention is not an issue because RLS gets a shared lock on an individual record (rather than an entire data set) when doing a read. A multiple-cluster SDIDS can be useful to help improve your overall database performance, especially if you store significant amounts of searchable data.

For information on setting up a multiple-cluster SDIDS, see "Working with SDIDSs" on page 176.

Multiple BLX-SP Support

You can have more than one BLX-SP on a single z/OS system. This can be convenient for organizational or migration reasons. Unless you are migrating from a previous release of the product, or you have very unusual needs, you should not need to have more than one BLX-SP on the same processor. A single BLX-SP can support over 1500 users and hundreds of databases. Therefore, it is generally better to have one BLX-SP with a large region size than several smaller (region size) BLX-SPs.

Performance should not be an issue if multiple BLX-SPs share access to the same Tivoli Information Management for z/OS data sets through sysplex data sharing. In previous releases of the product, the sharing of data sets through Multisystem Database Access could have had a significant negative impact to performance. With sysplex data sharing, there will most likely be no noticeable performance impact if BLX-SPs on the same or different systems share data sets. The need to share data sets between BLX-SPs on the same system is usually unnecessary, however, because a single BLX-SP can support many users and databases.

Sysplex Data Sharing

Before Tivoli Information Management for z/OS Version 7.1, users on different systems could share access to a Tivoli Information Management for z/OS database through a feature called Multisystem Database Access (MSDA). Currently, data sharing is supported only through use of a parallel sysplex. The MSDA feature is no longer supported.

Tivoli Information Management for z/OS not only provides the capability for multiple BLX-SPs, it also enables those BLX-SPs to share access to common Tivoli Information Management for z/OS data sets. Data sharing is supported as long as all sharing BLX-SPs are on z/OS systems running in the same parallel sysplex, and Tivoli Information Management for z/OS is running with sysplex support activated. If you decide to use sysplex data sharing, be sure to read “Setting Up for Sysplex Data Sharing” on page 151.

Factors that can influence the performance during sharing include:

- The number of changes that a transaction makes to VSAM data sets.

  Transactions that file new records, update records, copy records, or delete records are normal sources of high VSAM change rates. Transactions of these types may individually take a long time if they do an unusually large number of changes. In a non-sysplex environment, if a workload has an unusually high percentage of these transaction types (even if individually they are small changes), the net impact can overutilize the SDIDS enqueue, and thus affect performance of all users. Use the COGENQ parameter in this situation to help reduce the number of SDIDS enqueues. In a sysplex, the SDIDS enqueue contention problem does not exist, so there is no need to use the COGENQ parameter.
- The number of reads for some transactions. Large reports, some sorting of search results lists, and large API searches are all examples of things that can cause very large numbers of reads.
- The number of sharing systems (or LPARs) in the global network.
- In a GRS ring, the settings of global resource sharing parameters such as RESMIL and ACCELSYS. A GRS star can also provide better performance than a GRS ring.
- The number of RLS cache set and cache structures, and the size of those structures. Generally, performance is better if you have more cache structures and larger cache structures because more data can be cached.

SRCHLIMIT Function

The SRCHLIMIT parameter of the BLGCLUST macro enables you to define the enqueues (in a non-sysplex environment) and searching of the SDIDS. (This parameter, introduced in Information/Management Version 6.1, replaced the SRCHENQ parameter available in Information/Management Version 5.1.) This parameter enables users to specify how many SDIDS reads to do under each enqueue during range reads. Range reads of the SDIDS are necessary during search processing (including search portions of reports) when search arguments specify ranges of values either explicitly or with a period (.) or asterisk (*) character in the arguments. SRCHLIMIT prevents individual search transactions from dominating the SDIDS enqueue and locking out other users who needed the SDIDS for their transactions. It also releases the SDIDS enqueue during range reads of the SDIDS caused by sorts of search results lists that had more than SORTPFX N3 matches.

The following subparameters are available:

- ENQUE specifies the number of records that a search or report can read from the SDIDS before the SDIDS is released to other users. This parameter replaces the SRCHENQ parameter and applies only to a non-sysplex environment. The enqueue parameter is ignored when sysplex mode is enabled because dataset enqueues are not needed.
- SRCHWARN enables users to specify a number. If a search or sort requires range reads of the SDIDS and the number of reads required exceeds this number, then the user who submitted the search receives a warning that the search caused an excessive number of SDIDS range reads. The search or sort is not interrupted, but the user is informed at the completion of the transaction that the search or sort exceeded the warning limit for SDIDS reads. This can be an important new tool in helping to identify and avoid transactions that misuse search or sort arguments. The SRCHWARN parameter is used in either a sysplex or non-sysplex environment.
- SRCHEND stops any search or sort that exceeds the specified number of SDIDS range reads. This provides an additional level of protection against misuse of search arguments or sort specifications that impair performance for everyone. The SRCHEND parameter is used in either a sysplex or non-sysplex environment.

Remote Client File Transactions

Remote client applications perform similar to host API transactions, but are associated with an increase in end user response time consistent with workstation communications and processing. The cost of the client platform and APPC or TCP/IP related portions of the transaction varies with the speed and availability of the machine running the client platform and with the host priorities and resources supporting APPC or TCP/IP.

However, based on measurements that were taken, it was found that these costs need not be great. On even a moderately powerful workstation processor, the increase may be very small for a workload that consists entirely of record creation transactions, even when the application is in the background rather than the foreground. Concurrent workloads in other tasks under client platforms may only moderately increase response time unless a completely processor-bound task is allowed in the foreground.

Search and retrieve transactions generally require additional processing time on the workstation because of their larger transmission and data handling requirements. Such transactions have longer response times with a remote client application than with a host API. Even though increases in response times occur from remote client applications, there may be no increase in resulting enqueue contention on the host.

Performance Enhancements

The following section briefly introduces recent changes and improvements that can influence performance in this version of Tivoli Information Management for z/OS. In most cases, the changes provide significant performance advantages, but some new features add additional workload or otherwise influence performance.

Support for Parallel Sysplex

If you run Tivoli Information Management for z/OS in a sysplex, you can take advantage of VSAM RLS. The use of RLS should significantly improve the performance of Tivoli Information Management for z/OS, especially if you are sharing databases (VSAM data sets). Because RLS enables a more precise lock on data in a data set (data is locked at the record level rather than as a data set enqueue), you can achieve a greater interleaving of user transactions with the database than if you used VSAM NSR/LSR. This locking granularity provides advantages in both shared and nonshared database environments. RLS also provides a common cache for the sysplex in the coupling facility, which provides a performance boost over the cross-system buffer invalidation used by Multisystem Database Access in previous releases of the product. For more information about planning for and installing Tivoli Information Management for z/OS in a sysplex, refer to “Setting Up for Sysplex Data Sharing” on page 151.

Performance Recommendations

The following items are provided as a quick check to help you verify that the most important performance factors are considered. Although individual situations may vary, the items that are usually the most important to performance are earlier in the list. These items are not fixed rules for all your Tivoli Information Management for z/OS implementations but are simply important things to consider when performance is important. A more detailed discussion of the performance aspects of many of these recommendations can be found earlier in this chapter.

General Recommendations

Regardless of how you run Tivoli Information Management for z/OS (sysplex or non-sysplex), note the following recommendations:

- **Use cache and DFW for the SDDS and the SDIDS if possible.** While these do not reduce the number of execute channel programs (EXCPs), they can sharply reduce the time it takes to complete them. Cache and DFW can be important ways to reduce enqueue contention.
- **Eliminate most of the physical I/O to your read panel data sets.** Tivoli Information Management for z/OS offers two ways to do this—user panel buffers and VSAM buffers.

  Note: The term VSAM buffers can mean either LSR buffers if you are running without sysplex support enabled, or RLS cache if you are running in a sysplex environment. The concepts related to caching of VSAM data are the same, however.

  The number of user panel buffers can now be specified in the session-parameters member. These buffers enable users to retain their most frequently accessed panels in compressed form in user storage. If the needed panel is in the buffer, then the user’s session does not have to request a panel from the VSAM buffer. If a user needs a panel that is not in the user panel buffers, then the user’s session must request a panel from VSAM. VSAM does not need to physically get the panel if it is already in the VSAM buffers. You can tune your panel buffers for the best balance for your situation. User buffers are faster and use less processor time but more storage. VSAM buffers use a little more processor time, but less storage, because all users can share them. The most important thing is to minimize the physical I/O for your panel data sets by one or both methods. The method you choose is mostly a function of whether storage or the processor is more important to your situation.
- **Eliminate any unnecessary cognizing.** You can use PMF to specify whether a field is cognized. Every actual content for every cognized field results in an SDIDS record which points to the related SDDS records. Whenever an SDDS record is created, updated, copied or deleted, all affected SDIDS records must be read in, changed, and rewritten.

You can eliminate much of this I/O by using PMF to tell Tivoli Information Management for z/OS not to cognize fields that you do not need to use as search criteria. This can greatly reduce the I/O and enqueue contention for the SDIDS.

You can eliminate cognizing s-words while continuing to cognize related p-words. This valuable method cuts I/O delays on file transactions.

- **Place your key Tivoli Information Management for z/OS data sets on volumes that minimize contention for the volumes and their controllers.**
  - This is very important for the SDIDS. Isolate the SDIDS on its own volume or place it on a volume that has very little other activity. Just like the SDDS, the SDIDS can be spread over multiple volumes. If possible, spread the SDIDS across several low contention volumes so that shared accesses from multiple users can proceed simultaneously. This becomes more valuable in workloads with a high percentage of read activity compared to write activity.
  - Usually, the SDDS is not as busy as the SDIDS, but it is still advisable to make sure that contention is minimized by isolating and spreading the SDDS over low contention volumes, if possible.
  - The panel data sets are usually not as critical here because most I/O to them is avoided by using user panel buffers or VSAM buffers.
- **Put Tivoli Information Management for z/OS into a link pack area if possible so that all users can share the common code.** This is especially important where memory is limited and where many Tivoli Information Management for z/OS users are active.
- **Avoid using STEPLIBs and JOBLIBs whenever possible.** These may be necessary for some users, but for users who want the best performance, avoiding these can improve performance.
- **Minimize the number of data sets concatenated in your STEPLIBs, JOBLIBs, and link lists.** Searching these directories can significantly delay some transactions. Keep the concatenations as short as possible and order them in such a way that frequently accessed items can be found early in the concatenation sequence. Keeping concatenations short is especially important when you use the link pack area for sharing common code. These directories must be searched before it is determined that the needed item is in the link pack area.
- **Consider giving your Tivoli Information Management for z/OS users and the BLX-SP a relatively high processor dispatching priority.** Because Tivoli Information Management for z/OS processor needs are interrupted by frequent I/O and enqueues, the impact on lower priority users should be modest, and it should help your Tivoli Information Management for z/OS users complete their I/O and release their enqueues.
- **Avoid transactions that must read hundreds or thousands of SDDS records**. Such transactions can include large API searches or large reports.
- **Avoid large range searches and large range sorts wherever possible.** Large range searches result when search arguments specify, either explicitly or with periods (.) or asterisks (*), ranges that require many thousands of SDIDS records to be read. Sorting of search results lists by any field containing thousands of different values can also cause large numbers of SDIDS records to be read in some cases. Avoiding such inappropriate searches or sorts can dramatically improve the performance of the search or report transactions, and it can also reduce the contention for the SDIDS, thus reducing response times for other transactions. See “SRCHLIMIT Function” on page 68 for more information.

- **Consider not reusing SDDS position numbers.** By not reusing SDDS position numbers, you can avoid having to sort the search results list to display system-assigned RNIDs in order. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information on storing records in system-assigned RNID order.
- **Use pre-started API sessions for faster API session initialization.** If you are using client API applications with Tivoli Information Management for z/OS, you can take advantage of an MRES parameter to pre-start API sessions. Using pre-started API sessions, clients may achieve a much faster response time when initially connecting to Tivoli Information Management for z/OS through the MRES. If you have client applications that typically connect to the MRES using the same control parameter data block values for the HL01 initialization transaction, you should consider setting up the MRES with pre-started API sessions. For more information on pre-starting API sessions, refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide*.
- **Do not add Tivoli Information Management for z/OS resource names in the inclusion or exclusion lists in the GRSRNLnn member in SYS1.PARMLIB.**

  Whether or not you use Tivoli Information Management for z/OS in sysplex mode, or global resource serialization, there should be no Tivoli Information Management for z/OS resources in the resource name lists (RNLs) for global resource serialization. If you inadvertently specify resource names for inclusion or exclusion and use the same resource names as are currently used by Tivoli Information Management for z/OS, poor performance or data corruption can result. See "In a Sysplex with Sysplex Mode Enabled" on page 73 for other recommendations on what to do in a sysplex environment for global resource serialization.

  Note: The details of INCLUDE and EXCLUDE can vary for alternative serialization products. Check the rules for your serialization tool to avoid unnecessary global enqueues.

  For a list of resource names used by Tivoli Information Management for z/OS, see "Resource Names That Tivoli Information Management for z/OS Enqueues On" on page 351.

In a Non-Sysplex Environment

- Make sure your VSAM buffer specifications are reasonable:
  - **Specify sufficient LSR buffer pools to eliminate all physical reads of the index components for the SDIDS, SDDS, all read panel data sets.** You can determine how many CIs are required by looking at output from the IDCAMS LISTCAT command. Divide the high used relative byte address (RBA) of the whole index component (not one of the extents) by the CI size for the index to determine the total number of CIs. For the SDDS and SDIDS, add enough buffers to the total to accommodate reasonable growth. (Usually 10 percent provides plenty of growth without undue memory use.)
  - **Specify as many SDIDS buffers for the SDIDS data component as you can afford.** The SDIDS consists of mostly very small records, with a relatively high read activity. Because the SDIDS usually has the highest activity of all the Tivoli Information Management for z/OS data sets, reducing the time it takes to complete the reads directly improves not only individual response times, but overall throughput.

Although the percentage of reads found in the buffers varies with the workload, you might find it possible to obtain, for example, a 60 percent hit ratio even with only 20 percent of your CIs in buffers. Because the total size of the SDIDS data component is relatively small, and it does not increase rapidly as the SDDS grows, you may find it to your advantage to buffer quite a lot of it.

- **Specify a relatively small number of buffers for the SDDS data component.** The records are usually large, and there are many of them, so buffering too many can be counterproductive. But, because the access is not totally random, having a few (1 percent or so) probably yields a match ratio that is well worth the memory. You may find that even 1 percent of the CIs is too much memory to allocate for very large databases. In this case, consider allocating at least 100 buffers for the SDDS data component of any performance critical databases.

  Many transactions read several SDDS records. For example, a search or scroll usually reads a full screen of records (usually between 14 and 44). You might consider specifying enough SDDS buffers to enable several users to be searching or scrolling without causing other short term transactions to lose the benefits they enjoy from buffering. This suggestion could probably be satisfied by specifying at least enough SDDS buffers to equal five times your screen-size.

- **Make sure your enqueue interleaving is specified in the most efficient manner.**
  - Set the SRCHLIMIT parameter to release the SDIDS enqueue frequently during range reads. Usually, setting the *enque* subparameter to 100 provides the best performance. You must specifically set this parameter to obtain the enqueue interleaving you want because the default enables range reads to dominate the enqueue. Do not set the *enque* subparameter to less than 100 because very low values cause significant additional I/O.
  - Tune your file priorities with the COGENQ parameter. This parameter enables you to greatly improve file response times for selected users by enabling them to cognize more affected records under each SDIDS enqueue. COGENQ defaults to 10 for all users, but consider increasing it for users who need fast file response times and decreasing it for users whose file response times are less important.

- **Try setting up a multiple-cluster SDIDS to obtain more concurrent processing of enqueues.** For customers with large amounts of data, multiple-clusters can offer a real bonus in database performance. For a single system database, minimal CPU overhead should be expected with the additional enqueue/dequeue activity. Searchable words are cognized in sorted order, so no more than one enqueue is acquired and released for each SDIDS cluster when a record is filed. The same is true for sequential reads of the SDIDS when records are searched. (This only applies to a non-sysplex environment.) Multiple clusters can also reduce I/O delay for both sysplex and non-sysplex environments, if the data sets reside on different volumes.

In a Sysplex with Sysplex Mode Enabled

- **Use a GRS star rather than a GRS ring**: It is recommended that you use a GRS star for a parallel sysplex because a GRS star generally provides better performance than a GRS ring. If you must use a GRS ring, set the ACCELSYS and RESMIL parameters as follows. If you are using a GRS star, the recommendation for the RESMIL and ACCELSYS parameters do not apply.

  If you are using a GRS ring:

ACCELSYS Set the ACCELSYS under GRS (or equivalent) to an appropriate value that speeds up global enqueue processing.

RESMIL Set the RESMIL parameter in GRS (or similar parameter if you are using another global resource serialization tool) as low as you can afford. Tivoli Information Management for z/OS uses global enqueues and record-level locks to protect the integrity of its data sets while multiple systems access them. Low RESMIL values speed up the global enqueue process but cost extra processor power. You must tune this parameter to make this trade-off in your best interests.

- **RLS cache sets and sizes:** The VSAM recommendation is that the total size of the cache structures should be at least equal to the total size of all the LSR buffer pools that were in use for the same set of data sets. Greater performance can be achieved by increasing the size of the structures to allow more buffering to occur. Additionally, all cache structures should be defined with an initial size of no less than 2MB. For more information, refer to the *DFSMS/MVS DFSMSdfp Storage Administration Reference*.
- **BLX lock structure size:** The IXCMIAPU sample job provided with Tivoli Information Management for z/OS shows the size of the BLX lock structure you can use.

  A formula that will help determine the size of the BLX lock structure can be found in “Setting Up for Sysplex Data Sharing” on page 151.
- **IGWLOCK00 size:** To estimate its size requirements in megabytes, use the following formula (note that a megabyte is 1 048 576 bytes in this case). This formula is described in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*. Refer to that manual for more details on calculating the IGWLOCK00 size.

  ```
  10M * number_of_systems * lock_entry_size
  ```

  In this formula, `number_of_systems` is the number of systems in the parallel sysplex, and `lock_entry_size` is the size of each lock entry. This value depends on the MAXSYSTEM value that is specified to the IXCL1DSU Couple Data Set format utility.

  To determine the actual lock entry size for the different MAXSYSTEM setting values, refer to the description on how to define the CF lock structure in the *DFSMS/MVS DFSMSdfp Storage Administration Reference.*
- **VSAM extended format data sets:** RLS and Tivoli Information Management for z/OS support extended format data sets, including compressed data sets.
- **XCF:** Tivoli Information Management for z/OS sends messages with a maximum length of 256 bytes and has very low message traffic. Choose an XCF transport class accordingly.
- **GRS:** Some Tivoli Information Management for z/OS resources are enqueued with a scope of SYSTEMS when operating in sysplex mode. (See “Resource Names That Tivoli Information Management for z/OS Enqueues On” on page 351 for a list of resources and their scopes.) If Tivoli Information Management for z/OS is running on one system only, consider adding these resource names to the SYSTEMS Exclusion RNL to reduce enqueue delay. You must remove these entries if you run Tivoli Information Management for z/OS on more than one system.

5

Using VSAM Resources in a Non-Sysplex Environment

Important

The information in this chapter applies to you only if you are using or intend to use Tivoli Information Management for z/OS in a non-sysplex environment. When you install Tivoli Information Management for z/OS in a sysplex and enable sysplex support, VSAM resources are not defined in the BLX-SP VSAM resource definition member to support nonshared resources (NSR) or local shared resources (LSR) as described in this chapter. Instead, coupling facility structures are defined to support VSAM record-level sharing (RLS). Where possible, you should install Tivoli Information Management for z/OS in a sysplex to take advantage of the improved performance and simplified administration benefits that a sysplex offers. For information about how VSAM resources are used in a sysplex, see “Setting Up for Sysplex Data Sharing” on page 151.

This chapter discusses how Tivoli Information Management for z/OS uses VSAM resources. It provides:

- An introduction to VSAM resources in Tivoli Information Management for z/OS
- An explanation about how VSAM resources are defined in a BLX-SP environment
- Recommendations on when to use LSR and when to use NSR
- An example scenario for defining LSR buffer pools
- A VSAM buffer pool definition checklist

This chapter assumes that you have a working knowledge of the following:

- VSAM shared and nonshared resources. Refer to *DFSMS/MVS Using Data Sets* for details.
- The VSAM BLDVRP macro, which builds VSAM resource pools. Refer to *DFSMS/MVS Macro Instructions for Data Sets* for a description of this macro and its syntax.

This chapter is designed to help you:

- Understand how VSAM resources are used by Tivoli Information Management for z/OS in a non-sysplex environment
- Plan your Tivoli Information Management for z/OS environment

Using VSAM Resources for Tivoli Information Management for z/OS Data Sets

VSAM uses buffers and control blocks for its input/output (I/O) operations. You can use VSAM LSR and NSR for Tivoli Information Management for z/OS VSAM data sets. However, you cannot use VSAM global shared resources (GSR) because the BLX-SP environment does not support them.

You can share VSAM resources among data sets by building resource pools.

Defining VSAM Resources in the BLX-SP Environment

VSAM resources for Tivoli Information Management for z/OS are defined in the BLX-SP VSAM resource definition member and allocated in the BLX-SP. The shared resource pools that are allocated are exploitable for use by all users' address spaces connected to the BLX-SP. You can also define multiple shared resource pools in the BLX-SP environment.

The VSAM resource information that you must define to the BLX-SP are:
- The number of VSAM placeholders required for NSR
- The VSAM data sets that use VSAM NSR
- The VSAM LSR pools
- The connection of the VSAM resources to the VSAM data sets
- The type of the VSAM data set (numbered or key sequenced)

You can use the Installation Tailoring Facility to create your VSAM resource definition members. See "Using the Installation Tailoring Facility" on page 185 for information on using the Installation Tailoring Facility.

Note: LSR keyword specifications (BLGPARMS and BLGCLDSN macros) and the shared resource pool definitions (BLDVRP macros) contained in a Tivoli Information Management for z/OS session-parameters member are ignored if specified.

Using Resource Pools

Define separate resource pools for the following Tivoli Information Management for z/OS VSAM data sets:
- RPANLDS
- SDDS
- SDIDS
- SDLDS (if used)

If your installation has heavy online report activity, a separate resource pool can be defined for the dictionary (DICTDS). If the majority of the report activity is performed off hours, the dictionary can share the SDDS resource pool. If you decide to use LSR buffering for one or more of your SDLDSs, define a separate resource pool for them. All of the SDLDSs can share the same resource pool.

If your installation plans to use multiple databases, define separate resource pools for each SDDS and each SDIDS used, unless activity in those databases is isolated by time or shift, or performance is not a concern for the databases sharing the pools.

Note: Only 31 key-sequenced data sets can share the same resource pool.

Using VSAM Placeholders

When LSR pools are used, the number of VSAM placeholders is specified as part of the shared resource pool definition. The number of placeholders is the value specified for the STRNO keyword of the BLDVRP macro, which defines the shared resource pool.

When VSAM data sets use NSR, it is necessary to provide the BLX-SP with the number of placeholders required for the data sets. When NSR is used, the number of placeholders required for all VSAM data sets that use NSR is defined by the value specified for the PLACES keyword in the BLXNSR macro.

VSAM limits the number of placeholders specified for a data set or shared resource pool to a value of 255. The number of placeholders required is not the number of active users' address spaces that can access the data sets connected to the resource pool, but the number of positions in the resource pool that VSAM must maintain at any given time. Because only one user's address space is allowed write access to a data set at a time, this number should reflect the number of concurrent read accesses that can occur at any given time. It is highly unlikely that all users' address spaces would attempt to read from data sets connected to the *same* resource pool at the same time.

If you do not specify enough placeholders, the user must wait until a placeholder is freed for use. If you are competing with only yourself for placeholders, you can encounter a deadlock situation. Consider this: you have a resource pool with ninety placeholders and you have a multiple-cluster SDDS database sharing the same resource pool. If an attempt is made to use BLGUT7 to copy the multiple-cluster SDDS database, deadlock occurs when BLGUT7 attempts to open the ninety-first data set because all the placeholders are being used.

Notes:

1. For a data set that is write-only, such as the SDLDS, you can use fewer placeholders because only one user is allowed access at a time.
2. You should always allow at least one placeholder (at a minimum) per data set in an LSR buffer pool.
3. One additional placeholder is needed for VSAM use to process CI and CA splits.

   For example, assume a 5-cluster SDDS:

   ```
     5  (one for each SDDS cluster - #2 above)
     1  (reserved for VSAM - #3 above)
   ____
     6  (MINIMUM to avoid deadlocks)
   ```

   Twenty-four (four times this number) would be recommended to avoid the need for placeholder waits. Refer to the use of the QUERY command, described in the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for more information on displaying VSAM statistics such as placeholder users and the number of placeholder waits.

Recommendations

We recommend that you use LSR for most key-sequenced VSAM data sets being accessed because this results in better performance. To get the best performance improvements in Tivoli Information Management for z/OS, it is strongly recommended that you use LSR for at least the SDDS and the SDIDS. See "When You Can Use LSR" on page 78 for further details.

The use of NSR is more efficient for relative record data sets, such as the SDLDS, because of the way Tivoli Information Management for z/OS uses these data sets. LSR does not usually improve the performance of the SDLDS. See "When You Can Use NSR" on page 79 for further details.

When You Can Use LSR

Use the LSR keyword in the BLXDSN macro to specify LSR for a data set. The value you specify for the LSR keyword is the identifier (ID) of the shared resource pool to connect to the data set. LSR is especially attractive for key-sequenced VSAM data sets in the BLX-SP in that all connected users' address spaces share in its benefits without the added address space storage requirement incurred with LSR pools.

The use of LSR in the BLX-SP can significantly improve the performance of users' address spaces connected to the BLX-SP. Therefore, it is strongly recommended that you use LSR with Tivoli Information Management for z/OS in a non-sysplex environment. The LSR facility of VSAM permits data buffers and control blocks to be shared among open VSAM data sets to reduce the amount of storage required to access the data sets individually. The use of LSR also enables VSAM to maintain a part of a data set in storage, which can reduce physical I/O to the data set. In the BLX-SP, use LSR to reduce physical I/O, but not to reduce the storage requirements of the BLX-SP. Sharing buffer pools of the same size between VSAM data sets can adversely affect the benefit of using shared resources. Because the order of VSAM data set access can vary according to the function performed, use separate buffer pools within a shared resource pool for each VSAM data set. For data sets used primarily as read-only (such as a panel data set), the ideal situation is to have a buffer pool defined large enough so that only one physical read operation is necessary to access a record and to enable the record to remain in the buffer pool for subsequent accesses. You can exploit this reduction of I/O when buffer pools are large enough to satisfy a working set.

VSAM requires LSR use for cross-memory support. This enables access to a BLX-SP VSAM data set under control of the user's address space task. When NSR is specified for data set access, VSAM requires that all requests be issued by a BLX-SP task. All requests for NSR data sets are serialized through one BLX-SP task. Consequently, LSR is required for multithread processing. The resources required for an LSR pool are called a working set.

Working Set

The working set for each VSAM LSR pool and each component within the pool must be determined to make efficient use of LSR. The primary factor that determines your working set for an LSR pool is the number of concurrent accesses of the data sets connected to the pool. This factor also determines the number of placeholders required for the LSR pool. The working set is also influenced by the number of CIs that are to be maintained in storage to make efficient use of LSR by the data sets connected to the LSR pool. These two factors determine the size of the buffer pool that must be allocated. Each key-sequenced VSAM data set consists of two components, the data component and the index component. Working sets need to be established for each component of the data set.

Index Component Working Set

The number of CIs in an index component is small relative to the data component. The working set for an index component is the number of CIs in the index component. You should define buffer pools in your LSR definition that are large enough to contain the index components for each key-sequenced VSAM data set assigned to this shared resource pool. A shared resource index pool should be defined for the shared resource pool to avoid buffers defined for index CIs being used by data component CIs of the same or smaller CI size. If this is a read/write data set, you should make allowances in the buffer pool for additional index CIs that are created as new records are added to the data component of the data set.

Data Component Working Set

Determining the buffer pool size required for the data component of a data set is not as straightforward as that of an index component. The working set of a data component depends upon the contents of the data set and the use of the data set. The minimum buffer pool size that can be allocated for the data component of any VSAM data set is the number of positions that are to be maintained in the shared resource pool plus one.

Read/write data sets that are randomly accessed, such as the SDDS and the dictionary, reap small benefits, if any, when a buffer pool larger than the minimum size is allocated for their data components. The primary advantage gained by the use of LSR for these data sets is having the index components resident. Some benefit can be derived by buffering small percentages of the data component because a few CIs exhibit high frequencies of use and are likely to be found in even a small buffer pool.

A high-activity data set benefits when a buffer pool larger than the minimum size is allocated for the data component. Because the SDIDS is the data set with the highest activity, performance gains should be enhanced by having a buffer pool defined that enables a large portion of the data component to remain resident.

Panel data sets can benefit when a buffer pool larger than the minimum size is allocated for the data component. A complex analysis of panel flows and panel sizes is required to determine the additional storage working set above that required for concurrent user access. In most cases, it is beneficial to use panel buffering in the user’s address space rather than attempt to use LSR in the BLX-SP to eliminate panel I/O. Still, with a reasonable number of CIs buffered, LSR can be a good second tier defense against physical I/O. If you have limited storage, you may find that decreasing the number of panel buffers and using LSR buffers as backup buffers can save enough storage for some users to make it a good trade-off with respect to I/O delays. You may need to experiment many times before making this complex tuning decision.

When You Can Use NSR

The situations when you do not want to use LSR are due to performance considerations only, and do not affect the normal operation of users’ address spaces or the BLX-SP. Using NSR is more efficient for relative record data sets, due to the manner in which Tivoli Information Management for z/OS uses these data sets.

In a non-sysplex environment, you can use NSR for any Tivoli Information Management for z/OS VSAM data set. You can choose to use NSR when any of the following conditions exists:
- Single task access serialization does not affect performance
- VSAM data set access is low
- Using LSR for a data set makes no measurable difference in the performance of the users’ address spaces
- A VSAM data set is part of a test or development database
- A VSAM data set is only used as a write panel data set

Notes:

1. We recommend that you use NSR only if one or more of these conditions exists. Otherwise, using NSR can severely degrade performance. To enhance performance, we recommend that you use LSR.
2. LSR cannot be used when a VSAM data set is accessed in load (create) mode. NSR is automatically used when Tivoli Information Management for z/OS utilities accessing data sets in load mode are running. If LSR was specified for the data set, the data set is reaccessed using LSR when the utility finishes.

The following special conditions that restricted using LSR for a VSAM data set prior to Information/Management Version 5.1 do not apply in the BLX-SP environment. This is because you now define the shared resource pools in the BLX-SP instead of the user's address space.

- Logging on after address space abnormally ends
- Using the ISPF split-screen facility
- Coexisting with other programs using LSR
- Using the NetView Hardware Monitor Interface

Defining LSR Buffer Pools

This section gives you an example of how you can define your LSR buffer pools if you decide to use LSR for your Tivoli Information Management for z/OS applications.

When a VSAM data set is opened for which you requested LSR, VSAM connects the data set's control blocks with a buffer pool of the appropriate size. Both the index and data components of each key-sequenced data set must be taken into account when defining buffer pools within a shared resource pool. If buffers of the proper size do not exist in the shared resource pool definition, VSAM uses the next larger buffer size defined in the shared resource pool. In addition, VSAM uses a larger buffer when all buffers of the proper size are in use.

The required sizes of the buffers are determined by the CI sizes of the data sets for which you want to use LSR. The number of buffers assigned to a buffer pool is determined by the working set required for the data set. The two factors of the working set size are the number of concurrent users and additional storage required to enhance performance. If you already defined your data sets, you can determine their CI sizes by using the Access Method Services (AMS) LISTCAT command for each data set.

If you did not define your data sets yet, or if you plan to redefine your data sets, you may want to specify certain CI sizes using the Installation Tailoring Facility. See "Using the Installation Tailoring Facility" on page 185 for information on using the Installation Tailoring Facility. If you change CI sizes, make sure you also change your LSR definition to reflect the changes of the CI sizes. Define CI sizes to match the requirements of the data set and not to eliminate CI size conflicts among key-sequenced data set components or their working sets. The analysis utilities—BLGUT20, BLGUT21, and BLGUT22— can assist you in determining the most efficient CI sizes for a VSAM data set.

Plan to define separate LSR pools for each SDDS and SDIDS. Read panel data sets can share a common LSR pool. Define both an index and data pool for the LSR pool for key-sequenced data sets. The number of buffers of a particular size that you choose is determined by the working set of each component for the data sets for which you want to use LSR.

Figure 7 shows how you can use the AMS LISTCAT command to collect data on allocated data set sizes.

In this example, two Tivoli Information Management for z/OS databases are defined, one for production and the other for test and development. The production database includes a log, while the other one does not. Two panel data sets are defined for both production and test and development use, one containing Tivoli (IBM) panels and the second contains your modified panels. In addition, another panel data set is defined solely for test and development use. Two write panel data sets are defined for PMF-authorized users.

The data set information that the AMS LISTCAT commands generate is used to create the LSR definition worksheet shown in Table 4.

```
LISTCAT ENTRY('BLM.IBMBPNLS')     ALL  /* Base Panel Data Set        */
LISTCAT ENTRY('BLM.READ.PANELS')  ALL  /* Modified Panel Data Set    */
LISTCAT ENTRY('BLM.JILLS.PANELS') ALL  /* Write Panel Data Set       */
LISTCAT ENTRY('BLM.DICT')         ALL  /* Dictionary Data Set        */
LISTCAT ENTRY('BLM.SDDS')         ALL  /* Production SDDS            */
LISTCAT ENTRY('BLM.SDIDS')        ALL  /* Production SDIDS           */
LISTCAT ENTRY('BLM.SDLDS')        ALL  /* Production SDLDS           */
LISTCAT ENTRY('BLM.TEST.PANELS')  ALL  /* Test/Development Panels    */
LISTCAT ENTRY('BLM.JACKS.PANELS') ALL  /* Write Panel Data Set       */
LISTCAT ENTRY('BLM.TEST.SDDS')    ALL  /* Test/Development SDDS      */
LISTCAT ENTRY('BLM.TEST.SDIDS')   ALL  /* Test/development SDIDS     */
```

Figure 7. Example: AMS LISTCAT Commands to Collect Data Set Information

From the LISTCAT output listing, extract the following information:
- The CI size for the data component
- The high used relative byte address (RBA) of the data component
- The CI size for the index component
- The number of records in the index component

The number of CIs in the index component is the number of records in the index component. The number of CIs in the data component of a key-sequenced data set is not equivalent to the number of records in the data component because a CI in the data component can contain multiple records or only a piece of a record when it is longer than a CI. The number of CIs in the data component of a key-sequenced data set can be computed by dividing the high used RBA by the CI size. The number of CIs in a relative record data set is equivalent to the number of records in the data set.

Assume the information extracted with the LISTCAT command (in Figure 7) provided the values shown in Table 4.

Table 4. Example: LSR Definition Worksheet

Data Set Name[1]	Data CI[2] Size	Data CIs[3]	Index CI Size	Index CIs
Tivoli panels	4096	2660	1536	20
Read panel	4096	300	1536	3
Write panels	4096	10	1536	1

Table 4. Example: LSR Definition Worksheet (continued)

Data Set Name[1]	Data CI[2] Size	Data CIs[3]	Index CI Size	Index CIs
Dictionary	4096	546	1536	4
Production SDDS	4096	1638	2048	53
Production SDIDS	2048	868	3584	32
Production SDLDS	4096	6000	N/A	N/A
Test panels	4096	300	1536	3
Write panels	4096	10	1536	1
Test SDDS	4096	220	512	5
Test SDIDS	2048	160	512	1

Notes: [1]The information given in this table is used to define the LSR buffer pools shown in Figure 26 on page 308 through Figure 30 on page 311. The resulting LSR buffer pool definitions are used to define the VSAM resource definition member to the BLX-SP. See Figure 31 on page 312. [2]CI = control interval [3]All data control intervals calculated with high used RBA and CI sizes.

The first step is to review the allocations of all VSAM data sets and ensure that the optimum CI size was chosen and used for each component of each VSAM data set. The data set analysis utilities, BLGUT20, BLGUT21, and BLGUT22, can aid you in this process. If you do not have an existing database, only BLGUT22 may be useful.

If you change the CI size, then after you make the change, check (using the AMS LISTCAT command) that VSAM used the value that you specified; in some cases, VSAM overrides the value that you specify. If VSAM has done this for you, you can coax VSAM into giving you what you want by changing either the data component's average record size, or by changing the type of unit on which you allocated the data set.

Assume the CI sizes in Table 4 on page 81 match the requirements of your data sets. You can eliminate from consideration those data sets for which NSR is used. Little or no performance improvement is gained in using LSR for write panel data sets, and NSR is more efficient for relative record data sets, so you can eliminate defining an LSR pool for the two write panel data sets and the SDLDS. Because fewer users access the test and development database, fewer placeholders and fewer buffers are required to satisfy the LSR requirement. Examining the list of VSAM data sets that are defined, you must define five LSR pools:

- The production SDDS
- The production SDIDS
- The test and development database
- The production panel data sets
- The test and development panel data sets

In addition, you have three data sets that use NSR.

You need one more piece of information before you can define your LSR pools. The information required is the number of positions you require to be maintained in these resource pools by VSAM.

Initially, the number of positions required is somewhere within the range of 25 to 30 percent of the number of users for low-activity data-set-connected resource pools and somewhere within the range of 40 to 60 percent of the number of users for high-activity data-set-connected resource pools. You can monitor both the maximum number of positions used and the maximum number of waits required for shared resource placeholders with the BLX-SP QUERY operator command. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for a description of this command. Appropriate adjustments can be made to the number of placeholders assigned as your installation stabilizes.

Assume that 90 users' address spaces can be connected to the BLX-SP for production activities. Again, you are not too concerned with the number of users, but with their activity. The initial value of positions that are maintained in the low-activity data sets connected to a shared resource pool in this example is 30. The initial value of positions that are maintained in the high-activity data sets connected to a shared resource pool in this example is 45. This gives you the numbers of placeholders required for the production shared resource pools.

Assume also that 20 users' address spaces can be connected to the BLX-SP for test and development activities. The initial value of positions that are maintained in the low-activity data sets connected to a shared resource pool in this example is 5. The initial value of positions that are maintained in the high-activity data sets connected to a shared resource pool in this example is 10. This gives you the number of placeholders required for the test and development shared resource pool.

VSAM Buffer Pool Definition Checklist

The dependence of the BLX-SP upon VSAM resources and their effect upon the performance of the users' address spaces require your continuing attention to the following points when using Tivoli Information Management for z/OS in a non-sysplex environment:

- Define separate index and data pool for key-sequenced data sets.
- Ensure that the optimum CI size is defined for each component of each VSAM data set using LSR. *Remember, if a buffer of the exact CI size is not available, VSAM uses the next larger buffer size available.*
- When you change CI sizes of a data set, ensure that your LSR definition reflects the changes.
- When you change the key length of a data set (SDDS or SDIDS), ensure that your LSR definition reflects the change.
- Make allowances in shared resource pool definitions for new records added to the data set.
- Set the CI size of the dictionary data component the same size as that selected for a panel data set data component so that they can share the same buffer pool.
- Allocate buffer pools for index components large enough to contain the entire index with allowances for growth.
- Monitor both the maximum number of positions used and the number of waits required for shared resource placeholders with the BLX-SP QUERY operator command and adjust the number of placeholders for the LSR buffer pools. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for a description of this command.
- Use NSR only where serial execution is appropriate.
- Check (using the AMS LISTCAT command) that VSAM has used the value that you specified for a CI size.

- Investigate the use of Hiperspace™ buffers for shared resource pools.
- If a Tivoli Information Management for z/OS database is used both as a read/write and read-only database, you must consider both read/write and read-only activity when allocating placeholders in the shared resource pool.
- The minimum buffer pool size that can be allocated for the data component of any VSAM data set is the number of positions that are to be maintained in the shared resource pool plus one.

6

Determining Storage Requirements

This chapter provides information to help you determine storage requirements for Tivoli Information Management for z/OS. Formulas are provided to calculate storage requirements for:

- The user's address space
- The BLX-SP

This chapter assumes that you have a working knowledge of:

- VSAM shared and nonshared resources as used in a non-sysplex environment. Refer to *DFSMS/MVS Macro Instructions for Data Sets* for details.

  It is assumed that you understand how Tivoli Information Management for z/OS uses VSAM resources in a non-sysplex environment. See "Using VSAM Resources in a Non-Sysplex Environment" on page 75.
- VSAM record-level sharing in a parallel sysplex. Refer to the description of administering VSAM record-level sharing in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

  It is assumed that you understand how Tivoli Information Management for z/OS uses VSAM resources in a sysplex. See "Setting Up for Sysplex Data Sharing" on page 151.

Calculating the User's Address Space Storage Requirements

Tivoli Information Management for z/OS does not change z/OS system storage requirements. For planning purposes, the virtual storage required for Tivoli Information Management for z/OS is approximately:

- The number of panel buffers specified in the PNLBCNT keyword of the BLGPARMS macro in the session-parameters member (for panel buffers), times
- 2 KB (KB equals 1024 bytes). This value must be higher if you use large control or help panels. Plus
- 1000 KB, plus
- Five times the maximum LRECL for the SDIDS (for database processing buffers), plus
- Two times the maximum LRECL for the SDDS, plus
- The virtual storage required by TSO and ISPF

In addition, Tivoli Information Management for z/OS requires approximately:
- 50 KB of virtual storage for problem, change, and configuration functions
- 500 KB of virtual storage for PMF functions

The Integration Facility requires no additional storage.

Calculating the BLX-SP Region Size

Materials required to calculate the variables comprising the BLX-SP region size will depend on the environment in which you install Tivoli Information Management for z/OS.

In a non-sysplex environment

The materials required are:

- LISTCAT output of data sets or a personal knowledge of the maximum record sizes of your VSAM data set as well as the CI sizes.
- BLDVRP macro definitions of the LSR pools defined for the BLX-SP. Refer to *DFSMS/MVS Using Data Sets* for a description of the BLDVRP macro.
- A knowledge of the data sets that the BLX-SP is required to support and their connection to specific LSR or NSR.

The BLX-SP region size can be calculated using the following formula.

```
Region Size = 5MB + P + D + S + B + N

(rounded up to next MB, where MB equals 1 048 576 bytes)
```

P Space required by the local shared resource (LSR) buffer pools defined for the BLX-SP
D Space required for data sets that the BLX-SP is required to support
S Space required for the number of strings (VSAM placeholders) that are defined for the BLX-SP
B Space required for input/output (I/O) buffers in the BLX-SP
N Space required for nonshared resource (NSR) buffers in the BLX-SP.

In a sysplex (with sysplex mode enabled)

The material required is:

A knowledge of the data sets that the BLX-SP is required to support and their connection to specific LSR or NSR.

The BLX-SP region size can be calculated using the following formula.

```
Region Size = 5MB + D

(rounded up to next MB, where MB equals 1 048 576 bytes)
```

D Space required for data sets that the BLX-SP is required to support

Sample BLX-SP Region Size Calculation – Non-Sysplex

The following sections explain how to calculate these variables using the example scenario that is described in "Defining LSR Buffer Pools" on page 80 and implemented in Figure 31 on page 312.

Calculating P

Obtain the value of *P* by calculating the space required to satisfy the values specified for the BUFFERS keywords in each of the BLDVRP macros and totaling these values for all the BLDVRP macros included in your LSR definition.

For example, using the example shown in Figure 31 on page 312, *P* has the value shown below:

```
Pool 0       Data  (4096x31) =  126976   Index  (2048x53) =  108544
Pool 1       Data (2048x500) = 1024000   Index  (3584x32) =  114688
Pool 2       Data  (4096x36) =  147456   Index  (1536x30) =   46080
Pool 3       Data  (4096x 6) =   24576   Index  ( 512x 5) =    2560
Pool 4       Data  (2048x50) =  102400   Index  ( 512x 3) =    1536
                               -------                       ------
   Subtotals                   1425408                       273408

P value               1425408 + 273408 = 1698816
```

Calculating D

Obtain the value of *D* by multiplying the number of data sets that the BLX-SP is required to support by 2000 and adding 5000 to the result.

For example, counting the data sets in the example shown in Table 4 on page 81, yields 11 data sets that are to be supported by the BLX-SP. *D* has the value shown below:

```
D value         (11 x 2000) + 5000 = 27000
```

Calculating S

Obtain the value of *S* by calculating the space required to satisfy the values specified for the STRNO keywords in each of the BLDVRP macro and totaling these values for all BLDVRP macros included in your LSR definition. The resulting sum is added to the value specified for the PLACES keyword of the BLXNSR macro to determine the number of strings BLX-SP is required to support. The final number of strings is multiplied by 1000 to determine the value of *S*.

For example, using the example shown in Figure 31 on page 312, *S* has the value shown below:

```
Pool 0            STRNO  = 30
Pool 1            STRNO  = 45
Pool 2            STRNO  = 35
Pool 3            STRNO  =  5
Pool 4            STRNO  = 10
BLXNSR macro      PLACES = 20
                           ---
   Subtotal                145

S value         145 x 1000 = 145000
```

Calculating B

Obtain the value of *B* by calculating the space required to satisfy the I/O buffer requirements of the BLX-SP. You must first determine the largest maximum record length for each LSR pool in your LSR definition and for all data sets using NSR. The maximum record length and CI size of a data set are not equivalent values.

Multiply the largest maximum record length for each LSR pool by the value the STRNO keyword of the BLDVRP macro. Multiply the largest maximum record length for each LSR pool by the value of the PLACES keyword of the BLXNSR macro. Add the values for each calculation. The result of this addition is added to 4000 to determine the value of *B*.

For example, using the example shown in Figure 31 on page 312, *B* has the value shown below:

```
Pool 0      STRNO  = 30     maxlrecl = 4089     30 x 4089 = 122670
Pool 1      STRNO  = 45     maxlrecl = 2041     45 x 2041 =  91845
Pool 2      STRNO  = 35     maxlrecl = 2041     35 x 2041 =  71435
Pool 3      STRNO  =  5     maxlrecl = 4089      5 x 4089 =  20445
Pool 4      STRNO  = 10     maxlrecl = 2041     10 x 2041 =  20410
```

```
NSR         PLACES = 20     maxlrecl = 4089     20 x 4089 =  81780
                                                            ------
   Subtotal                                                 408585

B value     408585 + 4000 = 412585
```

Calculating N

Obtain the value of *N* by multiplying the CI size of each data set using NSR by 3.

For example, the three data sets in the example shown in Figure 31 on page 312 use NSR and are supported by the BLX-SP. *N* has the value shown below:

```
BLM.SDLDS           control_interval_size = 4096 x 3 = 12288
BLM.JILLS.PANELS    control_interval_size = 4096 x 3 = 12288
BLM.JACKS.PANELS    control_interval_size = 4096 x 3 = 12288
                                                       -----
N value                                                36864
```

Calculating Region Size for Non-Sysplex

Using the values obtained for the variables, calculate the BLX-SP region size required to support the sample configuration, as shown below:

```
Region Size = 5M + 1698816 + 27000 + 145000 + 408585 + 36864

Region Size = 5M + 2316265  or 8M
```

Calculating Region Size for Sysplex

The sysplex formula for the BLX-SP region size is

```
Region size = 5MB + D
```

where

D Space required for data sets that the BLX-SP is required to support.

7

Ordering the Necessary Hardware and Software

This chapter describes the hardware and software requirements for running Tivoli Information Management for z/OS Version 7.1. It is designed to help you plan your system environment for the installation of Tivoli Information Management for z/OS.

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.

Hardware Requirements

Base Product

Tivoli Information Management for z/OS Version 7.1 requires a host processor that accommodates the software environment described in “Software Requirements” on page 92.

The display station must be supported by ISPF, and must have a minimum screen size of 24 lines by 80 characters.

Tivoli Information Management for z/OS has no dependencies on any new or changed hardware equipment.

The following hardware requirements are for optional functions, components, or features of Tivoli Information Management for z/OS.

Sysplex

To use Tivoli Information Management for z/OS in a sysplex environment, you need a parallel sysplex with at least one Coupling Facility running at control code level 2 at a minimum to make use of VSAM record level sharing and other enhancements. To share data sets across multiple OS/390 systems, you must have a parallel sysplex. The Multisystem Database Access (MSDA) feature provided in previous releases of the product is no longer supported.

Reports

If your installation requires DBCS support, the display station must support the entry and display of DBCS characters.

Printing Reports with DBCS Characters

For printing reports containing DBCS characters, you need a system printer that supports the printing of such data.These printers include Advanced Function Printing™ (AFP™) printers and standard line printers.

Desktop

To use or customize the Tivoli Information Management for z/OS Desktop, the following hardware is recommended at a minimum:

- IBM Personal Computer or any compatible system unit that supports a 133 MHz CPU or higher
- 48 MB or more of memory
- 20 MB fixed disk space (for the Desktop and the Desktop Toolkit)

Note: 1 MB equals 1 048 576 bytes, 1 KB equals 1024 bytes.

HLAPI/2 Client

Each HLAPI/2 client workstation requires the following hardware:

- An IBM personal computer or compatible system unit capable of running OS/2 WARP® 4, and either Communications Manager/2 Version 1.1 or IBM TCP/IP
- One or more fixed disk drives with sufficient capacity to contain your version of OS/2, Communications Manager/2 or TCP/IP, and the disk storage requirements as specified below
- A communication option capable of supporting APPC LU 6.2 or TCP/IP communication to one or more MVS systems running a Tivoli Information Management for z/OS server

The amount of fixed disk space needed by the HLAPI/2 depends on which parts of the product you install and how you install them. You can only install the HLAPI/2 on an HPFS drive. When you install the HLAPI/2, the disk space needed for each component is:

- Installation and Maintenance utility, 1.4 MB
- Run time parts, 610 KB
- Toolkit parts, 260 KB

When you install the HLAPI/2 from a local area network (LAN) server, the numbers are the same, but the Installation and Maintenance utility is only temporarily copied to your workstation, and then it is deleted.

HLAPI for Windows NT Client

The HLAPI/NT client consists of two parts:

- Requester
- Client interface

Both the Windows® NT requester and client interface must be run on the same machine. The following hardware is required:

- An IBM personal computer or compatible system unit capable of running Microsoft® Windows NT 4.0 or Windows 2000 Professional
- One or more fixed disk drives with sufficient capacity to contain your version of Windows, and the disk storage requirements as specified below
- Token-Ring Adapter Card and network or a communication option capable of supporting APPC LU 6.2 or TCP/IP communication to one or more OS/390 systems running a Tivoli Information Management for z/OS server

The amount of fixed disk space you need to install the HLAPI/NT is:

- Installation utility, 520 KB
- Requester, 1.5 MB
- Toolkit, 10.3 MB

Additional disk space requirements are as follows:

- 6 MB if you are using the Open Database Connectivity (ODBC) driver for Tivoli Information Management for z/OS
- 8.5 MB if you are using the Tivoli Information Management for z/OS interface to Tivoli Inventory

HLAPI/AIX Client

The HLAPI/AIX client consists of two parts:

- Requester
- Client interface

Both the AIX requester and client interface can be run on the same machine or on different machines.

To use the HLAPI/AIX client, you need:

- An RS/6000® machine capable of running IBM AIX

Machines that run the requester require either APPC/APPN or TCP/IP connectivity to the OS/390 host and TCP/IP connectivity to the machines that run the client interface. Machines that run the client interface (this can be the same machine as the requester) require TCP/IP connectivity to the machine that runs the requester.

The amount of fixed disk space you need to install the HLAPI/AIX is:

- Requester, approximately 310 KB
- Client, approximately 2340 KB

HLAPI/HP Client

The HLAPI/HP client consists of two parts:

- Requester
- Client interface

Both the HP requester and client interface can be run on the same machine or on different machines.

To use the HLAPI/HP client, you need:

- An HP Series 700 or 800 workstation capable of running HP-UX Version 10 (up to and including Version 10.2). HP-UX includes TCP/IP.

The amount of fixed disk space you need to install the HLAPI/HP is:

- Requester, approximately 150 KB
- Client, approximately 2350 KB

To use the optional HLAPI for Java provided with the client, you need one of the following:

- HP 9000 Enterprise Business Server
- HP 9000 Workstation
- HP Visualize Workstation

HLAPI/Solaris Client

The HLAPI/Solaris client consists of two parts:

- Requester
- Client interface

Both the Solaris requester and client interface can be run on the same machine or on different machines.

To use the HLAPI/Solaris client, you need:

- A Sun SPARCstation workstation capable of running Solaris Version 2.5.1 (which includes TCP/IP)

The amount of fixed disk space you need to install the HLAPI/Solaris is:
- Requester, approximately 160 KB
- Client, approximately 375 KB

HLAPI/USS Client

The HLAPI for OS/390 UNIX System Services client (HLAPI/USS) must be installed on a host system that can run OS/390. A TCP/IP connection is required between the following:
- The MVS systems running the requester and the MRES with TCP/IP servers
- The MVS systems running the client interface and the requester

HLAPI/CICS Client

The HLAPI/CICS client must be installed on a host system that can run CICS/ESA. If Tivoli Information Management for z/OS and the HLAPI/CICS client do not reside on the same MVS system, you must have a communications link between the two systems.

Problem Service

Additional disk space requirements are:
- For AIX: 12 MB
- For Windows NT or Windows 2000: 7 MB

Software Requirements

Base Product

The minimum software requirements for using the base Tivoli Information Management for z/OS product are defined in this section. Specific functions of Tivoli Information Management for z/OS may require additional products or other levels of the products as noted below. Tivoli Information Management for z/OS is executed as an application in the MVS element of the OS/390 operating system.
- OS/390 Version 2 Release 8 (5647-A01)
- Data Facility Sort (DFSORT™) or a compatible sort and merge licensed program
- OS/390 Security Server (RACF) or equivalent
- TSO Extensions (TSO/E) for 3270 access

Note: The address spaces that APIs run in do not require Interactive System Productivity Facility (ISPF) and a sort program.

The following software requirements are for optional components or features of Tivoli Information Management for z/OS.

Note: Where a requirement for the Java Development Kit (JDK) is specified, the JDK can be obtained from Sun Microsystems, Inc.

DB2 Extract Facility

To use the DB2 Extract Facility you must have the following software:

- IBM Database 2 (DB2) Version 3 (5685-DB2) or a subsequent release

Electronic Mail Notification and Escalation

To use the Notification Management facility to send e-mail messages to an MVS TCP/IP SMTP server, you must configure the SMTP component of TCP/IP.

Graphical User Interface and 3270 User Interface

To use the ISPF graphical user interface (GUI) mode to display Tivoli Information Management for z/OS panels, you must have ISPF and TSO/E operational on the same OS/390 system as Tivoli Information Management for z/OS. In addition, you must have any software that is required by ISPF for running an application in ISPF's GUI mode. Refer to the *ISPF User's Guide* for more information.

To access Tivoli Information Management for z/OS using 3270 you must have ISPF and TSO/E.

Data Reporting

To use the Open Database Connectivity (ODBC) driver for Tivoli Information Management for z/OS, the following are required:

- HLAPI/NT
- Microsoft Windows NT 4.0 or Windows 2000 Professional
- An ODBC-enabled workstation application

Reports can also be produced from a workstation using Tivoli Decision Support for Information Management (5697-IMG).

Support for the host graphics function of the Report Format Facility, which uses the Graphical Data Display Manager (GDDM), is not included. Customers requiring graphics reports can use the ODBC driver or use Tivoli Decision Support for Information Management.

Desktop

To use or customize the Tivoli Information Management for z/OS Desktop, the following software is required:

- Microsoft Windows NT 4.0, Windows 95, Windows 98, or Windows 2000 Professional

The Java runtime environment needed to run the Desktop is provided with the Desktop: Java 2 Runtime Environment, Standard Edition, Version 1.3 (J2RE).

Integration Facility Interfaces

All Integration Facility interfaces are optional. However, to use an Integration Facility interface, the corresponding products are required:

- NetView Version 3 for OS/390 (5655-007) or a subsequent release
- Operations Planning and Control/ESA (OPC/ESA) Version 1 Release 3 (5695-007) or a subsequent release
- Service Level Reporter (SLR), Version 3 Release 3 (5665-397), or a subsequent release

NetView Bridge Adapter

To use the NetView Bridge Adapter function in Tivoli Information Management for z/OS, the following programs are required:

- NetView Version 3 for OS/390 (5655-007) or a subsequent release
- OS PL/I Library, Version 2 Release 3 (5668-911) or a subsequent release. If you are using NetView for OS/390 Version 1 Release 1 (5697-B82) or later, you must use Language Environment® (available with OS/390)

NetView AutoBridge

To use the Tivoli Information Management for z/OS-Netview AutoBridge, the following program is required:

- NetView Version 3 for MVS/ESA (5655-007) (or a subsequent release)

HLAPI/2 Client

Each HLAPI/2 client workstation requires OS/2 WARP Version 4.

To use the optional HLAPI for Java provided with the client, you must have JDK Version 1.1.8 or higher.

To use a RES or MRES with APPC, you must have the following software:

- Communications Manager/2 Version 1.1 or higher (required for the APPC protocol)

To use an MRES with TCP/IP, no additional software is required. TCP/IP is provided with OS/2 WARP .

HLAPI for Windows NT Client

Each Windows NT machine that runs any part of the HLAPI/NT requires the following software:

- Microsoft Windows NT 4.0 or Windows 2000 Professional
- If you install the version of HLAPI/NT that supports both TCP/IP and APPC, you must install and configure the APPC client software such as the client software provided with IBM Communications Server for Windows NT Version 5.0 or Microsoft SNA Server Version 2.11 or higher

Microsoft Windows NT 4.0 includes support for TCP/IP.

To use the optional HLAPI for Java provided with the client, you must have JDK Version 1.1.8 or higher.

HLAPI/AIX Client

Each RS/6000 machine that runs any part of the HLAPI/AIX requires the following software:

- IBM AIX Version 4.2 (5765-C34) or Version 4.3

To use SNA, each RS/6000 machine that runs the requester options of HLAPI/AIX to communicate with either a RES or an MRES with APPC requires IBM AIX SNA Server/6000 Version 2.1 (5765-247) or a subsequent release.

To use the REXX HLAPI/AIX, each RS/6000 machine that runs REXX HLAPI/AIX requires IBM AIX REXX/6000 (5764-057).

To use the optional HLAPI for Java provided with the client, you must have JDK Version 1.1.8 or higher.

HLAPI/HP Client

Each HP machine that runs any part of the HLAPI/HP requires the following software:

- HP-UX Version 10 (up to and including Version 10.2)

To use the optional HLAPI for Java provided with the client, you must have JDK Version 1.1.8 or higher, and HP-UX Version 10.2.

HLAPI/Solaris Client

Each Sun machine that runs any part of the HLAPI/Solaris requires the following software:

- Solaris Version 2.5.1

To use the optional HLAPI for Java provided with the client, you must have JDK Version 1.1.8 or higher, and Solaris Version 2.5.1.

HLAPI/USS Client

To use the HLAPI/USS client, OS/390 UNIX System Services must be configured to start up in "full function" mode. (Refer to *OS/390 UNIX System Services Planning* for more information.)

HLAPI/CICS Client

The HLAPI/CICS client requires the following software:

- CICS/ESA Version 4 Release 1 (5655-018)

  Note: The sample CICS application that is shipped by Tivoli requires either Language Environment (available with OS/390) or VS COBOL II Version 1.4 to run. HLAPI/CICS itself does not require VS COBOL II.

Connectors to the World Wide Web

REXX Web connector for MVS

The software requirements for using the REXX web connector for MVS are:

- A Web browser
- IBM Library for SAA® REXX/370 Release 3 for MVS/ESA (5695-014)

REXX Web connector for OS/2

The software requirements for using the REXX web connector for OS/2 are:

- A Web browser
- OS/2 WARP Version 4
- Lotus Domino™ for OS/2
- HLAPI/2

REXX Web connector for OS/390

The software requirements for using the REXX web connector for OS/390 are:

- A Web browser

To use the Java applets for data field validation, you need:

- REXX Web connector for MVS or REXX Web connector for OS/390
- A client Web browser that supports both Java Version 1.1.8 or higher and JavaScript calling applets compiled with JDK Version 1.1.8 or higher
- A Web server that is capable of serving Java applets
- Supplied Java applets stored on a Web server

TEC Event Adapter

The Tivoli Information Management for z/OS TEC Event Adapter requires the following software:

- Tivoli Event Integration Facility for z/OS (available with Tivoli Information Management for z/OS as FMID H256100)

Integration with Other Tivoli Products

If you are using Tivoli Information Management for z/OS in a Tivoli management software environment, you must have the following software installed as applicable. Refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications* for more information.

- Tivoli Decision Support: To use Tivoli Decision Support for Information Management, the following software is required:
  - Microsoft Windows NT 4.0 or later
  - Tivoli Decision Support for Information Management (5697-IMG)
  - Tivoli Decision Support 2.1 (installation of Cognos PowerPlay and Crystal Reports may also be required)
  - HLAPI/NT
  - ODBC driver for Tivoli Information Management for z/OS
- Tivoli Enterprise Console integration facility: To send events to Tivoli Information Management for z/OS through the Tivoli Enterprise Console, you must have the following installed:
  - Problem Service
  - Tivoli Enterprise Console Version 3 Release 1 or higher
- Tivoli Inventory: To use an extract of Tivoli Inventory data in Tivoli Information Management for z/OS, you must have the following:
  - Tivoli Inventory Version 3.6.2
  - HLAPI/NT
- Tivoli Service Desk Bridge: To exchange records with the Tivoli Service Desk through Tivoli Service Desk Bridge, you need:
  - Tivoli Service Desk 6.0
  - HLAPI/NT

  Optionally, to handle Tivoli Service Desk Bridge errors, you need the Tivoli Information Management for z/OS TEC Event Adapter.
- Tivoli Software Distribution: Tivoli Information Management for z/OS HLAPI client features can be installed using Tivoli Software Distribution (HLAPI/2, HLAPI/NT, HLAPI/AIX, HLAPI/HP, and HLAPI/Solaris). The HLAPI/CICS and HLAPI/USS are installed using SMP/E.

  To distribute software upon approval of a Tivoli Information Management for z/OS change request, you need:
  - Tivoli Software Distribution Version 3.1
  - Tivoli Enterprise Console Version 3.1
  - Tivoli Information Management for z/OS TEC Event Adapter

Problem Service

To use Problem Service, you need:
- For AIX:
  - IBM AIX Version 4.2 (5765-C34) or Version 4.3

  - HLAPI/AIX
  - Tivoli Management Environment 3.1 or higher
  - Tivoli Application Development Environment (if you are writing an application to use Problem Service)
- For Windows NT:
  - Microsoft Windows NT 4.0
  - HLAPI/NT
  - Tivoli Management Environment 3.1 or higher
  - Tivoli Application Development Environment (if you are writing an application to use Problem Service)

8

Assigning and Scheduling Personnel and Systems

This chapter discusses:
- Assigning personnel to the various tasks involved in running Tivoli Information Management for z/OS
- Scheduling education and training for Tivoli Information Management for z/OS
- Implementing a test system for Tivoli Information Management for z/OS

This chapter assumes that you are familiar with the tasks that are involved in installing and running Tivoli Information Management for z/OS. This chapter is designed to help you plan your resources for Tivoli Information Management for z/OS.

Assigning Personnel

To plan for Tivoli Information Management for z/OS, it is important that you involve those who are responsible for the functional areas that use the product. These people must be available to represent their departments and make their concerns known during the planning. This helps to ensure that planning decisions, especially those that can affect existing procedures or policies, take place.

Choose a project administrator or planner to coordinate the installation planning activities. This person can continue to be responsible for the program after it is installed and in use. In addition, select a coordinator for each functional area that uses Tivoli Information Management for z/OS.

You can select one functional area to serve as a pilot project. By doing so, you can validate the initial system definitions, reports, data flow, and support procedures before you make Tivoli Information Management for z/OS generally available to your entire organization.

The project administrator can distribute the responsibilities for Tivoli Information Management for z/OS as follows:

- Assign a system analyst or programmer to maintain the operating system control program and to install Tivoli Information Management for z/OS. See “Overview for Planning and Installing” on page 1 for installation instructions.
- Assign a system analyst or programmer to set up and manage the following as applicable:
  - Set up any necessary coupling facility structures to run Tivoli Information Management for z/OS in a parallel sysplex or to share Tivoli Information Management for z/OS databases across multiple z/OS systems. For information on

setting up for data sharing or the GRS environment for sharing databases across multiple z/OS systems, see “Setting Up for Sysplex Data Sharing” on page 151.

- Assign a system analyst or programmer to set up and manage the APPC/MVS environment that is required for using the Remote Environment Server functions. Refer to the *Tivoli Information Management for z/OS Client Installation and User’s Guide* for information on setting up APPC/MVS for the Remote Environment client/server functions.

- Assign a program administrator to do the following:
  - Manage the dictionary data set, panels, and dialogs. See “Defining the DICTDS” on page 289 and refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
  - Set privilege classes for users. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.
  - Manage notification and escalation. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.
  - Control the use of PMF and terminal simulator panels (TSPs or TSXs). Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.
  - Create data model records as necessary. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.
  - Customize use of the Tivoli Information Management for z/OS Desktop, if applicable, and help deploy the Desktop at user workstations. Refer to the *Tivoli Information Management for z/OS Desktop User’s Guide* for more information.
  - Set up a knowledge base for improved problem resolution searching, if desired. Create a text search index to use with the IBM Intelligent Miner™ for Text.

- Assign a database administrator to set up access to the data sets, maintain the databases, and run the utilities. See “Defining Tivoli Information Management for z/OS Data Sets” on page 277 and refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for further details.

- Assign host and client application programmers to create and manage programs used as Tivoli Information Management for z/OS exit routines and to write applications for use with the application program interfaces. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for details about the APIs. Refer to the *Tivoli Information Management for z/OS Client Installation and User’s Guide* for details about the Tivoli Information Management for z/OS client APIs.

- Assign a system administrator to integrate Tivoli Information Management for z/OS with other Tivoli applications if you are using Tivoli Information Management for z/OS in a Tivoli Management software environment. Refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications* for more information.

- Assign a programmer with REXX and HTML skills to design and create applications that use the Tivoli Information Management for z/OS Web connector. Java skills and a knowledge of Web browsers and Web servers are desirable. Also, consult with your company’s Information Technology security officer to review any Internet and intranet security concerns your company may have.

- Assign a system programmer to create TSPs or TSXs to simulate Tivoli Information Management for z/OS sessions, customize panels, and create reports. Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*, *Tivoli Information Management for z/OS Panel Modification Facility Guide*, and *Tivoli Information Management for z/OS Data Reporting User’s Guide* for details.

- Assign a DB2 administrator to facilitate the use of the DB2 Extract Facility. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for details.

See Table 24 on page 401 for additional duties that you can assign.

Scheduling Education and Training

Establish an education plan early in the implementation process. All personnel should know how to use and maintain Tivoli Information Management for z/OS effectively so they understand the purpose of the product and how it affects their roles in the data processing environment.

After your system is ready, you should educate your users through on-the-job training or formal education. You should develop and maintain documentation and procedures based on how you tailor your system.

Implementing a Test System

After you install Tivoli Information Management for z/OS according to the instructions given in “Overview for Planning and Installing” on page 1, and all modifications to your system are made and tested, you can make the system available to a small group of users. As the first users, they can help identify any problems with the system and procedures. They can also suggest additional modifications and educational requirements.

You can control access to the data in the test system through the use of privilege classes because each user must have display authority to view any records except SRC, ALIAS, and COMMAND records.

9

Migrating from Previous Versions

This chapter specifies the tasks and considerations for migrating from previous versions of Tivoli Information Management for z/OS, including Tivoli Service Desk for OS/390 Version 1.2 and Information/Management licensed programs.

Note: Organizations currently using older versions not listed here should contact Tivoli support personnel for migration information.

If you are a new user, go to "Overview for Planning and Installing" on page 1.

Tivoli Information Management for z/OS Version 7.1 Changes

"What's New in Tivoli Information Management for z/OS?" on page xix lists the key changes associated with this release. From an installation perspective, note the following actions you *must* perform with this release or that you *should consider* performing if you currently have earlier versions of Tivoli Information Management for z/OS or its predecessor products installed.

Spanned data sets no longer supported

Because it is unnecessary to use them, spanned VSAM data sets are not supported. If you try to open a spanned VSAM data set, an error message is displayed. Use IDCAMS to delete and redefine the data set without the SPANNED attribute. After the data set is created with IDCAMS, use the BLX-SP REALLOC command (described in *Tivoli Information Management for z/OS Operation and Maintenance Reference*) to reallocate the data set.

Migrate to sysplex if you are using Multisystem Database Access

If you are currently using Multisystem Database Access to share databases across multiple systems, you must now use sysplex data sharing or stop sharing databases. You must have a parallel sysplex with a coupling facility to continue sharing databases.

- For all BLX-SPs, modify the BLX-SP parameters member in the data set specified on the BLXPRM DD statement in your BLX-SP procedure:
  - Add the SYSPLEX=YES parameter. SYSPLEX=YES enables sysplex mode for the BLX-SP and its users.
  - Remove the VSAMRESOURCES parameter. The VSAM resource definition member name (for example, BLXVDEF) is ignored when running Tivoli Information Management for z/OS in sysplex mode.
  - Remove the DESTNAMES parameter. The DESTNAMES parameter is not supported.

For an example of the BLX-SP parameters member, see BLX1SH in the SBLMSAMP data set.

- Because VSAM RLS requires use of a LOG parameter, you must migrate the VSAM data sets from your current version of Tivoli Information Management for z/OS to SMS-managed clusters with the LOG(NONE) parameter. You can use the IDCAMS ALTER or IDCAMS DEFINE CLUSTER command to define clusters with the LOG(NONE) parameter. As an alternative, you can run the BLGALTER job provided in the SBLMSAMP data set to migrate existing VSAM data sets to be RLS enabled.

  Note: The following samples in the SBLMSAMP data set are also updated to illustrate use of the LOG(NONE) parameter.

  BLGDATAB – Define the Tivoli Information Management for z/OS database
  BLGDICT – Define the dictionary data set
  BLGRPNL – Define read panel data sets
  BLMWPNL – Define write panel data sets

  Additionally, RLS does not support data sets defined with the IMBED option. If you have data sets defined with the IMBED option, you must create new ones with the NOIMBED option, and then run IDCAMS REPRO to copy the data from the old data sets to the new ones.

  Refer to the *DFSMS/MVS Access Method Services for the Integrated Catalog Facility* for more information about using IDCAMS.

- Perform the sysplex setup tasks as described in “Enabling Sysplex Mode” on page 156. These tasks may be performed by your z/OS systems programmer and include the following:
  - Creating RLS share control data sets.
  - Setting up a CFRM policy.
  - Setting up the sysplex policy couple data sets.
  - Updating the SYS1.PARMLIB data set with COUPLExx and IGDSMSxx members.

Load the base records provided with Tivoli Information Management for z/OS.
The base records that are shipped with Tivoli Information Management for z/OS must be loaded because certain base functions in this release are dependent on their existence. For example, to take advantage of the enhanced change approver processing, or to use the universal time processing feature, you must have the base data model records loaded. For instructions on loading the base records, see “Loading Records Provided with Tivoli Information Management for z/OS” on page 219.

Consider enabling data-entry panels for new user option
If you have created your own data-entry panels or have customized the panels provided with Tivoli Information Management for z/OS, you should consider enabling these panels to allow entry field numbers to be hidden. A new screen control option provided in the user profile (session control defaults) enables users to hide field numbers on data-entry panels. The base panels supplied with Tivoli Information Management for z/OS are already enabled, but you must migrate any

panels that have been built using previous releases of the product. You should do this only if you want to enable the option to hide the entry field numbers. If users do not mind seeing field numbers on panels, you can leave your panels alone. To enable the data-entry panels, bring up each panel in the update function of the Panel Modification Facility (PMF) and file the panel. No actual panel changes are necessary. The PMF file process automatically enables the panel.

Note: If you use the ISPF GUI to interact with the database, you should *not* use the option to hide entry field numbers because it is not compatible with the ISPF GUI feature.

Date and time processing is changed

In this release, changes were made to how dates and times in records are processed.

- The way dates and times are stored in the SDDS is changed. Dates and times are now stored in internal format.

  The internal and external time-of-day format are still the same (HH:MM) (unless you have written your own time conversion routine, in which case your time format is different).

  Dates were previously stored in external date format in the SDDS. Now, all dates are stored in the SDDS in internal format. Dates from records created in prior versions (in external format) will be converted automatically to internal format when the record is read for any reason (for example, when the record is displayed, updated, or shown on a report). This change was made so that new features, such as the user's ability to select an external date format, will work with records created in prior versions of the product. The first time the record is filed in Tivoli Information Management for z/OS Version 7.1, all dates in the record are stored in internal date format. The record can no longer be accessed properly by prior versions of the product.

- The user profile now allows users to select their desired external date format from a list of 22 supported formats. This change enables users to alter the way dates appear on the Tivoli Information Management for z/OS panels and reports and other output. If no selection is made in the user profile, the default external date format specified in the session parameters through the BLGPARMS DATEFMT keyword is used.

  Users can specify a preferred external date format by selecting the User and Database Defaults section of the user profile. The preferred date format selection takes effect for all records created or updated under this release or under previous releases.

BLGPARMS DATECNV keyword is changed; new keywords are available

Some changes were made to the keywords and parameters of the BLGPARMS session parameters macro. The DATECNV keyword is changed, and the DATEFMT, ODATEFMT, TIMEZONE, and OTIMEZONE keywords are new. Familiarize yourself with the changes, make the necessary adjustments to your session parameters, and then reassemble your session-parameters members.

- The DATECNV keyword specifies the date conversion routine name only. It no longer specifies the internal or external date format. A separate required keyword, DATEFMT, is provided to specify external date formats. Internal date format is now YYYY/MM/DD for all users; there is no option to change the internal date format. Details about the DATECNV keyword are available on page 322.

Also, the ASIS and PRIMARY parameters that were formerly entered in the fifth position of DATECNV are no longer supported. These parameters previously specified whether Tivoli Information Management for z/OS should convert dates entered by users into a primary external date format. Now that dates are stored in internal format and users can enter dates in 22 different formats, these parameters are no longer necessary. If you reassemble your session parameters and include either the ASIS or PRIMARY parameter, an assembly error occurs. Therefore, you should remove ASIS or PRIMARY from the DATECNV keyword before reassembling.

- The DATEFMT keyword is a required keyword that specifies the default external date format to be used by Tivoli Information Management for z/OS when processing dates. See page 320 for more details about DATEFMT.
- The ODATEFMT keyword is an optional keyword you can use to specify the external date format of dates in old records. ″Old″ records are existing records in your database that were created with versions of Tivoli Information Management for z/OS (or its predecessor products) before Version 7.1. The ODATEFMT parameter is only necessary if there are dates in your database with an external date format that is different from the default external date format specified on DATEFMT. See page 326 for more information about ODATEFMT.
- Optionally, to enable universal time processing, specify the TIMEZONE keyword and a time zone symbol. The TIMEZONE keyword is described on page 332.

  If you are not implementing universal time processing, it is recommended that your Tivoli Information Management for z/OS administrator use PMF to remove the User’s time zone field from panel BLG0P700, User and Database Defaults, so that users do not get frustrated by attempting to use a field that will not work in their environment.

  Note: If you use the Integration Facility you should not implement universal time processing. Unpredictable results could occur if you enable the TIMEZONE keyword and use the Integration Facility. If you use the Integration Facility, you should also consider using PMF to remove the User’s time zone field from panel BLG0P700.
- Additionally, the OTIMEZON keyword is available as an optional keyword that specifies the time zone associated with ″older″ records in your database. (In this case, ″older″ records are those records that do not have universal time data for a field that is a universal time field. This includes all records established before Tivoli Information Management for z/OS Version 7.1, and also those records that are filed before the field is defined as a universal time field.) The OTIMEZON keyword enables Tivoli Information Management for z/OS to migrate older records as they are being accessed to use the enhanced date processing functions. If you are enabling universal time processing and have older records in your database that you want users to be able to process, and the time zone for those records is different from that which you are specifying on the TIMEZONE keyword, specify the OTIMEZON keyword. Otherwise, the value specified for TIMEZONE will be used. For more information about the OTIMEZON keyword, see page 326.
- The following table provides some examples of how parameters are now specified:

If you had this previously:	Specify this now:
Internal date format of YYYY/MM/DD, primary external date format of MM/DD/YYYY (second value) and no secondary date format: `DATECNV=(BLGCDATS,YYYY/MM/DD,MM/DD/YYYY)`	`DATECNV=BLGCDATS,   (or omit for default)` `DATEFMT=MM/DD/YYYY`

Internal date format of YYYY/MM/DD, primary external date format of MM/DD/YYYY (second value) and secondary date format of MM/DD/YY (third value) and conversion of dates entered to primary external date format:
`DATECNV=(BLGCDATS,YYYY/MM/DD,MM/DD/YYYY,MM/DD/YY,PRIMARY)`
	`DATECNV=BLGCDATS,  (or omit for default)`
	`DATEFMT=MM/DD/YYYY,`
	`ODATEFMT=MM/DD/YY`

Records for a company with two geographic locations in different time zones (Eastern and Mountain, U.S.) all reflected the Eastern time format.	If you want users in the Eastern time zone to work business as usual, but yet have users in the Mountain time zone see times that reflect their own time zone: `TIMEZONE=ET` Users in the Mountain time zone can specify MT as the time zone format preference in the user profile.
Records entered were all associated with Eastern time zone (ET) times for a U.S.-based company. However, a company merger has since occurred with a company based in the U.K. Information processing headquarters is in the U.K. but older U.S. records must still be processed. All users are in the U.K.	`TIMEZONE=WET,` `OTIMEZONE=ET`

For more information about date-related migration tasks, see "Migration Considerations" on page 234.

Session members can use different time-of-day formats

Different session members can now use different time-of-day formats. The formats supported are the default HH:MM format, or a format you specify through your own conversion exit routine. Previously, you could have only one time-of-day format specified for all session members using a database.

Session members can now use different external date formats and time zones as well. With this feature, a session member for Boston and another for Frankfurt can be set up, even if the members are using a single database. Remember to use ODATEFMT and OTIMEZON to reflect the date format and time zone of records which were created previous to the installation of Tivoli Information Management for z/OS Version 7.1.

Modify change request panels to include revised control panel flow

The flow on some change request panels is changed in the Management application. Change approver and reviewer data is collected as list processor data for new records or for existing records that do not have approver or reviewer data specified. List processor panel BLGLAPVR is displayed for the entry of approver data and panel BLGLREVR is displayed for the entry of reviewer data during change record entry and update processing. If existing change records have approver or reviewer data already specified, the data-entry panel flow is unchanged and the list processor panels are not displayed.

Note: To use this enhancement, you must load the base data model records which are provided with Tivoli Information Management for z/OS. For more information about loading base data model records, see “Loading Records Provided with Tivoli Information Management for z/OS” on page 219.

If the ;COPY command is used to create a new change request record and the copied data contains non-list approver and reviewer data, the approver and reviewer data for the new record is collected as non-list data.

The changes are summarized below:

- Panel BLG1A121, Change Record File Processing:

  Two FLOW lines are added for TSXs BLGTX121 and BLGTXCAP.
- Panel BLG0CU01, Change Request Summary (entry):

  When option 5 (Approver data) is selected, the panel flows to control panel BLG1AAPU instead of BLG0C500. When option 6 (Reviewer data) is selected, the panel flows to control panel BLG1AREU instead of BLG0C700.
- Panel BLG0CU00, Change Request Summary (update):

  When option 5 (Approver data) is selected, the panel flows to option panel BLG0C015 instead of BLG0C500. When option 6 (Reviewer data) is selected, the panel flows to control panel BLG1AREU instead of BLG0C700.
- Panel BLG0S020, Change Summary Display:

  When option 8 (Approver display) is selected, the panel flows to control panel BLG1AAPD. When option 9 (Reviewer display) is selected, the panel flows to control panel BLG1ARED. In addition, the selection s-word 0CFA XIMDIAAPR0 is added to option 8 (Approver display), which allows panel BLGLAPST to be re-displayed with the updated approval status when program exit BLG02041 completes.

When a change approver is added to the list of approvers on panel BLGLAPVR, a default status of PENDING is set for the approver. The change approver can accept or reject the request by selecting option 8 on BLG0S020 Change Summary Display panel and typing an ’A’ or an ’R’ as appropriate on list processor panel BLGLAPST.

Note: These changes do not apply to or affect the Integration Facility.

If you have modified any of these panels (BLG1A121, BLG0CU01, BLG0CU00, or BLG0S020), or if you have created your own customized copies of these panels, you must modify them to flow to the new control panels and add the new selection s-word (for ’A’ or ’R’) to panel BLG0S020. For *each* panel, you must add the new TSXs, update the selections, and test the modifications. The following steps describe this process:

1. Add the New TSXs
   a. Select **PMF**.
   b. Select **Panel update** on the PMF options panel.
   c. Enter the name of the panel to modify on BLM8CU00 (BLG1A121 or your panel name).

   d. On BLM8CU60, select **Abstract**. The panel you requested is displayed for update.
   e. Type **control** on the command line and press Enter.
   f. On BLM1TSCU, insert a FLOW control line and the FLOW control line that has BLG1ACAN as the false target.
   g. On BLM8CU63, select **Control flow processing**.
   h. On BLG8CU6A, change the Function code to 002B, code-invoke TSP, clear the True target panel name, and enter BLGTX121 in the **Program exit/TSP name** field.
   i. Type **end,end** and press Enter.
   j. On BLM1TSCU, type **R** on the FLOW control line you just created.
   k. On BLM8CU63, select **Control flow processing**.
   l. On BLG8CU6A, change the **Program exit/TSP name** field to BLGTXCAP.
   m. Type **end,end,end,end** and press Enter.
   n. On BLM8CU60, select **File**.
2. Update the selections.
   a. Select **PMF**.
   b. Select **Panel update** on the PMF options panel.
   c. Enter the name of the panel to modify on panel BLM8CU00 (BLG0CU01, BLG0CU00, BLG0S020, or your panel name).
   d. On panel BLM8CU70, select **Externals**. The panel you requested is displayed for update.
   e. Type **control** on the command line, move the cursor to the number for the Approver selection, and press Enter.
   f. On panel BLM8CU73, select **Panel flow processing**.
   g. On panel BLM8CU7A, change the name of the target panel to the new control panel:
      - If the current target panel is BLG0C500 and you are updating BLG0CU01, change it to BLG1AAPU.
      - If the current target panel is BLG0C500 and you are updating BLG0CU00, change it to BLG0C015.
      - If the current target panel is BLG0M500, change it to BLG1AAPD.
   h. Type **end** and press Enter.
   i. If the panel you are modifying is *not* BLG0S020 or a copy of this panel (Change Display Summary), go to step j.

      If the panel you are modifying *is* BLG0S020 or a copy of this panel (Change Display Summary), perform the following additional steps:
      1) On BLM8CU73, select **Data collection processing**.
      2) On BLM8CU7B, change the **Structured word index** field to 0CFA (Approver Entry).

3) Type **end** and press Enter.

4) Go to step j.

j. Type **end** and press Enter.

k. Type **control** on the command line, move the cursor to the number for the Reviewer selection, and press Enter.

l. On panel BLM8CU73, select **Panel flow processing**.

m. On panel BLM8CU7A, change the name of the target panel to the new control panel:

- If the current target panel is BLG0C700, change it to BLG1AREU.
- If the current target panel is BLG0M700, change it to BLG1ARED.

n. Type **end,end,end** and press Enter.

o. On panel BLM8CU70, select **File**.

3. Test the modifications.

a. Copy the modified panels to your test read panels data set.

b. Create, update, and display change request records to ensure the Approver and Reviewer selections display the desired panel.

The report format tables (RFTs) BLMPRCR, BLMZZ13, BLMZZ14, and BLMZZ34, provided with Tivoli Information Management for z/OS, are updated to include the change approver and reviewer list field data. If you have modified these RFTs or have created RFTs based on these formats, you may need to update them to display the list field data.

Note about Searching: Quick search panels BLG0F590 (Change Approver Data Inquiry) and BLG0F890 (Change Control Data Inquiry) do *not* need to be changed to support the list data. The s-words and p-words added for the new change approver and change reviewer fields associated with list processor panels are the same or similar to the values used for the structured fields when list data is not used. Your existing searches should find all occurrences of the approver and reviewer data, regardless of whether or not the approver or reviewer data was entered through a list processor panel. However, you can search specifically on the new change fields if desired. New change fields are listed in the following table:

Field Name	P-word
Approver pending approved rejected	 SPAR/ SAAR/ SRAR/
Approval status	APST/
Approver class	CLAP/
Reviewer	CLAX/

For example, the following searches return change records associated with approver or reviewer data entered as either list or non-list data:

- `;SEARCH SP**/HELPDESK` finds all changes pending approval by the HELPDESK privilege class name.
- `;SEARCH SA**/HELPDESK` finds all changes that have been approved by HELPDESK.
- `;SEARCH SR**/HELPDESK` finds all changes that have been rejected by HELPDESK.
- `;SEARCH CLAX**/HELPDESK` finds all changes assigned to HELPDESK for review.

If needed, you can retrieve only those change records that are associated with change approver or reviewer list data. For example:

- `;SEARCH APST/PENDING` finds all changes pending approval for records entered through BLGLAPVR only.
- `;SEARCH SPAR/HELPDESK` finds all changes pending approval by HELPDESK (BLGLAPVR only).

Migrate privilege class records to use new list processor panels

The panels used to collect and display the list of eligible users of a privilege class are changed in the System application. List processor panels are now used to collect and display the data for new privilege class records. List processor panel BLGLJ300 (General User Authorization Table) is displayed for the entry of user IDs associated with a privilege class. For existing privilege class records, the data-entry panel flow is unchanged and the BLGLJ300 list processor panel is not displayed.

If desired, you can run TSX BLGTPRIV to migrate all your existing privilege class records to use the new list processor panels. By running this TSX, you can ensure that the same set of panels is used to enter or display data related to users of a privilege class. This panel consistency can make it easier to work with the records. TSX BLGTPRIV can be run from the command line or in batch mode. It will update all the privilege class records in your database that contain eligible user data in the old format and convert the data to the new list processor format.

You can modify a copy of the sample JCL shipped for BLHRCDSL and use that copy to run TSX BLGTPRIV. BLGTPRIV has no parameters, so modify the copy to appear as follows:

```
IPSTART BLGINIT PARMS(SESS(00) CLASS(MASTER) TSP(BLGTPRIV) IRC(;QUIT))
```

Before deciding whether to use BLGTPRIV, you should be aware of the following:

- The user running TSX BLGTPRIV (either interactively or in batch made) must be in a privilege class that has, at minimum, DBADMIN and class authorities.
- BLGTPRIV will *not* update Integration Facility (IIF) privilege class records.
- It is not necessary to to run BLGTPRIV if you have already customized your privilege class panels to use list processor data for the eligible user list.

Desktop users – migrate to new version

If you are currently using the Java-based Desktop application, you should do the following:

- If not already loaded, load the base data model records provided in this version of Tivoli Information Management for z/OS. If you made any modifications to the records in the previous release and want to keep the changes, you must update the Version 7.1 data model records with your changes. Although the data model records provided previously will still work, it is recommended that you use the new records. For instructions on how to load the base data model records, see "Loading Records Provided with Tivoli Information Management for z/OS" on page 219.

- Before installing Desktop Version 7.1, uninstall the previous version. Instructions are provided in the *Tivoli Information Management for z/OS Desktop User's Guide*.
- Use the configuration editor to rebuild the jhd.properties file. The properties file constructed under Desktop Version 1.2 is not upwardly compatible.
- Hierarchy files constructed with the previous version of the Desktop are upwardly compatible. There is no need to rebuild them.

Migrating from Earlier Versions

This section summarizes functions and actions to consider if you are migrating from previous versions of the product to Tivoli Information Management for z/OS Version 7.1. You should review the information in "Tivoli Information Management for z/OS Version 7.1 Changes" on page 103 and the following table, and check off the items that apply to you, based on the release you are migrating from (as shown in the far right column). **In Table 5, an *X* in the "New" column identifies items that are new with this Version 7.1, or that you must address if you are installing Version 7.1.** If you have kept up with installing the latest releases of the product, you need only pay attention to the items with an *X* in the "New" column. Otherwise, if you are migrating from older releases, you should review all topics to understand what changes took place to determine what to do. Where necessary, the topics in Table 5 are discussed in further detail in the text that immediately follows the tables.

Table 5. Migrating from Previous Versions

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
You can perform the following action in Step 4 of the general installation checklist in Chapter 1.				
Multisystem Database Access	MSDA not supported – use sysplex	If you used the MSDA facility to enable users on different BLX-SPs to share Tivoli Information Management for z/OS databases (VSAM data sets) concurrently, you must now run Tivoli Information Management for z/OS in a parallel sysplex environment with a coupling facility to continue sharing data sets. For more information, see "Setting Up for Sysplex Data Sharing" on page 151.	X	Information/Management 6.1, 6.2; TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2
Shared databases	Resource name changes	Note the following resource name changes: ■ The BLXMSDA and BLXVRFY resource names are no longer used. ■ A new BLGSDIDS resource name has been added for use in a sysplex environment. ■ The BLXDASDS resource name is not used in a sysplex environment. These changes are incompatible with previous versions. If you share databases between systems running different levels of Tivoli Information Management for z/OS or its predecessor products, you will need to migrate all systems to this release. See "Resource Names That Tivoli Information Management for z/OS Enqueues On" on page 351 for details.	X	All previous releases
You can perform the following action in Step 5 of the general installation checklist in Chapter 1.				

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Dynamic load library	Running Tivoli Information Management for z/OS from link pack area	If you run Tivoli Information Management for z/OS from the link pack area (LPA), you must modify the ISPF TSO command table ISPTCM. See "Modifying the ISPTCM Table" on page 21 for complete information on running from the LPA.		All previous releases
You can perform the following action in Step 6 of the general installation checklist in Chapter 1.				
BLX-SP parameters member	Modified	If you intend to run Tivoli Information Management for z/OS in a parallel sysplex (to take advantage of VSAM RLS or to use shared Tivoli Information Management for z/OS databases), you must include the SYSPLEX=YES parameter in your BLX-SP parameters member to enable sysplex mode. (You can also use the Installation Tailoring Facility to enable sysplex mode.) The DESTNAMES parameter which was previously used with Multisystem Database Access is no longer supported. The VSAMRESOURCES parameter is ignored when sysplex mode is enabled.	X	All previous releases
BLX-SP parameters member	Modified	The APISECURITY parameter was added to activate security checking for APIs. You need to add this required parameter before use. Refer to the *Application Programming Interface Guide* for details. The MAILQ and MAILQWAITTM parameters (optional) were added to support queueing of e-mail notification messages. Refer to ″Using Notification Management″ in the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for details. **Note:** The BLX-SP parameters member did not exist before Information/Management Version 5.1. If you are migrating from versions that predated Version 5.1, become familiar with parameters and create a BLX-SP parameters member. You can also use the Installation Tailoring Facility to create this.		Information/Management 5.1, 6.1, 6.2

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Date format	Run BLGUT17	Tivoli Information Management for z/OS 7.1 has implemented a number of date and time-of-day enhancements. If you are currently using two or more external date formats, it is recommended that you run BLGUT17 before or during the installation process in order to standardize the date formats your organization uses. Information on running BLGUT17 can be found in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.	X	All previous releases
Date format	DATEFMT is used for default external date format specification; DATECNV is for date conversion routine name only; ODATEFMT is available to automatically convert old date records	Review changes made to the BLGPARMS macro keywords that support date processing (see "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318). All panels and PIDTs shipped with Tivoli Information Management for z/OS support the entry of dates in any of the 22 external date formats. Users have the option to change date formats by making a selection in the User and database defaults option in the user profile.	X	Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2
Date format	Users can select different date formats through user profile	Because users can set different date formats through the user profile, any TSPs or TSXs that you have written that enter or process dates must support different external date formats for different users. All processing in the TSP or TSX should use internal format and should convert to and from the user's external format when interacting with the database. The TSPs or TSXs should be written to use the 10-character internal date format (YYYY/MM/DD) for processing date data. Use the BLGIDATE and BLGEDATE user exits to convert the internal format date from or to the user's local date format. The BLGIDATE user exit converts a date from the user's local date format to internal format. The BLGEDATE user exit converts a date from internal format to the user's local date format. For more information about these user exits, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.	X	Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Date format	Internal date format must be set to YYYY/MM/DD	For a description of what you need to do to change an existing database to use this expanded format, see the explanatory note on date formats which follows this table. **Note:** You only need to change if this format if you are currently using internal date format YY/MM/DD.	X	All previous releases
Logical database partitions	The Tivoli Information Management for z/OS database can be divided into multiple logical partitions.	Determine if you want to use logical database partitions and reassess privilege classes as appropriate. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information. The BLGPARMS macro has a GBLPID parameter (optional) to enable you to specify the name of a global partition if desired.		All previous releases
SDIDS	Data structure changed	Because the data structure of the SDIDS changed, you must, as part of installing this release, rebuild or migrate your SDIDS using the BLGUT1 or BLGUT1M utility program respectively. (This does not apply to TME 10 Information/Management Version 1.1 or Tivoli Service Desk for OS/390 Version 1.2 users, who do not need to rebuild the SDIDS.) You must also specify either the 18-byte or 34-byte SDIDS key, since only those two keys are supported. See "SDIDS Keys" on page 42 for more information.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3
SDIDS	Data structure changed	The JCL for the BLGUT1 utility has changed. The BLGSD and BLGSI DD statements are no longer supported. BLGUT1 requires a session to determine which SDDS and SDIDS to use. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details. If you use these statements, delete them and specify a valid/appropriate session.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Session parameters	**BLGPARMS macro modified**	A TIMEZONE keyword was added to enable you to implement universal time processing as an option. See "Implementing Universal Time Processing" on page 251 for more information. BLGPARMS macro keywords are also listed in "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318.	X	All previous releases.
Session parameters	BLGPARMS macro modified	A MODELDB keyword was added to enable you to specify a particular data model database and trigger character to identify data attribute records. If you elect to use data model records, specify this parameter. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details.		All previous releases
Session parameters	Enhanced options. Some parameters removed.	Become familiar with the current parameters. Parameters that you may have used in the past may have changed or become unavailable in this release. You must reassemble your session-parameters members using the latest macros in this version. See "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318 for more information.	X	All previous releases
VSAM data set characteristics	VSAM spanned records are no longer supported. Support is provided for extended attribute data sets.	You must rebuild or migrate your SDIDS using the BLGUT1 or BLGUT1M utility program respectively, because the internal data structure of the SDIDS has changed. The SDIDS can consist of multiple clusters. The number of records in your database is not limited to the SDIDS maximum record size. Determine if you can take advantage of multiple clusters to improve overall performance. There is no longer a need to make use of VSAM-spanned records, which are no longer supported. Only the 18- and 34-byte SDIDS key lengths are supported. Also, starting with Version 7.1, VSAM extended attribute data sets are now supported.	X	Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
		■ Determine what key length to use (18 or 34) ■ Allocate a new SDIDS with KEY(18 0) or KEY(34 0) and up to a RECORDSIZE(64 32752). Do not specify the SPANNED keyword. See sample member BLGDATAB in the SBLMSAMP sample library for an example. ■ Change the VSAM resources shared pool for your SDIDS to have KEYLEN=18 or KEYLEN=34. If you are not sharing databases, see sample BLXVDEF in Figure 31 on page 312. ■ Build your new SDIDS from your SDDS by using the BLGUT1 utility or the BLGUT1M migration utility. Guidance on setting up multiple clusters is available in “Working with Multiple-Cluster SDIDSs” on page 177. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using these utility programs.		

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
VSAM data set characteristics	Modified	Before you use previous version data sets with this new release, use the IDCAMS LISTCAT command to display the data set's attributes. If the data set's attributes (KEY, LOG, NOIMBED, SPANNED, and SHAREOPTIONS) are not compatible with this new release, use the IDCAMS ALTER command to change the attributes so that they are compatible. In some cases you can use the IDCAMS ALTER command to migrate a data set. Other times you have to define a new data set with the correct attributes and then migrate the data using either IDCAMS REPRO or the appropriate Tivoli Information Management for z/OS utility. If you change the key length and are not sharing databases, you may need to update the BLXVDEF resource definition member also. Refer to the *DFSMS/MVS Access Method Services for VSAM Catalogs* for information on the AMS (IDCAMS) commands. See "Defining Tivoli Information Management for z/OS Data Sets" on page 277 for information on current VSAM data set attributes. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on maintaining the data sets using the Tivoli Information Management for z/OS utility programs.	X	Information/Management 4.1, 4.2, 4.2.2, 5.1
VSAM resources	Each resource pool can be shared by 31 key-sequenced data sets.	Create VSAM resource definition members if you are not running Tivoli Information Management for z/OS in a sysplex. (That is, you are using NSR or LSR and not VSAM RLS.) RLS does not use VSAM resource definitions. If you are migrating from Information/Management Versions 6.1 through TME 10 Information/Management 1.1, or Tivoli Service Desk for OS/390 Version 1.2, be aware that SHARE=YES is no longer supported on the BLXDSN macro.	X	Information/Management 4.1, 4.2, 4.2.2

9. Migrating from Previous Versions

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
VSAM sequence numbers	Sequence numbers can be reused	If you want to continue reusing VSAM sequence numbers as records are deleted, you need to run the BLGUT9 utility to set the sequence number reuse option on for your database; otherwise, sequence numbers will not be reused and new records created in the database will be stored in record number ID (RNID) order. You should run the BLGUT9 utility after running the BLGUT1 utility only if you want to reuse sequence numbers. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for more details on the BLGUT9 utility. The *Tivoli Information Management for z/OS Operation and Maintenance Reference* describes a process you can follow to sort existing records in your database so that the entire database will remain in system-assigned RNID order. Having your database in RNID order can help to eliminate the need for users to sort search results lists.		All previous releases
You can perform the following actions in Step 10 of the general installation checklist in Chapter 1.				
Read panel data set	Data set supports 10-character external date format	The base product panels (in the SBLMPNLS data set) and PIDTs (SBLMFMT) are shipped to support any of the supplied external date formats. Decide if you want to use these panels with expanded (10 character) date fields as they exist. If you do, run the BLGUT6 utility to load the panels. If you need to convert the date fields to some other format, refer to "Enabling Alternative Date and Time-of-Day Formats" on page 227 for more information.		Customers using unmodified product panels (all previous releases)
You can perform the following actions in Step 15 of the general installation checklist in Chapter 1.				
ISPF command table	BLG0CMDS changed	If you modified the BLG0CMDS command table, you need to carry your changes forward into this latest version. You can use ISPF Dialog Tag Language (DTL) to accomplish this. Definitions for the following command names were added: UP, DOWN, RIGHT, LEFT, END, and HELP.	X	Information/Management 6.1, 6.2, 6.3;TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2
ISPF command table	New ISPF command table member BLG0CMDS for the enhanced panel style	Add BLG0CMDS to the ISPTLIB concatenation if you select the enhanced panel style. See "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199 for more information.		Information/Management 4.1, 4.2, 4.2.2, 5.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
ISPF keylists	BLG0KEYS unchanged	If you modified the BLG0KEYS keylist member, you need to carry your changes forward into this new release. You can use ISPF Dialog Tag Language (DTL) to accomplish this.	X	Information/Management 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
ISPF keylists	New ISPF keylist member BLG0KEYS for the enhanced panel style	Add BLG0KEYS to the ISPTLIB concatenation if you select the enhanced panel style. See "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199 for more information.		Information/Management 4.1, 4.2, 4.2.2, 5.1
ISPF panels	BLGISPFx panels unchanged	If you modified ISPF panels, you need to carry your changes forward into this new release. You can use ISPF Dialog Tag Language (DTL) to accomplish this. You must use the Tivoli Information Management for z/OS Version 7.1 level of these panels. Check for old copies of panels, especially BLGISPFD in ISPPLIB and ensure only the Version 7.1 panels are available. To be sure you've used the correct version of these panels, issue the HELP STATUS command. The WINDOW LEVEL should be JOYB103. Information/Management Version 6.1 introduced the enhanced panel style. The BLGISPFE, BLGISPFA, and BLGISPFI panels were added for the enhanced style.	X	Information/Management 6.1, 6.2, 6.3;TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2
ISPF panel style	Choose either the standard or enhanced panel style	Copy the panel style you choose into BLGISPFM. See "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199 for more information.		Information/Management 4.1, 4.2, 4.2.2, 5.1
You can perform the following action in Step 20 of the general installation checklist in Chapter 1.				
User-defined profile variables	Profile variables	BLGPVARS goes into ISPLLIB concatenation.		Information/Management 4.1, 4.2, 4.2.2
You can perform the following actions in Step 21 of the general installation checklist in Chapter 1.				
Macros	ISPF/PDF editor	If you are using the ISPF/PDF editor, you must include the ISPF BLGISMAC edit macro in a data set in each Tivoli Information Management for z/OS user's SYSPROC concatenation.		Information/Management 4.1, 4.2, 4.2.2
		You must also copy ISPF panel BLM@EDIT to the ISPF panel data set. This panel enables you to change the fields that are displayed on a freeform text panel when a user chooses the PDF editor.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
PDF editor	BLM@EDIT panel	You must copy ISPF panel BLM@EDIT to the ISPF panel data set. This panel enables you to change the fields that are displayed on a freeform text panel when a user chooses the PDF editor.		Information/Management 5.1, 6.1
You can perform the following action in Step 22 of the general installation checklist in Chapter 1.				
Invocation CLIST	TSXs are available for use	Update the invocation CLIST to allocate the BLGTSX data set (required). Refer to "Sample CLISTs to Start Tivoli Information Management for z/OS" on page 207, and the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* for more information on TSXs.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.2
You can perform the following action in Step 25 of the general installation checklist in Chapter 1.				
USERS record	Required for notification	Tivoli Information Management for z/OS uses a USERS record to assist in the notification function. The USERS record was changed in Information/Management Version 6.3. If one does not currently exist, you must create a USERS record to make full use of notification. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information on creating this record. **Note:** Your existing USERS record can be used with this new release.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2
You can perform the following actions after Tivoli Information Management for z/OS is installed and customized.				
API security	Enhancement	Determine if there are any existing API applications that can benefit from a security enhancement which allows the application ID to be changed on each transaction. If so, make the appropriate changes. Details are provided in the *Tivoli Information Management for z/OS Application Program Interface Guide*.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Command record	Migrate to Tivoli Information Management for z/OS Version 7.1	If you are using Information/Management Version 6.3 and have created COMMAND record(s), update the COMMAND record(s), run TSP BLGTCMIG, and file the record(s). BLGTCMIG contains a WORDFIX control line necessary to correct COMMAND records before they are used in Tivoli Information Management for z/OS Version 7.1. For more information on running TSPs, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.		Information/Management 6.3
Commands	TABLE	The minimum truncation for the TABLE command is now TA. Modify any TSPs and SRCs that use the old minimum truncation value of T.		Information/Management 4.1, 4.2, 4.2.2, 5.1
Database	SDDS Key Format	If you are using a key 8 format SDDS, now is the time to consider switching to a key 7 format to take advantage of new functions. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for a discussion of the advantages. Use the BLGUT7 utility to convert the SDDS to a key 7 format. The key 8 format may not be supported in future releases.		All previous releases
HLAPI	Conversation sharing disabled	Conversation sharing is automatically disabled when you start the requester program for the HLAPI/2, HLAPI/NT, HLAPI/UNIX (AIX, HP, or Solaris), or HLAPI/USS clients. To enable conversation sharing, specify the IDBSHARECMS keyword in the client API system profile. Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for more information about using this keyword.		Information/Management 6.1, 6.2, 6.3
HLAPI	HL01 initialization	For maximum performance of API applications using the HLAPI, consider increasing the TABLE_COUNT value to include inquiry PIDTs, which can now be cached. Refer to the description of the HL01 transaction in the *Tivoli Information Management for z/OS Application Program Interface Guide*.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
HLAPI	HL11 record inquiry	An API error is no longer returned by the HLAPI when an application attempts to perform an HL11 record inquiry transaction without supplying a search selection criteria on the input PDB. If no search selection is supplied, all records in the database are returned that match the record type defined by the PIDT or data view record specified on the inquiry.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3
HLAPI	Changed linkage requirements	Review description of HLAPI which follows this table to see if your applications need to be link-edited differently or to see if your applications require modification.		Information/Management 4.2.2
HLAPI	New return and reason codes for existing condition	A return code of 4 and a reason code of 14 now appear when a request for record retrieval (HL06) specifies TEXT_OPTION=YES, TEXT_MEDIUM=D, and an input PDB chain that lists one or more freeform text items.		Information/Management 4.2, 4.2.2, 5.1, 6.1
HLAPI	PIDTs were modified to support DBCS data	Use BLGUT8 to migrate your PIDTs. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information on using BLGUT8.		Information/Management 4.2, 4.2.2, 5.1
HLAPI	PIDTs were renamed	All PIDT sample names starting with TSO were renamed. See Table 23 on page 360 for the changed names.		Information/Management 4.2, 4.2.2, 5.1, 6.1
HLAPI	High memory support	Decide whether your company will benefit from the HLAPI returning data above the 16 MB line. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for complete information about using this enhancement.		Information/Management 4.2, 4.2.2, 5.1, 6.1
List processor	Line commands changed	The line commands for list processor panels are changed for Information/Management Version 6.2 and subsequent releases. For a complete description of the changes, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*. You should inform your users of these changes. You might also need to update any TSPs that use line commands on list processor panels.		Information/Management 4.2, 4.2.2, 5.1, 6.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
LLAPI	Suppression of audit data	If you use LLAPI transaction T100 (Retrieve Record) and wish to continue receiving audit data associated with freeform text, you may need to modify your application. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information about T100.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
LLAPI	Changed linkage requirements	Review the LLAPI description which follows this table to see whether your applications require modification or need to be link-edited differently.		Information/Management 4.2, 4.2.2
LLAPI	Expanded freeform searches	If you have applications that use the LLAPI and want to perform freeform searches using individual freeform arguments, modify your declaration of the program interface argument table (PIAT) in your application and recompile your application to include the modified PIAT.		Information/Management 4.2, 4.2.2, 5.1
LLAPI	High memory	Decide whether your company will benefit from the LLAPI returning data about the 16 MB line. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for complete information about using this enhancement.		Information/Management 4.2, 4.2.2, 5.1, 6.1
LLAPI	PICA	If you have applications that use the LLAPI, review the new fields in the PICA and ensure that your application initializes them appropriately. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for the structure of the PICA.		Information/Management 4.2, 4.2.2, 5.1, 6.1
LLAPI	PIDTs were renamed	All PIDT sample names starting with TSO were renamed. See Table 23 on page 360 for the changed names.		Information/Management 4.2, 4.2.2, 5.1, 6.1
LLAPI	Value change for command processing detection entry	The value for command processing detection in the user profile determines how an assisted-entry command reply is handled. The default value of this entry is changed from PROMPT to DATA during initialization of the LLAPI.		Information/Management 4.2, 4.2.2, 5.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Mixed case field data	New support for field-level data	By default, field data is stored and cognized in uppercase. Now, using PMF, you can collect, store, cognize and display data in mixed case. See the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on how to specify how data should be collected and stored.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.3; TME 10 Information/Management 1.1
MRES	Easier startup	If you use Multiclient Remote Environment Servers, specify your MRES startup parameters in a data set rather than inline in your cataloged procedure. Many of the startup parameters have been renamed. Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for more information about specifying parameters in the new format.		Information/Management 6.2, 6.3; TME 10 Information/Management 1.1
MRES	TCP/IP address space name	If you are using the MRES with TCP/IP, you no longer need to specify the TCP/IP address space name parameter in your startup parameters. Your RACF systems programmer must add the MRES started procedure name to the RACF STARTED class, specifying a user ID that has an OMVS segment. (Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for more information about using an MRES with TCP/IP.)		Information/Management 6.2, 6.3; TME 10 Information/Management 1.1
MRES	Pre-started sessions for faster startup	For faster initialization, consider pre-starting the API sessions by specifying the appropriate parameters in the MRES startup parameters data set. Refer to the *Tivoli Information Management for z/OS Client Installation and User's Guide* for more details on configuring and running the MRES.		Information/Management 6.2, 6.3; TME 10 Information/Management 1.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
MRES	API Security	If you are using a pre-started MRES session and APISECURITY=ON is specified in the BLX-SP parameters member, your RACF system programmer must add the user ID associated with the MRES to the access list of the general resource used as the application ID. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information about API security.		Information/Management 6.2, 6.3; TME 10 Information/Management 1.1
NetView Hardware Monitor Interface	BLMVATSR and BTNCNMBD modified	Carry forward any customizing you did in previous versions. See the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications* for more customization information.		Information/Management 4.1, 4.2, 4.2.2, 5.1
Panels	Updated compression algorithm	If you have modified file-time processing, review your panels. The last dialog end collected for each record type must also have an s-word index of 0CF1.		Information/Management 4.1
Panels	Updated panels	Review changes you made to Tivoli-supplied panels and use PMF to carry them forward. See "New, Changed, and Removed Panels" on page 369 for a list of panels that are new, changed, or removed. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details about panels and PMF. The product panels are shipped with dates in 10-character external date format. If you need to apply your changes to panels with 8-character date formats instead, you must use the BLGUT6M migration utility, or PMF, to change the panels to support 8-character date fields.	X	All previous releases
Privilege Class Authorities	Additional authorities	Add new authorities to existing privilege class records. You do not need to delete and recreate these records. Privilege class records for Information/Management Versions 4.1 through 4.2.2 work with this new version, but users do not have authority for new functions until the privilege class records are updated.		Information/Management 4.1, 4.2, 4.2.2

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Program exit BLG01385	Authorization code 0001	Two program exits now govern how list processor rows are sorted: BLG01385 (ascending sort order) and BLG01386 (descending sort order). Formerly, the BLG01385 program exit was used to specify a descending sort order. If you used the BLG01385 program exit in the past, your panels will automatically be migrated to use BLG01386 instead. If you are accustomed to using the BLG01385 program exit with authorization code 0001 in the control flow processing of your panels, you should now use BLG01386 to achieve a descending sort order. More information about these program exits is available in the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
Program exit BLG01439	Authorization code	Do not specify an authorization code value of 0004 on your control panel. This value is now reserved.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3
Report Format Facility	PUT statements changed	The PUT statements in the supplied RFTs were changed to accommodate a LENGTH(10) instead of a LENGTH(8) for date fields. The column placement of output generated by the reports may shift to accommodate this change. Review your programs and modify as necessary.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3
Report Format Facility	Use exits called by the Report Format Facility CALL command must now be link-edited as reusable (REUS).	Link-edit user exits again. Otherwise, the MVS LINK macro that is used loads a new copy of the exit every time the exit is called. This may cause an ABEND to occur.		Information/Management 4.1, 4.2, 4.2.2
Stored response chains	Search argument changes for different SDIDS key lengths	SRCs containing fully qualified search arguments written for an SDIDS with a 16- or 18-byte key may produce different search results if the key length of the SDIDS is changed to 34 bytes. If you want to use the 34-byte key, revise your SRCs to do searching. Refer to the *Tivoli Information Management for z/OS User's Guide* for a discussion of how the SDIDS key affects searching.		All previous releases

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Terminal Simulator Panels (TSPs)	AutoBridge users	If you use the AutoBridge PostProcessor feature, you must, if you haven't already done so, modify the BLGAPI00 TSP to delete the branch control line just before the EYMSP010 USEREXIT. Removing the branch line will enable the PostProcessor to detect and process AutoBridge-created records. Details are provided in the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
Terminal Simulator Panels (TSPs)	TSPs BLGAPI00, BLGAPIDI, BLGAPI05 modified	TSPs BLGAPI00 (LLAPI router TSP for panel processing) and BLGAPIDI (LLAPI router for bypassing panel processing) are modified to invoke user exit BLGYITSP. If you are using and have customized these TSPs, you should review them and make changes if necessary. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for details on these TSPs. Also, if you have an API application that updates records and does not use bypass panel processing, or if you use the Archiver function, review changes to API TSP BLGAPI05 and make any necessary changes to your user-modified copy of TSP BLGAPI05.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Terminal Simulator Panels (TSPs)	Support for multiple external date formats	Because users can set different date formats through the user profile, any TSPs or TSXs that you have written that enter or process dates must support different external date formats for different users. All processing in the TSP or TSX should use internal format and should convert to and from the user's external format when interacting with the database. The TSPs or TSXs should be written to use the 10-character internal date format (YYYY/MM/DD) for processing date data. (Alternately, the option to use multiple formats should be taken away from the user by removing the field from BLG0P700). Use the BLGIDATE and BLGEDATE user exits to convert the internal format date from or to the user's local date format. The BLGIDATE user exit converts a date from the user's local date format to internal format. The BLGEDATE user exit converts a date from internal format to the user's local date format. For more information about these user exits, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.	X	Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1; Tivoli Service Desk for OS/390 1.2

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Terminal Simulator Panels (TSPs)	Notification TSPs/TSXs modified	If you use Notification Management and have customized any of the following TSPs/TSXs, review your TSPs/TSXs and update them as necessary for modifications related to the Assignee Name field: TSPs – for problem records – BLGTSPPE, BLGTSPPU; for change records – BLGTSPCE, BLGTSPCU; for activity records – BLGTSPAE, BLGTSPAU; for Integration Facility – BLGTCN06, BLGTPN02; TSX – for e-mail notification –BLGTXNOT. The Assignee Name field in problem, change, and activity records is now a string field that accepts blanks. The length of the Assignee Name field was also changed to hold up to 40 characters (including blanks). However, the base product panels were not changed to show a longer field length. The notification TSPs/TSXs were modified to search for the Assignee Name as string data (up to 40 characters including blanks) first. If no string data is found, a search is performed for non-string field data. For example, in BLGTSSPE, review the following: lines 28, 29 (added), line 32 (changed), lines 41, 42, and 45 (added). Similar changes were made to the other TSPs listed to search for the Assignee Name as string data first before searching for it as non-string data. In BLGTXNOT, review the additions made (lines 175 and 176). For more information about using Notification Management, refer to Appendix D of the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
Terminal Simulator Panels (TSPs)	Escalation TSPs modified	If you use Escalation Management and have customized TSPs BLGESC04 and BLGESC06, then review these TSPs and update them as necessary. For more information about using Escalation Management, refer to Appendix D of the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Terminal Simulator Panels (TSPs)	TSP user exits can now use 31-bit addressing (AMODE=31)	Review your TSP user exits and make changes if necessary. Enhanced TSP control lines became available with Information/Management Version 5.1. Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* for details on these changes.		Information/Management 4.1, 4.2, 4.2.2
Terminal Simulator Panels (TSPs)	TSPs using ISPEXEC	You may need to use a TSP LINK control line instead of the TSP ISPEXEC control line to run a REXX program in most cases. Review any TSPs that use an ISPEXEC control line with the syntax SELECT CMD(rexxpgm), where rexxpgm is a REXX program. Change the ISPEXEC control line to a TSP LINK control line and use the rexxpgm name in the 'Terminal simulator name' field of the TSP LINK control line.		Information/Management 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3; TME 10 Information/Management 1.1
User profiles	Enhanced profile options	Become familiar with the options and modify if necessary. Values from your Information/Management Version 4 or later profile are carried forward into your new profile the first time you access the new release. Refer to the *Tivoli Information Management for z/OS User's Guide* for more information.		Information/Management 4.1, 4.2, 4.2.2, 5.1
User profiles	Fields on panel BLG0P100 are rearranged and renumbered	Two versions of panel BLG0P100 are available. The default is shown in Figure 8 on page 138. You can use PMF to replace the default with an optional panel that is not renumbered (member BLG0P101 in SBLMPNLS—see Figure 9 on page 139). Be sure to update any TSPs or final responses of SRCs that specify profile option numbers. It is recommended that you use s-words instead of field numbers when you create TSPs.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1
Utility BLGUT1	DD statements deleted	The BLGSD and BLGSI DD statements were deleted in TME 10 Information/Management Version 1.1. The SDIDS and SDDS are determined from the session-parameters member. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* and the description of the BLGCLUST macro in this manual for more details.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2, 6.3

Table 5. Migrating from Previous Versions (continued)

Topic	New / Modified Function	Migration Action	New	Applies to those Migrating From
Utility BLGUT4	DD statements changed	The BLGBKTM and BLGBKIN DD statements were added in Information/Management Version 6.3. Review your JCL and make any necessary corrections. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details.		Information/Management 4.1, 4.2, 4.2.2, 5.1, 6.1, 6.2
Web applications	SBLMHTML changed	The data set SBLMHTML has been renamed SBLMHTMV, and the data set's attributes have been changed for greater flexibility. SBLMHTMV is now variable-blocked with an LRECL of 512. If you have been using Web applications with Tivoli Information Management for z/OS prior to installing Version 7.1, you may need to change references in the directory you created for IBM-supplied HTML. For more information, see Chapter 12 of the *Tivoli Information Management for z/OS World Wide Web Interface Guide*.	x	All previous releases

The following section provides additional information or notes on the migration topics listed in the table.

BLX-SP parameters member

Information/Management Version 5.1 introduced the BLX-SP parameters member. When new parameters for the BLX-SP parameters member are provided, you do not have to assemble the BLX-SP parameters members because they are read by BLX-SP during its initialization phase. Refer to the online introduction of the Installation Tailoring Facility or see "Defining BLX-SP Parameters Members" on page 343 for a description of these parameters.

Note: Although it is not necessary to assemble the BLX-SP parameters members, you must, if you are using VSAM NSR or LSR, create a VSAM resource definition member, assemble and link-edit it. VSAM resource definition members are not used if you are installing Tivoli Information Management for z/OS in a sysplex environment and using VSAM RLS.

Date formats

The DATEFMT keyword on the BLGPARMS macro is required if DATECNV=BLGCDATS or is omitted. Before you can reassemble the session-parameters members, you must explicitly specify a default external date format on the DATEFMT keyword if you use the BLGCDATS date conversion routine (or take it as a default). For instructions on specifying date formats for your session-parameters members, see "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318 for details.

The only available internal date format is YYYY/MM/DD. To change an existing database that is currently using the YY/MM/DD internal date format to use the expanded internal date format (YYYY/MM/DD), you must:

- Reassemble all session members.
- Delete and redefine the SDIDS. (The SDIDS must be empty in order to run BLGUT1.)
- Run BLGUT1 with one of the reassembled session members.
- Modify any TSPs, invocation IRCs, and SRC final responses that specify freeform search arguments to use the expanded date format.
- Inform your users that freeform search arguments including dates must use the expanded format. For example,

```
SE DATD/1995/01/01 -2001/12/31
```

HLAPI and LLAPI

If your HLAPI application or low-level API uses a load and call macro, it must be link-edited AMODE (31) RMODE (ANY). If it uses the MVS LINK macro instead, these addressing and residency mode requirements do not apply.

AMODE
Addressing mode

RMODE
Residency mode

Mixed case field data

You can collect, store, and display field information in mixed case. You can store field data in these formats:

- Uppercase
- Lowercase
- First character uppercase, rest lowercase
- By a pattern
- Exactly as entered by the user

These choices apply to field data and string data (e.g., a Description field), but not to freeform text data.

By default, the panels shipped with Tivoli Information Management for z/OS collect data in uppercase. If you want to take advantage of mixed case support, you should:

- Identify the fields for which you want to store data in mixed case in the SDDS. Be careful not to change any fields that collect data such as record IDs (RNIDs), z/OS data set names, or partitioned data set member names, which must be in uppercase. (Unpredictable results could occur if these particular fields are in mixed case.)
- Modify the assisted-entry panels associated with the chosen fields. In some cases, there are several assisted-entry panels associated with a field (for example, one for data collection, one for search argument collection, and so on). PMF reports such as the P-Word List, P-Word Cross Reference, and S-Word Cross Reference may help you to determine what panels need to be changed.

- Ensure that any programs which process data retrieved from Tivoli Information Management for z/OS are updated, if necessary, to properly handle mixed case data for the selected fields. For example, if your organization makes use of API programs to extract host data to produce reports, you need to ensure those programs are also compatible.
- If you choose to cognize data in mixed case, ensure that all RFTs, TSPs/TSXs, control panels, and API programs that perform searches or tests on the data are updated to do mixed case searches or tests.
- If necessary, use WORDFIX control lines in a TSP to change the case of data which is already in the database in uppercase.
- Also, if you cognize data in mixed case, ensure that users who may use freeform or combined searches to retrieve the data are informed of the changes so that they can alter their search arguments if necessary. The *Tivoli Information Management for z/OS User's Guide* provides more details on entering such search arguments.

For instructions on how to validate the case at data entry, store data, and cognize data in mixed case, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

Panels

If you used PMF to modify a panel that is changed by Tivoli Information Management for z/OS Version 7.1 you must review your changes and carry them forward into the current panel set. See "New, Changed, and Removed Panels" on page 369 for a list of panels that have recently been added, changed, or removed. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details about panels and PMF.

Read panel data sets

The base product panels are shipped in a single read panel data set (SBLMPNLS) and support all external date formats listed in "External Date Format" on page 231. The steps you need to take to install the panels will depend on the date formats you need in your environment (the steps are summarized below). "Enabling Alternative Date and Time-of-Day Formats" on page 227 contains more detailed instructions.

Guidelines on date processing for new customers are available in "Four-Digit Year Considerations" on page 237.

- To use the panels as shipped (uncustomized):

  Run the BLGUT6 utility to load the read panel data set (assuming you want to use the NN/NN/NNNN validation pattern).

  The normal SMP/E installation you perform as described in the *Program Directory* handles installation of the PIDTs.
- To convert from 10- to 8-character date formats:
  1. Run the BLGUT6M migration utility to load the read panel data set and convert the date fields to 8 characters. Use the BLGDATE8 sample JCL with the BLGUT6M utility.
  2. Update the example text shown on assisted-entry panels.

3. If you use APIs, run the BLGUT6F utility to offload the modified panels, and then run the BLGUT8 utility to rebuild the PIDTs to use 8-character dates. (If you create PIDTs using data model records, you do not have to rebuild them.)

   As an alternative, you can change the API applications to specify a particular date format. For information on setting the API option to have all dates converted to or from a particular format, regardless of the date format used by the Tivoli Information Management for z/OS host database, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
4. Disable any date formats that your panels do not support. Modify the data attribute record for the date format profile field (BLG&DFMT) to remove the unsupported formats.

Note: If you convert from 10– to 8–character date formats, you will limit the ability of your users to choose some date formats.

- To convert to some other date format:
  1. Copy the BLGDATE8 sample JCL and update the `LENGTH` and `VALIDATION` input statements to reflect your format.
  2. Run the BLGUT6M migration utility to load the read panel data set and change the date field lengths on data entry and table panels, and validation patterns in assisted-entry panels.
  3. If you use APIs, run the BLGUT6F utility to offload the modified panels, and then run the BLGUT8 utility to rebuild the PIDTs to use the correct character length for dates. (If you create PIDTs using data model records, you do not have to rebuild them.)

     As an alternative, you can change the API applications to specify a particular date format. For information on setting the API option to have all dates converted to or from a particular format, regardless of the date format used by the Tivoli Information Management for z/OS host database, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
  4. Update the example text shown on assisted-entry panels.
  5. Migrate your offloaded panel data set using the BLGUT6M utility.

  Note: If you convert to some other date format, you will limit the ability of your users to choose some date formats.
- To modify your customized panels to support all date formats:
  1. Identify what you need to change on your panels.
  2. Offload your customized panel data set by using the BLGUT6F utility.
  3. Load the base product panels using the BLGUT6 utility.
  4. Migrate your offloaded panel data set using the BLGUT6M utility.

See “Modifying Customized Panels to Use 10-Character Date Fields” on page 237 for more detailed instructions.

Update any report format tables that you have created, APIs, TSPs, etc. that use dates to handle dates in the new format. The base product RFTs are already updated to reflect 10-character dates.

Inform your users in advance of the change so they know what to expect.

If you have existing records in your database that use an external date format that is different from the default external date format specified on the BLGPARMS DATEFMT keyword, you can specify the BLGPARMS ODATEFMT keyword to have Tivoli Information Management for z/OS automatically recognize the dates in those records so that they can be processed with the latest date processing enhancements.

In summary, if you are using modified panels, you need to perform one of the following sets of operations:

- Modify your panels to accept all of the product-supported date formats by doing all of the following:
  - Change all date fields on data entry panels, table panels, RFTs, and so on, to 10 characters (if they are not already).
  - Change validation patterns on all date assisted-entry panels so that any valid date format can be entered. It is suggested that you use IIV63 to allow for possible new formats in the future, but IIV9 would work for all current formats.
  - Change the text and help information for all date assisted-entry panels so as to make them independent of any one date format. See the following panel BLG60CCD for an example.

```
      + BLG60CCD ------------ PROBLEM OCCURANCE DATE ------------- DATO/-+
      |                                                                  |
      | USE....Enter the date that the problem occurred or was detected. |
      |                                                                  |
      | FORM...Date in your external format (e.g. MM/DD/YYYY) -or-       |
      |        = for today's date or an offset from today's date.        |
      | NOTE...Enter ;HELP STATUS to find your external date format.     |
      |                                                                  |
      | EXAMPLES: May 27, 2001..Reply...05/27/2001, etc.                 |
      |           Today.........Reply...=     1 month ago...Reply...=-1M |
      |           Yesterday.....Reply...=-1   2 years ago...Reply...=-2Y |
      |           2 weeks ago...Reply...=-2W                             |
      |                                                                  |
      |                                                                  |
      +---------------------- REPLY AS ILLUSTRATED---------------------+




===>
```

- Limit the user's options for changing the date format by doing one of the following:
  - Remove **option 5. User date format** from profile panel BLG0P700.

- Modify the list of valid formats in attribute record BLG&DFMT to only include those supported by your panels. For example, if you have 8–character date fields on your data entry panels, you would remove all formats that are longer than 8 characters.

Ensure the DATEFMT keyword is specified with the correct external date format.

User-defined profile variables

Before you can create user-defined profile variables, the ISPF BLGPVARS panel must have already been created by the Tivoli Information Management for z/OS system administrator. The ISPF BLGPVARS panel must be included in a data set in the ISPPLIB concatenation. For further details about this panel, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

User profiles

Note: This applies to Information/Management Versions 4.1 through 6.1 only. Refer to the *Tivoli Information Management for z/OS User's Guide* for more information on the latest user profile options.

The new BLG0P100 panel is as follows. Be sure to update any TSPs or final responses of SRCs that specify profile option numbers. It is recommended that you use s-words instead of field numbers when you create TSPs.

```
BLG0P100                    SESSION DEFAULTS                 USER ID: HELENA

Enter session default data; cursor placement or input line entry allowed.

 ------------- INVOCATION -------------   ------- OUTPUT DESTINATION --------
  1. Application.........<R> MANAGEMENT   31. Print................. ______
  2. Class.................. ________     32. Standard report....... ______
  3. SRC.................... ________     33. Draw.................. ______
  4. Bypass copyright?...<R> NO_

 -------- COMMAND PROCESSING ----------   --------- SEARCH OPTIONS ----------
 11. Detection...........<R> PROMPT_      41. Default panel......... ________
 12. BACK operation......<R> PROCESSED    42. Quick search?......<R> NO_
 13. RECALL operation....<R> CMDLINE      43. User line commands.<R> RUN_
 14. RECALL stack depth..<R> 10
 15. PRINT operation.....<R> KEEP
                                          --------- EDITOR OPTIONS ----------
 ----------- MISCELLANEOUS ------------   51. Editor selection...<R> INFO
 21. PF key data.........<R> SUFFIX_      52. INFO for SRCs/TSPs?<R> YES


    When you finish, type END to save or CANCEL to discard any changes.

===>
```

Figure 8. BLG0P100 Panel

The following panel contains the old numbering scheme for the profile option fields. If you want to use this panel, use PMF to copy panel BLG0P101 into your modified panel data set as panel BLG0P100.

```
BLG0P101                    SESSION DEFAULTS                 USER ID: HELENA

Enter session default data; cursor placement or input line entry allowed.

 ------------- INVOCATION -------------   ------- OUTPUT DESTINATION --------
  1. Application.........<R> MANAGEMENT    6. Print................. ______
  2. Class.................. ________      7. Standard report....... ______
  3. SRC.................... ________     12. Draw.................. ______
 17. Bypass copyright?...<R> NO

 -------- COMMAND PROCESSING ----------   --------- SEARCH OPTIONS ----------
  4. Detection...........<R> PROMPT_      13. Default panel......... ________
  5. BACK operation......<R> PROCESSED    14. Quick search?......<R> NO_
 10. RECALL operation....<R> CMDLINE      18. User line commands.<R> RUN_
 11. RECALL stack depth..<R> 10
 19. PRINT operation.....<R> KEEP
                                          --------- EDITOR OPTIONS ----------
 ----------- MISCELLANEOUS ------------   15. Editor selection...<R> INFO
  9. PF key data.........<R> SUFFIX_      16. INFO for SRCs/TSPs?<R> YES


    When you finish, type END to save or CANCEL to discard any changes.

===>
```

Figure 9. BLG0P101 Panel

Additional Considerations for Migrating to This New Release

Compatibility with Information/Management and Information/System Version 4.2 and 4.2.2

Please also consider the following information pertaining to your specific release.

- Information/MVS databases and the Information/Access product database (previously known in an earlier release of Information/Management as database 6) are not supported. User-defined format databases (previously known as the Information/MVS format - databases 1, 2, and 3) are still supported. In Tivoli Information Management for z/OS Version 7.1, database 0 is also available as a user-defined format database. Database 6 is now used as a Tivoli Information Management for z/OS read-only database and is reserved for Tivoli Inventory use only.
- A report that is running cannot lock the SDDS for longer than the time it takes to process a single record.
- To ensure compatibility between Information/System Version 4.2.2 and this new release, you must do the following:
  - If you are not installing Tivoli Information Management for z/OS in a sysplex, you should define, assemble, and link-edit a VSAM resource definition member to specify VSAM options for Tivoli Information Management for z/OS, including LSR pool definitions and lists of data sets for which LSR is used. This information was in the user's session-parameters members (BLGSESaa) in Information/System Version 4.2.2.

    Refer to the online introduction of the Installation Tailoring Facility or see "Defining VSAM Resources to the BLX-SP in a Non-Sysplex Environment" on page 301 for more information about VSAM resource definition members.
  - Assemble and link-edit all user session-parameters members again.

- Define a started task procedure. See "Starting Tivoli Information Management for z/OS" on page 203 for instructions.
- Modify z/OS system parameters, where required. Do a system IPL before accessing databases with Tivoli Information Management for z/OS Version 7.1 for the first time. This step is required to start BLX-SP.

- If your SDDS consists of one VSAM cluster and you want to turn it into a multiple-cluster SDDS (either during initial installation of the product or at a later date), you must run the database migration utility (BLGUT7) against your current SDDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for instructions on using BLGUT7. A key length of 7 is recommended for the SDDS so that the BLGUT23 utilities can be used. If your SDDS is currently key length 8, you should consider changing to key length 7 when you have a need to run the BLGUT7 utility (such as when moving from one to multiple clusters).

  You must modify, reassemble, and relink-edit user session-parameters members and the VSAM resource definition member at the time of this migration.

  Note: The modification, reassembly, and relink-edit of the VSAM resource definition member does not apply if you are installing Tivoli Information Management for z/OS in a sysplex and using sysplex support.

- At least one of the Tivoli Information Management for z/OS panel style members must be placed in the ISPPLIB concatenation. See "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199.
- Panels you created that have DBCS data must be loaded using BLGUT6 with the DBCS keyword specified.
- Panels you created that do not have DBCS data can be processed in the old format by Tivoli Information Management for z/OS. If you want to convert them to the new format, load them using BLGUT6 with the NODBCS keyword specified (or accept it as the default).

  Note: You can also use BLGUT6F to download panels, and then use BLGUT6 with the NODBCS (DBCS, if there is DBCS data) keyword specified to replace the panels.

- If you decide to use graphic character substitution, you must review and modify existing reports or API programs so that they reflect the substitutions. You must also make sure that characters in the reports that come with Tivoli Information Management for z/OS are changed to reflect the substitutions. See "Graphic Character Substitutions" on page 46 for more information about character substitution.
- Review how time is validated with program exit BLG01052 and ensure the validation patterns are correct in assisted-entry panels associated with time fields. Time fields with p-word TIM/ are validated as HH:MM. Time fields with any other p-word are validated as DD:HH:MM. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on this program exit.

Compatibility with Information/Management Version 5.1

- At least one of the panel style members must be placed in the ISPPLIB concatenation. See "Selecting the Tivoli Information Management for z/OS Panel Style" on page 199.
- Panels you created that have DBCS data must be loaded using BLGUT6 with the DBCS keyword specified.

- Panels you created that do not have DBCS data can be processed in the old format by Tivoli Information Management for z/OS. If you want to convert them to the new format, load them using BLGUT6 with the NODBCS keyword specified (or accept it as the default).

  Note: You can also use BLGUT6F to download panels, and then use BLGUT6 with the NODBCS (DBCS, if there is DBCS data) keyword specified to replace the panels.
- Review your VSAM data set share options and make the appropriate changes. Tivoli Information Management for z/OS enforces a SHAREOPTIONS value of 1 for nonshared data sets. If you are sharing data sets in a sysplex, be aware that Tivoli Information Management for z/OS and VSAM RLS ignore the SHAREOPTIONS value.
- Verify the CI size on your VSAM data sets using IDCAMS LISTCAT (or another equivalent tool). Ensure the CI sizes shown by IDCAMS LISTCAT are used in your VSAM resource definition file.
- Review how time is validated with program exit BLG01052 and ensure the validation patterns are correct in assisted-entry panels associated with time fields. Time fields with p-word TIM/ are validated as HH:MM. Time fields with any other p-word are validated as DD:HH:MM. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on this program exit.

Compatibility with Information/Management Version 6.1

- Review how time is validated with program exit BLG01052 and ensure the validation patterns are correct in assisted-entry panels associated with time fields. Time fields with p-word TIM/ are validated as HH:MM. Time fields with any other p-word are validated as DD:HH:MM. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on this program exit.

10

Setting Up Your BLX-SP

This chapter provides information for setting up your BLX-SP. The following tasks are described:

- Modifying the Program Properties Table
- Adding a Tivoli Information Management for z/OS load library to the APF list
- Defining subsystems for BLX-SPs
- Defining a BLX-SP procedure
- Defining a BLX-SP parameters member
- Defining a VSAM resource definition member

  Note: You do not have to define a VSAM resource definition member if you are installing Tivoli Information Management for z/OS in a sysplex and using sysplex support.

- Defining multiple BLX-SPs
- Starting and stopping a BLX-SP

This chapter assumes that you have a working knowledge of your z/OS system environment. Refer to the *OS/390 MVS Initialization and Tuning Reference* and *OS/390 MVS: System Commands* for more information.

This chapter is designed to help you enable your z/OS system for Tivoli Information Management for z/OS.

Programming Interface information

Modifying the Program Properties Table

You must add an entry to the Program Properties Table to make the BLX-SP program entry nonswappable. To do this, add the following to the appropriate SCHEDaa members of SYS1.PARMLIB:

```
PPT PGMNAME(BLXSSP00)  /* BLX SERVICE PROVIDER                    */
        NOSWAP         /* PROGRAM IS NOT SWAPPABLE                */
        KEY(8)         /* PROGRAM IS A KEY(8) TASK                */
```

If you do not make the BLX-SP program entry nonswappable, users who attempt to access the BLX-SP may hang or receive an ABEND or experience unpredictable results.

Note: To implement the changes to the Program Properties Table, you must either IPL the system or change the SYS1.PARMLIB SCHEDaa member dynamically using the SET command. Check with your system programmer to identify other members you can modify dynamically.

Adding a Tivoli Information Management for z/OS Load Library to the APF List

To define a Tivoli Information Management for z/OS load library to the MVS element of z/OS, specify it as an authorized program facility (APF) library in the appropriate IEAAPFaa or PROGaa members of SYS1.PARMLIB. Each entry in an IEAAPFaa or PROGaa member includes the data set name (*dsn*) and the volume serial number (*volser*) of the library.

You may have one or more libraries that contain Tivoli Information Management for z/OS load modules. The data set name of your main Tivoli Information Management for z/OS load library can be found by checking DDDEF SBLMMOD1 using SMP/E. If you have multiple BLX-SPs, then the name of each data set that contains a BLXSSINM module (see "Defining a Subsystem to MVS" for information about this module) must also be added to your APF list.

Note: Changes to the PROGAA member can be implemented with the SET command. Changes to the IEAAPFAA member require an IPL of the system. Refer to the *OS/390 MVS Initialization and Tuning Reference* for more information about the SYS1.PARMLIB data set and its members and parameters.

Defining Subsystems for BLX-SPs

During Tivoli Information Management for z/OS installation, you must identify each of your BLX-SPs as an MVS subsystem. You must give each of your BLX-SPs a subsystem name, define each subsystem to MVS, and define each subsystem to its respective BLX-SP.

Naming a Subsystem

The subsystem names you use must follow these naming rules:

- The name must have 4 characters.
- The first character of each name must be an alphabetic or national (#, $, @) character.
- The remaining characters can be alphabetic, numeric, or national characters.

Notes:

1. Use the BLX-SP subsystem name as the first 4 characters in the names of each BLX-SP parameters member that you use with the subsystem.
2. If you want to use the same name for the BLX-SP procedure and the BLX-SP subsystem, specify the SUB=JESx parameter on the START command for the procedure (where JESx is the name of your JES subsystem). See "Starting and Stopping the BLX-SP" on page 148 for the syntax of the START command.

Defining a Subsystem to MVS

The following example defines a subsystem, named BLX1, to the MVS system. The statement contains only the subsystem name and must be added to the appropriate IEFSSNaa members of SYS1.PARMLIB. Do this step for each subsystem using the following syntax depending on your level of z/OS.

```
BLX1

- or -

SUBSYS SUBNAME(BLX1)
```

Do not specify an initialization routine in the IEFSSNaa member for the subsystems. Subsystem initialization is deferred until the BLX-SP starts.

Note: To implement most changes to the SYS1.PARMLIB, you will have to IPL the system. You can modify some SYS1.PARMLIB members dynamically; those do not require you to IPL the system. Check with your system programmer to identify those members.

Defining a Subsystem to the BLX-SP

All Tivoli Information Management for z/OS initialization code uses a load module named BLXSSINM that defines the subsystem to be used by the BLX-SP and all user Tivoli Information Management for z/OS sessions. A sample of this module is provided in the SBLMMOD1 library.

The sample uses the subsystem name BLX1. If you have just one BLX-SP, you can use this sample module. The sample is already assembled and link-edited for you. If you have multiple BLX-SPs, you must define, for each BLX-SP, a unique load library containing, at a minimum, a copy of the BLXSSINM module. Make sure that the load libraries are specified in the STEPLIB DD statements of the respective BLX-SP procedures. See “Defining a BLX-SP Procedure” on page 146 for more information about the STEPLIB DD statement. See “Performance Recommendations” on page 70 for performance considerations for the STEPLIB DD statement.

You can also specify the load libraries in the link list. However, if you have multiple BLX-SPs on your system, you must also specify STEPLIB DD statements in your BLX-SP procedures. Otherwise, the system uses the first BLXSSINM module that the link list points to.

Note: You can put a BLXSSINM module in the link pack area, but, because the load modules for the BLX-SPs all have the same name, only one module can reside in the link pack area.

BLXSSINM is an assembler language module that contains only the subsystem name of your BLX-SP. The source code (shown below) for this module is in the SBLMSRC1 library.

```
BLXSSINM  CSECT
          DC        CL4'BLX1'      Define the BLX subsystem name.
          END
```

Make sure that each copy of BLXSSINM contains a different 4-character subsystem name. To change the subsystem name that is in the BLXSSINM file, do the following:

- Modify the BLXSSINM source code.
- Reassemble the source (no macros required).
- Relink-edit the text into BLXSSINM and place the module in the appropriate load library. Use the following parameters:

  ```
  PARM=(XREF,LIST,MAP,RENT,REUS,NCAL,'SIZE=(512K,128K)','AMODE=31','RMODE=ANY')
  ```

Specifying a Subsystem

To specify a subsystem for Tivoli Information Management for z/OS, specify the subsystem name in either the **Target server subsystem name** field of the Installation Tailoring Facility or the CAS keyword of the BLGPARMS macro when you define your session-parameters members. If you do not specify a value, the default value, *BLX1*, is used. For more information about defining a session-parameters member, see “Working with Session-Parameters Members” on page 164. For more information on the BLGPARMS macro, see “BLGPARMS Macro — Defining Tivoli Information Management for z/OS’s Operating Characteristics” on page 318.

Each user (interactive users, APIs, batch jobs, and so on) of a subsystem must also be sure that when its Tivoli Information Management for z/OS session is started, the BLXSSINM load module for that subsystem is in one of the following:

- A load library that is allocated to the ISPLLIB ddname
- A load library that is defined in a STEPLIB DD statement
- A load library that is in the system link list
- The link pack area

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

- The JCL that you use to run the utility must reference the load library that contains the BLXSSINM load module for the desired subsystem.
- 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.

Defining a BLX-SP Procedure

You must define a procedure for each BLX-SP in SYS1.PROCLIB. If your BLX-SP subsystem is named BLX1, you can use the following sample procedure, BLX1PROC (provided in the SBLMSAMP library). If your BLX-SP subsystem is named something else, use the Installation Tailoring Facility to create a procedure.

```
//BLX1PROC  PROC PRM=00
//BLXSPCAS  EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB   DD   DISP=SHR,DSN=BLM.SBLMMOD1  APF AUTHORIZED
//BLXPRM    DD   DISP=SHR,DSN=BLM.SBLMSAMP  BLX-SP PARMS
```

A description of the JCL statements in the sample follows:

BLX1PROC

The *PRM=00* parameter specifies the suffix of the name of the BLX-SP parameters member that is used. For example, if your BLX-SP parameters member name is BLX100, then your suffix name is 00. If you specify this parameter in the START command for the BLX-SP procedure, the value specified in the command overrides the value specified here.

Note: Your BLX-SP parameters member must be in the data set specified by the BLXPRM DD statement.

The default value for PRM is 00.

BLXSPCAS

The *PGM=BLXSSP00* parameter specifies the BLX-SP program. You must use *BLXSSP00* as the parameter value. The *TIME=1440* parameter tells the operating system not to time out the BLX-SP. If this parameter is not specified, the BLX-SP will receive an out-of-time ABEND.

STEPLIB

This DD statement specifies the data sets that contain the load modules used by the BLX-SP. You must specify the data sets in a STEPLIB DD statement if you do not specify them in the link list or if you do not have the load modules in the link pack area. The data sets must be APF libraries.

If you have multiple BLX-SPs, ensure that the load library containing the BLXSSINM module you want to use is concatenated to the STEPLIB DD statement before the main Tivoli Information Management for z/OS load library (SBLMMOD1). Otherwise, the system reads in the sample copy of BLXSSINM that is in the main load library. The following is an example of what the STEPLIB DD can contain:

```
//STEPLIB   DD   DISP=SHR,DSN=BLM.BLXLIB2
//          DD   DISP=SHR,DSN=BLM.SBLMMOD1
```

BLXPRM

This DD statement specifies the data set that contains the BLX-SP parameters member for this BLX-SP. You must use *BLXPRM* as the ddname. The data set that you specify must be a PDS.

Defining a BLX-SP Parameters Member

The BLX-SP parameters member is a standard text file that contains the operating parameters you want your BLX-SP to run with. You can use the Installation Tailoring Facility to create a BLX-SP parameters member, or see “Defining BLX-SP Parameters Members” on page 343 for information on defining a BLX-SP. See “Using the Installation Tailoring Facility” on page 185 for information on using the Installation Tailoring Facility.

The BLX-SP parameters member must be created before you start your BLX-SP.

Note: The BLX-SP parameters member naming rules changed for Version 6.1. Instead of BLXPRMaa, the names are now *ssssaa*, where *ssss* is the name of the BLX-SP subsystem, and *aa* is any 1 or 2 alphabetic, numeric, or national (#, $, @) characters. However, the *aa* value must match the PRM designation in the BLX-SP procedure.

Defining a VSAM Resource Definition Member

Non-Sysplex Environment

The task of defining a VSAM resource definition member applies only if you are installing Tivoli Information Management for z/OS in a non-sysplex environment. If you are installing Tivoli Information Management for z/OS in a sysplex and using sysplex support, there is no need to define a VSAM resource definition member because VSAM resources are managed directly through DFSMS/MVS.

VSAM resources for a BLX-SP are defined in the VSAM resource definition member (see “The VSAM Resource Definition Member – Non-Sysplex” on page 25 for more information). Multiple BLX-SPs can share a VSAM resource definition member.

You can use the Installation Tailoring Facility to create a VSAM resource definition member, or see “Defining VSAM Resources to the BLX-SP in a Non-Sysplex Environment” on page 301 for information on defining a VSAM resource definition member. See “Using the Installation Tailoring Facility” on page 185 for information on using the Installation Tailoring Facility.

The VSAM resource definition member load module must be created before you start a BLX-SP.

Defining Multiple BLX-SPs

You must have a separate BLX-SP parameters member, a BLX-SP subsystem (with a unique 4-character subsystem name), and a BLX-SP procedure for each of your BLX-SPs. You must have one program properties table entry on each z/OS system that uses Tivoli Information Management for z/OS. Use the Installation Tailoring Facility to define a separate BLX-SP parameters member and BLX-SP procedure for each of your BLX-SPs (see “Using the Installation Tailoring Facility” on page 185). It is strongly recommended that you not use BLXn for any extra subsystems you define because Tivoli could use them in a possible future release.

Starting and Stopping the BLX-SP

The following example of the MVS system operator START command starts a BLX-SP. An explanation of the example follows the example text:

```
S BLX1PROC,SUB=JES2,PRM=01
```

S The MVS system operator START command.

BLX1PROC
The BLX-SP procedure for this BLX-SP.

SUB=JES2
A parameter that specifies the JES subsystem.

Note: This parameter is necessary in the START command only when you specify a procedure name that is the same as a subsystem name on your system.

PRM=01
An optional parameter that specifies the suffix of the name of the BLX-SP parameters member that is used. Use this parameter to override the PRM parameter value that is specified in the BLX-SP procedure. In this example, *01* overrides the value in the BLX-SP procedure.

An example of the MVS system operator STOP command follows. An explanation of the example follows the example text:

```
P BLX1PROC
```

P The MVS system operator STOP command.

BLX1PROC
The BLX-SP procedure for the BLX-SP.

When the BLX-SP stops, all Tivoli Information Management for z/OS sessions end. You can set a delay time to let your users get to a stopping point so they do not lose any data (see "Defining BLX-SP Parameters Members" on page 343).

For more information about the MVS system operator commands, refer to *OS/390 MVS: System Commands*.

End of Programming Interface information

11

Setting Up for Sysplex Data Sharing

This chapter describes how to set up Tivoli Information Management for z/OS for sysplex data sharing. You must set up for sysplex data sharing if:

- You intend to use Tivoli Information Management for z/OS in a parallel sysplex and take advantage of sysplex functions.
- You want to share Tivoli Information Management for z/OS databases. That is, users on different BLX-SPs need to concurrently share Tivoli Information Management for z/OS VSAM data sets.

What is a Sysplex?

A *sysplex* is a set of MVS systems that communicate and cooperate with each other through certain hardware and software components and software services to process workloads. The products that make up a sysplex work together to provide higher availability, easier systems management, and improved growth potential over a conventional computer system of comparable processing power. You can run Tivoli Information Management for z/OS in a non-sysplex or z/OS Parallel Sysplex environment.

A sysplex can be a base sysplex or a *parallel sysplex*. A parallel sysplex has one or more coupling facilities that enable multiple central processor complexes to simultaneously process a workload. By allowing two or more processors to share the same data you can maximize performance while minimizing cost, improve system availability and concurrency, expand system capacity, and configure your system environment more flexibly.

Note: Throughout this document, when you see the term sysplex, understand it to mean a parallel sysplex (a sysplex with a coupling facility). Tivoli Information Management for z/OS provides support for parallel sysplex only.

Tivoli Information Management for z/OS exploits the benefits of a parallel sysplex environment in the following ways:

- It takes advantage of an improved VSAM processing design. In a sysplex, the VSAM access mode used is record-level sharing (RLS). In a non-sysplex environment, the access mode is nonshared resources (NSR) or local shared resources (LSR).
- A sysplex offers performance benefits, especially if you are sharing Tivoli Information Management for z/OS databases. VSAM data sets are opened directly by the Tivoli Information Management for z/OS user rather than by the BLX-SP. Because RLS locks data at a more granular level, the performance of the Tivoli Information Management for z/OS database is improved overall, regardless of whether or not you share databases.

- When users issue a VSAM request, a single internal code path is used to access both shared and nonshared data sets. RLS handles all cross-memory requirements more reliably and efficiently than the BLX-SP does in a non-sysplex environment.
- The need for setup of APPC when using shared databases is eliminated. With RLS, applications running on more than one Tivoli Information Management for z/OS system can read from and write to the same set of data concurrently. VSAM handles all cross-system buffer invalidation.
- Operator commands can be entered on just one system in a shared database complex and take effect across the sysplex. For example, you can enter the BRDCST operator command to send a message to all Tivoli Information Management for z/OS users across the entire sysplex. The FREE and REALLOC commands can be used to free and reallocate a VSAM data set throughout the sysplex. You can also use the QUERY command to query the status of one or all VSAM data sets in the sysplex. (For more information about these commands, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.)

This chapter describes what you need to know about setting up Tivoli Information Management for z/OS to run in a sysplex whether or not you will be sharing Tivoli Information Management for z/OS databases. Use of a sysplex is optional; however, if you intend to share databases, you must install Tivoli Information Management for z/OS in a sysplex because VSAM RLS is required to share VSAM data sets.

Note: If you are not installing Tivoli Information Management for z/OS in a sysplex and exploiting sysplex services, you must use VSAM NSR or LSR instead of RLS, and you must define a VSAM resource definition member as described in “Defining VSAM Resources to the BLX-SP in a Non-Sysplex Environment” on page 301.

This chapter does not explain everything a systems programmer should know about installing or operating a z/OS system in a sysplex. Many publications are available in the z/OS library that address those topics, including the following publications which may be useful if you are not already familiar with z/OS data sharing and parallel processing or VSAM data sets:

- *OS/390 Parallel Sysplex Overview* – For a discussion of sysplex concepts.
- *OS/390 MVS Setting up a Sysplex* – For information about installing and operating a z/OS system in a sysplex.
- *OS/390 MVS Planning: Global Resource Serialization* – For information about serializing access to resources
- *OS/390 MVS Initialization and Tuning Reference* – For information about starting MVS and coding system parameters.
- *DFSMS/MVS DFSMSdfp Storage Administration Reference* – For information about the DFSMVS/MVS Storage Management Subsystem (SMS) and setting up RLS.
- *DFSMS/MVS® Using Data Sets* – For information about defining and working with VSAM data sets.

Understanding Sysplex Components

In a multiple system environment, systems are managed through the *cross-coupling facility* (XCF) services. XCF is a software component of z/OS. The XCF provides the services that allow authorized programs on z/OS images in a multisystem environment to communicate (send and receive data) with programs on the same z/OS image or other z/OS images in the sysplex. The services provided by XCF are the cornerstone of the sysplex; without it, there would be no sysplex at all. Tivoli Information Management for z/OS uses XCF to enable one BLX-SP to communicate with another BLX-SP on the same or another z/OS system in a sysplex.

However, the heart of parallel sysplex is the data-sharing technology based on the *coupling facility*. Unlike the XCF, which is software, a coupling facility consists of both hardware and software in your sysplex. The coupling facility is a special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex. It enables high-performance data sharing and rapid recovery from failures. A parallel sysplex uses one or more coupling facilities to simultaneously process a particular workload. Tivoli Information Management for z/OS provides parallel sysplex data sharing by exploiting the coupling facility.

Storage in a coupling facility is divided into objects called structures, which are used to implement data sharing and high-speed serialization. Structure types are cache, list, and lock, each providing a specific function to Tivoli Information Management for z/OS. The coupling facility is managed through a policy called the coupling facility resource management (CFRM) policy. CFRM allows you to specify how a coupling facility and its resources are to be used at your installation. In a CFRM policy, you supply information about each coupling facility and each coupling facility structure that you plan to use. As part of setting up Tivoli Information Management for z/OS to run in a parallel sysplex, your z/OS systems programmer must define the coupling facility structures and sysplex policy couple data sets to be used at your installation. In addition, if you are migrating from previous releases of Tivoli Information Management for z/OS, there are some other tasks to perform, such as migrating your exiting BLX-SP to use new parameters for sysplex support and migrating your existing VSAM data sets to be enabled for record level sharing.

Figure 10 shows an example of Tivoli Information Management for z/OS in a non-sysplex environment.

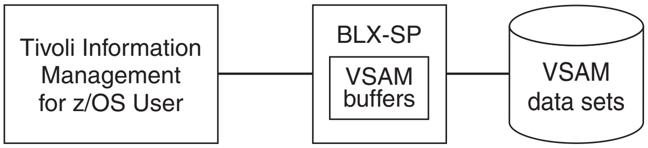

Figure 10. Example of Tivoli Information Management for z/OS in a Non-Sysplex Environment

In a non-sysplex environment:

- There is no coupling facility.
- VSAM I/O processing is performed through the BLX-SP using VSAM buffers.
- VSAM processing is done through nonshared resources (NSR) or local shared resources (LSR).
- Multiple BLX-SPs or systems cannot share a common database.
- Data locking mechanism locks the entire VSAM data set.

Figure 11 on page 155 shows an example of a sysplex setup. In contrast, in a sysplex:

- A coupling facility is required.

- VSAM I/O processing is handled more reliably by VSAM, not the BLX-SP. Information on VSAM data sets is kept in the SMSVSAM data space and in lock and cache structures in the coupling facility rather than in BLX-SP VSAM buffers.
- VSAM processing is done through RLS. The RLS caches control intervals, and the cache is in the SMSVSAM data space and in coupling facility cache structures. VSAM uses RLS serialization to lock individual records within VSAM data sets and to achieve buffer coherency. RLS locks take the place of enqueues at the data set level most of the time when sysplex mode is enabled. Because RLS gets a shared lock on a record when doing a read, fewer Tivoli Information Management for z/OS enqueues are required in a sysplex. The greater degree of locking granularity provided enables you to have better performance because more user transactions can take place simultaneously with the database.
- The BLX-SP requires less storage because there are no LSR buffers and users require less cross-memory storage.
- The cross-system extended services (XES) component of z/OS enables Tivoli Information Management for z/OS to take advantage of data sharing services (data locking) through the coupling facility. VSAM RLS uses a lock structure (IGWLOCK00) to do record locking, and cache structures to cache records in the coupling facility. In addition, the BLX-SP has its own lock structure to facilitate use of the FREE, REALLOC, and QUERY operator commands. Both lock structures are necessary for Tivoli Information Management for z/OS data sharing.
- The XCF component of z/OS allows Tivoli Information Management for z/OS on one system to communicate with Tivoli Information Management for z/OS on another system. For example, the BRDCST command can be used to send messages to users on one BLX-SP or all BLX-SP in a sysplex.
- You can share Tivoli Information Management for z/OS databases. Databases can be shared only if sysplex mode is enabled.
- It is easier to administer the Tivoli Information Management for z/OS database and expand for growth. For example, if you want to expand your system, you can add a BLX-SP without having to define VSAM resource definitions. The coupling facility structures are used instead. You can also add the BLX-SP without having to stop and restart Tivoli Information Management for z/OS. Also, if you are sharing databases, you do not have to set up APPC/MVS.
- If you are already using Tivoli Information Management for z/OS, you must initially migrate existing VSAM data sets and BLX-SPs to exploit sysplex. You must also set up coupling facility structures to support data sharing and communication if you do not already have them.

An example of a Tivoli Information Management for z/OS shared database environment is shown in Figure 12 on page 155.

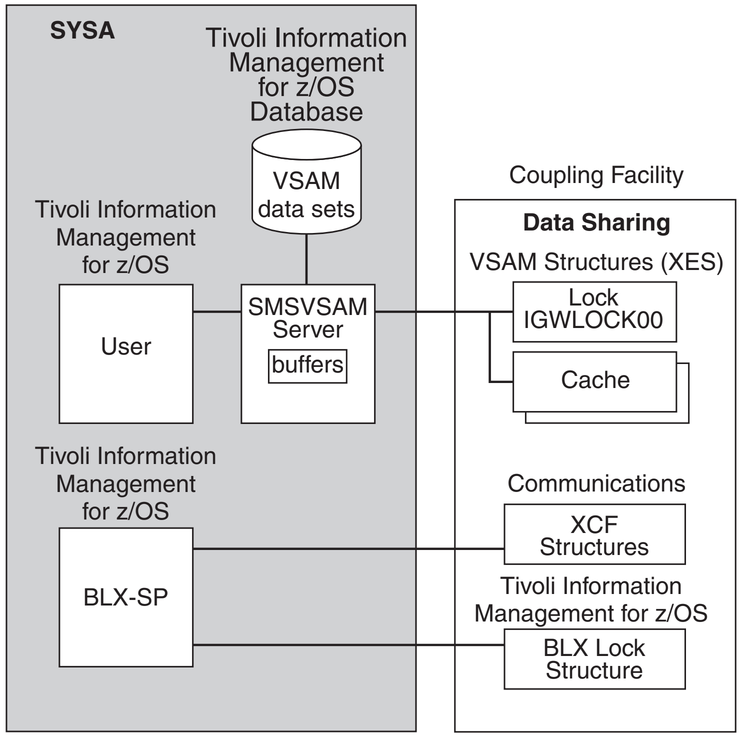

Figure 11. Example of Tivoli Information Management for z/OS in a Parallel Sysplex

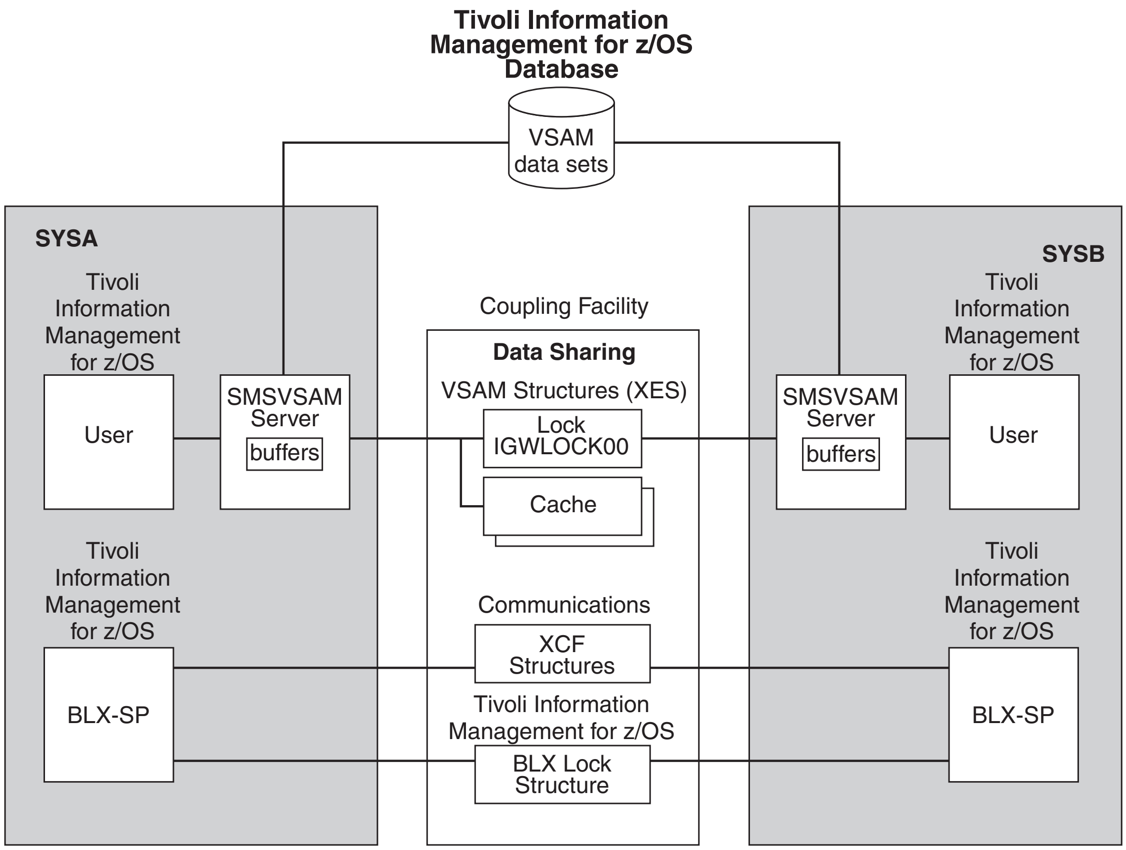

Figure 12. Example of Tivoli Information Management for z/OS with Shared Databases in a Parallel Sysplex

Enabling Sysplex Mode

The information in this section describes the tasks involved to enable Tivoli Information Management for z/OS to exploit z/OS sysplex services. If you are migrating from a previous release of Tivoli Information Management for z/OS, you should note the additional migration tasks that are required; these additional tasks are discussed below under the **If migrating** notes.

- If you have not already set up your BLX-SPs, you must define and set up BLX-SPs on each system in your sysplex. (See "Setting Up Your BLX-SP" on page 143.) For all BLX-SPs, ensure that the SYSPLEX=YES parameter is included in the BLX-SP parameters member in the data set specified on the BLXPRM DD statement in your BLX-SP procedure. SYSPLEX=YES enables sysplex mode for the BLX-SP and its users. For an example of the BLX-SP parameters member, see BLX1SH in the SBLMSAMP data set.

  If migrating: When sysplex mode is enabled, any existing BLX-SP VSAMRESOURCES parameters are ignored. The DESTNAMES parameter is not supported. If you have a DESTNAMES parameter, you *must* remove it to avoid an error.

- Define the session-parameters members on each system in your sysplex. There is no difference in the session parameters for shared databases and nonshared databases. See "Defining Tivoli Information Management for z/OS Session-Parameters Members" on page 317 for information about this task. When using the BLGCLUST macro to define the database, note that the COGENQ keyword is not used in a sysplex and is ignored if specified.

- **If migrating:** VSAM data sets must be managed by SMS to use VSAM RLS. Because RLS requires use of a LOG parameter, you must migrate the existing VSAM data sets from your current version of Tivoli Information Management for z/OS to SMS-managed clusters with the LOG(NONE) parameter. You can use the IDCAMS ALTER or IDCAMS DEFINE CLUSTER command to define clusters with the LOG(NONE) parameter. As an alternative, you can run the BLGALTER job provided in the SBLMSAMP library to migrate existing VSAM data sets to be RLS enabled. BLGALTER is a sample IDCAMS ALTER job.

  Also, data sets must be defined with the NOIMBED parameter. RLS does not support use of the IMBED parameter. If current data sets are defined with IMBED, migrate them by defining new data sets with the NOIMBED parameter, and then copy the data set (for example, with IDCAMS REPRO).

  The following samples in SBLMSAMP are also updated to illustrate use of the LOG(NONE) parameter.

  BLGDATAB – Define the Tivoli Information Management for z/OS database
  BLGDICT – Define the dictionary data set
  BLGRPNL – Define read panel data sets
  BLMWPNL – Define write panel data set

  Refer to the *DFSMS/MVS Access Method Services for the Integrated Catalog Facility* for more information about using IDCAMS.

- Set up RLS as described in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*, in the section that describes "Administering VSAM Record-Level Sharing."

As part of this setup, have your storage administrator or systems programmer set up storage management definitions for RLS. You can use the ISMF panel interface of DFSMS/MVS to do this. Refer to the *DFSMS/MVS DFSMSdfp Storage Administration Reference* for more information about this.

All systems sharing data sets must be part of the same SMS complex.

Additionally, create RLS share control data sets. This process is also described in the ″Administering VSAM Record-Level Sharing″ section of the *DFSMS/MVS DFSMSdfp Storage Administration Reference*. A sample job, BLXRLSCD, is provided in the Tivoli Information Management for z/OS SBLMSAMP sample library to define RLS share control data sets.

- Have your z/OS systems programmer set up the CFRM policy to be used. The CFRM policy defines how the coupling facility and its resources should be used. You can update an existing policy with the IXCMIAPU administrative data utility provided with z/OS. A sample IXCMIAPU job is provided with Tivoli Information Management for z/OS in the SBLMSAMP sample library to help you define coupling facility structures (see member BLXCFSTR). In addition to providing sample JCL to define the structures necessary for RLS, BLXCFSTR defines a lock structure exclusively for Tivoli Information Management for z/OS which must have these attributes:
  - Name – The Tivoli Information Management for z/OS lock structure must be named BLXLOCK00.
  - Size – The size of BLXLOCK00 is calculated using this formula:

    ```
    INITSIZE = D * E
    ```

    where

    D the total number of VSAM data sets that will be accessed by all Tivoli Information Management for z/OS users throughout the sysplex, rounded up to the next power of 2.

    E the size (in bytes) of a single lock table entry, determined by calculating

    ```
    1 + (number of BLX-SPs in the sysplex + 1)/8
    ```

    and then rounding this up to the next power of 2.
  - Exclusion list – There are no exclusions; that is, there are no structure names that cannot be shared with the same coupling facility.
  - Persistence – The lock structure does not remain allocated in the coupling facility if there are no active connectors to it.
  - Rebuild– The lock structure cannot be rebuilt by an operator.
  - Connectivity – Ensure that IGWLOCK00 and BLXLOCK00 have universal connectivity so they can be accessed from all systems in the sysplex.

    Note: You must define IGWLOCK00 if you are setting up RLS for the first time.
  - Coupling facility requirement – The coupling facility must have a coupling facility control code of 2 at a minimum.
- Have your z/OS systems programmer include the appropriate COUPLExx and IGDSMSxx members in the z/OS SYS1.PARMLIB data set. SYS1.PARMLIB members contain values that MVS uses as input during system initialization to define the

characteristics of the system. Tivoli Information Management for z/OS has no specific recommendations for required parameters for COUPLExx; however, it is recommended that you specify **RLSINIT(YES)**.

What are Shared Databases?

A shared Tivoli Information Management for z/OS database is a database that can be accessed by users on different BLX-SPs at the same time. The VSAM data sets that make up the database can reside in the same z/OS system or on multiple z/OS systems. The following VSAM data sets are eligible for sharing when a Tivoli Information Management for z/OS database is shared:

- SDDS data sets – the data records
- SDIDS data sets – the index for searching
- SDLDS data set – the optional log for storing backup copies of the SDDS
- DICT data set – the definition for p-words, s-words, and validation patterns associated with the data
- RPANLDS data sets – the panels used in production
- WPANLDS data set – the data set containing Tivoli Information Management for z/OS panels that you create or modify using PMF.

Note: In previous releases of Tivoli Information Management for z/OS, the facility that enabled data sharing to take place in Tivoli Information Management for z/OS was called *Multisystem Database Access* (MSDA). MSDA no longer exists in Tivoli Information Management for z/OS, but the concept of sharing data or sharing databases still exists. To share databases, a parallel sysplex is required and sysplex mode must be enabled for Tivoli Information Management for z/OS. By exploiting a sysplex environment, Tivoli Information Management for z/OS no longer requires APPC for data sharing. The data and index buffers are maintained in the SMSVSAM data space and in the coupling facility. SMSVSAM handles any necessary communication with the coupling facility to maintain integrity of the VSAM buffers.

No other programs or non-Tivoli Information Management for z/OS utilities can share the data sets with Tivoli Information Management for z/OS unless they access the data sets through the Tivoli Information Management for z/OS APIs. As an exception, under RLS, if read integrity is not required, a SHAREOPTIONS(2) data set can be opened in read-only mode by a nonshared resource application (such as IDCAMS REPRO).

Enabling Shared Databases

The information in this section describes the tasks involved to enable Tivoli Information Management for z/OS to share databases in a sysplex. To share VSAM data sets in a Tivoli Information Management for z/OS database, you must perform the following tasks:

- Set up for sysplex mode as described in “Enabling Sysplex Mode” on page 156.
- If you are using a product other than the GRS component of z/OS to manage enqueues, note the resource names listed in “Resource Names That Tivoli Information Management for z/OS Enqueues On” on page 351 for Tivoli Information Management for z/OS.

After you have defined a shared database on all the systems that share it, you can use it. z/OS Security Server (RACF) (or a similar security program), and DFSMS/MVS, and at least one of the shared BLX-SPs must be running on the systems sharing the database for users to access the database. If you have multiple BLX-SPs sharing a database and one of them goes down, the others can still access the database.

Defining the BLX-SP Procedure

To illustrate how to define the BLX-SP procedure, suppose that you are setting up shared databases on three z/OS systems (SYSTEM1, SYSTEM2, and SYSTEM 3). SYSTEM1 contains a BLX-SP cataloged procedure named BLXSHR. This BLX-SP procedure defines a database of VSAM data sets that you plan to share with two other systems, SYSTEM2 and SYSTEM3. Figure 13 illustrates the BLX-SP procedure for BLXSHR. This procedure is provided in the SBLMSAMP library in member BLXSHR.

```
//BLXSHR    PROC PRM=SH
//BLXSPCAS  EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB   DD   DISP=SHR,DSN=BLM.SBLMMOD1  APF AUTHORIZED
//BLXPRM    DD   DISP=SHR,DSN=BLM.SBLMSAMP  BLX-SP PARMS
```

Figure 13. Sample BLX-SP Procedure

This BLX-SP uses the BLX-SP parameters member (BLX1SH) illustrated in Figure 14 on page 160. Use this same procedure and BLX–SP parameters member on SYSTEM2 and SYSTEM3.

Global Resource Serialization Considerations

A GRS complex, which consists of one or more systems connected by communications links, is required for a sysplex. It enables you to serialize data across multiple systems while maintaining data integrity. It is assumed that a GRS complex has already been established during z/OS installation. If a GRS complex has not been set up, refer to *OS/390 MVS Planning: Global Resource Serialization* for more information. This section describes some setup details that are specific to the use of Tivoli Information Management for z/OS in a GRS complex, but it does not describe how to set up a GRS complex.

Modifying the GRSRNLaa Member of SYS1.PARMLIB

Before you modify the GRSRNLaa member of the SYS1.PARMLIB, review the VSAM recommendations for RNL in *OS/390 MVS Planning: Global Resource Serialization*.

Also, check the GRSRNLaa member for any of the resource names listed in “Resource Names That Tivoli Information Management for z/OS Enqueues On” on page 351. Normally, none of these names should be present in GRSRNLaa. The only exception would be if Tivoli Information Management for z/OS is running in sysplex mode and not sharing data sets across systems. In that case, performance may be improved by adding the following resource names to the SYSTEMS Exclusion RNL: BLGAPI, BLGDICTN, BLGPMFPU, BLGRNID, BLGSEQN, BLGUT4, BLGVCGZR, and BLGSDIDS.

Important!
You must remember to remove these names if you decide to share data sets across systems. If you do not remove them, serialization errors can occur and the integrity of your Tivoli Information Management for z/OS database may be compromised.

Modifying the BLX-SP Parameters Member

Figure 14 is a sample BLX-SP parameters member for a BLX-SP running in sysplex mode.

```
/*********************************************************************/
/*                                                                   */
/*         BLX-SP OPERATING PARAMETERS                               */
/*                                                                   */
/*********************************************************************/

 BLXPRM                            /* SPECIFY BLX-SP PARAMETERS     */

     /**********************************************************/
     /*                                                        */
     /*   BLX-SP SYSPLEX OPTION                                */
     /*                                                        */
     /**********************************************************/

        SYSPLEX=YES,               /* Enable sysplex data sharing   */

     /**********************************************************/
     /*                                                        */
     /*   BLX-SP TRACE OPTIONS                                 */
     /*                                                        */
     /**********************************************************/

        TRACE=OFF,                 /* DON'T PRODUCE TRACE OUTPUT    */

     /**********************************************************/
     /*                                                        */
     /*   BLX-SP LOG OPTIONS                                   */
     /*                                                        */
     /**********************************************************/

        LOG=ON,                    /* PRODUCE LOG INFORMATION       */
        LOGSYSOUT=A,               /* JES SYSOUT CLASS FOR LOG DS   */
        LOGLINES=0,                /* MAX # OF LINES IN A LOG DS    */
```

Figure 14. Sample BLX-SP Parameters Member (BLX1SH) for Sharing Data Sets (Part 1 of 2)

```
/**********************************************************/
/*                                                        */
/*   BLX-SP SHUT DOWN OPTIONS                             */
/*                                                        */
/**********************************************************/

   SHUTDOWNWT=00050000,        /* SHUTDOWN WAIT TIME HHMMSSTH   */
   SHUTDOWNTFY=00001000,       /* SHUTDOWN NOTIFY WT HHMMSSTH   */

/**********************************************************/
/*                                                        */
/*   BLX-SP MESSAGE ROUTING OPTIONS                       */
/*                                                        */
/**********************************************************/

   WRITEOPER=1,                /* DEFAULT WTO ROUTING CODE      */

/**********************************************************/
/*                                                        */
/*   BLX-SP API OPTIONS                                   */
/*                                                        */
/**********************************************************/

   APISECURITY=XXX,            /* Replace XXX with ON or OFF    */
   APICHKOUTLIM=00000000;      /* HHMMSSTH - NO LIMIT SET       */

/**********************************************************/
```

Figure 14. Sample BLX-SP Parameters Member (BLX1SH) for Sharing Data Sets (Part 2 of 2)

12

Evaluating Base Product Tailoring Considerations

This chapter discusses factors concerning the Tivoli Information Management for z/OS base product you should consider before you install Tivoli Information Management for z/OS.

This chapter is designed to help you tailor Tivoli Information Management for z/OS.

Working with Translate Tables

If you want to use non-Latin alphabetic characters to enter data into the database for other than freeform text, you must install the non-Latin translate table. You must choose the tables to use when you install Tivoli Information Management for z/OS; you cannot change the translate table at a later time.

To collect, store, or display data in mixed case, Tivoli Information Management for z/OS requires use of the Latin translate tables.

When you install the Latin translate tables:

- All lowercase ward 42 DBCS and SBCS alphabet characters entered at the workstation automatically translate to uppercase for all commands, but not for freeform text and panel externals. Data is not automatically translated. The translation of data is controlled by the setting of the Collected Data Case field on the assisted-entry panel for the field. You can use the Installation Tailoring Facility to tailor the uppercase translate table that Tivoli Information Management for z/OS uses for translating your Latin characters to uppercase. See “Using the Installation Tailoring Facility” on page 185 for information on using the Installation Tailoring Facility. Do not change the uppercase translate table unless you have special national language requirements.
- You can use the Latin translate tables with non-Latin alphabet workstations if you translate all panels to uppercase when you load the VSAM panel data set and if you enter all data (except freeform text and external panel changes) in uppercase Latin.

When you install the non-Latin translate tables:

- All characters translate to themselves. Thus, the **Display uppercase** and **Output in uppercase** fields in the Tivoli Information Management for z/OS profiles for your users are not used.
- Ward 42 DBCS alphabet characters are translated to uppercase alphabet characters.
- All commands must be entered as uppercase Latin alphabet characters to ensure that Tivoli Information Management for z/OS recognizes them.

- You cannot use the mixed case data collection and processing features provided in Tivoli Information Management for z/OS.
- You must use uppercase Latin alphabetic characters for the alphabetic characters entered on those panels whose validation pattern did not change to an I pattern. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on validation pattern characters and how to change the validation patterns on panels.
- You must use uppercase Latin alphabetic characters for all p-words and s-words entered into the dictionary. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on validation pattern characters and how to change the validation patterns on panels.

To use the non-Latin translate tables, you must ensure that:

- All workstations on which Tivoli Information Management for z/OS runs use the same character set. If only some of the workstations have non-Latin alphabetic character sets, Latin alphabet workstations cannot read data entered on the non-Latin alphabet workstations.
- You do not plan to convert to Latin alphabet workstations in the future, or to display data from the database on Latin alphabet workstations or printers. If you enter non-Latin characters into your database, Latin alphabet workstations or printers cannot read them.
- You translate all of the Tivoli-supplied panels to uppercase when loading them into VSAM panel data sets. To do this, use BLGUT6 and specify the UPPERCASE keyword.
- You change the panel validation patterns to I patterns for all fields that are to accept non-Latin alphabet characters and non-Latin DBCS characters. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on validation pattern characters and how to change the validation patterns on panels.

When you enter special characters for nonprefixed string data fields, such as the **Description abstract** field, the special characters are translated to blanks before the data is cognized. Thus, if 123-4567 were entered in the **Description abstract** field, the dash would be translated to a blank and both 123 and 4567 would be cognized (written to the SDIDS data set) and would be searchable, but 123-4567 would not be searchable. This process is called *blank substitution*.

To look at the Latin and non-Latin translate tables used for display and the Latin and non-Latin blank substitution translate tables, see “Translate Tables” on page 353.

Note: If you require special national language support, you can also define, at installation, additional uppercase or lowercase characters that should be treated as valid characters by Tivoli Information Management for z/OS. You can define these characters at installation through use of the optional features option of the Installation Tailoring Facility.

Working with Session-Parameters Members

The session parameters define certain processing options and all the data sets that Tivoli Information Management for z/OS requires for a Tivoli Information Management for z/OS session. Most of the data sets used by the Tivoli Information Management for z/OS program are VSAM data sets. The report format table data sets (RFTDS) and output listings created by the report or print functions are examples of exceptions. By using the session parameters provided by Tivoli Information Management for z/OS, you can tailor Tivoli Information Management for z/OS sessions for individual users. By defining more than one set of session-parameters members, you enable Tivoli Information Management for z/OS to start with different operating characteristics for different users or with different characteristics for different sessions for the same user. Each session-parameters member can define only those databases needed by a certain set of users. Thus, you can have several session-parameters members, each one identifying the databases and panel data sets needed by different groups of users.

A session-parameters member identifies the following:
- One dictionary data set
- One report format table data set
- One write panel data set (for use with PMF)
- One or more read panel data sets
- The databases you want to use
- The data sets associated with each database
- The BLX-SP you want to use
- Other processing characteristics, such as a control panel to use upon entry to the system, default date format, or whether universal time processing should be enabled.

For a list of the available session parameters that you can define, see the “Defining Tivoli Information Management for z/OS Session-Parameters Members” on page 317.

Consider defining a separate session-parameters member to use when running the Tivoli Information Management for z/OS utilities. If you attempt to recover a database from the SDLDS while using a session-parameters member that has an SDLDS, BLGUT3 attempts to log the recovery transactions. This is not something that you want to do because it conflicts with the intended purpose of BLGUT3 (for an explanation, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*). Therefore, define a separate session-parameters member without an SDLDS. You can also use a separate session-parameters member for the BLGUT1 utility, although this is not required. The advantage is that you can use a session-parameters member that has a different sort routine specified. To take advantage of this option on BLGUT1, you must overwrite the SDDS and SDIDS cluster names with DD statements. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details.

The following considerations apply to parameters that are specified in the session-parameters member.

Working with Panel Buffers

You can define how many panels are kept in the individual user’s address space. You specify this value either in the **Number of panel buffers** field when you define a session-parameters member using Installation Tailoring Facility or in the PNLBCNT keyword of the BLGPARMS macro (see page 327). If this value is kept at a sufficiently high number, the number of read panel data sets and their placement is not very significant.

SORTIN, SORTOUT, and SORTWK01 Data Sets

Three data sets are used in conjunction with:

- The SORTIN primary and secondary allocation percentage fields
- The number of matches in the search results list

- The size and number of the sort fields for the report (as defined in the Report Format Table (RFT).

The values specified for SORTIN, SORTOUT, and SORTWK01 determine both the data set block size and the space allocation information for the SORTIN, SORTOUT, and SORTWK01 data sets. The formula for this computation follows. All division is integer division; that is, any remainder from the division is truncated. Determine the SORTIN LRECL (logical record length) from the number and size of the sort fields specified by the RFT. Divide the *trksize* value by the SORTIN LRECL to determine the number of logical records that fits in a physical record. This number of logical records multiplied by the SORTIN LRECL becomes the data control block's (DCB) block size (BLKSIZE) used during dynamic allocation of the SORTIN data set.

```
Step 1.  # logical records = trksize / SORTIN LRECL
Step 2.  DCB BLKSIZE = # logical records x SORTIN LRECL
```

You must request space allocation in blocks rather than in tracks or cylinders. Your requested block size is the value computed for the DCB BLKSIZE.

To determine the requested number of primary blocks, multiply the number of search results list matches by the SORTIN LRECL size. This gives the approximate number of bytes needed if each search results list entry produces exactly one SORTIN logical record. (However, this is not always true, as discussed in the paragraph following Step 4.) To determine the number of blocks that are necessary to contain all the data in the primary extent of the data set, divide this total number of bytes in the SORTIN data set by the DCB BLKSIZE value and add one to the result.

```
Step 3.  Approximate # bytes = # of search matches x SORTIN LRECL
Step 4.  # of blocks  = (approximate # bytes / DCB BLKSIZE) + 1
```

As previously mentioned, one search results list match can produce multiple SORTIN logical records. The number produced is a function of the number of times a particular RFT sort field appears in the Tivoli Information Management for z/OS SDDS data record, and whether there are multiple RFT sort fields for which this occurs. The total number of logical records written to SORTIN is a function of how many times this condition occurs for each record reported.

Therefore, multiply the number of primary blocks (as calculated in Step 4) by the value you specify in the user's profile field, primary SORTIN allocation percentage (shown as PSIAP in Step 5, below), divided by 100, and increased by one. The resulting value becomes the primary space allocation request number.

```
Step 5.  # of blocks allocated = ((# of blocks x PSIAP) / 100) + 1
```

The following example contains sample values for the variables used in steps 1 through 5:

```
trksize = 55996
SORTIN LRECL = 80
number of search results matches = 50
PSIAP = 123
1.  # logical records = 55996 / 80                equals 700
2.  DCB BLKSIZE = 700 x 80                        equals 56000
3.  Approximate # bytes = 50 x 80                 equals 4000
4.  # of blocks  = (4000 / 56000) + 1
    equals roughly 1.07,  truncated to 1
5.  # of blocks allocated = ((1 x 123) / 100)) + 1
    equals 1.23 + 1, which equals 2
```

In this case, the primary space allocation is for two blocks.

Calculate the secondary space allocation request number in a similar manner, using the user's profile value for the secondary SORTIN allocation percentage.

Calculate the DCB parameters for the SORTOUT data set in a manner similar to that of the SORTIN data set. The SORTOUT LRECL is the same as the SORTIN LRECL. However, you can compute the space allocation information exactly, because the actual number of SORTIN logical records is available by the time the SORTOUT data set is dynamically allocated. Therefore, the space allocation request for SORTOUT data set requires only a primary space allocation.

Allocate the SORTWK01 data set using the same space allocation block size as that used for the SORTIN data set. However, use only one-third the amount of space needed for the primary and secondary quantities for the SORTIN data set.

For all three data sets, use the space allocation keyword CONTIG to ensure that you allocate contiguous sections of external storage for the data sets. For the SORTWK01 data set, use the space allocation keyword, ROUND, to ensure that you allocate the data set on a cylinder boundary. These keywords enhance the performance of the Tivoli Information Management for z/OS report generator. Do not use the ROUND keyword for the SORTIN and SORTOUT data sets because, in most cases, you allocate these data sets to a VIO device. Even if you do not use VIO, you allocate excess direct access space, which is unusable by other functions in your installation while a Tivoli Information Management for z/OS report is running. This outweighs the performance improvement obtained for most reports.

SORT Routine

Figure 15 contains the parameter lists that Tivoli Information Management for z/OS passes to the SORT routine that you indicate in your session-parameters member. Tivoli Information Management for z/OS calls the SORT routine twice; therefore, Figure 15 shows both parameter lists. Refer to the *DFSORT Application Programming Guide* for the standard interface to the IBM DFSORT program (5740-SM1).

```
*******************************************************
*        INTERFACE FOR FIRST CALL TO SORT
*******************************************************
ADLST    DC    AL2(LISTEND-LISTBEG)   Param list length
LISTBEG  DC    A(SORTA1)              beginning address of sort stmt
         DC    A(SORTA1E)             end address of sort stmt
         DC    A(RECA1)               beginning addr of record stmt
         DC    A(RECA1E)              end addr of record stmt
         DC    F'0'                   addr of E15 routine
         DC    V(BLG0Z3X1)            addr of E35 routine
         DC    C'SRT1'                ddname for SORT1
LISTEND  EQU   *
SORTA1   DC    C' SORT FIELDS=(1,8,BI,A),'  sort control stmt
         DC    C'FILSZ=E3800000,SKIPREC=       ,' (continued)
         DC    C'DYNALLOC=SYSDA'              (continued)
         DC    CL149' '                       (continued)
SORTA1E  DC    C' '                         end of control stmt
RECA1    DC    C'RECORD TYPE=F,LENGTH=8'    record control stmt
RECA1E   DC    C' '                         end of record control stmt
```

Figure 15. Parameter Lists for SORT Routine (Part 1 of 2)

```
*******************************************************
*       INTERFACE FOR SECOND CALL TO SORT
*******************************************************
ADLST    DC    AL2(LISTEND-LISTBEG)   Param list length
LISTBEG  DC    A(SORTA2)              beginning address of sort stmt
         DC    A(SORTA2E)             end address of sort stmt
         DC    A(RECA2)               beginning addr of record stmt
         DC    A(RECA2E)              end addr of record stmt
         DC    F'0'                   addr of E15 routine
         DC    V(BLG0Z3X2)            addr of E35 routine
         DC    C'SRT2'                ddname for SORT2
LISTEND  EQU   *
SORTA2   DC    C' SORT FIELDS=(1,8,BI,A),'  sort control stmt
         DC    C'FILSZ=E3000000,'                 (continued)
         DC    C'DYNALLOC=SYSDA'                  (continued)
SORTA2E  DC    C' '                         end of control stmt
RECA2    DC    C'RECORD TYPE=F,LENGTH=8'    record control stmt
RECA2E   DC    C' '                         end of record control stmt
```

Figure 15. Parameter Lists for SORT Routine (Part 2 of 2)

You can allocate a data set with the ddname of BLGSMSG in your TSO logon procedure to receive any SORT/MERGE program product messages generated during your use of Tivoli Information Management for z/OS. This data set is optional.

Working with Databases

Tivoli Information Management for z/OS works with two types of databases, whose contents vary as follows:

- A Tivoli Information Management for z/OS-format database consists of an SDDS, SDIDS, and optionally, an SDLDS. Tivoli Information Management for z/OS identifies these databases as database 4, 5, 6, 7, 8, or 9. Database 5 is the only database that you can write to; the remainder are read-only databases. Database 6 is reserved for Tivoli Inventory and must be defined with an SDIDS key of 34.
- A user-defined format database consists of only an SDDS and an SDIDS. User-defined format databases are read-only databases. Tivoli Information Management for z/OS identifies user-defined format databases as database 0, 1, 2, or 3.

  To create user-defined format databases, do the following:

  1. Allocate the VSAM data sets for the SDDS and SDIDS for your user-defined format database.
  2. Create or update a session-parameters member to contain the name of the SDDS and SDIDS for your user-defined format database. See the Installation Tailoring Facility or “Defining Tivoli Information Management for z/OS Data Sets” on page 277 for details on setting up the SDDS and SDIDS.
  3. Create a sequential data set containing the entries for your user-defined format database.
  4. Run the BLGOZUD utility to load your user entries from the sequential data set into your SDDS and SDIDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details about BLGOZUD.

Working with VSAM Data Sets

When working with VSAM data sets, remember that:

- RLS requires SMS-managed data sets.
- Extended attribute data sets are supported when running Tivoli Information Management for z/OS in either sysplex or non-sysplex mode.
- Spanned data sets are not supported.

Review the following considerations before defining your Tivoli Information Management for z/OS VSAM data sets.

Using Imbed or Noimbed

Starting with DFSMS/MVS Version 1.5, the *Imbed* option is no longer supported when defining VSAM data set indexes. If *Imbed* is specified, it is ignored. Therefore, you should use *Noimbed* when defining VSAM data set indexes. *Noimbed* is also required for VSAM RLS.

When you define your VSAM clusters using the Installation Tailoring Facility, a value of *Noimbed* indicates the lowest level index records (sequence set) of the data set are written once and reside with the rest of the index levels.

Using Replicate or Noreplicate

When you define your VSAM clusters using Installation Tailoring Facility, a value of *Replicate* specified in the **Index track records** field indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, *Replicate* can reduce the rotational delay of the disk.

However, *Noreplicate* may be the better choice for Tivoli Information Management for z/OS VSAM data set indexes:

- If you are using LSR and have allotted enough buffers to contain all of the data set index CIs then, after the first access, no physical reads of the index from the DASD take place. *Replicate* has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- If you are using a controller cache function, *Replicate* can be counterproductive. Because entire tracks are buffered into the controller cache, *Replicate* can cause cache memory in the controller to unnecessarily store the replicated index records.

For these reasons, *Noreplicate* is preferred for Tivoli Information Management for z/OS data set indexes. Consider using *Replicate* for the index only if you are not buffering most of the index through LSR and if you are not using a controller cache function for the data set.

Specifying VSAM Data Set Space

You can specify space in terms of records, tracks, or cylinders. By specifying the allocation in terms of numbers of records, the characteristics for the device on which the data set is allocated are not important. However, disk fragmentation and other space problems can occur when space is allocated by records or tracks. If you know the characteristics of the device on which the data set is allocated, you may want to convert the number of records or tracks to cylinders.

Your organization may require the allocation to be in certain units of allocation. To arrive at the number of tracks or cylinders required, use the number of records calculated for the data set. When you allocate specific data sets, use the example provided in *DFSMS/MVS Using Data Sets*.

Working with SDDSs

The SDDS is a VSAM key-sequenced data set. You define this data set by using the Installation Tailoring Facility or the Access Method Services (AMS) DEFINE command. In the AMS DEFINE command, you must specify the maximum and average record sizes and the amount of disk storage to allocate.

When you allocate the SDDS, you must also choose which format of the SDDS you are allocating. Use the KEYS keyword to specify the format of the SDDS. Specifying a key length of 8 (8 bytes) allocates the SDDS for use as key 8 format. You can select a key length of 7 (7 bytes) for allocating the SDDS format. This is known as key 7 format. Key 7 format generally provides better performance, and is the recommended format for Tivoli Information Management for z/OS. Key 7 format is required for some new functions such as the BLGUT23 series of utilities that provide a backup mechanism that can be used without requiring users to be off the system.

Note: SDDSs for databases 0, 1, 2, and 3 must have a key length of 8.

For more information about the key 7 and key 8 formats, see the following discussions on maximum record size and DASD space for the SDDS.

The following list is a set of recommended values to use to calculate the size of the data component for your SDDS. Use these values in the formula provided in *DFSMS/MVS Using Data Sets* in the section that shows a sample calculation of space allocation for a VSAM key-sequenced data set. After you create a representative number of records for all record types, run BLGUT20 to get statistical data to help you determine whether the values you originally chose for the SDIDS are accurate or need redefining.

Control interval size

For optimal performance and space utilization, the CI size should be large enough to result in a single Tivoli Information Management for z/OS logical record being read or written in one I/O operation, yet allow efficient utilization of the device. The BLGUT20 utility indicates what the average SDDS record size is. This average record size value should be rounded up to the next multiple of 2048. Since the SDDS can be very large, it is desirable to further adjust the CI size to a higher value that is also a multiple of 2048 to achieve efficient track utilization for the device type used for the SDDS. The CI size (and the maximum record size) can be increased at any time simply by using AMS REPRO. Refer to the description of the BLGUT7 utility in the *Tivoli Information Management for z/OS Operation and Maintenance Reference* if you want to decrease the CI size or record size.

See “Calculating the Size of Your SDDS Records” on page 171 if you cannot use the BLGUT20 utility to determine the average SDDS record size.

Maximum record size

When you use a key 8 format or key 7 format, the maximum record size should be 7 bytes less than the CI size so that Tivoli Information Management for z/OS can use the maximum available space to block logical records. For example, for the CI size of 2048, use a maximum record size of 2041. If you increase the CI size, always increase the record size at the same time. This is true even when you increase the CI size to achieve a more effective CI size for the device. Any maximum record size (CI size minus 7) greater than the average will provide optimal performance. SDDS records are variable length up to the maximum record size. Any record written that is larger than the maximum record size is segmented into multiple physical records before it is written. The maximum record size (along with the CI size) can be increased at any time simply by using AMS REPRO. Refer to the description of the BLGUT7 utility in the *Tivoli Information Management for z/OS Operation and Maintenance Reference* if you want to decrease the record size or the CI size.

The Installation Tailoring Facility automatically sets your maximum record size equal to the CI size minus 7 (for both key formats).

Average record size
Run the BLGUT20 utility to determine the average record size. If this utility cannot be run, see “Calculating the Size of Your SDDS Records”.

DASD space for the SDDS
Next, consider the amount of space required for the SDDS. If you know the average record size and the desired number of records you would like to have the SDDS contain (being sure to consider your anticipated growth), you can calculate the amount of space needed by multiplying those two values. Then, convert that number to tracks or cylinders based on your device type. Be sure to select a CI size that allows an efficient use of the device. Quarter and half track allocations are typically optimum for both device utilization and performance.

Calculating the Size of Your SDDS Records

If you have a previous version of Tivoli Information Management for z/OS installed and have records established, you can run BLGUT20 to determine your average record size and record size distribution. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information about BLGUT20.

If you used PMF to modify the Tivoli Information Management for z/OS panels or to create your own panels, the amount of data collected by the prompting sequence can be considerably different from the amount of data collected by the unmodified Tivoli Information Management for z/OS prompting sequence panels. For example, if you modified the panels to reduce problem entry to a single data-entry panel containing only a few fields, you reduced the amount of data written to the database.

If you think the amount of data your panels collect is considerably different, use the following formula to calculate the average size of your Tivoli Information Management for z/OS logical records:

```
avgsize: = (rspcnt + auto) x 35 + (txtcnt x avgline) + (jrnlcnt x 44)
```

avgsize
Average size of your Tivoli Information Management for z/OS logical records at your organization.

rspcnt
Average number of panel responses contained in your logical record. Each field on a data-entry panel counts as two panel responses. Consider also the data collected automatically by TSPs.

auto
All entries added at file time or by TSPs.

txtcnt
Average number of text lines entered on the freeform text entry panels for a Tivoli Information Management for z/OS logical record.

avgline
Average length of a text line. Tivoli Information Management for z/OS provides a 60-character input line on which you can enter text. You also have the option of using the ISPF/PDF editor, which accepts up to 80 characters of text per line.

jrnlcnt
Average number of journalized entries for a logical record. A journal entry is made only for selected fields, but one is made every time you modify those fields. If you decide to journalize many fields or if you modify your journalized fields many times, you must increase this number. A typical value for the number of times a logical record's fields are journalized ranges from 5 to 50 or more.

Working with Multiple-Cluster SDDSs

To use multiple VSAM clusters, follow the naming conventions pertaining to multiple SDDS clusters (see the online help information for the Installation Tailoring Facility or "Defining the SDDS" on page 278) and specify the TRIGGER character when you define your session-parameters member (see the Installation Tailoring Facility or "BLGCLUST Macro — Defining a Database" on page 333).

If you plan to use multiple clusters, define them using the instructions given in the Installation Tailoring Facility or under "Defining the SDDS" on page 278, because all of the SDDS clusters are closely related and you specify only the first cluster name to Tivoli Information Management for z/OS. If you want to convert your one SDDS cluster database to a multiple SDDS cluster database, you must run the BLGUT7 utility. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for details about running BLGUT7.

The following section describes some factors you should consider when deciding whether to use multiple SDDS clusters.

Factors to Consider

- How large (cylinders/total bytes) is your current SDDS and is it expected to grow significantly?
- Are the advantages of using multiple clusters important?

  The advantages are:
  - Some performance improvement. In a non-sysplex environment with only one cluster, if a user updates the SDDS, all other users wait until that user completes. Waiting becomes more important when you have many active users (interactive, batch, or API). With multiple clusters, users can access other SDDS clusters while updates are occurring on another cluster.
  - More capacity. One SDDS cluster is limited to 4 gigabytes. Each additional cluster used adds another 4 gigabytes.
  - Reduced time needed for backups. Assuming you have a sufficient number of devices (normally tape drives), you can submit multiple jobs and have each job back up one of the clusters. Since the jobs can run simultaneously, you can reduce the overall time it takes to do a backup.

The disadvantage is:

- More clusters to manage.

- How often would you want to repeat this process?
  - The database is unavailable while BLGUT7 runs.
  - If you have 5 clusters and decide to convert to 6, you will have to perform all the conversion steps necessary, just as if you went from 5 to 10 clusters, but with only 20% more added capacity. However, if you convert from 5 to 10 clusters, database capacity increases by 100%.
  - Will you have enough DASD to do the conversion? You will need to have twice the amount of DASD tied up while you are converting—all the DASD for the current SDDS cluster(s), plus roughly the same amount for the new SDDS clusters. For example:

```
Assume:        Currently a single cluster that uses
               1000 cylinders

Converting to: 4 clusters

Then:          Define 4 new 250-cylinder clusters.
Result:        1000 + 250 + 250 + 250 + 250 = 2000 cylinders needed
               until BLGUT7 completes (the original 1000
               cylinders can be deleted after converting).
```

Note: The BLGUT23 series of utilities can be used instead of the BLGUT7 utility if DASD space is a concern. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using these utilities.

Determining Number of Clusters

To help determine how many clusters to use, do an IDCAMS LISTCAT on your current SDDS cluster(s).

- Look at the HI-USED-RBA value. How close to 4 gigabytes (4 294 967 295) is the HI-USED-RBA?
- How many new records do you expect? Remember that you can search using prefix DATE/ (date entered) to determine how many records per month or year you have created. Also, the BLGUT20 utility provides information on the size of your records.

If your HI-USED-RBA is less than 536 870 912 (half a gigabyte) you will most likely not want to do anything.

If your HI-USED-RBA is less than 2 147 483 648 (2 gigabytes) per SDDS cluster, depending on how the database is expected to grow, you might benefit from multiple clusters.

If your HI-USED-RBA is greater than 2 147 483 648 (2 gigabytes) per SDDS cluster, you should benefit from multiple clusters in performance and receive 4 gigabytes relief.

If you decide to increase the number of clusters, choose a number that would result in the new clusters being in the range of 536 870 912 (half a gigabyte) to 1 073 741 824 (1 gigabyte):

```
Assume:   Currently a single cluster with HI-USED-RBA
          of 3221225472
```

```
3221225472 divided by 1073741824 = 3

3221225472 divided by  536870912 = 6
```

You should define at least 3 clusters at a minimum. Define 4 or 5 clusters if you do not expect much growth in the database. Define 6 or more if you expect the database to grow over the next several years. Keep in mind that you will have to manage (back up) these data sets.

```
Assume:   Currently a 5-cluster SDDS

          SDDS$01 HI-USED-RBA of 2051235489
          SDDS$02 HI-USED-RBA of 2200929432
          SDDS$03 HI-USED-RBA of 1981778632
          SDDS$04 HI-USED-RBA of 2221225472
          SDDS$05 HI-USED-RBA of 1887331010

          Total number of bytes 10342500035
          (10.3 gigabytes)
```

SDDS$04 is the largest value, so using it for our calculations:

```
2221225472 divided by 1073741824 = 2.06

2221225472 divided by  536870912 = 4.12
```

Doubling the number of clusters to 10 would result in each of the 10 SDDS clusters having approximately 1 gigabyte of data. Converting to 20 clusters (4 x 5) would mean each cluster would have approximately half a gigabyte. You would need to manage 20 clusters, which might not be worth the effort unless the database is expected to grow beyond 40 gigabytes (4 gigabytes x 10 clusters).

Ten to 12 clusters would be a good choice, unless over the long term, the size was expected to grow well beyond 40 gigabytes.

Converting from Single to Multiple Clusters

If you are converting to multiple clusters, follow this procedure:

1. Define the new clusters.

   Reduce the CYLINDERS of the new SDDS using the same factor you chose for the number of SDDS. If you are doubling (converting from 5 to 10), use half the number of cylinders for each new SDDS. If you are converting from 1 to 4 clusters, each new SDDS will have only one-fourth of the data of the current SDDS, so each new SDDS will need only 25% of the cylinders used by the old cluster.

   Note: Use the same CISIZE, RECORDSIZE, and KEY values as your existing SDDS This will help improve the performance of BLGUT7. You could use the BLGUT20 utility to analyze the SDDS and choose to use a new record size, but BLGUT7 will take approximately twice as long to process.

   If you are not already using an SDDS key format of 7, it is now recommended that you do so. Key format 7 is needed to support the database backup solution provided with the BLGUT23 utilities. If you are running BLGUT7 to change the number of clusters, use the opportunity to change the SDDS to a key format of 7.

2. Update the VSAM resource definition (BLXVDEF) if you have a non-sysplex environment. If you are using sysplex support, skip this step and the next and proceed to step 4.
   - Add a new BLXDSN macro for each of the new clusters.

     Note: All the SDDS clusters can share the same LSR buffer pool. Since you would normally want to have the new clusters use the same LSR pool as the existing SDDS cluster(s), use the existing BLXDSN macro as a model for the new data sets, and change only the data set name.
   - Reassemble/re-link the VSAM resource definition.
3. Stop and restart the BLX-SP to pick up changes to the VSAM resource definition.

   Note: You can use the ADDVDEF command to temporarily add the SDDS clusters. You will not have to stop and restart the BLX-SP; however, you will still need to update the VSAM resource definition so that the next time the BLX-SP is stopped and restarted, the correct VSAM resource definition would be used.
4. Add TRIGGER to the BLGCLUST macro in your session members BLGSESXX.
5. Update BLGUT1.
   - Add OTRG to BLGUT1.
   - Verify that SESS=XX is now the session member you updated or created in the previous step.

   Note: You do not have to run BLGUT1 after running BLGUT7; however, this is the ideal time to make this change so that BLGUT1 is ready when needed.
6. Using the BLX-SP FREE command:
   - Free the SDIDS.
   - Free the old SDDS.
7. Using the BLX-SP REALLOC command:
   - REALLOC the old SDDS.

     Note: Use the UTIL parameter of the REALLOC command.
8. Run BLGUT7. Be sure to use the NEWTRIG parameter and verify that NEWSDDS and OLDSDDS are correct.
9. After BLGUT7 completes:
   - Use the BLX-SP FREE command to free the old single-cluster SDDS.
   - Use the BLX-SP REALLOC command to reallocate the SDIDS.
   - Use the new session member to access the new multiple-cluster SDDS. Do several searches and display several records to verify the database is behaving normally.
10. Create or update your backup jobs and procedures.
11. After you complete your tests:
    - (Non-sysplex only) Remove the BLXDSN macro for the old cluster in the VSAM resource definition.
    - (Non-sysplex only) Assemble/link-edit the VSAM resource definition.
    - Delete the old SDDS.

Note: The change to the VSAM resource definition will not be used until the BLX-SP is stopped and restarted. Leaving the BLXDSN macro in the VSAM resource definition will not affect anything.

Working with SDIDSs

With Information/Management Version 1.1, changes were made to the internal data structure of the SDIDS to support the use of multiple clusters. Multiple SDIDS clusters are optional, but they can help to reduce the contention for data stored in the SDIDS. Because the clusters are logically independent and can be accessed in parallel, they allow concurrent activity and can help to improve your overall database performance. In addition, other changes affecting the SDIDS include:

- The maximum amount of data you can store in the SDIDS is increased from 4 to 400 gigabytes. This change has no effect on the number of records you can store in your SDDS.
- The only SDIDS key lengths supported are 18 and 34. The 16- and 32- bytes keys are no longer available.

  Note: Database 6 is reserved for Tivoli Inventory and must be defined with an SDIDS key of 34.
- Only advanced compression is used.

This section provides guidelines you can follow to determine the maximum record length of the SDIDS. It also contains examples of how to set up a multiple-cluster SDIDS if you choose to use multiple clusters.

Maximum SDDS Records Formula (18-, 34-byte keys)

With the 18- or 34-byte key lengths, the maximum record size of the SDIDS does not limit the number of records that an SDDS may have from a practical point of view. The minimum number of SDDS records supported is over 119 million when using a SDIDS maximum record size of 505 bytes. The maximum is over 8 billion SDDS records when using a SDIDS maximum record size of 32 752. You will likely run out of physical space to store the SDDS records before you reach any limit imposed by the maximum record size of the SDIDS using the 18- or 34-byte SDIDS key length. Also, the 18- and 34-byte SDIDS key lengths completely eliminate the need for using spanned records.

In general, the sole consideration for determining the SDIDS record size is the CI size. Therefore, it is highly recommended that you use a half-track CI size, since this is the largest possible CI size that provides efficient track utilization for a storage device. This varies depending on DASD manufacturer, so check the documentation provided with your device for more details. Tests have shown that anything other than half-track blocking can seriously impact performance and DASD utilization.

To get a general idea of how many records your SDDS can contain, use the following formula:

```
sddsrecords = ((sdidsmaxrecsize - SDIDS key length - 16) x 8) x 32767
```

sddsrecords
Maximum number of records the SDDS can hold.

sdidsmaxrecsize
Specified in the **Maximum record size** field when you use the Installation Tailoring

Facility to define an SDIDS or in the RECORDSIZE keyword when you use the AMS DEFINE command. The value must be from 505 to CI size minus 7, up to a maximum of 32 752. The CI size can be 512 to 16 384 in multiples of 512, or it can be 2048 to 32 768 in multiples of 2048.

Effects of SDIDS Key Length Settings on Searches

The SDIDS key length can be set to 18 or 34 bytes. With a key length of 18, your search argument can be up to 16 characters long. With a key length of 34, your search argument can be up to 32 characters long. The extra two bytes are used by Tivoli Information Management for z/OS internally to prevent spanning and are not included as part of the searchable argument. The 34 key length enables you to do more exact searches, because you are able to search on more characters in the argument.

For instance, suppose you have many people in your company with long names and similar names (for example, Maryann Hoffman and Marylou Hoffman) and the practice at your company is to enter names without spaces. With a 18-byte SDIDS key, you get up to 16 characters of comparison on a search argument including the prefix:

```
search pers/hoffmanmaryann

search pers/hoffmanmarylou
```

In this example, the 16 characters are:

```
pers/hoffmanmary
```

With an 18-byte SDIDS key, if you enter more than 16 characters Tivoli Information Management for z/OS truncates the characters after the 16th character, and returns only those records that match up to 16 characters. Because Tivoli Information Management for z/OS is not able to distinguish between the two names, you will not get just the particular record you were looking for.

You could, through use of the 34-byte key, enter PERS/HOFFMANMARYANN and get a more exact match. A 34-byte key enables you to search more explicitly on the longer names. Tivoli Information Management for z/OS uses the additional characters beyond the 16th character to do searching, and does not truncate the search argument starting with the 17th character. Therefore, the search PERS/HOFFMANMARYANN would yield just Maryann's data records.

Report format tables, stored response chains, and terminal simulator panels or EXECs that do searches can also be affected by the key length defined for the SDIDS. If you decide to change the SDIDS key length, remember that you must also run the BLGUT1 or BLGUT1M utility.

Backup Recommendation

When the database grows so large that it requires the largest possible CI size, it is time to consider backup and recovery.

- An SDIDS MAXLRECL of 32 752 enables appropriate blocking on tape
- An SDIDS MAXLRECL of 32 752 is the largest record possible for the REORG utilities

Both of these values require a CI size of 16(2048)=32 768.

Working with Multiple-Cluster SDIDSs

Tivoli Information Management for z/OS allows you to set up multiple-cluster SDIDSs in addition to multiple-cluster SDDSs. If you store significant amounts of searchable data and are looking for ways to improve your overall database performance, you should consider setting up a multiple-cluster SDIDS. Multiple clusters allow enqueues for data to be obtained and released independently, thereby reducing contention for the SDIDS for more efficient throughput. With multiple clusters, you can store up to 400 gigabytes of SDIDS data.

If you plan to use multiple clusters, we recommend that you start with 10 (or more) SDIDS clusters, and then over a period of weeks add or consolidate SDIDS clusters.

The output from the BLGUT21 analyze utility can be used to select the starting key for each BLGCLKEY macro. Simply browse the output and issue a DOWN nnnnn command where nnnnn is one-tenth of the number of records in the current single cluster SDIDS. Depending on the key located through this simple technique, you may want to select a key in that general area that is more meaningful (such as a new prefix or s-word grouping). The key located will be used on the BLGCLKEY macro. After you use the new SDIDS configuration for a period of time, you can run the BLGUT21 utility again with the output from the BLX-SP operator command QUERY,TYPE=IO to adjust the number of SDIDS clusters. You should also plan on repeating this process over several weeks to tune the values used on the BLGCLKEY macro. If you prefer to start with fewer or more than 10 SDIDS clusters, modify this simple technique or select starting keys using any method you desire.

Note: The TYPE=IO parameter of the QUERY command is supported in a non-sysplex environment only.

If you are converting to multiple clusters, follow this procedure:

1. Define the new SDIDS clusters.

   Reduce the number of CYLINDERS for the new SDIDS using the same factor you chose for the number of SDIDS clusters. If you are doubling (converting from 1 to 2), use half the number of cylinders for each new SDIDS. If you are converting from 1 to 10 clusters, each new SDIDS should start with only one-tenth of the data of the current SDIDS, so each new SDIDS will need only 10 percent of the cylinders used by the old cluster.

   Note: After the new SDIDS clusters have been loaded with data, you can use the IDCAMS LISTCAT command to determine the actual amount of space each SDIDS used. Then, adjust the number of CYLINDERS using the DEFINE CLUSTER command to accurately match the actual space needed.

2. Update the VSAM resource definition. (This step applies only in a non-sysplex environment. If you are using sysplex support, skip to step 4.)

   Add a new BLXDSN macro for each new cluster. You may want to actually define more BLXDSN macros than you actually have clusters for, since it is likely that you may want to add more clusters to your SDIDS during the SDIDS tuning process. These additional BLXDSN macros will have no effect until the actual clusters they represent are used. However, having the extra BLXDSN macros will prevent you from having to stop and restart the BLX-SP. If you prefer, you can use the ADDVDEF command to add additional data sets.

   Note: All the SDIDS clusters can share the same LSR buffer pool. Since you would normally want to have the new clusters use the same LSR pool for the existing SDIDS cluster, use the existing BLXDSN macro as a model for the new data sets, and change only the data set name.

Reassemble and re-link the VSAM resource definition.

3. Stop and restart the BLX-SP to pick up changes to the VSAM resource definition. (This step applies only in a non-sysplex environment. If you are using sysplex support, skip this step.)

   Note: You could use the ADDVDEF command to temporarily add the SDIDS clusters. You would not have to stop and restart the BLX-SP; however, you would still need to update the VSAM resource definition so that the next time the BLX-SP is stopped and restarted, the correct VSAM resource definition is used.

4. Update your session-parameters members (BLGSESaa):
   - Add or update the BLGCLUST macro TRIGGER keyword to use multiple clusters.
   - Add the IDSKEYP keyword to the BLGCLUST macro in your session-parameters members.
   - Add BLGCLKEY macros to define the starting key ranges for each of the SDIDS clusters.
   - Assemble your updated session-parameters member(s) and link-edit them again.
5. Verify that your JCL for the BLGUT1 utility is correct so that BLGUT1 will be ready when needed. Verify that SESS=aa is now the session-parameters member you updated or created in the previous step.

   Consider setting up multiple BLGUT1 jobs. (For details, refer to the description of the CLUSTER parameter of the BLGUT1 utility in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.) Multiple BLGUT1 jobs can help to reduce the amount of time needed to rebuild multiple SDIDS clusters if a rebuild is necessary.

   Note: You do not have to run the BLGUT1 utility to create a multiple-cluster SDIDS. The BLGUT1M migration utility can be used to create the multiple-cluster SDIDS.

6. Using the BLX-SP FREE command:
   - Free the old SDIDS.
   - Free the SDDS(s).
7. Using the BLX-SP REALLOC command:
   - Reallocate the old SDIDS with the UTIL parameter.
   - Reallocate the new SDIDSs with the UTIL parameter.
8. Run the BLGUT1M migration utility to create the new multiple-cluster SDIDS.

   On future runs of BLGUT1M, you will need to use the OTRG= parameter to specify that the input SDIDS is composed of multiple clusters.
9. After the BLGUT1M migration utility completes:
   - Use the BLX-SP FREE command to free the old single-cluster SDIDS.
   - Use the BLX-SP REALLOC command to reallocate the SDDS(s).
   - Use the BLX-SP REALLOC command to reallocate the new SDIDSs.

- Use the new session-parameters member to access the new multiple-cluster SDIDS. Do several searches and display several records to verify that the database is behaving normally.

10. Create or update your backup jobs and procedures.
11. After you complete your tests, delete the old SDIDS.
12. Use the BLX-SP command QUERY,TYPE=IO to monitor the activity over several days or a week. This step applies only in a non-sysplex environment.

    Note: Do not use the RESET option on the QUERY,TYPE=IO command unless you are recording the values to acquire a large enough sample on which to base your adjustments. It is best to allow the values to accumulate over several days without resetting.

Tuning Your Multiple-Cluster SDIDS

After you have collected your sample of database activity, the primary goal is to balance the SDIDS clusters so that the PUT activity is consistent as possible across the SDIDS clusters. The GET activity can also be considered; however, GET activity is performed under a share enqueue which allows multiple users to access the cluster simultaneously.

Consolidate clusters with low write activity with adjacent clusters. For cluster with high PUT activity, add additional clusters to divide the key ranges. This is especially important for clusters with both high PUT and high GET activity. The size (cylinders or total number of records) of a cluster is not important. It is quite possible that, after this process is repeated several times, you will find that some clusters have relatively few records with high activity, and other clusters have large numbers of records with low activity. An exact balance may not be possible. An effective balance should be achieved by tuning your clusters three times over a period of time after initially creating the multiple-cluster SDIDS. The BLGUT21 analyze utility can be used to generate output that can identify records that are updated frequently. The UPDTE COUNT column indicates the number of times an SDIDS record has been updated. It is reset to zero each time the BLGUT1 or BLGUT1M utility is run. For details on running these utilities, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

If you are not using sysplex support, after you have achieved the desired balance, remove the BLXDSN macro for any unnecessary SDIDS cluster in the VSAM resource definition, and then assemble/link-edit the VSAM resource definition. Adjust the number of cylinders allocated by the AMS DEFINE CLUSTER command to match each SDIDS cluster’s actual needs.

Note: The change to the VSAM resource definition is not used until the BLX-SP is stopped and restarted. Leaving the BLXDSN macro in the VSAM resource definition will not affect anything, and could be useful if adjustments are needed later.

If you are using sysplex support, there is no need to redefine and assemble the VSAM resource definition.

Over time, or if the number of records in your database changes significantly, you may want to review the activity and make additional adjustments as necessary.

Example of a Multiple-Cluster SDIDS

As mentioned previously, you can use the BLGUT21 utility as a tool to help collect statistics on SDIDS activity. You should monitor the performance results of your clusters over a period, and adjust the structure as needed until you find the most ideal arrangement for your environment.

To illustrate one possible arrangement of a multiple-cluster SDIDS, suppose you wanted to separate s-words from cognized words starting with lowercase letters, and split the cognized word alphabet in half. You could set up the clusters as described in the following table. As an arbitrary arrangement, this division is not recommended specifically. It is simply an illustration of how you can set up multiple clusters.

Table 6. Sample Arrangement for Multiple-Cluster SDIDS

Cluster #	VSAM Data Set	Contents	Key Specified with BLGCLKEY macro	Internal Key Range Used
1	`BLM.PROD#01.SDIDS`	Cognized words starting with national characters or lowercase EBCDIC characters	Do not code BLGCLKEY for cluster 1.	`'0000'x` to `'B9FF'x`
2	`BLM.PROD#02.SDIDS`	S-words in the range starting with BA or BB, including `MASTER_BIT_LIST` and `LASTENTRYNUMBER`	`KEY=BA,TYPE=HEX`	`'BA00'x` to `'BBFF'x`
3	`BLM.PROD#03.SDIDS`	All remaining s-words	`KEY=BC,TYPE=HEX`	`'BC00'x` to `'C0FF'x`
4	`BLM.PROD#04.SDIDS`	Cognized words starting with character ’A’ through ’RM’	`KEY=A`	`'C10000'x` to `'D9D4FF'x`
5	`BLM.PROD#05.SDIDS`	All cognized words starting with character ’RN’	`KEY=RN`	`'D9D500'x` to `'D9D5FF'x`
6	`BLM.PROD#06.SDIDS`	All remaining cognized words	`KEY=RO`	`'D9D600'x` to `'FFFFFF'x`

The table shows that six clusters will be set up, with five of those clusters containing key value definitions.

Note: Each SDIDS cluster, except the first SDIDS cluster, must have a starting key value defined for it. Do not define a starting key value for the first SDIDS cluster; X'00' is predefined as the starting key value for this cluster.

The remaining key values determine which index information is contained in each cluster. Key values can be entered using the BLGCLKEY macro. The control record in each cluster contains the key range information for that cluster.

Using this example, you would specify the following input on your session-parameters member macros and keywords:

```
BLGSES00 CSECT
         BLGPARMS ...
DBS5     BLGCLUST NAME=5,TRIGGER=(#,1,6)
                  SDDS=DBS5DDS,
                  SDIDS=DBSIDS5,IDSSKEYP=KEY,
                  ...
DBS5DDS BLGCLDSN DSN=BLM.PROD.SDDS
DBS5IDS BLGCLDSN DSN=BLM.PROD#01.SDIDS
                  ...
KEY1    BLGCLKEY KEY=BA,TYPE=HEX
KEY2    BLGCLKEY KEY=BC,TYPE=HEX
KEY3    BLGCLKEY KEY=A
KEY4    BLGCLKEY KEY=RN
KEY5    BLGCLKEY KEY=RO
        ...
END
```

Note that the name of the first SDIDS cluster only is specified in the session-parameters member. Initialization takes care of connecting all remaining clusters. The name of the multiple-cluster SDIDS must conform to the rules currently used to name a multiple-cluster SDDS (that is, it should contain a trigger character followed by two numeric characters). Also, in the `TRIGGER` keyword, you should always specify one more than the number of BLGCLKEY macros you have for the count of SDIDS clusters.

Working with SDLDSs

The SDLDS is a preformatted relative record data set. Each record is fixed in length. This eliminates the need for VSAM to update certain information in the catalog. Thus, in the event of a system failure, the data in the SDLDS remains usable for reconstructing the SDDS if the SDDS becomes unusable.

You must offload the SDLDS to a backup sequential data at regular intervals. You can use the BLGUT4 utility, the Automatic Log Save Facility, or the DB2 Extract Facility to offload the SDLDS.

The Automatic Log Save Facility and the DB2 Extract Facility automatically offload the SDLDS for you at the interval you specify. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information about using the Automatic Log Save Facility and the DB2 Extract Facility.

In addition to offloading the SDLDS, the BLGUT4 utility also resets the SDLDS so you can reuse it. If the SDLDS fills up before it is normally offloaded, you cannot write to the SDDS. You must run the BLGUT4 utility at that time to reset the SDLDS. This unscheduled use of BLGUT4 can be a severe interruption to the Tivoli Information Management for z/OS users.

The BLGUT4 utility uses MVS data spaces to offload the SDLDS. Before running BLGUT4, you may need to update the MVS installation exit IEFUSI to increase the limit on the size and number of data spaces allowed. See *OS/390 MVS Installation Exits* for more information about IEFUSI.

You should allocate enough space to the SDLDS to prevent it from becoming full before it is normally offloaded. The amount of space required depends on the number of SDDS records that are written or deleted between offload operations. For example, if you offload the SDLDS every 24 hours (probably on third shift), you must allocate enough space to allow for 24 hours of activity.

The size of the SDLDS does not affect performance, so having an SDLDS that is larger than necessary will not decrease throughput.

You can use the same calculations that you used for computing the SDDS record size in "Working with SDDSs" on page 170 in the SDLDS calculation. The difference is that the SDDS space has to account for all of your records, while the SDLDS only has to account for the records between offloads. However, a new record is written to the SDLDS every time an SDDS record is updated. Therefore, you can have multiple instances of the same record in the SDLDS. For example, if you create a Tivoli Information Management for z/OS logical record and then update it several times, the record appears in the SDLDS as many times as it was updated. You can run DBCLEANUP periodically throughout the day to see the number of SDLDS positions used and available. For instructions on using DBCLEANUP, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

13

Using the Installation Tailoring Facility

This chapter discusses how to tailor Tivoli Information Management for z/OS using the Installation Tailoring Facility. This chapter assumes that you have planned:

- How your base Tivoli Information Management for z/OS system is to be configured
- The options and features that you want to install with Tivoli Information Management for z/OS

This chapter is designed to help you use the Installation Tailoring Facility.

Note: For a description of the tasks available through the Installation Tailoring Facility, see “The Installation Tailoring Facility” on page 28.

Allocating the Installation Tailoring Facility Data Sets

You must prepare your z/OS environment for running the Installation Tailoring Facility before you use it. You must allocate the SBLMEXEC data set to one of the following ddnames:

- SYSPROC
- SYSEXEC

  If you use SYSEXEC, you must also issue the following command:

  ```
  EXECUTIL SEARCHDD(YES)
  ```

Note: When you allocate the SBLMEXEC data set to either SYSPROC or SYSEXEC, be sure that SBLMEXEC is compatible with the other data sets in the SYSPROC or SYSEXEC concatenations. Check for the following possible incompatibilities:

- Whether the data sets are sequential data sets or partitioned data sets
- Whether the record formats of the data sets are variable or fixed. SBLMEXEC is preallocated as fixed block (FB) 80 (refer to the *Program Directory* for information on Tivoli Information Management for z/OS data set allocations). If the other data sets in your concatenation have different record formats, either change the allocation of SBLMEXEC or copy the members of SBLMEXEC into a data set that has a compatible record format.

Tivoli Information Management for z/OS runs as an application under the Interactive System Productivity Facility (ISPF) Licensed Program. The Installation Tailoring Facility automatically allocates SBLMINST to the following ddnames for you:

- ISPMLIB
- ISPPLIB
- ISPSLIB
- ISPTLIB
- ISPTABL

The Installation Tailoring Facility creates and uses two data sets to save information and members that are generated during an Installation Tailoring Facility session. The SBLMTABL contains tailoring specifications that you provide during an Installation Tailoring Facility session. The SBLMJCL1 data set contains the JCL and text members that are generated during the Installation Tailoring Facility session.

The Installation Tailoring Facility creates these data sets the first time you use the Installation Tailoring Facility. The Installation Tailoring Facility automatically accesses the data sets during subsequent uses of the facility.

Starting the Installation Tailoring Facility

You can start the Installation Tailoring Facility by issuing the BLGISTRT command from your ISPF menu or from the TSO READY prompt. The BLGISTRT REXX EXEC uses the ISPF LIBDEF service to define the Installation Tailoring Facility application data sets and to display the Installation Tailoring Facility main menu.

During initialization, the Installation Tailoring Facility identifies itself to ISPF as a new dialog application with the application name of BLGI. ISPF creates a profile called BLGIPROF. When the PDF editor is used to display JCL and text members, a BLGIEDIT member is created. Installation Tailoring Facility uses the BLGIPROF member to save values, such as the user's JOB statement and customization data set names.

Starting the Installation Tailoring Facility from an ISPF Menu

To enable your ISPF menu panel to start the Installation Tailoring Facility, specify the following command as a selection in the ISPF menu panel definition:

```
'CMD(BLGISTRT hlqualif)'
```

hlqualif

The high-level qualifier that the SBLMINST data set was given during SMP/E installation. This parameter is optional. If you omit this parameter, then the default value, *BLM.V2R1M0*, is used.

Note: The Installation Tailoring Facility uses the value that you specify for *hlqualif* as the high-level qualifier for all Tivoli Information Management for z/OS data sets that it references when it generates JCL statements such as SYSLIB and SYSLMOD. If your Tivoli Information Management for z/OS data sets are split among multiple high-level qualifiers, you must review the JCL that the Installation Tailoring Facility generates, check the high-level qualifiers of the Tivoli Information Management for z/OS data sets, and manually replace all incorrect high-level qualifiers in the JCL.

If you start the Installation Tailoring Facility from the ISPF menu panel, you can take advantage of ISPF functions by using the ISPF split screen facility; however, you cannot use the ISPF functions if you start from the TSO command line. Refer to the *ISPF Dialog Developer's Guide and Reference* for information on modifying ISPF menu panel definitions.

Starting the Installation Tailoring Facility from the Command Line

To start the Installation Tailoring Facility using the BLGISTRT command, issue the following command from your MVS command line (be sure you allocated data set SBLMEXEC to either SYSPROC or SYSEXEC).

```
BLGISTRT hlqualif
```

hlqualif

The high-level qualifier that the SBLMINST data set was given during SMP/E installation. This parameter is optional. If you omit this parameter, then the default value, *BLM.V2R1M0*, is used.

Note: The Installation Tailoring Facility uses the value that you specify for *hlqualif* as the high-level qualifier for all Tivoli Information Management for z/OS data sets that it references when it generates JCL statements, such as SYSLIB and SYSLMOD. If your Tivoli Information Management for z/OS data sets are split among multiple high-level qualifiers, you must review the JCL that the Installation Tailoring Facility generates, check the high-level qualifiers of the Tivoli Information Management for z/OS data sets, and manually replace all incorrect high-level qualifiers in the JCL.

Error Recovery

The Installation Tailoring Facility modifies some of your ISPF profile values and the TSP profile prefix value. If you encounter a problem while using the Installation Tailoring Facility and it does not end correctly, you must reset the TSO profile value as follows:

```
TSO PROFILE PREFIX(sysuid)
```

You may also need to reset some of your ISPF profile values (for example, the command line placement).

Using Online Help Information

```
        Information Management for z/OS Installation Tailoring Facility

Type an option number, and then press Enter.

_  1. Getting Started
   2. Installing the Base Product
   3. Installing Optional Features
   4. Adding to or Changing the Base Product
   5. Working with Customized JCL/Text Members











Command ===> ______________________________________________________________
 F1=Help      F2=Split     F3=Exit      F9=Swap     F11=Retrieve F12=Cancel
```

Figure 16. The Installation Tailoring Facility Main Menu

The Installation Tailoring Facility provides the following types of online help information:

Online introduction

Provides a detailed explanation of the Installation Tailoring Facility. The online introduction can be accessed by selecting option 1, Getting Started, from the Installation Tailoring Facility main menu (see Figure 16).

Extended help
Provides information about the purpose and content of an Installation Tailoring Facility panel. To obtain extended help:
- Press the Help function key (F1 or F13) when the cursor is not on an entry field.
- Press the Ex Help function key (F2 or F14) while field help is being displayed.
- Press the Ex Help function key (F2 or F14) while message help is being displayed.

Field help
Provides an explanation and syntax information for each field. Required fields and default values are specified. Field help is obtained by pressing the Help function key (F1 or F13) when the cursor is on an entry field.

Message help
Provides explanations for Installation Tailoring Facility messages. Message help is obtained by pressing the Help function key (F1 or F13) when a message is displayed.

Help for Keys
Provides a listing of the function keys available to the user and an explanation of the function. Help for Keys is obtained by pressing the Keyshelp function key (F9 or F21) from any help panel.

Entering Values on Installation Tailoring Facility Panels

The Installation Tailoring Facility panels contain display-only fields, entry fields, and single- and multiple-selection fields:

- Display-only fields cannot be changed; they provide additional information about the panel's content.
- Entry fields may or may not be prefilled with values. You can change the value in the field if the field is prefilled. See "Default Values".

  Use field help to help you determine the valid values for a particular field. If you see a plus sign (+) displayed next to an entry field, you can place your cursor on the field and press the Prompt function key (F4 or F16). A list of possible values are displayed for you to choose from.
- With a single-selection fields, type the number of the item that you want. A number may already be prefilled for you. See Default Values.
- Use a slash (/) to select one or more items in multiple-selection fields.

Default Values

Some of the entry fields and single-selection fields are prefilled with default values. These default values are either Tivoli Information Management for z/OS or Installation Tailoring Facility default values.

- A Tivoli Information Management for z/OS default value is a value that Tivoli Information Management for z/OS uses if you do not specify your own value. You can change these values to ones that are more suitable for your installation. However, if a field does not require a value, and you delete the prefilled default value, the default value is still used.
- An Installation Tailoring Facility default value is a value that the Installation Tailoring Facility uses if you do not specify your own values. You can change these values to

ones that are more suitable for your installation. However, if a field does not require a value and you delete the prefilled default value, the default value is still used.

Note: The field names of the fields that require values are highlighted on the Installation Tailoring Facility panels.

Using the Installation Tailoring Facility Program Function Keys

The Installation Tailoring Facility defines keys that perform certain functions. The Exit and Cancel functions take you through multiple panels, and the Bkwd and Fwd functions take you through multiple pages of a single panel.

The following program function keys are available from most panels:

F1 or F13 (Help)

- Displays a field help panel when the cursor is on a specific field
- Displays an extended help panel when the cursor is not on a specific field
- Displays a help panel when a message is displayed on the panel.

F2 or F14 (Split or Ex help)

- Splits the screen when the Split function is available
- Displays an extended help panel when the Ex help function is available.

F3 or F15 (Exit)

- Returns to the Installation Tailoring Facility main menu when the cursor is on a menu selection panel
- Returns to a menu selection panel when the cursor is on an entry panel

  Note: Any information that you entered is not saved.
- Returns to the entry or menu selection panel when the cursor is on a help panel.

F4 or F16 (Prompt or Window)

- Makes the list of prompt choices appear when the Prompt function is available
- Enables you to move a pop-up panel to a different position on the screen when the Window function is available.

F7 or F19 (Bkwd)

- Scrolls backward through the panel one page at a time.

F8 or F20 (Fwd)

- Scrolls forward through the panel one page at a time.

F9 or F21 (Swap or Keyshelp)

- Toggles between the two parts of a split screen when the Swap function is available
- Displays the Help for Keys panel when the Keyshelp function is available.

F11 or F23 (Retrieve)

- Recalls the last command you entered on the ISPF command line and places it on the command line for you when the Retrieve function is available.

F12 or F24 (Cancel)

- Returns to the previous panel. If the current panel is an entry panel, then any information that you entered is not saved.

14

Securing Tivoli Information Management for z/OS Information

This chapter describes how to use the z/OS Security Server (which includes Resource Access Control Facility (RACF) to secure your Tivoli Information Management for z/OS information.

This chapter assumes that you have a working knowledge of RACF. For more information about RACF, refer to the RACF documentation (see the Bibliography).

This chapter is designed to help you protect your Tivoli Information Management for z/OS information.

Programming Interface information

Using RACF to Protect Tivoli Information Management for z/OS VSAM Data Sets

RACF provides three types of access: READ, UPDATE, and NONE. Two RACF Class Descriptor Table entries (INFOMAN and GINFOMAN) are supplied by Tivoli Information Management for z/OS for defining access to Tivoli Information Management for z/OS VSAM data sets. These table entries can be used only in a non-sysplex environment. When sysplex mode is enabled, user access to data sets is controlled solely through the use of data set profiles.

INFOMAN
A RACF resource class. Use this resource class to define profiles for specific VSAM data sets in a non-sysplex environment.

GINFOMAN
A RACF resource class group representing a set of INFOMAN profiles. Use this resource class group to define access to multiple VSAM data sets in a non-sysplex environment. When an Tivoli Information Management for z/OS user is authorized to access a GINFOMAN group, RACF permits the user to access all of the INFOMAN profiles defined in that GINFOMAN group.

The GINFOMAN resource class group can be used by a system administrator to simplify access to Tivoli Information Management for z/OS databases. Instead of having to authorize users to access each individual INFOMAN profile, you can use GINFOMAN to group INFOMAN profiles together. Then you only need to give your users authorization to access the GINFOMAN group name instead of all the individual INFOMAN profiles.

The INFOMAN and GINFOMAN entries specify a maximum length of 39 characters for a data set name. Use these two names when defining access for Tivoli Information Management for z/OS VSAM data sets. The system administrator is responsible for defining and controlling access to all VSAM data sets.

For Tivoli Information Management for z/OS, the following requirements must be met when using RACF to protect Tivoli Information Management for z/OS VSAM data sets:

- System administrators must permit their user IDs UPDATE access to all the resources and data sets that they need to perform their tasks, and READ access to all the resources and data sets that are used by other users.
- Protect all Tivoli Information Management for z/OS VSAM data sets with a universal access of NONE. However, system administrators must have READ or UPDATE access to the data sets.
- For each started task (BLX-SP) and Multiclient Remote Environment Server (MRES), you must create a RACF started task table entry and a RACF user ID to associate with it.

  Note: See “Defining a BLX-SP Procedure” on page 146 for an example of the started task.
- The BLX-SP must be permitted CONTROL authority to all of the Tivoli Information Management for z/OS VSAM data sets that users of that BLX-SP require access to.
- A profile for each Tivoli Information Management for z/OS VSAM data set must also be defined to RACF in the INFOMAN resource class.
- Permit each Tivoli Information Management for z/OS user either READ or UPDATE access to the profiles defined in the INFOMAN resource class for the data sets they are authorized to use.
- The BLX-SP determines the level of access defined for a specific user to a specific Tivoli Information Management for z/OS VSAM data set by querying entries in the INFOMAN RACF resource class.
- The GINFOMAN class enables an organization to define multiple profiles in the INFOMAN resource class with a single command. If this is used, it is also possible to permit a single user to have access to all of the profiles in that group with a single command.

If your organization has a large number of Tivoli Information Management for z/OS users, the suggested method is to create a group for each set of users with similar access requirements. The group is put in the access lists instead of a large number of users. This method makes profile maintenance simpler.

Example Scenario

The following scenario takes you through the commands for creating a RACF group, for creating RACF profiles, and for giving RACF access for a database.

Creating a Group

Your system administrator, Pat, wants to create a group named PROBLEM with an owner of USERPAT (Pat’s user ID) and a superior group of SYS1, a predefined RACF group. To do this, Pat uses the following command:

```
ADDGROUP PROBLEM OWNER(USERPAT) SUPGROUP(SYS1)
```

To connect users USER1, USER2, and USER3 to the PROBLEM group, Pat uses the following command:

```
CONNECT (USER1 USER2 USER3) GROUP(PROBLEM) OWNER(USERPAT)
```

Creating a Generic Profile

Pat wants to protect a database that consists of the following data sets:

- BLM.DATABASE.SDDS
- BLM.DATABASE.SDIDS
- BLM.DATABASE.SDLDS

To create a generic profile to protect this database, Pat uses the following command:

```
RDEF INFOMAN BLM.DATABASE.* NOTIFY OWNER(USERPAT) UACC(NONE)
```

In the following command, Pat gives the PROBLEM group UPDATE access to the database by giving the group UPDATE access to the generic profile:

```
PERMIT BLM.DATABASE.* CLASS(INFOMAN) ACCESS(UPDATE) ID(PROBLEM)
```

Creating a Resource Group Profile

To create a resource group profile named DATAB1 to protect the database, Pat uses the following command:

```
RDEF GINFOMAN DATAB1 ADDMEM(BLM.DATABASE.SDDS BLM.DATABASE.SDIDS
 BLM.DATABASE.SDLDS) NOTIFY OWNER(USERPAT) UACC(NONE)
```

In the following command, Pat gives the PROBLEM group UPDATE access to the database by giving the group UPDATE access to DATAB1:

```
PERMIT DATAB1 CLASS(GINFOMAN) ACCESS(UPDATE) ID(PROBLEM)
```

Note: You must activate the INFOMAN class when using resource group profiles (that is, GINFOMAN). Otherwise, GINFOMAN profiles are not used. Use the following command to activate INFOMAN:

```
SETROPTS RACLIST(INFOMAN)
```

Profiles are loaded into storage when they are first accessed. If profiles are updated in the INFOMAN or GINFOMAN classes, they are not effective until they are reloaded into storage. Use the following command to tell RACF to reload the profiles when they are next accessed:

```
SETROPTS RACLIST(INFOMAN) REFRESH
```

End of Programming Interface information

Defining a Profile in the RACF Started Class

If you want to provide security for your shared database systems and MRES, you must define a profile in the RACF started class for each BLX-SP and MRES on that system that shares data sets.

Granting Read Access to Tivoli Information Management for z/OS Data

If users are to be given READ access for any Tivoli Information Management for z/OS data, the session-parameters member they use must specify RDONLY=YES for that data. Otherwise, UPDATE authority is requested.

Ensuring the Proper Usage of TSP and TSX Control Lines

After installing Tivoli Information Management for z/OS, you should take appropriate action to ensure data integrity and security. This section describes security measures you can use to protect against the misuse of certain sensitive functions of the Tivoli Information Management for z/OS program.

There are certain control lines you can use in a TSP or TSX that require precautions against their unauthorized use. The following is a discussion of these control lines and some suggestions on how to prevent their unauthorized use. Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* for details on TSPs and TSXs.

ADDSDATA

The ADDSDATA control line enables you to add data to records in the Tivoli Information Management for z/OS database. It is the TSX equivalent of the add function of the TSP WORDFIX control line. You cannot be protected from damaging your database if you make a mistake when using this control line.

You must limit the use of this function to someone with extensive knowledge of Tivoli Information Management for z/OS. Because this control line uses the same load module (BLGMWDFX) as WORDFIX, you can restrict the use of ADDSDATA by moving the BLGMWDFX load module to a RACF-protected data set.

This control line can be used with TSXs only.

DELSDATA

The DELSDATA control line enables you to delete data from records in the Tivoli Information Management for z/OS database. It is the TSX equivalent of the delete function of the TSP WORDFIX control line, but it also provides the ability to delete an entire structured description entry (SDE) with a single control line call. You cannot be protected from damaging your database if you make a mistake when using this control line.

You must limit the use of this function to someone with extensive knowledge of Tivoli Information Management for z/OS. Because this control line uses the same load module (BLGMWDFX) as WORDFIX, you can restrict the use of DELSDATA by moving the BLGMWDFX load module to a RACF-protected data set.

This control line can be used with TSXs only.

FLATTEN

The FLATTEN control line enables you to take a Tivoli Information Management for z/OS record and reformat it into a form that you can transport outside the database.

FLATTEN does not check your authority to access a record. After a record is flattened, you have access to information in the record that you might not be able to examine online. Therefore, if you had no access authority to display a record, you could flatten it and, with some difficulty, look at the information in the record.

In addition, you could also manipulate information in the flattened record via a user exit from a USEREXIT control line or a separate program.

This control line can be used with both TSPs and TSXs.

Note: Modifications made to flattened or unflattened records are done at your own risk and are not supported by Tivoli. To manipulate records outside of Tivoli Information Management for z/OS, refer to the use of:

- The Archiver function (*Tivoli Information Management for z/OS Program Administration Guide and Reference*)
- Terminal simulator EXECs (*Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*)
- APIs (*Tivoli Information Management for z/OS Application Program Interface Guide*)

UNFLATTEN

The UNFLATTEN control line enables you to take a flattened Tivoli Information Management for z/OS record and restore it to a database.

UNFLATTEN does not check your authority to create a record. It is possible for you to update information in the flattened record prior to unflattening it, either with a user exit specified in a USEREXIT control line or by a separate program.

This control line can be used with both TSPs and TSXs.

In addition, if you have not set up your databases and procedures correctly, it is possible to contaminate your database. The unflattened record is restored to an equivalent database (the database in the current session that has the same database identifier—specified on the NAME parameter in the BLGCLUST macro—as the original database from which the record was flattened). If you process the TSP and specify the wrong session-parameters member, you can unflatten the record into the wrong database.

Note: See “BLGCLUST Macro — Defining a Database” on page 333 for a description of the BLGCLUST macro.

If the database into which records are unflattened is logically partitioned, the primary partition name (primary partition ID) from the privilege class of the user performing the UNFLATTEN becomes the owning partition name of the unflattened records. If the user’s privilege class does not contain a primary partition name, the unflattened records will not be assigned an owning partition name. The *Tivoli Information Management for z/OS Program Administration Guide and Reference* contains additional information about logically partitioned databases.

Note: You should exercise extreme caution when unflattening records with system-assigned (numeric) record number IDs. When unflattened records are to retain their original record number IDs, the last entry number for each database partition is not affected by the UNFLATTEN process. It is possible to unflatten a record with a system-assigned record number ID that is higher than the last entry number value. If new records are created, they may have the same record number ID (obtained from the last entry number value) as an unflattened record.

WORDFIX

The WORDFIX control line enables you to change records in a Tivoli Information Management for z/OS database. This function circumvents normal panel flow processing and validation. Because there is no validation done at processing time, you cannot be protected from damaging your database if you make a mistake. You must limit the use of this function to someone with extensive knowledge of Tivoli Information Management for z/OS.

This control line can be used with TSPs only.

Preventing the Unauthorized Use of Control Lines

To prevent the unauthorized use of these control lines, you must do the following:

- Place the load modules (listed in Table 7) in RACF protected data sets. This enables anyone with PMF authority to create a TSP with the TSP control lines in it. However, only users with RACF authority to access the data set containing the load modules can process the TSP. Any attempt to access the load modules without proper RACF access authority results in a RACF authorization error.
- Place the panels used to create the control lines in RACF protected data sets. This limits the usage of these control lines to those who have RACF access authority to the data sets containing those panels. Any attempt to access the panels without proper RACF access authority results in a RACF authorization error.

You can combine both the above options to achieve greater security.

- In addition to using the previously described options for data security, you can use PMF to remove these control lines from the TSP create dialogs, so that no one can use them.

Table 7 shows the control lines, their load modules, and associated panels where applicable.

Table 7. Control Lines, Their Load Modules, And Associated Panels

Line	Load Module	Panel
FLATTEN	BLGMFLA	BLM8CU9Z
UNFLATTEN	BLGMUFL	BLM8CU9X
WORDFIX	BLGMWDFX	BLM8CU9M
ADDSDATA	BLGMWDFX	–
DELSDATA	BLGMWDFX	–

Using Command Aliases/Authorization to Restrict Commands

If you have a need to restrict the usage of certain standard Tivoli Information Management for z/OS commands, such as the VIEW command to view the internals of a record, you can define a command alias identical to the command name and set an authorization code for it.

Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information on defining command aliases and authorization.

15

Preparing to Start Tivoli Information Management for z/OS

This chapter describes tasks for you to perform before users can start Tivoli Information Management for z/OS. The following tasks are described:

- Loading the read panel data set
- Using multiple read panel data sets
- Loading the dictionary data set
- Formatting the SDLDS
- Selecting the Tivoli Information Management for z/OS panel style
- Installing the graphical user interface
- Defining report format table data sets
- Setting up a remote printer

15. Preparing to Start

Loading the Read Panel Data Set

Before any users can use Tivoli Information Management for z/OS, you must first load the read panel data set. To load the read panel data set (RPANLDS), use the BLGUT6 or BLGUT6M utility program. The procedures for using these utility programs are described in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

| **BLGUT6** Enables you to load the read panel data set and use the panels as they are shipped (uncustomized). The panels support the entry of date fields in any of the 22 supported external date formats. Users can enter dates in the external date format specified by their session parameters (BLGPARMS DATEFMT keyword), or they can select their own date format preference through the user profile. (For a description of date processing, see “Enabling Alternative Date and Time-of-Day Formats” on page 227.)

BLGUT6M Enables you to load the read panel data set but also change the field lengths and validation patterns used by the shipped panels. For example, you can use BLGUT6M to convert the date fields on the shipped panels from 10 to 8 characters or to some other date format.

If you are migrating from a previous release and have not already reviewed the migration information about selecting date formats, see “Migrating from Previous Versions” on page 103 and page 135 for more information. If Tivoli Information Management for z/OS is being installed for the first time and you would like to review information about selecting and specifying date formats, see “Enabling Alternative Date and Time-of-Day Formats” on page 227.

If you use a non-Latin translate table, be sure to translate all of the Tivoli-supplied panels to uppercase when loading them into the data set: specify the UPPERCASE keyword when using BLGUT6.

Note: You must start the BLX-SP before you can use the Tivoli Information Management for z/OS utilities.

Using One or More Read Panel Data Sets

For maintenance reasons, it is best to have two read panel data sets: one for Tivoli-supplied panels and one for user-modified panels. You get better performance, but you use more DASD space when you use only one read panel data set for production. If this is what you want, do the following:

1. Create at least three read panel data sets:
   - One (for example, RPANEL0) to contain the Tivoli base panels
   - A second (for example, RPANEL1) to contain the Tivoli panels and your modified panels
   - A third (for example, RPANEL2) or multiple panel data sets to test your modified panels
2. Load the Tivoli base panels into RPANEL0 (keep this data set as a backup).
3. Copy the Tivoli base panels from RPANEL0 into RPANEL1 (copy only once).
4. Test your new and modified panels in RPANEL2.
5. Copy your tested panels from RPANEL2 into RPANEL1.
6. Use RPANEL1 as your production read panel data set.

You use less DASD space, but you do not get better performance when you use two read panel data sets for production. If this is what you want, do the following:

1. Create at least three read panel data sets:
   - One (for example, RPANEL0) to contain the Tivoli base panels
   - A second (for example, RPANEL1) to contain your new and modified panels
   - A third (for example, RPANEL2) or multiple panel data sets to test your modified panels.
2. Load the Tivoli base panels into RPANEL0.
3. Test your new and modified panels in RPANEL2.
4. Copy your tested panels from RPANEL2 into RPANEL1.
5. Use RPANEL1 and RPANEL0 as your production read panel data sets. Concatenate RPANEL1 before RPANEL0 in the session-parameters member that you use in production.

Note: Do not use IDCAMS REPRO to copy panels from one VSAM panel data set to another. Using REPRO to copy panels can damage the panel data set that the panels are being copied into. If message BLM04098W is issued for a panel, that panel is permanently damaged. Instead, use the panel copy function in PMF, or use BLGUT6F to copy the panels to a partitioned data set (PDS), and then use BLGUT6 to copy the panels from the PDS to the target VSAM panel data set.

For details about using PMF, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*. For information about using BLGUT6 and BLGUT6F, refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

Whether you decide to use one or two read panel data sets for production, you can use the RACF program or its equivalent, to protect RPANEL0 and RPANEL1. Give your users read access only, and give your system administrator update access to those panel data sets. Or, you can specify the RDONLY keyword in the session parameters on the BLGCLDSN macro for these panel data sets.

For more information about read panel data sets, see “Multiple Read Panel Data Sets” on page 43.

Loading the Dictionary Data Set

If you use a DICTDS, you must load it before you use it. Use the BLGUT5 utility program to load the DICTDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using BLGUT5.

Note: You must start the BLX-SP before you can use the Tivoli Information Management for z/OS utilities.

Formatting the SDLDS

If you use an SDLDS with your database, you must format it before you use it. Use the BLGUTR utility program to format it. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using BLGUTR.

Note: You must start the BLX-SP before you can use the Tivoli Information Management for z/OS utilities.

Selecting the Tivoli Information Management for z/OS Panel Style

You must decide which Tivoli Information Management for z/OS panel style your users see. You can choose either the standard panel style that previous versions of Tivoli Information Management for z/OS used or the enhanced panel style. For more information on the enhanced panel style, refer to the *Tivoli Information Management for z/OS User’s Guide* and the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

To select the panel style for your Tivoli Information Management for z/OS installation, do the following:

1. Copy one or more of the following members of the SBLMSAMP library to a data set that is concatenated to the user’s ISPPLIB ddname:
   - BLGISPFD—the standard panel style
   - BLGISPFE—the primary window for the enhanced panel style
   - BLGISPFI—the inquiry window for the enhanced panel style
   - BLGISPFA—the administration window for the enhanced panel style
2. Create the default panel style member by copying either BLGISPFD or BLGISPFE to a new member named BLGISPFM. Copy BLGISPFM to a data set that is concatenated to the user’s ISPPLIB ddname.

3. If you want to use one of the enhanced panel styles, you must also copy the following members of the SBLMSAMP library to a data set that is concatenated to the user's ISPTLIB ddname:
   - BLG0CMDS
   - BLG0KEYS

Tivoli Information Management for z/OS defaults to the panel style that BLGISPFM contains when it starts a Tivoli Information Management for z/OS session. If you select the enhanced panel style, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information on switching between the various styles of enhanced panels.

Installing the Graphical User Interface

You can display the Tivoli Information Management for z/OS panels in workstation windows by:

- Installing the ISPF Client/Server (ISPF C/S) component on each workstation. Refer to the *ISPF User's Guide* for information on installing the ISPF C/S component.
- Installing any software required by ISPF for running an application in ISPF's GUI mode. Refer to the *ISPF User's Guide* for more information on software requirements.

Note: To use GUI mode for AIX or HP clients, you must have ISPF installed and running.

The ISPF GUI mode should not be confused with the with Java-based Tivoli Information Management for z/OS Desktop or any Desktop applications provided with Tivoli Information Management for z/OS. The Tivoli Information Management for z/OS Desktop uses a different graphical user interface and is an alternative way to access the Tivoli Information Management for z/OS database. For more information about the Desktop GUI, refer to the *Tivoli Information Management for z/OS Desktop User's Guide*.

Defining Report Format Table Data Sets

You must allocate a report format table data set (RFTDS), when you start a Tivoli Information Management for z/OS session, if you want to print Tivoli Information Management for z/OS records or reports or use API tables or RDMTs. (See "The Report Format Table Data Set" on page 24 for more information about RFTDSs.) The RFTDS is a PDS with a logical record length of 80 and a fixed blocked record format. Tivoli Information Management for z/OS supplies predefined report format tables (RFTs), API tables, and RDMTs in the SBLMFMT data set. You can use this data set as your RFTDS.

Note: If you create or modify the members of the SBLMFMT data set, give the new or altered members new names. This ensures that you do not write over Tivoli-supplied members that you may want to keep.

You may want to create additional RFTDSs if you define your own RFTs, modify copies of the Tivoli-supplied RFTs, or create your own API tables or RDMTs. The first RFTDS you use in a Tivoli Information Management for z/OS session must be allocated in the session-parameters member. In the session-parameters member, you specify the RFTDS name and the ddname to allocate the data set to. If you allocate additional RFTDSs, you must allocate them in some other way, for instance, in a CLIST (see Figure 17 on page 208). Create them with the same logical record length (LRECL) and record format (RECFM) attributes. If you do concatenate the RFTDSs, the block sizes of all the concatenated data sets do not have to be the same. The data sets must be concatenated to the ddname specified in the session-parameters member.

Setting Up a Remote Printer

The output destination for print output and report output can be defined in a user's profile. If it is not defined, a user can select the destination when issuing the PRINT or REPORT command.

Note: Refer to the *Tivoli Information Management for z/OS User's Guide* for details about defining output destinations. Refer to the *Tivoli Information Management for z/OS Data Reporting User's Guide* for further details about defining output destinations specifically for reports. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for specifications about the APIPRINT print data set.

Sending Output to a Remote Printer

You can set up an 8-character nickname that JES recognizes as a remote printer. You must set up some parameters in HASPPARM that match this nickname. Set up the destination as remote so that you can use the nickname in Tivoli Information Management for z/OS. This nickname can also represent a parameter, such as a room number, bin information, or accounting information, that you want to pass to JES for printing on a report.

If you use a TSP print on a remote printer, you must preallocate the SYSOUT with JCL and set the PROFILE print destination to go to that ddname. The TSP sends the job to the remote printer after the Tivoli Information Management for z/OS session ends and the ddname is freed.

When SYSOUT is selected as the print output destination, Tivoli Information Management for z/OS uses DYNAMIC ALLOCATION to get a SYSOUT data set allocated. SYSOUT must be set to *class*.

Printing on a VTAM-attached Printer

To print on a VTAM®-attached printer, define that printer to JES as a node (you do this with JES328X). Then in Tivoli Information Management for z/OS, identify the node as a remote device. You can also send output to a user ID on a remote system by updating your user profile or by using the fields on the SYSOUT panel.

16

Starting Tivoli Information Management for z/OS

This chapter describes alternate ways to start Tivoli Information Management for z/OS:
- From ISPF
- In batch mode
- In batch mode with GUI communications for a workstation client
- From NetView

Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* and the *Tivoli Information Management for z/OS Client Installation and User's Guide* for information on starting the APIs.

This chapter assumes that you:

- Have a working knowledge of at least one of the following:
  - ISPF and TSO CLISTs. Refer to the *ISPF Dialog Developer's Guide and Reference* and *TSO/E CLISTs* for more information.
  - MVS Job Control Language (JCL). Refer to the *OS/390 MVS JCL User's Guide*.
  - NetView. Refer to *NetView Operation*.
- Finished tailoring your installation of Tivoli Information Management for z/OS (see "Overview for Planning and Installing" on page 1).

Note: Before you start Tivoli Information Management for z/OS, you must start the BLX-SP (see "Starting and Stopping the BLX-SP" on page 148).

Starting Tivoli Information Management for z/OS from ISPF

Tivoli Information Management for z/OS runs as an application under the Interactive System Productivity Facility (ISPF) Licensed Program.

Under ISPF, you can start Tivoli Information Management for z/OS in one of two ways:
- By issuing the ISPSTART command
- By selecting Tivoli Information Management for z/OS from your ISPF menu

Issuing the ISPSTART Command

You can start Tivoli Information Management for z/OS by issuing the ISPSTART command with the PGM(BLGINIT) operand. Refer to *ISPF Dialog Developer's Guide and Reference* for complete information on the ISPSTART command. See "ISPSTART Syntax" on page 204 for an example of the ISPSTART command.

Note: Even though Tivoli Information Management for z/OS runs as a separate application under ISPF, it is not necessary to specify NEWAPPL(BLG0) on the ISPSTART command; Tivoli Information Management for z/OS automatically supplies this information for you.

Allocating Data Sets

When you start Tivoli Information Management for z/OS with the ISPSTART command, you must:

- Preallocate all of the data sets that ISPF uses for its processing (that is, ddnames ISPPLIB, ISPMLIB, ISPSLIB, ISPTLIB, and ISPPROF).
- Do one of the following with the Tivoli Information Management for z/OS load modules; be sure you include the BLXSSINM module for the target BLX-SP subsystem:
  - Place them in an ISPF link library and preallocate the library to ddname ISPLLIB.
  - Define them in a STEPLIB DD statement.
  - Place them in a system link list.
  - Place them in the link pack area.

You do not need to preallocate any of the Tivoli Information Management for z/OS VSAM data sets because Tivoli Information Management for z/OS performs this dynamically during initialization.

Note: If you are using non-VSAM data sets, such as output file BLGFLOW, you must allocate them. RFT data sets can be preallocated or not (see “Defining Report Format Table Data Sets” on page 200).

You can perform these allocations automatically by placing them in the user’s TSO logon procedure using DD statements or in a TSO CLIST using TSO ALLOCATE commands. Both methods handle all the allocations, so you do not have to manually allocate each data set.

If you provide a CLIST, consider having the CLIST let the user specify the needed parameters for the PARM keyword of the ISPSTART command. See “Sample CLISTs to Start Tivoli Information Management for z/OS” on page 207 for a sample CLIST for processing Tivoli Information Management for z/OS.

You can allocate an optional data set with the ddname of BLGSMSG in your TSO logon procedure to receive any Sort/Merge licensed program messages generated during your use of Tivoli Information Management for z/OS.

ISPSTART Syntax

The syntax of the ISPSTART command follows:

```
ISPSTART  PGM(BLGINIT)
          [PARM(  [CLASS(name)]
                  [SRC [(name)]  | NOSRC]
                  [IRC(immediate response chain)]
                  [TSP(name)]
                  [SESS(suffix)]  )]
          [GUI(LU:display|IP:display|,NOGUIDSP)]
          [TITLE(title)]
          [GUISCRW(screen-width)]
          [GUISCRD(screen-depth)]
```

Optionally, you can specify PARM and one or more of the following keywords. If you specify any of these keywords incorrectly, Tivoli Information Management for z/OS ignores the error. Your session continues as though you did not specify the keyword or data. You can enclose the keywords in single quotation marks so that the ISPSTART command ignores any special characters that are used as keyword values. See the ISPSTART command in Figure 19 on page 212. If you enter a keyword more than once, or specify mutually exclusive keywords, the last keyword remains in effect. Keywords must be SBCS strings:

CLASS(name)
Specifies the privilege class name that you want to operate under. The name is a mixed string of 1 to 8 bytes, the first of which must be an SBCS alphabetic or shift-out (SO) character. SBCS characters in the rest of the string must be alphanumeric or /, $, #, or @. If you use a code page different from the default, use the X'5B' code point instead of the $. Tivoli Information Management for z/OS verifies that the class exists and that you are eligible to use the class.

If you omit the CLASS keyword, the default privilege class defined by your profile is used. If you do not define a class in your profile, if your user ID is not in any privilege class, or if it is in more than one class, you receive a message. If only one such privilege class is located, it becomes the current class that you run under. Alternatively, you can choose the CLASS option on the Primary Options Menu to select a class to run under.

If you do not run under a privilege class, you cannot create, update, or delete any Tivoli Information Management for z/OS records, nor can you display them (with the exception of SRC records).

If the class name you specify (or the invocation class in your user profile) is not valid, you can continue your session without having any authorities.

SRC(name)
Specifies the name of the stored response chain (SRC) to process. The name is a mixed string of 1 to 8 bytes, the first of which must be an SBCS alphabetic or shift-out (SO) character. SBCS characters in the rest of the string must be alphanumeric or /, $, #, or @. If you use a code page different from the default, use the X'5B' code point instead of the $. Tivoli Information Management for z/OS verifies that the SRC is valid for processing from the Primary Options Menu and for the default application setting. If the SRC is valid, the chain processes immediately. If not, the Primary Options Menu appears and you receive a message.

If you specify SRC without a name, a list of all valid SRCs appears on the screen. You can select one of these SRCs by number and immediately start the SRC. If you decide not to start an SRC, enter the END command to display the Primary Options Menu.

If you omit both the SRC and NOSRC keywords, the default invocation SRC (if one exists) in your profile is processed. This SRC must be valid from the primary options menu and for the default application setting. Otherwise, the Primary Options Menu appears and you receive a message.

This keyword and the NOSRC keyword are mutually exclusive.

NOSRC
Specifies to ignore the invocation SRC defined by your profile. This keyword and the SRC keyword are mutually exclusive.

IRC(immediate response chain)
Specifies the immediate response chain (IRC) to process from the Primary Options Menu. The responses can include mixed strings.

If you also specify the SRC keyword, or if your profile defines an invocation SRC and you do not specify the NOSRC keyword, the data in the IRC is treated as responses to diverted panels in the SRC. Any responses remaining in the chain after the SRC completes are processed as a normal IRC. The maximum allowable length of the IRC that follows an SRC is 512 characters.

TSP(name)
Specifies the name of the TSP panel, TSX EXEC, or alias to process after initializing Tivoli Information Management for z/OS. TSP names must be 8 SBCS alphanumeric characters, the first of which must be alphabetic. TSX or alias names can be 1 to 8 SBCS alphanumeric characters and must also start with an alphabetic character.

The specified TSP, TSX, or alias is processed before any IRCs or SRCs.

SESS(suffix)
Specifies the suffix for the session-parameters member that your user uses for Tivoli Information Management for z/OS. Specify the suffix as 1 or 2 SBCS alphanumeric characters. If you specify a single digit, it is right justified with a leading zero. If you specify a single alphabetic or national character, it is left justified with a trailing blank.

The suffix appends to BLGSES to form the member name.

If you omit the SESS keyword, the default suffix is 00.

To display Tivoli Information Management for z/OS panels in a graphical workstation window, use the GUI parameter to start Tivoli Information Management for z/OS in the ISPF GUI mode. For more information on using ISPF's GUI mode, refer to the *ISPF User's Guide* and the *ISPF Dialog Developer's Guide and Reference*.

GUI(LU:display|IP:display|,NOGUIDSP)
Identifies the workstation that the Tivoli Information Management for z/OS panels are to appear on.

LU:display
Specifies the workstation's APPC network name.

IP:display
Specifies the workstation's Internet Protocol (IP) address.

NOGUIDSP
Specifies that you want to make a connection to the workstation, but **do not** want ISPF to display in GUI mode.

Note: This parameter is only valid if you have specified an **LU** or **IP** parameter. In other words, you can have any one of the following situations:
- You specify **LU:display** *or* **IP:display** without the NOGUIDSP parameter
- You specify **LU:display, NOGUIDSP**
- You specify **IP:display, NOGUIDSP**

TITLE(title)
Specifies the text that appears on the title bar of the workstation window. The title can have a maximum length of 255 characters, and it is truncated at display time without notice to the user.

GUISCRW(screen-width)
Specifies a screen width different from that of the emulator or real device from which you enter the ISPSTART command. If you specify GUISCRW but do not specify GUISCRD, the depth of the screen is that of the emulator or real device.

Note: If GUISCRW is different from the emulator or real device and GUI initialization fails, ISPF does not initialize.

GUISCRD(screen-depth)
Specifies a screen depth different from that of the emulator or real device from which you enter the ISPSTART command. If you specify GUISCRD but do not specify GUISCRW, the width of the screen is that of the emulator or real device.

Note: If GUISCRD is different from the emulator or real device and GUI initialization fails, ISPF does not initialize.

When the ISPSTART command finishes processing, the first Tivoli Information Management for z/OS panel displayed is the product proprietary statement panel, BLG00002, unless the user profile is modified to bypass it (refer to the *Tivoli Information Management for z/OS User’s Guide* for information on modifying a user profile).

Note: Initialization IRCs, SRCs, and TSPs are processed after the product proprietary panel appears and the user returns from it.

Sample CLISTs to Start Tivoli Information Management for z/OS

Figure 17 on page 208 is a CLIST that passes the SESS, CLASS, SRC, NOSRC, IRC, and TSP parameters to the ISPSTART command. Data set names are for illustration only; replace them with the names appropriate to your organization. When you run Tivoli Information Management for z/OS, concatenate the SBLMSAMP data set name to the ISPPLIB DD statement in your ISPF procedures and CLISTs, or copy the appropriate panel style and related members from SBLMSAMP to a data set that is already concatenated to the ISPPLIB ddname. See “Selecting the Tivoli Information Management for z/OS Panel Style” on page 199.

```
PROC 0 SESS() CLASS() SRC() IRC() TSP() LIST NOSRC GUI()
/*********************************************************************/
/* CLIST TO START Tivoli Information Management for z/OS VERSION 7.1  */
/*********************************************************************/
IF &LIST = LIST THEN CONTROL LIST
CONTROL NOMSG

/* FREE ALL FILES
FREE FI(ISPLLIB,ISPMLIB,ISPPLIB,ISPSLIB,ISPTLIB,ISPPROF)
CONTROL MSG

/* ALLOCATE DATA SET FOR TSP TRACE
ALLOC DA(*) FI(BLGTRACE) SHR

/* ALLOCATE FLOW DATA SET
ALLOC DA(*) FI(BLGFLOW) SHR

/* ALLOCATE DATA SET FOR TSP PRINT
ALLOC DA(*) FI(SYSPRINT) SHR

/* ALLOCATE DATA SET WHERE YOUR TSX REXX EXECS RESIDE
ALLOC FI(BLGTSX) DA('blm.rexx.execs' +
                    'BLM.SBLMTSX') SHR REUSE
/* ALLOCATE MULTIPLE RFTDSS (OVERRIDES RFTDS IN SESSION-PARAMETERS MEMBER)
ALLOC FI(RFTDD) DA('blm.report.dataset1' +
                   'BLM.SBLMFMT') SHR REUSE

/* ALLOCATE LIBRARY FOR ISPF DIALOG MANAGEMENT PANELS
ALLOC FI(ISPPLIB) DA('ISPF.PANELS')  SHR REUSE

/* ALLOCATE LIBRARY FOR ISPF DIALOG MANAGEMENT MESSAGES
ALLOC FI(ISPMLIB) DA('ISPF.MSGS')    SHR REUSE

/* ALLOCATE LIBRARY FOR ISPF DIALOG MANAGEMENT SKELS
ALLOC FI(ISPSLIB) DA('ISPF.SKELS')   SHR REUSE

/* ALLOCATE LIBRARY FOR ISPF DIALOG MANAGEMENT TABLES
ALLOC FI(ISPTLIB) DA('ISPF.TABLES')  SHR REUSE

/* EACH USER NEEDS HIS OR HER OWN ISPF DIALOG MANAGEMENT PROFILE DATA SET
ALLOC FI(ISPPROF) DA('&SYSUID.ISPF.PROF') OLD REUSE

/* ALLOCATE LIBRARY FOR Tivoli Information Management for z/OS LOAD MODULES
ALLOC FI(ISPLLIB) DA('BLM.SBLMMOD1') SHR REUSE

/* START INFORMATION MANAGEMENT for z/OS VERSION 7.1
ISPSTART PGM(BLGINIT) PARM(SESS(&SESS) CLASS(&CLASS)      +
         SRC(&SRC) IRC(&IRC) TSP(&TSP) &NOSRC) GUI(&GUI)  +
         TITLE(INFORMATION MANAGEMENT for z/OS)
```

Figure 17. Sample CLIST to Start Tivoli Information Management for z/OS

You can change the load library you are using for Tivoli Information Management for z/OS without exiting ISPF. One reason you might want to do this is to enable users to switch between multiple BLX-SPs. You might have a production BLX-SP (BLX1) and a test BLX-SP (BLXT). For users to access the production system, you allocate the SBLMMOD1 product library. To enable users to also access the test system, you allocate a library containing a BLXSSIMN module with the BLXT subsystem *before* the SBLMMOD1 product library. (See “Defining a Subsystem to the BLX-SP” on page 145 for more information.)

You must also define a session member that specifies the CAS=BLXT keyword. (See "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318 for more information.)

Figure 18 is a sample CLIST that demonstrates starting Tivoli Information Management for z/OS from within ISPF.

```
PROC 0 SESS() CLASS() SRC() IRC() TSP() NOSRC
/*************************************************************/
/*                                                           */
/*   CLIST TO START INFORMATION MANAGEMENT for z/OS VERSION 7.1       */
/*                                                           */
/*************************************************************/
IF &SYSISPF=ACTIVE THEN +
  DO
    /* ALLOCATE DATA SET FOR TSP PRINT */
    ALLOC DA(*) FI(SYSPRINT) SHR REUSE

    /* ALLOCATE MULTIPLE RFTDSS (OVERRIDES RFTDS IN SESSION-PARAMETERS MEMBER)*
    ALLOC FI(RFTDD) DA('blm.report.dataset1' 'BLM.SBLMFMT') SHR REUSE
    /* ALLOCATE DATA SET WHERE YOUR TSX REXX EXECS RESIDE
    ALLOC FI(BLGTSX) DA('blm.rexx.execs' +
                        'BLM.SBLMTSX') SHR REUSE

    /* ALLOCATE THE LOADLIB THAT CONTAINS THE BLXSSINM MEMBER AND */
    /* THEN SBLMMOD1 FOR THE PRODUCT CODE */
    ALLOC FI(BLGLLIB) DA('blm.loadlib' 'BLM.SBLMMOD1') SHR REUSE
    ALLOC FI(BLGPLIB) DA('BLM.SBLMSAMP') SHR REUSE
    ALLOC FI(BLGTLIB) DA('BLM.SBLMSAMP') SHR REUSE
    ISPEXEC LIBDEF ISPLLIB
    ISPEXEC LIBDEF ISPLLIB LIBRARY ID(BLGLLIB)
    ISPEXEC LIBDEF ISPPLIB
    ISPEXEC LIBDEF ISPPLIB LIBRARY ID(BLGPLIB)
    ISPEXEC LIBDEF ISPTLIB
    ISPEXEC LIBDEF ISPTLIB LIBRARY ID(BLGTLIB)
    IF &MAXCC<=4 THEN +
      DO
        /* INVOKE INFORMATION MANAGEMENT for z/OS VERSION 7.1 */
        ISPEXEC SELECT PGM(BLGINIT) PARM(SESS(&SESS) CLASS(&CLASS) +
               SRC(&SRC) IRC(&IRC) TSP(&TSP) &NOSRC)
      END
    ISPEXEC LIBDEF ISPLLIB
    ISPEXEC LIBDEF ISPPLIB
    ISPEXEC LIBDEF ISPTLIB
    FREE F(BLGTLIB BLGPLIB BLGLLIB)  END
ELSE +
  DO
    WRITE 'YOU MUST ALREADY BE IN ISPF TO USE THIS CLIST BECAUSE'
    WRITE 'IT USES ISPF DIALOG SERVICES.'
  END
EXIT
```

Figure 18. Sample CLIST to Start Tivoli Information Management for z/OS From Within ISPF. Lowercase entries in this example indicate that you can substitute variable data. For example, you can include user data sets in the allocation for report data sets or TSXs, and allocate additional load libraries.

Note: The following notes pertain to both Figure 17 on page 208 and Figure 18.

- You do not have to use an SRC, IRC, TSP, *or* NOSRC to start Tivoli Information Management for z/OS. The sample given in Figure 17 includes these to show all of the possible options you can use to start Tivoli Information Management for z/OS.

- A session-parameters member limits you to one RFTDS. If you require additional RFTDSs, you must allocate them in some other way, for instance, in a CLIST. Figure 17 on page 208 shows how to concatenate two RFTDSs. For more information on concatenating multiple RFTDSs, see “Defining Report Format Table Data Sets” on page 200.
- To access an RFTDS that is allocated in a CLIST (such as the one in Figure 17 on page 208) in a batch job, you must include a statement similar to the following in your batch job JCL:

```
//RFTDD DD DSN='rftds.name',DISP=SHR
```

  - *rftds.name* is the data set name of the RFTDS you want to access.

Making an ISPF Selection

Instead of starting Tivoli Information Management for z/OS with the ISPSTART command, you can create a selection on an ISPF menu panel to start Tivoli Information Management for z/OS. To do this, add a line to the body of the menu panel indicating the selection number to use for Tivoli Information Management for z/OS, and add the following to the)PROC section of the panel:

```
n,'PGM(BLGINIT) PARM(parameters)'
```

- *n* is the selection number chosen for Tivoli Information Management for z/OS
- *parameters* defines whatever Tivoli Information Management for z/OS parameters you want to use as defaults, as specified under “ISPSTART Syntax” on page 204.

Notes:

1. Consider making additional panel changes to enable the user to specify Tivoli Information Management for z/OS parameters when making the Tivoli Information Management for z/OS selection.
2. You must ensure that the ISPF load module, ISPLINK, is available to Tivoli Information Management for z/OS in a program library accessible through the MVS LOAD macro instruction. ISPF’s sample Primary Options Menu, ISR@PRIM, must be available in ISPF’s panel data set.

   You can do this in one of two ways:

   - Preallocate the libraries before you start ISPF in a logon procedure or CLIST.
   - Use ISPF LIBDEF service for ISPLLIB.

Setting up a Default Profile

During initialization, Tivoli Information Management for z/OS identifies itself to ISPF as a new dialog application with the application name of BLG0. ISPF then tries to make available the application profile pool, table BLG0PROF, from the user’s profile library or, if necessary, from the table input library. Because Tivoli Information Management for z/OS does not provide a default BLG0PROF member for installation, the first time a user starts Tivoli Information Management for z/OS, ISPF does not find BLG0PROF. So, ISPF provides the default application profile pool, ISPPROF, which contains a set of default PF key values, and Tivoli Information Management for z/OS provides default values for the Tivoli Information Management for z/OS user profile. Refer to the *Tivoli Information Management for z/OS User’s Guide* for a discussion of the Tivoli Information Management for z/OS profile and its default values.

You can provide a different set of defaults for first-time Tivoli Information Management for z/OS users by completing the following steps:

1. Following installation, start Tivoli Information Management for z/OS.
2. If you use the standard panel style, follow these steps to change the default PF key values:
   a. Issue the ISPF KEYS command from the Tivoli Information Management for z/OS primary options menu. Refer to the *Tivoli Information Management for z/OS User's Guide* for a description of how to use ISPF commands from Tivoli Information Management for z/OS.
   b. Set the PF key values to those default values you want to be available to first-time Tivoli Information Management for z/OS users.
   c. Issue the END command.

   Note: If you use the enhanced panel style, you can change the PF key values by changing the enhanced panel style keylists. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for information on the enhanced panel style keylists.
3. Select **PROFILE** from the Tivoli Information Management for z/OS primary options menu, or issue the PROFILE command.
4. Set the default values you want to be active for first-time Tivoli Information Management for z/OS users.
5. Select **Permanent profile end** from the PROFILE SUMMARY panel.
6. Issue the QUIT command to leave Tivoli Information Management for z/OS.
7. Copy the member BLG0PROF from your user's profile library, ddname ISPPROF, to the table input library, ddname ISPTLIB.

With member BLG0PROF in the table input library, the defaults you established become available to subsequent first-time Tivoli Information Management for z/OS users. For a discussion of some of the fields you can set in this manner, and for techniques for making some of these default values unchangeable by other Tivoli Information Management for z/OS users, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

The BLG0CMDS member contains command table information for the enhanced panel style. If you use the enhanced panel style, you must copy this member (as well as the BLG0KEYS keylist member of SBLMSAMP) into a data set that is allocated to ddname ISPTLIB. This member is provided in the SBLMSAMP library for Tivoli Information Management for z/OS.

Starting Tivoli Information Management for z/OS in Batch Mode

You can use the JCL shown in Figure 19 on page 212 to start Tivoli Information Management for z/OS in batch mode. The ISPSTART command is used in the JCL to start Tivoli Information Management for z/OS. See "ISPSTART Syntax" on page 204 for ISPSTART command syntax and keywords.

```
//jobname   JOB  (account),'name      ',
// CLASS=a,MSGCLASS=a,NOTIFY=userid,MSGLEVEL=(1,1)
//***********************************************************
//*  START ISPF COMMAND IN THE BACKGROUND
//***********************************************************
//stepname  EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=1048K
//** * *ALLOCATE PROFILES, PANELS, MSGS, PROCS, AND LOG * ****
//ISPPROF   DD DISP=SHR,DSN=userid.ispf.ispprof
//ISPPLIB   DD DISP=SHR,DSN=isp.sisppxxx
//ISPTLIB   DD DISP=SHR,DSN=isp.sisptxxx
//ISPMLIB   DD DISP=SHR,DSN=isp.sispmxxx
//ISPSLIB   DD DISP=SHR,DSN=isp.sispslib
//** * *ALLOCATE SYSPRINT DATA SET IF DSNAME IS DESIRED
//SYSPRINT  DD SYSOUT=*
//** * *ALLOCATE BATCH DATA SETS IF DDNAME IS DEFAULT   * ****
//*     ON PROFILE
//** * *ALLOCATE DATA SETS WHERE YOUR TSX REXX EXECS RESIDE
//BLGTSX  DD DISP=SHR,DSN=blm.rexx.execs
//        DD DISP=SHR,DSN=blm.sblmtsx
//*  * *ALLOCATE THE RFT DATA SETS
//RFTDD   DD DISP=SHR,DSN=blm.report.dataset1
//        DD DISP=SHR,DSN=blm.sblmfmt
//** * *ALLOCATE FLOW DATA SET IF DESIRED
//BLGFLOW DD SYSOUT=*
//** * *ALLOCATE TRACE DATA SET FOR TSP IF DESIRED      * ****
//BLGTRACE  DD SYSOUT=*
//** * *ALLOCATE DIALOG PROGRAM AND CLIST LIBRARIES     * ****
//ISPLLIB   DD DISP=SHR,DSN=isp.sisplpa
//          DD DISP=SHR,DSN=isp.sispload
//          DD DISP=SHR,DSN=isp.sispsasc
//          DD DISP=SHR,DSN=blgsesxx.load
//          DD DISP=SHR,DSN=blm.sblmmod1
//SYSPROC   DD DISP=SHR,DSN=isp.sispclib
//** * *ALLOCATE TSO BACKGROUND OUTPUT AND INPUT        * ****
//*     DATA SETS
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
  PROFILE PREFIX(userid)
  ISPSTART PGM(BLGINIT) PARM('IRC(rsp) SRC(srcname) TSP(tsp)')
/*
```

Figure 19. Sample JCL to Start Tivoli Information Management for z/OS in Batch Mode

Lowercase entries in the previous example indicate that you must substitute variable data. Most of the entries follow z/OS MVS or TSO conventions; however, the following entries have special meaning:

userid The TSO logon ID. The user ID specified with the PREFIX keyword must be identified in a Tivoli Information Management for z/OS privilege class. The user must specify this item.

BLGTSX
Identifies the data sets containing TSXs.

RFTDD
The ddname of the data sets containing your report format tables. You may need to modify RFTDD to use the ddname of the report data set specified in your users's session-parameters member. In this example, because no session-parameters member is JCL, the default session-parameters member, BLGSES00, is used.

rsp The prompting sequence responses.

srcname
The name of an SRC.

tsp The name of a TSP or TSX.

The JOB statement must match the standards defined for your organization. A JOBCAT DD statement may be required if the data sets are not in the master catalog. In addition, you may need to define the Tivoli Information Management for z/OS load modules on a STEPLIB DD statement if they are not in the system link pack area.

Error Processing

If an error occurs during batch processing, Tivoli Information Management for z/OS writes the message panel image and any existing message chain (displayed, in an interactive session) to the data set allocated to the SYSPRINT DD statement. If there is no SYSPRINT DD statement, Tivoli Information Management for z/OS writes the messages to a dynamically allocated SYSOUT class A data set. In either case, after closing and freeing the print data set, batch processing ends.

Note: Refer to the *Tivoli Information Management for z/OS Data Reporting User's Guide* for additional information on batch processing.

Writing to SYSPRINT

When Tivoli Information Management for z/OS writes to SYSPRINT, it formats the data using DCB information specified by the user on either a SYSPRINT DD statement (that is, LRECL or BLKSIZE) or a TSO ALLOCATE command. If the user specifies an LRECL without a BLKSIZE, Tivoli Information Management for z/OS sets the BLKSIZE to:

```
(14 * LRECL) + 4
```

If the user does not specify a BLKSIZE or an LRECL, the LRECL is set to:

```
(length of output message) + 4
```

and the BLKSIZE is set to:

```
(14 * LRECL) + 4
```

If the user specifies a BLKSIZE without an LRECL, the LRECL is set to whichever of the following statements is smaller:

```
(length of output message) + 4
```

or

```
BLKSIZE - 4
```

In any case, the LRECL must be less than or equal to (BLKSIZE - 4). If not, an ABEND will occur when the data set is opened because the data attributes are inconsistent.

The RECFM of the SYSPRINT data set must be VBA, and the DSORG must be PS.

Starting Tivoli Information Management for z/OS in Batch Mode with the GUI Parameter

ISPF allows you to enable a workstation user to communicate in GUI mode directly with an application running in batch mode. You can take advantage of this function with Tivoli Information Management for z/OS by setting the ISPF GUI parameter on the ISPSTART command, so that jobs running in the background can communicate with a GUI workstation. Some of the advantages of starting Tivoli Information Management for z/OS this way include:

- You can run Tivoli Information Management for z/OS without requiring an MVS emulator session.
- You can use an interactive TSO session independent of the batch Tivoli Information Management for z/OS session.
- A single interactive TSO session can be used to submit the batch JCL to start multiple Tivoli Information Management for z/OS background jobs. Each job can connect to the same workstation or to a different workstation. Additionally, each Tivoli Information Management for z/OS session can be associated with the same userid that submitted the job, or a different userid.

You can use the sample JCL shown in Figure 20 on page 215 to start Tivoli Information Management for z/OS in batch mode with the GUI parameter. The ISPSTART command is used in the JCL to start Tivoli Information Management for z/OS.

```
//jobname  JOB your-job-card,
//  USER=userid,PASSWORD=password
//*-------------------------------------------------------------------*
//* THIS IS A SAMPLE JOBSTREAM FOR RUNNING THE PRODUCT UNDER ISPF IN
//* A BACKGROUND JOB.
//* THE GUI PARAMETER IS SPECIFIED FOR CONNECTION TO A WORKSTATION.
//* UPDATE THE FOLLOWING WITH UPPERCASE TEXT:
//*
//*     FROM                    TO
//* ===================================================================
//* your-job-card        - YOUR JOB CARD PARAMETERS. USE THE
//*                        USERID= AND PASSWORD= PARAMETERS IF THE
//*                        USERID OF THE SESSION IS DIFFERENT FROM
//*                        THE USERID SUBMITTING THE JOB.
//* isp.sispxxxx         - THE ACTUAL NAMES OF THE ISPF DATA SETS
//*                        USED AT YOUR INSTALLATION.
//* blm.sblmxxx          - THE ACTUAL NAMES OF THE INFORMATION MANAGEMENT for z/OS
//*                        DATA SETS USED AT YOUR INSTALLATION.
//* blgsesxx.load        - THE ACTUAL NAME OF THE DATA SET THAT CON-
//*                        TAINS THE SESSION-PARAMETERS MEMBERS.
//* rftdd                - THE DDNAME OF THE REPORT DATA SET SPECIFIED
//*                        IN THE SESSION PARAMETERS MEMBER.
//* userid.ispf.ispprof  - THE NAME OF THE USER'S ISPF TABLE DATA
//*                        SET. NO OTHER BATCH OR INTERACTIVE SESSION
//*                        MAY USE THE DATA SET WHILE THIS JOB IS
//*                        ACTIVE.
//* SESS(xx)             - THE SUFFIX FOR THE SESSION-PARAMETERS MEMBER
//*                        USED BY INFORMATION MANAGEMENT for z/OS
//* batch-screen-depth   - THE MAXIMUM NUMBER OF LINES TO BE DISPLAYED
//*                        ON THE WORKSTATION SCREEN.
//* workstation-address  - THE TCP/IP IP ADDRESS (PREFIXED WITH IP:) OR
//*                        THE APPC NETWORK NAME (PREFIXED WITH LU:) OF
//*                        THE WORKSTATION.
//* title-bar-text       - THE TEXT TO BE DISPLAYED IN THE TITLE BAR
//*                        ON THE WORKSTATION.
//*-------------------------------------------------------------
```

Figure 20. Sample JCL to Start Tivoli Information Management for z/OS in Batch Mode with the GUI Parameter (BLGISGUI) (Part 1 of 2)

```
//BLGISGUI PROC
//ISPF    EXEC PGM=IKJEFT01,DYNAMNBR=50,REGION=4096K,TIME=1440
//STEPLIB  DD  DISP=SHR,DSN=isp.sisplpa
//         DD  DISP=SHR,DSN=isp.sispload
//         DD  DISP=SHR,DSN=isp.sispsasc
//ISPLLIB  DD  DISP=SHR,DSN=isp.sisplpa
//         DD  DISP=SHR,DSN=isp.sispload
//         DD  DISP=SHR,DSN=isp.sispsasc
//         DD  DISP=SHR,DSN=blgsesxx.load
//         DD  DISP=SHR,DSN=blm.sblmmod1
//ISPMLIB  DD  DISP=SHR,DSN=isp.sispmxxx
//ISPPLIB  DD  DISP=SHR,DSN=isp.sisppxxx
//ISPPROF  DD  DISP=SHR,DSN=&TABLE
//ISPSLIB  DD  DISP=SHR,DSN=isp.sispslib
//ISPTABL  DD  DISP=SHR,DSN=&TABLE
//ISPTLIB  DD  DISP=SHR,DSN=&TABLE
//         DD  DISP=SHR,DSN=isp.sisptxxx
//SYSEXEC  DD  DISP=SHR,DSN=isp.sispexec
//SYSPROC  DD  DISP=SHR,DSN=isp.sispclib
//BLGTSX   DD  DISP=SHR,DSN=blm.sblmtsx
//rftdd    DD  DISP=SHR,DSN=blm.sblmfmt
//BLGFLOW  DD  SYSOUT=*
//BLGTRACE DD  SYSOUT=*
//ISPLOG   DD  SYSOUT=*,DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//ISPLIST  DD  SYSOUT=*,DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DDNAME=SYSIN
//        PEND
//BLGISGUI EXEC PROC=BLGISGUI,TABLE='userid.ispf.ispprof'
//SYSIN    DD  *
  ISPSTART PGM(BLGINIT) PARM(SESS(xx)) +
    BATSCRD(batch-screen-depth) +
    GUI(workstation-address) TITLE(title-bar-text)
/*
//
```

Figure 20. Sample JCL to Start Tivoli Information Management for z/OS in Batch Mode with the GUI Parameter (BLGISGUI) (Part 2 of 2)

Lowercase entries in the previous example indicate that you must substitute variable data. The following entries have special meaning:

your-job-card
: The job card parameters associated with your Tivoli Information Management for z/OS batch job. Use the **USER=** and **PASSWORD=** parameters if the userid of the session is different from the userid submitting the job.

isp.sispxxxx
: The actual names of the ISPF data sets used at your installation. All of the data sets needed by ISPF must be specified on the appropriate DD statements.

blm.sblmxxx
: The actual names of the Tivoli Information Management for z/OS data sets used at your installation.

blgsesxx.load
: The actual name of the data set that contains the session-parameters member.

rftdd The DD name of the report data set specified in the session-parameters member.

userid.ispf.ispprof
: The name of the user's ISPF table data set. The user's table data set should be

unique; that is, each user should have a separate tables data set for batch which includes his or her ISPF profile information. It cannot be referenced by another ISPF or Tivoli Information Management for z/OS session while the background job is active.

SESS(xx)
The suffix for the session-parameters member used by Tivoli Information Management for z/OS.

batch-screen-depth
The maximum number of lines to be displayed on the workstation screen.

workstation-address
The TCP/IP address or APPC network name of the workstation. For example GUI(IP:2.33.44.5) represents a TCP/IP address. GUI(LU:NETWRKID.PARTNELU) represents an APPC network name.

title-bar-text
The text to be displayed on the title bar of the workstation window. The title can have a maximum length of 255 characters, and it is truncated at display time without notice to the user.

To establish a GUI connection with a background job:

1. Start the ISPF workstation agent.
2. Submit the batch JCL to start ISPF and Tivoli Information Management for z/OS.

To end the session, enter the Tivoli Information Management for z/OS QUIT command on the workstation. The background job will complete and end normally.

Starting Tivoli Information Management for z/OS from NetView

If you use NetView and Information/Management Version 5 or later or Tivoli Service Desk for OS/390 1.2, you can access Tivoli Information Management for z/OS from NetView by using NetView's Terminal Access Facility (TAF). Using TAF in a NetView address space, you can log on to Tivoli Information Management for z/OS running in a TSO address space. Figure 21 on page 218 shows an example of using TAF to link to Tivoli Information Management for z/OS.

```
Initial Usage - 8:30 A.M. - Begin a
NetView Hardware Monitor Interface session.
  ===> LOGON 'NetView userid'
  ===> 'NetView password'
        ...
        ...NetView Activities...
        ...
  ===> CLISTNAME1 (for: BGNSESS,FLSCN,APPLID=TSO,SRCLU=xxxxx,...)
  ===> LOGON 'TSO userid'
  ===> 'TSO password'
  ===> start Tivoli Information Management for z/OS
        ...
        ...Tivoli Information Management for z/OS Activities
        ...
  ===>"Disconnect" key, then see:
  TAF Disconnect Options Panel
    1. ...
    2. Return to NetView control mode
    3. ...
    4. ...
  ===> 2
     ...
     ...NetView Activities..., then proceed to Subsequent Usage
     ...
Subsequent Usage - 10:30 A.M. - following coffee break
     ...
     ...NetView Hardware Monitor Interface Activities...
     ...
===> CLISTNAME2 (for: RTRNSESS APPLIED=TSO,...)
     ...
     ...Tivoli Information Management for z/OS activities continued from initial session,
        beginning with the panel on which "disconnect" key was pressed.
     ...
===> "Disconnect" key, then see:
  TAF Disconnect Options Panel
    1. ...
    2. Return to NetView control mode
    3. ...
    4. ...
===> 2
     ...
     ...NetView Hardware Monitor Interface Activities...
     ...
Final Usage - 3:30 P.M. - Final session before end of day.
     ...
     ...NetView Activities...
     ...
  ===> CLISTNAME2 (for: RTRNSESS APPLID=TSO,...)
  ===> QUIT from Tivoli Information Management for z/OS
  ===> LOGOFF TSO
       ...
       ...NetView Activities...
       ...
  ===> LOGOFF NetView
```

Figure 21. Example: TAF Link to Tivoli Information Management for z/OS

Tivoli Information Management for z/OS data can also be accessed through the NetView Bridge Adapter. Refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications* for details.

17

Loading Records Provided with Tivoli Information Management for z/OS

This chapter provides information about loading data model records and other types of records provided in the SBLMRCDS data set in the Tivoli Information Management for z/OS database.

Loading Data Model and Other Records

Records that support certain base functions and optional features are provided with Tivoli Information Management for z/OS. These records include data model records (data attribute, data view, or validation records) and other types of records such as reference records. Records are provided to support the following:

- Base functions such as change approval processing
- Universal time processing (described in "Implementing Universal Time Processing" on page 251)
- E-mail message notification (described in the *Tivoli Information Management for z/OS Program Administration Guide and Reference*)
- People record panels (described in the *Tivoli Information Management for z/OS Program Administration Guide and Reference*)
- The Tivoli Information Management for z/OS Desktop (described in the *Tivoli Information Management for z/OS Desktop User's Guide*)
- The use of Tivoli Decision Support with Tivoli Information Management for z/OS data (described in *Tivoli Decision Support: Using the Information Management Guide*)
- The interface to Tivoli Inventory (described in the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*)
- The Tivoli Service Desk Bridge (interface to Tivoli Problem Management, described in the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*)

Records are loaded by running a batch job which calls the BLHRCDSL TSX, or by running the BLHRCDSL TSX interactively from the Tivoli Information Management for z/OS command line. For convenience, it is recommended that you submit a batch job to avoid having to scroll through the messages displayed on your screen if you run the TSX interactively.

All records are unflattened and loaded into Tivoli Information Management for z/OS database 5. If desired, you can set up a separate database to hold data model records only, so that the records in your normal read/write database are not mixed up with the data model records that are provided with Tivoli Information Management for z/OS. Instructions on how to do this are provided in "Loading Data Model Records into a Separate Database" on page 221.

Records are categorized into lists based on their content and function. When loading the records, you can specify the name of a list of records you want to load. The record lists and the records themselves are provided in the SBLMRCDS program data set. The SBLMRCDS program data set is created when you install Tivoli Information Management for z/OS through SMP/E as described in the *Program Directory*. (For more information about the program data sets created during installation, see "Tivoli Information Management for z/OS Program Data Sets and Sample Members" on page 359.)

Note: Although you are not required to load all the records, you *must* load a base set of records that consists of data model records and other types of records. In addition, if you intend to use some of the optional features that require some of these shipped records, you must load the appropriate records for those features.

The records provided in the SBLMRCDS data set include the following types of records:

- Data model records (data view records, data attribute records, or validation records)

  These records can be loaded so that they are available to users in a read/write database (database 5) or read/only database (database 4, for example). At a minimum you *must* load the required base data model records (listed in the BLHL4BAS member) provided with Tivoli Information Management for z/OS, or certain functions in Tivoli Information Management for z/OS will not work properly. If these records are not loaded, users will receive error messages if they try to use the panels that depend on the existence of data model records for processing.

  For a summary of the lists provided for data model records, see Table 8 on page 221. If you want to learn more about data model records in general (such as how to create your own), refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

- Other types of records, such as reference records

  These other types of records *must* be present in the user read/write database (database 5). At a minimum, you must load the base records listed the BLHL5BAS member.

To load both the minimum set of data model records and other types of records, you can specify the BLHLRBAS list when submitting the batch job or running the TSX to load the records. The BLHLRBAS list includes both the BLHL4BAS and BLHL5BAS lists.

Note: The user running the BLHRCDSL TSX or the BLHRCDSJ batch job must be in a privilege class that has, at a minimum, DBADMIN and class authorities.

Table 8. Lists of Records Provided with Tivoli Information Management for z/OS. At a minimum, the BLHLRBAS list should be specified. To load all records, use BLHLRALL.

List	Contents	Description	Lists included by this list
BLHLRALL	All DMR and non-DMR	Master list. Use to load all records – those required for base functions and optional features	BLHLRBAS BLGLRTDS BLHLRBRG BLHLRINV BLMLRDSK
BLHL4ALL	All DMR	Use to load only DMR required for base functions and optional features	BLHL4BAS BLGLRTDS BLHLRBRG BLHLRINV BLMLRDSK
BLHL5ALL	All non-DMR	Use to load only non-DMR required for base functions and optional features	BLHL5BAS
BLHLRBAS	Base DMR and base non-DMR (required)	Use to load the minimum set of records for the base product	BLHL4BAS BLHL5BAS
BLHL4BAS	Base DMR (required)	Use to load the minimum set of data model records for the base product	None
BLHL5BAS	Base non-DMR (required)	Use to load the minimum set of non-DMR records for the base product. Contents include TIMEZONE record for universal time processing and e-mail message models.	None
BLGLRTDS	DMR	Required only if using Tivoli Decision Support or the ODBC driver.	None
BLHLRBRG	DMR	Required only if using Tivoli Service Desk Bridge	None
BLHLRINV	DMR	Required only if using Tivoli Inventory	None
BLMLRDSK	DMR	Required only if using Desktop	None

Loading Data Model Records into a Separate Database

Lists that contain only data model records can be loaded into a database that users can access in read/write or read/only mode. If you do not want the shipped data model records to reside in the read/write database (database 5) by default, you can make them read/only for users by setting up a separate database for data model records. To do this, follow these procedures:

1. Create the database for the data model records. Create the SDDS and SDIDS at a minimum, and also the SDLDS (recommended). Define the VSAM data sets. If necessary (that is, if you are *not* using sysplex support), update the VSAM resource definition member.

2. Create the session-parameters member that will be used to load the database with the data model records. At a minimum, use the BLGPARMS macro to define the session parameters, and the BLGCLUST macro to define the database.

   Use the BLGCLUST macro to add the data model record database. You must specify NAME=5 as the external name of the read/write database to be able to load the data model records.
3. Update your users' session-parameters members so they can access the database you just created.
   a. Use the BLGPARMS macro and specify, in the MODELDB keyword, the name of the database containing the data model records.

      When supplying parameter values for the MODELDB keyword, also specify the trigger character that your administrators will use when creating their own data attribute records; for example: MODELDB=(4,#). Do not specify the ampersand character (&). The ampersand character is reserved for use by the data attribute records provided with Tivoli Information Management for z/OS.
   b. Use the BLGCLUST macro to add the data model record database. You must specify a NAME of 4, 7, 8, or 9 as the external name of the read-only database. Use the same name that you specified on the MODELDB keyword in Step 3a on page 222.
4. Proceed with submitting a batch job or running the BLHRCDSL TSX to load the data model records. Be sure to use the session-parameters member that you specified in Step 2 on page 222.

If you choose to separate data model records into their own databases, you must ensure that the base non-data model records are loaded into your read/write database so that they are accessible to users in read/write mode. You must submit a batch job or run the BLHRCDSL TSX again with the BLHL5BAS list specified at a minimum. Use the session-parameters member that you modified in Step 3 on page 222.

Loading Records into Database Partitions

If your database is partitioned and you want to load the records into a specific logical database partition, you must run load the data model records using a privilege class that has that partition specified as the primary partition ID. The primary partition becomes the owning partition of the unflattened records. If your privilege class does not contain a primary partition name (primary partition ID), the unflattened records will not be assigned an owning partition name. The *Tivoli Information Management for z/OS Program Administration Guide and Reference* contains additional information about logically partitioned databases.

Submitting a Batch Job

You can unflatten and load the records into your database by running a batch job which calls the BLHRCDSL TSX. You can use the BLHRCDSJ sample JCL in the SBLMSAMP library to load the records into Tivoli Information Management for z/OS database 5. The BLHRCDSL TSX is described in more detail in "Running the BLHRCDSL TSX" on page 223. An extract of the sample JCL follows:

```
//BLGRCDSJ EXEC PROC=BLGRCDSJ,TABLE='userid.ispf.ispprof'
//SYSIN    DD  *
  ISPSTART PGM(BLGINIT) PARM(SESS(xx) CLASS(MASTER) +
    IRC(RUN BLHRCDSL blm.sblmrcds list,;QUIT))
/*
//
```

Copy the sample JCL and follow the instructions in the sample to include the necessary parameters. For instance, specify the appropriate session-parameters member, SESS(xx), to use to load the records. By specifying the correct session, you can separate the data model records that get loaded onto database 5 into a read/only database for users.

The parameters you can include in the RUN BLHRCDSL statement are defined in "Running the BLHRCDSL TSX".

If you need to reload records, you can replace the existing records by adding the REPLACE parameter to the RUN statement in the JCL.

Running the BLHRCDSL TSX

You can unflatten and load the records into a Tivoli Information Management for z/OS database by running the BLHRCDSL TSX. From the Tivoli Information Management for z/OS command line enter the following:

RUN BLHRCDSL *dsname list* [REPLACE]

dsname

dsname is the fully qualified name of the SBLMRCDS program data set (entered without single quotation marks) at your location.

dsname and *list* are required parameters and must be specified in that order.

list *list* is the name of the member in that data set that contains the list of records you want to load. Specify one of the following 8-character values for *list*:

BLHLRALL

List of all the record lists. Specify BLHLRALL to load all of the records into your database. If BLHLRALL is specified, the BLHLRBAS, BLGLRTDS, BLHLRBRG, BLHLRINV, and BLMLRDSK lists are included. The records loaded include all data model records provided with Tivoli Information Management for z/OS.

If you need to specify more than one list yet exclude other lists, you can run the BLHRCDSL TSX for every list you want to include, or create your own list to use with the BLHRCDSL TSX. Refer to the format of BLHLRALL for the expected format of the list of lists member. Your list and the actual records should be in the same partitioned data set.

BLHL4ALL

List of all the data model records needed to support base functions and other optional features. If this list is specified, the following lists are included: BLHL4BAS, BLGLRTDS, BLHLRBRG, BLHLRINV, and BLMLRDSK.

BLHL5ALL

List of all the non-data model records needed to support base functions and other optional features. If this list is specified, this list is included: BLHL5BAS.

BLHLRBAS

List of the data model and other types of records needed to support base functions. If this list is specified, the BLHL4BAS and BLHL5BAS lists are included. At a minimum, BLHLRBAS should be specified if you are not loading all records. This functions supported through this list include: people record panels in Tivoli Information Management for z/OS, universal time processing (the TIMEZONE reference record), and e-mail message notification (generic message text records). If you are integrating with Tivoli Problem Management, or if you will use the Tivoli Information Management for z/OS Desktop, BLHLRBAS is required.

BLHL4BAS
List of the base data model records required at a minimum.

BLHL5BAS
List of the base non-data model records required at a minimum.

BLGLRTDS
List of the data model records for integration with Tivoli Decision Support or for use with the Tivoli Information Management for z/OS ODBC driver.

BLHLRBRG
List of the data model records for use with the Tivoli Service Desk Bridge (integration with the Tivoli Problem Management application of Tivoli Service Desk).

BLHLRINV
List of the data model records for integration with Tivoli Inventory.

BLMLRDSK
List of the data model records for the Tivoli Information Management for z/OS Desktop application.

REPLACE
Specify REPLACE to replace existing records if you already loaded them. REPLACE overwrites duplicate records (records with the same RNIDs). This parameter is optional.

As the TSX runs, a message is displayed for each record stored successfully in the Tivoli Information Management for z/OS database.

Moving Data Model Records

If, for some reason, you need to uninstall the data model records you loaded into a database, you can use the BLHRCDSU TSX to flatten and place the records into a partitioned data set. To run the BLHRCDSU TSX through a batch job, you can copy the BLHRCDSJ JCL, change the TSX name on the RUN statement from BLHRCDSL to BLHRCDSU, and submit the job. Alternatively, you can simply run the BLHRCDSU TSX interactively.

If desired, you can use the BLHRCDSU TSX to move your own data model records from one database to another (such as from a test database to a production database). Use the partitioned data set used by this TSX to received flattened records as the input to the BLHRCDSL TSX (the *dsname*) to load the records to another database.

The syntax of the TSX is:

RUN BLHRCDSU *dsname list* [NOHISTORY] [NOTEXT] [DELETE]

Run BLHRCDSU dsname list {NOHISTORY} {NOTEXT} {DELETE}

dsname
dsname is the fully qualified name of the partitioned data set (entered without single quotation marks) where flattened records will be written.

dsname and *list* are required parameters and must be specified in that order.

list *list* is the name of the member containing the list of records to flatten. This parameter is required. If you are unloading the records provided by Tivoli Information Management for z/OS, you should use the appropriate list names (see Table 8 on page 221). If you are unloading records created by administrators or users at your location, you should specify your own list names.

NOHISTORY
This parameter specifies that the history data for the records should not be copied when the flattened records are written to the partitioned data set. The default is to copy the history data with the records. This parameter is optional.

NOTEXT
This parameter specifies that freeform text data should not be copied when the flattened records are written to the partitioned data set. The default is to copy the freeform text data with the records. This parameter is optional.

DELETE
Indicates that the original records should be deleted from the database when the flattened records are written to the partitioned data set (PDS). By default (the DELETE parameter is omitted), original records are not deleted. This parameter is optional.

If you are using this TSX to flatten your own records into a list name of your choice, you can use one of the lists provided as an example of how to construct your own list.

This is an extract from a list provided by Tivoli Information Management for z/OS:

```
/*********************************************************************/
BLG$APST  BLG&APST  Approval Status
BLG$APVR  BLG&APVR  Approver
BLG$DFMT  BLG&DFMT  Profile External Date Format
```

The records are originally provided with a dollar sign trigger character ($) in the SBLMRCDS data set, but are actually loaded into your database with the ampersand trigger character, which is reserved.

This is an example of a list you can construct for your own records:

```
/*********************************************************************/
ABC#APST  ABC#APST  Approval Status
ABC#APVR  ABC#APVR  Approver
ABC#DFMT  APC#DFMT  Profile External Date Format
```

The first column represents the current name of the record. The second column represents the desired name of the record when it is copied into the target partitioned data set. (Your records will most likely have the same label in both columns.) You can also provide comments (optional) to describe the contents of the records.

A

Enabling Alternative Date and Time-of-Day Formats

This appendix describes the date and time-of-day formats available for use in Tivoli Information Management for z/OS, and explains how to enable Tivoli Information Management for z/OS to use alternative date and time-of-day formats. It also describes how to implement the optional universal time support in Tivoli Information Management for z/OS.

Date formats, time-of-day formats, and time zone are specified at the installation level through the use of BLGPARMS macro keywords:

- The DATECNV keyword of the BLGPARMS macro enables you to specify the name of the date conversion routine to be used by Tivoli Information Management for z/OS. This keyword is optional, and the default value for DATECNV is BLGCDATS, which is the name of the date conversion routine provided with Tivoli Information Management for z/OS. You can specify the name of a user-written date conversion routine if necessary. Typically, BLGCDATS should meet your date processing needs.
- The DATEFMT keyword of the BLGPARMS macro enables you to specify the external date format to be used at your installation by default for a session member. Tivoli Information Management for z/OS supports 22 external date formats. You can specify any of these formats as defaults for the session (although certain formats are recommended as described later in this appendix). Users can override the external date format specified on the DATEFMT keyword by selecting a different external date format in the user profile. They can also override the time zone for the session by selecting another time zone in the profile. An illustration of this option in the user profile is shown in "Specifying Preferences in the User Profile" on page 230. API applications can also override the external date format through the DATE_FORMAT parameter data block (PDB), and the time zone through the TIME_ZONE PDB.

  Additionally, if none of the 22 external date formats provided by Tivoli Information Management for z/OS is suitable for your needs, you can, as an alternative, specify your own date conversion routine for use by the session member. If you specify your own date conversion routine, all users *must* use the same external date format. This appendix describes the supported date formats and explains how to implement your own date conversion routine if a user exit is necessary.
- The TIMEZONE keyword of the BLGPARMS macro enables you to implement the optional universal time support feature of Tivoli Information Management for z/OS. If TIMEZONE is not specified, dates and times will be collected and displayed in the local time of the user who entered them and no universal time processing will be done.

Additionally, you have the option to define relationships between date fields and time fields. Only date and time fields that have been ″related″ can use universal time processing. Even if TIMEZONE is specified, unrelated date and time fields won't use universal time processing.

For more information about the BLGPARMS macro keywords, see "BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics" on page 318.

This appendix describes how to relate date and time fields and describes how searches performed by users are affected if universal time processing is enabled. Universal time processing is supported in the Management application of Tivoli Information Management for z/OS. It is not supported in the Integration Facility.

As a preview, it is important to be aware of the following:

- If you are *not* enabling universal time processing, then you should remove field **4. User's time zone** from profile panel BLG0P700. This field is intended to be used only if universal time processing is implemented at your installation. If this field is left on the panel with universal time disabled, then a user trying to set this field will encounter one of the following:
  - An error message because there is no TIMEZONE record, which is used as a validation record.
  - Confusion because entering a different time zone in this field has no effect (because the value is ignored except for universal time processing).
- If you *are* enabling universal time processing, you must do *all* of the following:
  - Add the TIMEZONE (and possibly OTIMEZON) keyword to the session member to enable universal time processing and to define the default time zone.
  - Load the TIMEZONE record provided with the product into database 5 and customize the record if desired. (For example, you may want to rename time zone symbols.)
  - Create a DATETIME record to define which of your date fields are paired with certain time fields. Converting a date or time to universal time requires both the date and time, because the code must know which fields are related.
  - Understand how universal time processing works and educate users about its use.

A detailed discussion of the above actions is found in "Implementing Universal Time Processing" on page 251.

Note: As this appendix explains, Tivoli Information Management for z/OS 7.1 has implemented a number of date and time-of-day enhancements. If you are currently using two or more external date formats, it is recommended that you run BLGUT17 before or during the installation process in order to standardize the date formats your organization uses. Information on running BLGUT17 can be found in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

Enabling Date Formats

Tivoli Information Management for z/OS users work with dates in two formats:

- External format — The format that users type and see when working with records on data entry or display panels, or in structured searches. The Tivoli Information Management for z/OS panels, as provided, display dates in a 10–character date format (for example, MM/DD/YYYY). An external date format is specified at installation through the BLGPARMS DATEFMT keyword. Users can override this format by selecting another supported format in the user profile.

  When a date is entered by a user, it is automatically converted from external date format to internal date format. The external date format is not stored as part of the record in the SDDS. It is collected while the record is in use, but removed when the record is filed.

- Internal format — The format used to perform freeform searches. The internal format is YYYY/MM/DD.

  The internal date format is stored in the SDIDS. It is also stored in the record in the SDDS. If universal time support is used, the internal date format is also stored in universal time in the SDDS.

How Data is Converted and Stored

Data values are entered by users and displayed in external format. When the data is entered by a user, it is automatically converted from that user's external format (for example, 05/27/2001) to internal format (2001/05/27). The dates and times are stored in the SDDS in *internal format*, regardless of any user preference. When records are read from the database, the date and time fields are automatically converted from internal format to the external format of the user reading the records (for example, to 27-05-2001) for presentation to the user. (Prior to Tivoli Information Management for z/OS Version 7.1, dates and times were stored in the SDDS in a single external format and all users saw the data in that format.)

Note: The change in how data is stored in the SDDS should not be important to you if you are currently using a previous version of Tivoli Information Management for z/OS and continue to use the same external date and time formats.

Whether or not you choose to use universal time processing (described in "Implementing Universal Time Processing" on page 251), dates and times are stored in the SDDS in internal format, and users can select the external date formats of their choice by updating their user profile. For a discussion of migration considerations, see "Tivoli Information Management for z/OS Version 7.1 Changes" on page 103 and "Migration Considerations" on page 234.

History data for dates and times is stored in internal format and converted to the user's external format upon display.

Selecting Date Formats for a New Installation

If you are a new customer to Tivoli Information Management for z/OS and are performing installation for the first time, you will most likely want to specify the following:

- `DATECNV=BLGCDATS`
- `DATEFMT=`*`extfmt`*

where *extfmt* is an external date format from the list of formats supported automatically by Tivoli Information Management for z/OS (see "External Date Format" on page 231 for a list). A format with a 4-digit year (such as MM/DD/YYYY) is recommended but not required.

Note: If any value other than BLGCDATS is specified for DATECNV, then users (interactive and API) cannot choose their own date format.

Specifying Preferences in the User Profile

Users can override the default external date format specified in the BLGPARMS DATEFMT keyword by selecting the **User and database defaults** option on BLG0PU00, the Profile Summary panel. On the resulting panel that displays (BLG0P700), users can specify an external date format. Additionally, they can optionally specify a time zone. For time zone processing to take place correctly, however, universal time processing must be enabled through the TIMEZONE keyword as described in “Implementing Universal Time Processing” on page 251. An example of the BLG0P700 panel follows:

```
BLG0P700                 USER AND DATABASE DEFAULTS        USER ID: SMITH

Enter user and database data; cursor placement or input line entry allowed.

            1. User's name............ SMITHSON_______
            2. User's department...... BBC________
            3. User's telephone....... 555-7878_____
            4. User's time zone....... ET______
            5. User's date format..... MM/DD/YYYY


            8. Database............<5>
            9. Logical files.......... __________________________

     When you finish, type END to save or CANCEL to discard any changes.









===>end
```

To restrict the list of acceptable external date formats in the user profile, the Tivoli Information Management for z/OS administrator can update data attribute record BLG&DFMT to remove any of the external date formats the users should not select. The BLG&DFMT data attribute record contains a list of the 22 external date formats supported by Tivoli Information Management for z/OS. It must exist in your database for users to set their own date format.

Using HELP STATUS to Show Date Formats

Users can enter the HELP STATUS command to see what specific external date formats and time zone are in use in their interactive session. For example, the output of HELP STATUS may show the following:

```
BLG1THLP               STATUS AND AVAILABLE COMMANDS              LINE 1 OF 35

VERSION = VxRxMx  RECORD ID = NONE  CLASS = MASTER  SUSPENSION LEVEL = 0
MODES = INQUIRY   SEQUENTIAL RECORD
DATABASE = 5   LOGICAL FILES = *
SESSION MEMBER = DD   WINDOW NAME = BLGISPFD   WINDOW LEVEL = JOYxxx
PRIMARY PARTITION = NONE   GLOBAL PARTITION = NONE
TIME ZONE FORMATS:    DEFAULT = ET      OLD = NONE
EXTERNAL DATE FORMATS:    DEFAULT = MM/DD/YYYY    OLD = MM/DD/YY
TRIGGER CHARACTERS = & #   DATA MODEL DATABASE = 5

COMMAND NAME      COMMAND DESCRIPTION

ARGUMENT        ADD/DELETE/MODIFY FREEFORM ARGUMENTS
BACK            BACK UP TO LAST DISPLAYED OR PROCESSED PANEL
CANCEL          TERMINATE THE CURRENT DIALOG WITHOUT SAVE
CHANGE          CHANGE SEARCH ARGUMENT OR AN SRC RECORD
COPY            COPY RECORD FROM THE CURRENT DATABASE
DELETE          DELETE RECORD FROM THE CURRENT DATABASE
DISPLAY         DISPLAY RECORD FROM THE CURRENT DATABASE
DROP            DROP USER-DEFINED LINE COMMAND
END             TERMINATE THE CURRENT DIALOG WITH SAVE
===>
```

As shown in the example, the external date formats are displayed in "default" and "old" date format. Prior to Tivoli Information Management for z/OS Version 7.1, you could specify up to two external date formats. If you needed users to enter *new* records using a particular format, that would be the *default format*. However, if you needed to retain support for *older* records that used some other date format, you would also specify an *old format* for those records. Starting with Tivoli Information Management for z/OS Version 7.1, the meanings of the default and old formats are changed somewhat because users can now enter whatever supported format they choose. Currently, the default external date format refers to the installation default external date format, or the date selection made in the user profile, which overrides the default external date format. The old format shows the value of ODATEFMT.

External Date Format

The Tivoli Information Management for z/OS panels as provided can accept and show dates in 10-character fields. You can, of course, change the panels to change the length of the fields if necessary. However, the 10-character date field enables all of the 22 supported external date formats to be entered by users. The external date formats (both default and old) supported by Tivoli Information Management for z/OS are:

```
MM/DD/YY      DD/MM/YY      YY/MM/DD      DDMMMYY
MM/DD/YYYY    DD/MM/YYYY    YYYY/MM/DD    DDMMMYYYY
MM-DD-YY      DD-MM-YY      YY-MM-DD      YYDDD
MM-DD-YYYY    DD-MM-YYYY    YYYY-MM-DD    YYYYDDD
MM.DD.YY      DD.MM.YY      YY.MM.DD
MM.DD.YYYY    DD.MM.YYYY    YYYY.MM.DD
```

A description of these formats follows:

MM The number of the month
MMM The abbreviation of the month (for example, JAN, FEB)
DD The number of the day in the month
DDD The Julian date
YY The last two digits of the year

A. Alternative Date and Time-of-Day Formats

YYYY All four digits of the year

Default Date Format

The external default date format:

- Cannot include commas because commas separate responses in response chains.
- Can include special characters (or blanks) if you modify the panels to accept them.

  Note: You may not want to use blanks because blanks act as word delimiters.
- Can be up to 64 characters long, including the length of the prefix.
- Must be specified on the DATEFMT keyword of the BLGPARMS macro (if you use the BLGCDATS date conversion routine) before you can reassemble session parameters members. If DATECNV is specified as any other value than BLGCDATS, then DATEFMT must be omitted.

When a date is entered by a user, it is automatically converted from external date format to internal date format by Tivoli Information Management for z/OS through a conversion program. The dates are stored in the SDDS in internal format, regardless of any user preference. When records are read from the database, the date fields are automatically converted from internal to external date format for presentation to the user.

All date processing is available for the supported external date formats, including range searching and date/time math.

Understanding Preferences

When deciding on an external date format for your installation, keep in mind that:

- MM/DD/YY or MM/DD/YYYY is the preferred format for the United States.
- DD/MM/YY or DD/MM/YYYY is the preferred format for Canada (English), Europe, and Latin America.
- YYYY-MM-DD is the preferred format for Canada (French).
- YYDDD and YYYYDDD are Julian date formats.

The external format you select and specify in the BLGPARMS macro is used by all users; however, users can override the format specified by selecting another external date format in their user profile. Validation of the data entered by users is performed using the user profile selection. If the user has not selected a preferred external date format in the user profile, validation of the data is done using the external date format specified at installation through the BLGPARMS macro.

Note: The assisted-entry panels provided with Tivoli Information Management for z/OS use the validation pattern IIV63 (effectively, no validation) to validate the data. This validation pattern enables users to enter a variety of supported external date formats. Validation of the data is performed by the date conversion routine.

Old Date Format

The old date format specifies the external date format used for records that were created with versions of the product prior to Version 7.1. It is specified on the ODATEFMT keyword of the BLGPARMS macro.

Processing Date Data with TSPs or TSXs

If you write TSPs or TSXs that process date data, you should write them so that they use the 10-character internal date format (YYYY/MM/DD) for processing date data. Use the BLGIDATE and BLGEDATE user exits to convert the internal format date from or to the user's local date format. The BLGIDATE user exit converts a date from the user's local date format to internal format. The BLGEDATE user exit converts a date from internal format to the user's local date format. For more information about these user exits, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

Specifying an External Date Format through the API

The Tivoli Information Management for z/OS program interface data tables that are provided to support API transactions with the Tivoli Information Management for z/OS database accept any of the external date formats supported by Tivoli Information Management for z/OS.

You can set an option (DATE_FORMAT) in your API to send or receive data in a particular format that may be needed by your API application. For example, if you want to ensure that your API programs will be unaffected by date changes in the Tivoli Information Management for z/OS database, you can specify that dates be received in a particular format by defining an application-specified format. For information on setting the API DATE_FORMAT option to have all dates converted to or from a particular format for the API application, regardless of the date format used by the Tivoli Information Management for z/OS host database, refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.

If the API application does not use the DATE_FORMAT option, Tivoli Information Management for z/OS uses the external date format specified in the BLGPARMS DATEFMT keyword.

Field Length on Panels

The length of date fields in the Tivoli Information Management for z/OS panel data set (SBLMPNLS) is 10 characters. The length supports the entry of 10-character external date formats (for example, MM/DD/YYYY) or the shorter external date formats (for example, YY-MM-DD or YYDDD). Program interface data tables (PIDTs) shipped with Tivoli Information Management for z/OS also support dates in a 10-character date format by default. If you want to use the base product panels with 8-character date fields instead, you can, at installation, convert the date fields on panels from 10- to 8-character format by running a special job provided for this purpose (see member BLGDATE8 in the SBLMSAMP sample library).

Note: If you change the fields to a shorter length, be aware that you will limit the flexibility users have when typing dates directly into fields.

The BLGUT6M migration utility can help you to change the field lengths and validation patterns on your panels if you need to change them. See "Modifying Date Formats on Panels" on page 236 for more information.

Selecting a Tivoli Information Management for z/OS External Date Format

To select one of the external date formats provided by Tivoli Information Management for z/OS:

1. Specify the appropriate format as the second parameter on the DATEFMT keyword of the BLGPARMS macro in the session-parameters member. For example:

```
DATEFMT=DD/MM/YYYY
```

indicates that the default external date format is DD/MM/YYYY.

2. Reassemble and relink the session parameters after the DATEFMT keyword is changed. To avoid assembly failures, be sure to do the following:
   - Type a supported date format on the DATEFMT keyword.
   - Include an external date format if you specify the BLGCDATS routine, or accept the default external date format.
   - Do not type a routine name other than BLGCDATS on DATECNV and specify an optional Tivoli Information Management for z/OS external date format on DATEFMT. You cannot mix a user-defined routine with a Tivoli Information Management for z/OS external date format.

     Note: If any value other than BLGCDATS is specified for DATECNV, then users (interactive and API) cannot choose their own date format.

Migrating Existing Panels

The date fields on product panels are 10 characters long. The recommended external date format length is 10 characters. If your existing panels are not currently using a 10-character external date format, you should do the following:

- If you are currently using 8-character date fields on data entry or table panels and wish to continue using 8-character date fields, ensure that the longer external date formats are not available to your users through a user profile selection. To restrict the list of acceptable external date formats in the user profile, update data attribute record BLG&DFMT to remove any of the external date formats the users should not select.

  It is not recommended that you delete the data attribute record from your database. Doing so will prevent users from being able to select a preferred date format.
- Consider migrating your panels to use 10-character date fields. To migrate panels, use the BLGUT6M migration utility.

  Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using the BLGUT6M utility.

Specifying a Unique Date Format

If you do not plan to use any of the external date formats provided by Tivoli Information Management for z/OS, and yet need to have your own variation, you can supply a user exit routine as described in “Implementing an External Date Format through User Exits” on page 242 and specify your own date conversion routine during the installation process.

Migration Considerations

The following activities may need to be performed by the PMF administrator or Tivoli Information Management for z/OS programmer:

- ODATEFMT keyword – Dates and times in existing records are automatically converted from the external format in which they were collected into the internal format when the record is read from the database. If you need to process records that were created or updated with a version prior to Tivoli Information Management for z/OS Version 7.1, you may need to specify the BLGPARMS macro ODATEFMT keyword if either of these two conditions is true:

- The length (in characters) of the external date format you previously used is different from the length of the value you will specify as the default external date format on the DATEFMT keyword. For example, if you previously used a 2-digit year external date format (DD/MM/YY) and had the following session parameters specified in a previous version:

  ```
  DATECNV=(BLGCDATS,YYYY/MM/DD,DD/MM/YY)
  ```

  And you intend to now use a 4-digit year external date format (MM/DD/YYYY), code the following:

  ```
  DATECNV=BLGCDATS,
  DATEFMT=DD/MM/YYYY,
  ODATEFMT=DD/MM/YY,
  ```

  The external date format length of DD/MM/YY is shorter than the new length of DD/MM/YYYY. You must include the ODATEFMT keyword parameter in your session parameters.

- You previously specified both a ″primary″ and a ″secondary″ external date format so that users could enter dates on panels in two different date formats. If your database contains records that were created this way, you should include the ODATEFMT keyword and specify the ″secondary″ external date format as the value for ODATEFMT. For example, if you had this previously:

  ```
  DATECNV=(BLGCDATS,YYYY/MM/DD,MM/DD/YYYY,MM/DD/YY)
                               |          |
                               |primary   |secondary
                                ext date   ext date
  ```

  You should now code this:

  ```
  DATECNV=BLGCDATS,
  DATEFMT=MM/DD/YYYY,
  ODATEFMT=MM/DD/YY,
  ```

  In this case, the dates associated with some of your older records do not match the default external date format on the DATEFMT keyword. Because they are different, you must specify the ODATEFMT keyword and supply the external date format used by the older records.

When processing dates in records that were produced prior to Version 7.1, Tivoli Information Management for z/OS checks the length of the date value in the record. If the length of the date value matches the length of the external date value specified for DATEFMT, the date is assumed to be in the format specified by DATEFMT. If the length is the same as that specified for the ODATEFMT value, the date is assumed to be in the format specified by ODATEFMT. Otherwise, the date is not converted and an error message is displayed.

When Tivoli Information Management for z/OS converts the dates in existing older records, it uses the date formats specified as BLGPARMS keyword parameters. It does not use the date format selection in the user profile to perform this internal conversion. However, when displaying the converted fields in older records to users, Tivoli Information Management for z/OS uses the user profile date format selection to display the record.

- TSXs or TSPs – If you have TSXs or TSPs that enter dates, you should ensure that they can support the potentially many external date formats various users may use. Write or modify the TSX or TSP so that it uses 10-character internal date format (YYYY/MM/DD) for doing any processing or date calculations. Use the BLGIDATE and BLGEDATE user exits to convert the date in internal format from or to the current user's external date format when necessary. For a description of these user exits, refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference*.

- API applications – Prior to Tivoli Information Management for z/OS Version 7.1, API applications may have passed dates in a ″secondary″ external date format. Because any supported external date format can be used, the concept of ″secondary″ external date format no longer exists. API applications that pass dates in what had been ″secondary″ external date format must be changed to use the DATE_FORMAT PDB to specify the format being used.

Modifying Date Formats on Panels

If necessary, you can shorten the date field on your data entry and table panels, although this is not recommended. Shorter fields will limit the date format options users can select through the user profile. If you decide to shorten the date fields regardless, you should remove the date formats that your fields will not support due to the shorter length. To do this, modify data attribute record BLG&DFMT and remove the unsupported formats.

If you decide to switch from 10 characters to a shorter length, run the BLGUT6M utility to load the panels in the SBLMPNLS data set and convert the date fields to the shorter length. You can use the JCL provided in member BLGDATE8 in the SBLMSAMP sample library to do the conversion. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for instructions on using the utility. Also, after installation, if you need to reassemble session-parameters members, be sure to specify the correct default external date format on the DATEFMT keyword of the BLGPARMS macro. You should also change both the visible and control information for the panels.

To identify panels containing dates, you can run a panel cross-reference report. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for instructions on running PMF reports.

Alternate Date Format for the Integration Facility

For the Integration Facility, the assisted-entry panels used in the problem create and problem inquiry dialogs are duplicated. Panels use a 10-character date field. Use BLGCDATS, the default date conversion user exit routine. You must also specify the external date format on the DATEFMT keyword of the BLGPARMS macro. Users can override the default external date format by making some other date selection in the user profile.

Note: If you use the Integration Facility, you should not implement universal time processing. Date processing results could be unpredictable if universal time processing is implemented. You (or your Tivoli Information Management for z/OS program administrator) should also consider using PMF to remove the **User's time zone** field on BLG0P700, the User and database defaults panel. This field is intended to be used only if universal time processing is implemented at your installation. To avoid frustrating users by displaying a field that will not work in the Integration Facility environment, you can update the panel to remove the field so that users cannot use it. For instructions on updating panels, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

Four-Digit Year Considerations

When selecting date formats for your installation, consider whether users want to see dates expressed with 4-digit years. The date fields on the panels shipped with Tivoli Information Management for z/OS are 10 characters long to accommodate entry of an external date format that uses a 4-digit year. If you have customized panels that do not use 10-character date fields, you can use the BLGUT6M migration utility to migrate your panels to use a new field length.

It is not necessary to change the external format to use a 4-digit year unless you want to work with dates before 1950 or after 2049. Tivoli Information Management for z/OS adds '19' to the front of years 50 through 99, and adds '20' to the front of years 00 through 49.

Although Tivoli Information Management for z/OS will work with a 2-digit year, there may be reasons you will still want to convert to a 4-digit year, such as organizational standards or interfaces or consistency with other products.

Changing to an Internal Format with a 4-Digit Year

If your database currently uses the YY/MM/DD internal date format, you must migrate it to the YYYY/MM/DD internal date format.

Follow these procedures to change your internal date format to YYYY/MM/DD:

- Modify and reassemble session members using Tivoli Information Management for z/OS Version 7.1.
- Modify any TSPs, RFTs, SRCs, API programs, or user exits that do freeform searches with dates to use a 4-digit year rather than a 2-digit year.
- Inform any users who do freeform searches with dates, or users who look at the glossary, that 4-digit years will be used.
- Delete and redefine the SDIDS. Run the BLGUT1 utility, pointing to one of the reassembled session-parameters members to reload the SDIDS with the new date format.

Modifying Customized Panels to Use 10-Character Date Fields

If you are like most users, you have already customized the product panels for your own use. If you want to expand the date fields on those panels and change the associated validation patterns, you can use PMF and/or the BLGUT6M migration utility to make the changes. Because you can make many changes efficiently and save some work in PMF, you should consider using the BLGUT6M utility to make these changes. You will need to update the date fields on each data entry panel, assisted-entry panel, or table panel. Details on how to use the utility are provided in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*, but before you run the utility, you should first gather the information you will need to be able to run the utility.

There are various ways you can extract the information you need to run the BLGUT6M migration utility. This section, intended for the Tivoli Information Management for z/OS program administrator, describes one way you can extract this information. Be aware that other methods exist, and use the approach that works best for you.

The following procedure involves identifying data needed to change panels, offloading your customized panel data set, loading the base product panels, and migrating your offloaded panel data set to the new date length. Follow this procedure to perform these tasks:

1. Identify the data entry panels, assisted-entry panels, s-word index and p-word index values associated with the date fields. To do this, you can create test records. In the test records, supply data for all date fields and required fields. File the records, display them, and issue the VIEW INTERNALS command to obtain the data entry and assisted-entry panel names and s-word index values associated with dates. (You can issue a PRINT ALL command from the View Internals Data panel to put this information in a data set, if desired.) The internal data shows the panel name and s-word value, as shown in the following sample extract:

```
 BLG1TVID                      VIEW INTERNAL DATA                      LINE

 PANEL     PANEL   REL  COG-   FLAGS     SWORD   STRUCTURED    PREFIX WORD OR
 NAME     TYP/RSP  LEV  NIZE  F  M  D    INDEX      WORD       VISIBLE PHRASE

 TME6RELF   A/ 2   00   B/U   0D/00/00   S0000   BC RELIEF     RELEF/N

 TME0B100   D/ 3   00   N/    41/00/00   S8002
*TME6DABG   A/ 2   00   B/U   0D/04/00   S0000   BC APARDATE   DATR/02/07/97

 TME0B100   D/ 4   00   N/    41/00/00   S803E

 TME0B200   D/ 2   00   N/    41/00/08   S8008
*TME6DDBG   A/ 2   00   B/U   0D/04/00   S0000   BC DSGNBEG    DATDB/02/07/97

*BLGCDTAD   C/ 2   00   B/U   0D/02/00   S802A   BC DSGNDEAD   DATE/02/17/97

 TME0B200   D/ E   00   N/    01/00/04   S000A
*TME1A111   C/ 2   00   B/U   0D/00/00   S0C35   BC IM00SDM00  DATM/05/09/97

 TME1A111   C/ 2   00   B/U   0D/00/00   S0C62   BC IM00STM00  TIMM/09:59
```

   In this example, asterisks were added to show the panels associated with date fields. Panel type ’A’ refers to an assisted-entry panel; panel type ’D’ is a data entry panel, and panel type ’C’ is a control panel. If you are unfamiliar with how to read or interpret internal data, refer to the “Interpreting Internal Data” section in the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for instructions.

2. Use PMF to obtain the prefix index (p-word index) values (perform a Panel Update for each assisted-entry panel, select Externals, type CONTROL and then enter 4 to view the Prefix and Content Validation dialog panel).

   When identifying the prefix index, ensure that you select the one associated with an actual validation pattern (for example, NN</>NN</>NN) and not the one associated with a predefined validation pattern, like =DATE.

   You should now have the following elements identified and recorded: panel name, s-word index, and current prefix index. For example:

```
Data Entry   S-word  Assisted-Entry  Current Prefix   New Prefix
Panel        Index       Panel           Index           Index
-----------  ------  --------------  --------------   ----------
TME0B100     S8002     TME6DABG          P00E8
etc.
```

3. Make a list of the prefix index values that begin with 0 to 7. Identify the new p-word index values supplied by Tivoli that correspond to the p-words used by your panels. To do this, you can use the list provided in the description of the BLGUT6M JCL sample in the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

   For example:

```
Data Entry   S-word  Assisted-Entry  Current Prefix  New Prefix
Panel        Index       Panel           Index          Index
-----------  ------  --------------  --------------  ----------
TME0B100     S8002      TME6DABG         P00E8          P0852
etc.
```

In addition, if you have any prefix index values that begin with 8 or a higher number that represent p-words you may have created (other than DATA/ through DATX/ which are reserved and shipped with the product), you can create a new p-word in the dictionary to obtain a new prefix index value.

4. Offload your modified VSAM panel data set to a partitioned data set (one at a time into separate partitioned data sets, if you have multiple panel data sets) by using the BLGUT6F offload utility.
5. Install the Tivoli Information Management for z/OS base product panels by running the BLGUT6 utility (not the BLGUT6M utility). The input to BLGUT6 is the BLM.SBLMPNLS data set.
6. Using the information you collected, run the BLGUT6M migration utility to replace the panels in the VSAM panel data set that you offloaded. Repeat this step for each customized panel data set you have. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for utility instructions.

   You can also refer to the SBLMSAMP library for sample JCL (member BLGUT6MJ). You must change the data set names for DD names STEPLIB, BLGPDS, BLGPNLS, and BLGDICT to correspond to data set names at your location. Include the LENGTH keyword and specify the new field length (for example, 10 characters) and VALIDATION keywords as necessary to change the validation patterns.

   Using the previous example, your JCL would include the following:

   ```
   MODIFY
   LENGTH S8002,10
   VALIDATION P00E8,P0852

   etc.
   ```

   The keyword value of P0852 represents the p-word index value of the validation pattern which should replace the old validation pattern in all assisted-entry panels. In this case, the new validation pattern (IIV63 for P0852) accepts any of the 22 external date formats supported by Tivoli Information Management for z/OS or any valid user-defined format.

   Note: If you created any additional date fields (other than those provided with the product), you must also add lines in your JCL to include the field lengths and validation patterns for the fields you created.
7. When the BLGUT6M migration utility runs, warning messages are generated if a field cannot be expanded due to insufficient space. This may occur with date fields appearing in the right-hand column of your panels, where expansion would cause the last date field position to be against the edge of your panel. If this occurs, use PMF to correct any panels that could not be expanded by BLGUT6M. You can use the AT and MOVEA commands to shift the date fields on the panels to provide room for expansion.
8. Use BLGUT6F to offload any corrected panels to a read panel data set.
9. Run BLGUT6M to migrate any corrected panels.

10. Visually inspect the data entry and table panels. Use the HELP VALIDATE command to inspect the validation patterns associated with the changed fields on your panels.

In addition to performing these tasks, you should ensure that:

- If you change the internal date format, be sure to run the BLGUT1 utility to update the SDIDS.
- Any report format tables that you have created, APIs, TSPs, etc. that use dates are updated to handle dates in the new format. The base product RFTs and PIDTs are already updated to handle 10-character date fields.

If your API application must receive and process dates in a particular external date format, ensure that the DATE_FORMAT option is specified in your API transactions with the database. The value specified for the DATE_FORMAT option should match one of the supported external date formats. If it does not matter what external date format is used by your API application, and no DATE_FORMAT option is specified, the external date format specified at installation on the BLGPARMS DATEFMT keyword is used by the API application. By setting the DATE_FORMAT option, you can also set up your API programs to be unaffected by date changes in the Tivoli Information Management for z/OS database. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information on defining date formats for use with the Tivoli Information Management for z/OS APIs.

As for TSPs or TSXs, if you use them to enter, retrieve, change or manipulate dates, be aware that the external date format can vary depending on who is running the TSP or TSX (the format selected in the user profile is in use if specified). If your TSPs or TSXs process date fields, you should modify them to perform all processing in internal format and to call the BLGIDATE user exit for data retrieval and BLGEDATE user exit for data entry. The BLGEDATE user exit converts a date from internal format to the current user's external format. BLGIDATE converts a date from the current user's external format to internal format (specifically, YYYY/MM/DD). If no date preference is specified in the user profile, the external date format specified on the DATEFMT keyword is used. Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* for a description of these user exits.

Converting Existing Records in Your Database

If you are changing the external format for an existing database, you will still have records in the database with dates in the original format. As described previously in "Migration Considerations" on page 234, dates in existing records are automatically converted from the external format in which they were collected to internal format when the record is read from the database (for example, when the record is displayed, updated, or printed on a report). As a result, there is no need to perform specific migration tasks.

If you are uncertain as to whether date records are being converted properly for your database, consider running the following test:

- Create a few test records with a date occurred (DATO/) of:
  - = (the current date)
  - Your birth date in 1997
  - Your birth date in 2023
  - Your birth date in 1923 (even if you were not born then)

- Your birth date in 2098

- Use the VIEW INTERNALS command to ensure the external date for the date occurred is stored in the correct 4-digit year format.
- Use the GLOSSARY command to view the SDIDS index to confirm that a corresponding date occurred entry exists for each of the dates you entered in the test records. The entries should be in the format YYYY/MM/DD. Ensure no glossary entries exist for dates in the format YY/MM/DD.
- Enter a search argument to ensure the records show up in the search results list. For instance, you should see a result for the 1997 record, the 2023 record, etc.

Implementing an External Date Format through User Exits

If you do not use a Tivoli Information Management for z/OS external date format and instead prefer to use your own date conversion routine, you can supply a user exit. Your user exit must convert the date entered by a user to Tivoli Information Management for z/OS's internal format and must convert a date in Tivoli Information Management for z/OS's internal format to the organization's external format of the date before display.

If you supply your own date conversion routine to use an alternate date format, be aware of the following:

- You must specify the same format when you run the Tivoli Information Management for z/OS utilities. Failure to do this can produce unpredictable results.
- All users of the session must use the same external date format. The user profile option to change external date formats is not supported if you supply your own date conversion routine. To avoid confusion, remove the date format entry field from panel BLG0P700.
- You cannot use the BLGPARMS macro ODATEFMT keyword to specify an external date format of old records for conversion purposes.

During its initialization processing, Tivoli Information Management for z/OS attempts to load your user exit routine. If it can be loaded, it is used for all date conversions performed by Tivoli Information Management for z/OS.

Programming Interface information

Tivoli Information Management for z/OS enters your program as a user exit from normal Tivoli Information Management for z/OS processing, so it must follow standard linkage conventions.

The contents of the general purpose registers, upon entry to your user exit, are shown in Table 9.

Table 9. Date Conversion General Purpose Register Contents

Register	Contents
Register 0	Unpredictable
Register 1	Address of a 4-word input parameter list
Registers 2–12	Unpredictable
Register 13	Address of a 72-byte register save area
Register 14	Return address
Register 15	Module entry point address (module return code on exit)

The following procedure outlines the steps you follow to implement a different external date format.

1. Write a user exit routine in the language of your choice. You must make this program reentrant. A sample program is in member BTNUDATE in the SBLMSAMP library.

   Requirements for the program you write depend on whether you use an existing database with records containing the old date format and on whether you plan to use multiple date formats within your databases.

If you are establishing a new database, the program you write must convert your organization's user-entered date to the Tivoli Information Management for z/OS internal format and convert the Tivoli Information Management for z/OS internal format back to your organization's external format. A switch setting determines which function is required.

Updating records in an existing database containing dates in the old format damages the data in your SDIDS unless your date conversion user exit routine can convert both the old and new formats to the Tivoli Information Management for z/OS internal format, and convert the Tivoli Information Management for z/OS internal format back to your organization's external format. Refer to the sample date conversion user exit routine prolog in member BTNUDATE in the SBLMSAMP library.

Note: If you included SAM or Network Problem Determination Application (NPDA) problem reporting in the database, the dates contained in the records created by SAM or NPDA are in the format MM/DD/YY. Therefore, your date conversion user exit routine must convert this format as well as your external date format if you intend to update these records.

Your program must do the following:

a. Get the input parameters that the address in Register 1 points to. The four parameters have slightly different descriptions, depending on the direction of the conversion. Table 10 describes the parameters. Those that apply to the external-to-internal format are in parentheses.

Table 10. Date Conversion Input Parameter Descriptions

Parameter	Length	Description
@SWITCH	Fullword	Address of a fullword, fixed, and signed, binary field. The switch indicates the direction of the conversion: 0 requests an internal-to-external format. 4 requests an external-to-internal format.
@EXTERNAL	Fullword	Address of 66 bytes consisting of: ■ A halfword signed binary number for the length of the date in the external format ■ A 64-byte character field for the date in external format.
@INTERNAL	Fullword	Address of an 8-byte character field that contains the internal date in YY/MM/DD format.
@INTERNAL2	Fullword	Address of 12 bytes consisting of: ■ A halfword signed binary number for the length in the date in the internal format. Only three values are accepted: • X'0' - ignore any date at the address in the @INTERNAL2 parameter • X'A' - @INTERNAL2 parameter contains a date in YYYY/MM/DD format ■ A 10-byte character field for the date in the internal format.

b. Perform date modification based on the switch setting (the @SWITCH parameter):

1) If the switch is 0, get the internal date (from either the @INTERNAL or @INTERNAL2 parameters), change it to the external format you want to use, and place the date's length and the date itself at the address in the @EXTERNAL parameter.

2) If the switch is 4, get the external date (the @EXTERNAL parameter), change it to internal format, and place the date's length (10) and the date in the YYYY/MM/DD format at the address in the @INTERNAL2 parameter. Any date at the address of the @INTERNAL parameter is ignored.

Figure 22 illustrates the parameter list (PLIST) structure for the user exit as it appears to an assembler-language routine. The address of the PLIST is found in general purpose register 1 (GPR1).

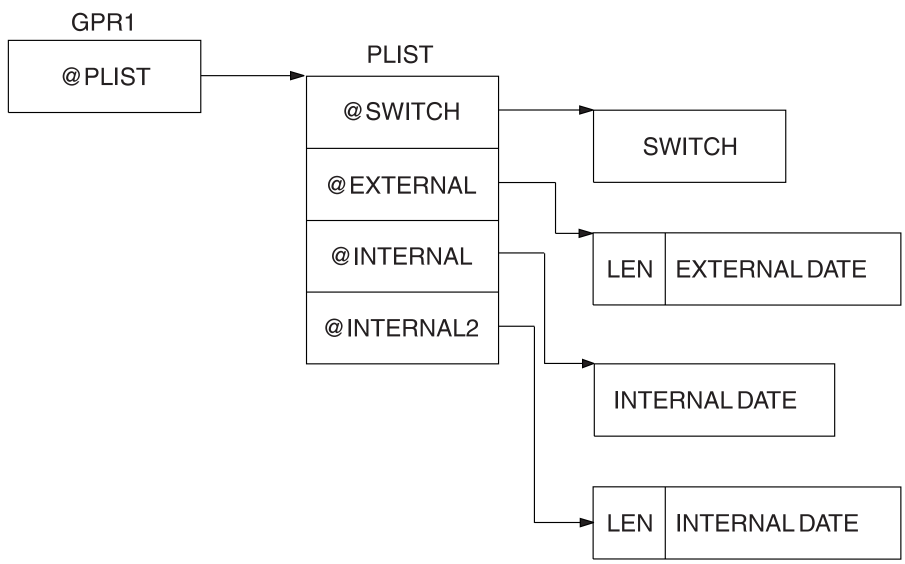

Figure 22. Input Parameter List for an Alternative Date Format User Exit

An external date format that is fewer than 64 characters long (including the prefix and a slash, if present) is left-justified with the remainder of the field padded with blanks. You cannot use an organization-defined external date format of more than 64 characters. The panels shipped with Tivoli Information Management for z/OS do not permit an organization-defined external date format of more than 10 characters (or 8 characters if you are using shorter date fields); you must change some panels using PMF if you want to use an external date format of more than 10 (or 8) characters.

c. Set the return code in register 15, and return to the calling program. Follow standard linkage conventions on exit from the program. For internal-to-external conversion, the only valid return code is:

0 Conversion successfully performed.

For external-to-internal conversion, the only valid return codes are:

0 Conversion successfully performed. The specified external date passed all organization-required validation checks.

16 Conversion failed. The user's date input is incorrect. Tivoli Information Management for z/OS notifies the user of the error and reprompts.

Any other return code causes an ABEND.

2. Use the Installation Tailoring Facility to specify the load module name of your user exit routine in your session-parameters member. Or use the DATECNV keyword in the BLGPARMS macro instruction. See page 322 for additional information.
3. Make the program module available at installation time. The AMODE of your date conversion user exit routine must be 31.
4. Use PMF to modify the data-entry or table panels to adjust the length of the fields as necessary. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on modifying panels.

Example of a Date-Conversion Routine — BTNUDATE

The BTNUDATE sample program (in the SBLMSAMP library) is an example of a date conversion user exit routine that is coded in assembler language. This program converts dates to and from the external format of DDMMMYY (MMM represents the first 3 characters in the month). It also supports a second external format for records that originate from an interface that contains a different external format to convert dates to the internal format. However, a display of an old record shows any dates in the old format. You can use the BTNUDATE example as a model for writing your user conversion routine.

You must link-edit this user exit routine with an addressing mode (AMODE) of 31.

Note: You must allocate storage to make the program reentrant.

Date Validations

Using your own date conversion user exit routine does not affect Tivoli Information Management for z/OS's validation of users' date entries. After a date is converted to the internal format you chose, Tivoli Information Management for z/OS validates it according to the date ranges specified in Table 11:

Table 11. Valid Date Ranges

Date Field	Valid Range
YY	00–99
YYYY	0000–9999
MM	01–12
DD	01–31, according to each month and year. For non-leap years, the DD value used with February cannot be greater than 28.

If the program exit or the Tivoli Information Management for z/OS validation check detects an error, the user receives an error message.

End of Programming Interface information

Enabling an Alternate Time-of-Day Format

You can use your own external time-of-day format by specifying an installation-supplied user exit routine during the Tivoli Information Management for z/OS installation process.

Note: All time-of-day entries created previously with Information/System Versions 1 through 4.2, Information/Management Version 5.1, and records created through the

NetView Hardware Monitor Interface, are displayed in the old HH:MM format. (The term *old* format refers to the format that was used when the records were created.) There is no utility program to convert the old time of day entries to the new format you choose. If NetView Hardware Monitor Interface records are updated, the time last altered appears in the current time-of-day format.

The Tivoli Information Management for z/OS internal time-of-day format is the same as the default external format:

```
HH:MM
```

HH The hour of the day
MM The minute of the hour

A colon (:) separates the hour from the minutes. The time-of-day format:

- Uses a 24-hour, 60-minute clock format
- Can be up to 30 characters, including the length of the prefix
- Can include special characters if you modify the panels to accept them

  Note: You may not want to use blanks because blanks act as word delimiters.
- Cannot include commas because commas separate responses in response chains.

Time-of-day data must be prefixed and the prefix must begin with the characters TIM. If you use your own user-written time-of-day conversion routine, the TIM prefix must not be used for any data other than time-of-day data.

Notes:

1. This enhancement only applies to time-of-day format. Durations and extended durations continue to be in the DD:HH:MM and DDDD:HH:MM formats.
2. If you plan to use the SETD Report Format Facility statement, your external time format must not be similar to the duration format, the extended duration formats, or your external date formats, so that the Tivoli Information Management for z/OS Report Format Facility can reliably detect that it is processing a time as opposed to a duration, extended duration, or date.

If you plan to use the default time-of-day format, you need not write a user exit routine. However, if you do not want to use the Tivoli Information Management for z/OS default time-of-day format, you must supply a user exit routine. Your user exit must convert the time-of-day entered by a user to Tivoli Information Management for z/OS's internal format and must convert a time-of-day in Tivoli Information Management for z/OS's internal format to the organization's external time-of-day format before display.

During its initialization processing, Tivoli Information Management for z/OS attempts to load your user exit routine. If it can be loaded, it is used for all time-of-day conversions performed by Tivoli Information Management for z/OS.

Programming Interface information

Tivoli Information Management for z/OS enters your program as a user exit from normal Tivoli Information Management for z/OS processing, so it must follow standard linkage conventions.

The contents of the general purpose registers, upon entry to your time-of-day user exit, are shown in Table 12.

Table 12. Time-of-Day Conversion General Purpose Register Contents

Register	Contents
Register 0	Unpredictable
Register 1	Address of a 3-word input parameter list
Register 2-12	Unpredictable
Register 13	Address of a 72-byte register save area
Register 14	Return address
Register 15	Module entry point address (module return code on exit)

Implementing a New External Time-of-Day Format

The following procedure outlines the steps to follow to implement a different external time-of-day format:

1. Write a user exit routine in the language of your choice. You must make this program reentrant. A sample program is in member BLGUTIMC in the SBLMSAMP library.

   Requirements for the program you write depend on whether you use an existing database with records containing the old time-of-day format and on whether you plan to use multiple time-of-day formats within your databases.

   If you are establishing a new database, the program you write must convert your organization's user-entered time-of-day to the Tivoli Information Management for z/OS internal format and convert the Tivoli Information Management for z/OS internal format back to your organization's external format. A switch setting determines which function is required.

   Updating records in an existing database containing times-of-day in the old format damages the data in your SDIDS unless your time-of-day conversion user exit routine can convert the new format to the Tivoli Information Management for z/OS internal format and can also recognize and support the old format. The old format does not need conversion but still must be returned to Tivoli Information Management for z/OS as the internal time. For your time-of-day conversion user exit routine to reliably detect whether it is processing a time in the old format or a time in the new format, your new format cannot be similar to an nn:nn format (where *nn* is a 2-digit numeric value).

   Note: If you included SAM or NPDA problem reporting in the database, the times-of-day contained in the records created by SAM or NPDA are in the format HH:MM. Therefore, your time-of-day conversion user exit routine must support this format as well as your external time-of-day format if you intend to update these records.

   Your program must do the following:

   a. Get the input parameters that the address in Register 1 points to. The three parameters have slightly different descriptions, depending on the direction of the conversion. Table 13 gives the parameter descriptions. Those that apply to the external-to-internal format are in parentheses.

A. Alternative Date and Time-of-Day Formats

Table 13. Time-of-Day Conversion Input Parameter Descriptions

Parameter	Length	Description
@SWITCH	Fullword	Address of a fullword fixed signed binary field. The switch indicates the direction of the conversion: 0 requests an internal-to-external format 4 requests an external-to-internal format
@EXTERNAL	Fullword	Address of 32 bytes consisting of: 1) A halfword signed binary number for the length of the time-of-day in the external format 2) A 30-byte character field for the time-of-day in your organization's external format.
@INTERNAL	Fullword	Address of a 5-byte character field for the time-of-day in the HH:MM internal format.

b. Perform time-of-day modification based on the switch setting (the @SWITCH parameter):

1) If the switch is 0, get the internal time (the @INTERNAL parameter), change it to the external format you want to use, and place the time's length and the time itself at the address in the @EXTERNAL parameter.

2) If the switch is 4, get the external time (the @EXTERNAL parameter), change it to internal format HH:MM, and place the internal format at the address in the @INTERNAL parameter.

Figure 23 illustrates the parameter list (PLIST) structure for the user exit as it appears to an assembler-language routine. The address of the PLIST is found in general purpose register 1 GPR1.

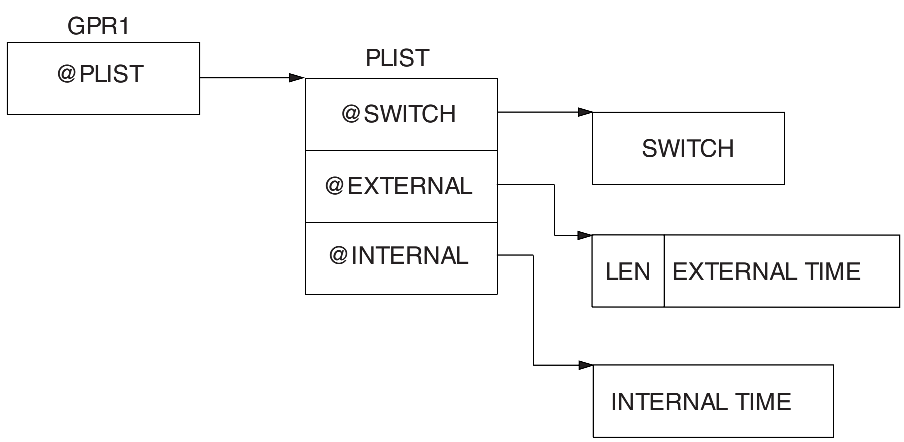

Figure 23. Input Parameter List for Alternate Time-of-Day Format User Exit

An external time format that is fewer than 30 characters long including the prefix data is left-justified with the remainder of the field padded with blanks. You cannot use an organization-defined external time-of-day format of more than 30 characters. The panels shipped with Tivoli Information Management for z/OS do not permit an organization-defined external time format of more than 5 characters; you must change some panels using PMF if you want to use an external time-of-day format of more then 5 characters.

c. Set the return code in register 15 and return to the calling program. Follow standard linkage conventions on exit from the program. For internal-to-external conversion, the only valid return code is:

0 Conversion successfully performed.

Any other return code causes an ABEND.

For external-to-internal conversion, the only valid return codes are:

0 Conversion successfully performed. The specified external time passed all organization-required validation checks.

16 Conversion failed. User's time input is incorrect. Tivoli Information Management for z/OS notifies the user of the error and reprompts.

Any other return codes cause an ABEND.

2. Use the Installation Tailoring Facility to specify the load module name of your user exit routine in your session-parameters member. Or use the TIMECNV keyword in the BLGPARMS macro instruction. See page331 for additional information.
3. Make the program module available at installation time. The AMODE of your time-of-day conversion user exit routine must be 31.
4. Modify the time panels using PMF. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details on modifying panels. A list of the panels appears in Table 14 on page 250.

Example of a Time-of-Day Conversion Routine

The BLGUTIMC sample program (in the SBLMSAMP library) is an example of a time-of-day conversion user exit routine that is coded in assembler language. The program converts times-of-day to and from a 12-hour time-of-day external format. It also supports a second external format of HH:MM for records that originate from an interface that contains the Tivoli Information Management for z/OS default HH:MM format to return as the internal format (no conversion is necessary). However, a display of an old record continues to show any times-of-day in the format in which they were entered. You can use the BLGUTIMC example as a model for writing your own time conversion program.

You must link-edit this conversion user exit routine with an AMODE of 31.

Note: You must allocate storage to make the program reentrant.

Modifying Time-of-Day Formats on Panels

In addition to writing the exit routine, you must modify the appropriate assisted-entry panels so your users can enter time correctly. Use PMF to change the visible portion and control portion of the Tivoli-supplied assisted-entry panels. If your new external format is nn:nn, you only need to change the visible portion.

Panels

If your organization adopts a new time-of-day format, you must modify the assisted-entry panels listed in Table 14.

Table 14. Panels Requiring Modification For Time-of-Day Format Changes

Panel Name	Field Name
BLG6ALTT	Time Altered
BLG6ASST	Time Assigned
BLG6CMPT	Time Finished
BLG6CRTM	Time Entered
BLG6OCCT	Problem Occurrence Time
BLG6REQT	Time Required
BLG6SCHT	Planned Start Time
BLG6TART	Planned End Time
BLG6TIMX	General Time
BLG6TSTA	Actual Start Time
BLG600CT	Time Closed
BLG7ALTT	Time Altered

Alternate Time-of-Day Format for the Integration Facility

If your organization also uses the Integration Facility, then in addition to the panels listed in Table 14, you must also modify the following Integration Facility assisted-entry panels for your new time-of-day format:

Table 15. Integration Facility Panels Requiring Modification For Time-of-Day Format Changes

Panel Name	Field Name
BTN6CMPT	Time Finished
BTN6OCCT	Problem Occurrence Time
BTN6SCHT	Planned Start Time
BTN6TART	Planned End Time
BTN6TIMX	General Time
BTN6TSTA	Actual Start Time
BTN600CT	Time Closed

Additionally, if you plan to use the Panel Set Create function of PMF, you must modify panels BLM6Y009 (Occurrence Time) and BLM6Y00G (Time Required).

Additional Panel Considerations

All of the assisted-entry panels listed in Table 14 and Table 15 are started through certain data-entry or selection panels. If your new external time-of-day format is greater than or fewer than 5 characters, be sure to modify the related data-entry panels. To identify all related panels, run a panel cross-reference report. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for instructions.

Time-of-Day Validations

Using your own conversion user exit routine does not affect Tivoli Information Management for z/OS's validation of users' time-of-day entries. After a time is converted to the internal format, Tivoli Information Management for z/OS validates it according to the ranges specified in Table 16 on page 251:

Table 16. Valid Time-of-Day Ranges

Date Field	Valid Range
HH	00–24
MM	00–59 **Note:** If HH is 24, then MM is 00.

If the program exit or the Tivoli Information Management for z/OS validation check detects an error, the user receives an error message.

End of Programming Interface information

Implementing Universal Time Processing

If your company or customer locations are spread geographically, you may want to consider implementing the universal time feature of Tivoli Information Management for z/OS. You can specify universal time processing by including the TIMEZONE keyword and time zone symbol on the BLGPARMS macro and by defining a relationship between date and time fields. The advantages of implementing universal time processing include the following:

- Users in different geographic areas who share a database can enter and view dates and times in their own time zones, regardless of how or where the records were entered. For example, if a user in Boston enters a problem record at the following date and time:

  ```
  Date: 04/30/2001   Time: 10:45
  ```

  a user in Italy can see the record as:

  ```
  Date: 30/04/2001   Time: 16:45
  ```

  The Boston time value of 10:45 is expressed as 16:45 for the user in Italy, reflecting the data in a time zone the user can readily understand.

  If the universal time feature is not used, the users in both countries would see the same value (10:45), which could make it more difficult for users in one geographic area to work with the records in the database if the format chosen is not native to their area.
- With universal time processing, you can find problems that occurred within a period of time easily, even if they were entered in different time zones. This is especially useful if you need to track service level agreements.
- For companies that use Tivoli Information Management for z/OS and undergo mergers, it is easier to handle differences in external date and time formats because large migration efforts are not required.

This section explains some of the concepts behind universal time processing and describes how to implement it in your Tivoli Information Management for z/OS environment.

Note: If you use the Integration Facility, you should not implement universal time processing. Date processing results could be unpredictable if universal time processing is implemented. You (or your Tivoli Information Management for z/OS program administrator) should also consider using PMF to remove the **User's time zone** field on BLG0P700, the User and database defaults panel. This field is intended to be used only if universal time processing is implemented at your installation. To avoid frustrating users by displaying a field that will not work in the Integration Facility environment, you can update the panel to remove the field so that users cannot use it. For instructions on updating panels, refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

Universal Time Concepts

Universal time processing in Tivoli Information Management for z/OS is based on Coordinated Universal Time. Coordinated Universal Time (UTC, or UT as it may be called in this manual) is a time base that is coordinated on a longitude, which is an imaginary line running from the North Pole to the South Pole. The universal time feature provided with Tivoli Information Management for z/OS is based on Gregorian time (hours and minutes). Universal time is coordinated on the Prime Meridian (Greenwich, UK). The Prime Meridian is at 0 degrees, and there are longitudinal lines every 15 degrees for a total of 24 (360 degrees). Because it is universal, universal time can be referred to by anyone on the globe.

By implementing universal time processing in Tivoli Information Management for z/OS, you can have users anywhere in the world work with records in the database in their own local time. This feature can help make it easier for users to work with the data, because they can understand date and time values from their own perspective. For example, suppose a system crash occurred at noon. Where is that ″noon″? Was it in the UK? In the US? In Australia? Noon can represent different times depending on the time zone. Knowing the time zone is essential to understanding what a particular time value means. It is also necessary to determine the sequence of events and to calculate the time difference between two events.

To illustrate this concept, suppose a system crash occurred at 8:00 a.m. in California. The problem is resolved 5 minutes later by someone in Boston. The Boston time is really 11:05 a.m. The duration of the outage is merely 5 minutes; but if you simply calculated the difference between the 8:00 a.m. and 11:05 a.m., you would think the outage was 3 hours and 5 minutes. If you did not have a universal way to calculate time, you could get a distorted picture of how long a problem is actually open.

Of course, you could require everyone to enter times using a single time zone, but that imposes a burden on all users not in that particular time zone. It may also lead to errors, especially if you have users in different countries which have different rules for Daylight Saving Time. The time difference between one country and another can change as often as four times a year if the two countries start and end Daylight Saving Time on different days. Therefore, it is important to know what Daylight Saving Times rules are in effect for your various geographies. With Tivoli Information Management for z/OS, you can use the Daylight Saving Time rules provided, or you can define your own rules as necessary.

How Data is Converted and Stored

Date values are entered by users and displayed in the user's local time. When universal time processing is enabled, the data entered by a user is automatically converted from that user's local time to universal time. The dates and times are stored in the SDDS in *universal time*, regardless of any user preference. When records are read from the database, the date and time fields are automatically converted from universal time to the local time of the user reading the records.

Whether or not you decide to implement universal time processing, users can select the external date format of their choice by updating their user profile. For a discussion of migration considerations, see "Tivoli Information Management for z/OS Version 7.1 Changes" on page 103 and "Migration Considerations" on page 234.

History data for dates and times is stored in universal time and converted to the user's local time upon display.

Date and time math program exits (including the calculation of duration, such as how long a problem was open) use universal time values, when available, in all calculations if universal time processing is specified. In addition, date and time calculations are adjusted for Daylight Saving Time if the duration is less than one day. For example, if Daylight Saving Time begins at 02:00 on April 3, then the duration from 04/02 20:00 to 04/03 08:00 is calculated as 11 hours (adjusted). Likewise, the duration from 04/02 20:00 to 04/03 19:59 is calculated as 22 hours, 59 minutes (adjusted). But the duration from 04/02 20:00 to 04/03 20:00 is calculated as 1 day (24 hours); this calculation is unadjusted because the parameters span a full day, even though the actual time elapsed is 23 hours. (If TIMEZONE is not specified, universal time values are unavailable. For non-universal time date and time fields, the values cannot be converted to the current user's time zone. The original local date and time stored in the record are used instead.)

Although users can select the external date format they want to see on panels, in reports, and in other output, they cannot control which form of the value is shown. That is, their user profile does not let them switch back and forth between seeing *universal time*, *original local time*, or *user's local time* for any given date or time field. (These terms are defined in "Terminology" on page 255.) The content of the field data is determined by the PMF administrator through panel definition, the author of an RFT, or the author of a TSX which uses the FINDSDATA control line.

Date Processing by PMF, RFTs, TSXs or TSPs, and FINDSDATA

When a record is processed, each date or time value exists in the record in the following forms:

- User's local – The value in the user's *external* format and, for universal time fields, the user's local time
- Original local – The value in *internal* format as it was entered (that is, with no time zone adjustment)

For fields defined as universal time fields, a third form is also present:

- Universal time – The value in *internal* format and universal time

If you use PMF, write RFTs to produce reports, or write or use TSXs or TSPs using the FINDSDATA control line, be aware that these functions by default use the user's local date and time forms only—not universal time or original local form. (Before Tivoli Information Management for z/OS Version 7.1, the original local form was used by default because it was the only form available.) Currently, all output (for example, screen display, RFT output, data retrieved through the API) for all date and time fields collected are in the external format of the user accessing the data. If the field is defined as a universal time field, these values will represent the user's local time; otherwise they will represent the data as it was entered into the record.

For example, suppose you are using FINDSDATA with the default search type (DATA) to find a date value. By default, FINDSDATA only looks at the user's local forms of dates and times (not the universal time and original local forms), so you must specify the user's local form of the date or time. In other words, if the user's date format is DD-MM-YYYY and you want to use the FINDSDATA control line to find a ″date occurred″ of any day in May

2001, you must use an argument containing DATO/**–05–2001. A FINDSDATA control line with an argument will not find any universal time or original local date values, even if the argument matches the data.

However, PMF administrators, RFT writers, and TSX writers can force a particular form of date and time field data on panels, RFTs, or in TSXs, regardless of what users set up as a preference in their user profiles. A particular format can be forced by doing the following:

- TSXs: To use the FINDSDATA control line to find data in a form other than user's local, you must specify a ″searchtype″ of UT to find the universal time form or OLOCAL to find the original local form. In these cases, FINDSDATA only checks the specified form of date and times for a match.
- Panels: A data-entry or table panel can be defined to display the original local or universal time forms of a date or time value. Use PMF to specify OLOCAL or UT, respectively, for ″Date/time display form″ in the field's control information.

  It is possible to show all three forms on a single panel. To do so, simply define three separate fields for the same piece of data, each with a different setting for the ″Date/time display form″ option.

  CAUTION! All values entered must be in user's local form (external format), regardless of the setting of ″Date/time display form.″ To avoid confusion, it is strongly recommended that you specify OLOCAL or UT only on display fields and not on entry fields. Specifying OLOCAL or UT on an entry field means the data will be displayed in a different form than it must be entered, which is confusing to users.
- RFTs: On a PUT statement, you can use the optional DTFORM keyword (date/time form) with a value of UT to display the universal time date or time on a report. You can also specify the OLOCAL value on this keyword to display the original local date or time. If you omit the DTFORM keyword, the user's local date or time will be displayed.

How Time Zones are Specified

As described in "Specifying Preferences in the User Profile" on page 230, users have the ability to override the external date format and time zone specified at installation for use by their session. The user profile option enables users to specify a preference of how dates should be displayed, and what user time zone should be used, so that it is easier to work with the data. In many cases, users will prefer to see the data in their own local external date format and time zone. This eliminates the need for users to remember, for example, that a particular company location is 4 hours ahead of their own local time. It also helps to eliminate the annoyance of having to interpret panel data in a foreign format.

You can specify the time zone Tivoli Information Management for z/OS should use in the following ways:

- You can code a particular time zone symbol for the BLGPARMS TIMEZONE keyword for a given session member. If you previously had the TIMEDEL parameter specified, you can remove it. The TIMEZONE keyword is required for universal time processing to occur. The time zone value you specify must match one of the time zone symbols in the time zone record. Only one time zone symbol should be coded in the TIMEZONE keyword.
- A user can select a time zone symbol in the user profile. Selection of a time zone through the user profile overrides the setting in BLGPARMS for the user.

- If you are interacting with the database through an API application, you can specify the time zone selection through the high-level API parameter data block (TIME_ZONE PDB) or PICATZON field in the PICA for the low-level API.

If you have records in your database that meet any of the following conditions, you should include an additional keyword in your session parameters, OTIMEZON. Include the OTIMEZON keyword to represent the time zone of the older records in your database. Tivoli Information Management for z/OS uses the value specified for OTIMEZON to convert dates and times in the older records to universal time. If OTIMEZON is not specified, the time zone specified on the TIMEZONE keyword is used for migrating old records.

- Records were created with versions of Tivoli Information Management for z/OS or its predecessor products before Version 7.1, and the dates and times in those records are for a time zone which is different from the default time zone coded in the TIMEZONE keyword.
- You have date or time fields in any records that were created before the dates and times were defined as a universal time (related) pairs. This includes:
  - Records created before Tivoli Information Management for z/OS Version 7.1
  - Records created with Version 7.1 before universal time processing was enabled
  - Records created with Version 7.1 after universal time processing was enabled, but before the fields were defined in the DATETIME record

Establishing a Relationship Between Date and Time

If you enable universal time processing by specifying the TIMEZONE keyword, you must also define a relationship between date fields and time fields in your records. Only the date and time fields that are related in *related pairs* (and therefore listed in a DATETIME record) are processed in universal time. The relationship also enables users across various geographic regions to see field data on a panel and know exactly what is represented by the field.

A related pair is a link between the date and time fields. This link is established primarily to identify the exact moment in time at which an event occurred (for example, Date occurred and Time occurred). This relationship is also necessary for any automated calculation of time differences to take place, such as for service level agreements. Without knowing that a particular date is associated with a particular time, it would be difficult to calculate, for example, open problem durations. The date and time data for related pairs is stored in the database in universal time but is displayed back to the user in the current user's time zone.

Terminology

At this point, you may be getting a bit confused about date and time processing. The following terms used in this appendix are defined here to help you understand how universal date and time processing works in Tivoli Information Management for z/OS. Reviewing these definitions before proceeding will make it easier for you to understand how to plan for and implement the following optional universal time functions in Tivoli Information Management for z/OS:

- Daylight savings time definition
- Time zone definition
- Related date and time fields

Related field
A date or time field that has a defined relationship to another time or date; for example, **Date occurred** and **Time occurred**. A date field is related to a time field only by the Tivoli Information Management for z/OS administrator, who must define the relationship by updating a special DATETIME record in the database. To convert universal time to a user's local time, date and time fields must be related.

Tivoli Information Management for z/OS provides a DATETIME record including most of the related fields that the base product uses. If you want fields related on your customized panels, you need to create a DATETIME record for those panels. Instructions are provided in "Relating Date and Time Fields" on page 262.

Independent field
A date or time field that has no defined relationship. No universal time processing is done for independent fields. Users viewing the data in an independent field will see the date and time in their external format, but the value will be the value that was entered, which may not be their local time.

Universal time (UT)
A worldwide common standard time, also sometimes called Greenwich mean time (GMT) or Zulu.

Original local date, original local time
The local date or time value that was *entered into a field* by a user. For example, if a user in Chicago types 05/15/2001 as the **Date occurred** and 11:30 as the **Time occurred**, that date and time is the original local date and original local time.

User's local date, user's local time
The date or time value in the local time of the user *currently viewing* the data. For example, if a user in Frankfurt is viewing the problem record opened by the user in Chicago on 05/15/2001 at 17:30 Chicago time, the Frankfurt user's local date and time is 05/16/2001 at 00:30.

Understanding Independent Fields

To get an understanding of how related fields are used, you should first understand how independent fields are processed. If date and time fields are independent (that is, they are not related because there is no DATETIME record entry defining them as part of a related pair, or the TIMEZONE keyword is not specified in the session parameters), then the processing of the data is as follows:

- When a user enters a value in a date field or time field, the data is collected two ways:
  - In internal format (original local)
  - In the external format as it was entered by the user

  Data is processed this way if it is entered by an interactive user, a control panel ADD line, an API create or update transaction, a WORDFIX replace function using validation, a WORDFIX add function, or a TSX ADDSDATA control line.
- When the record containing the date or time data is filed, the external format is removed so that only the internal format remains in the SDDS.
- When the record is read back in by a user (that is, when it is displayed, updated, or presented on a report), the internal format is converted to that user's external format. Both the external and internal formats are kept in the record while the record is in storage. The external format is removed when the record is filed.

The external format is displayed on panels and reports. The way a user performs a search is unaffected. If users are already familiar with Tivoli Information Management for z/OS, they are not required to do anything special to perform a search.

Searching Independent Field Data

Searching for independent field data is no different than searching for date or time fields in releases prior to Tivoli Information Management for z/OS Version 7.1. To search on date or time data, a user can enter a structured search or a freeform search. For example, in this illustration, a search is done for records containing both the date and time values shown, although the fields are not related:

```
se dato/2001/05/15 timo/12:30
```

Only the original local date and original local time are cognized (stored in the database for searching). Only those records having the specific date and time entries in this example are returned in the search results list.

Understanding Related Fields

Date and time fields are related if the TIMEZONE keyword is specified in the session parameters *and* there is an entry in the DATETIME record defining the date and time as a pair.

The determination of whether a field is independent or related is determined on a field-by-field basis. It is possible to have some dates and times defined as related pairs and others not defined (and therefore treated as independent fields).

If date and time fields are related, the processing of the data is as follows:

- When a user enters a value in a date field or time field, the data is collected *three* ways:
  - In internal format (universal time)
  - In internal format in the local time originally entered into the record (original local time)
  - In external format in the local time of the current user (user's local time)

  Data is processed this way if it is entered by an interactive user, a control panel ADD line, an API create or update transaction, a WORDFIX replace function using validation, a WORDFIX add function, or a TSX ADDSDATA control line.

  The universal date and time are determined by combining the entered value with the value of its related date or time field. The data is then adjusted by the universal time offset of the user's local time zone.
- When the record containing the date and time data is filed, the external format (the user's local external format) is removed so that only the two internal formats remain in the SDDS.
- When the record is read back in by a user (that is, when it is displayed, updated, or presented on a report), the internal format stored in universal time is converted to external format of the user viewing the data, in that user's time zone. All three formats are kept in the record while the record is in storage. If you were to issue a VIEW INTERNALS command, you would see all three formats. The user's local external format is removed from the record when the record is filed.

By default, the user's local date and time external formats are displayed on panels and reports. This can be changed by explicitly coding an option to show universal date/time or original local date/time values. Because both the original local values and universal time values are cognized, the way a user performs a search can vary depending on whether the user wants to search original local values or universal time values. Searching of related field data is described "Searching Related Field Data" on page 259.

Processing When Only One Field is Entered

If the user enters data in only one field of a related pair, there is not enough information for Tivoli Information Management for z/OS to calculate the universal time value. When this happens, the following assumptions are made:

- If only a date is entered, the universal time date is assumed to be the same as the user's local date.
- If only a time is entered, the time is converted to universal time assuming that Daylight Saving Time is not in effect. For example, a time of 11:00 entered in the ET time zone (US eastern, UT −5:00) when no related date exists will be converted to 16:00 universal time. If a date of 07/01/2001 is later added in the paired field, the time will be converted to 15:00 universal time to reflect the fact that Daylight Saving Time is in effect on 07/01/2001.

Scenario Showing Use of Related Fields

To understand how related date and time fields are collected and processed, consider this scenario:

- A user in Atlanta creates a new problem record at 9 p.m. that evening. She prefers to view the data in the traditional US format. She enters a date in format MM/DD/YYYY and time in HH:MM format. She is on the eastern coast of the US and therefore in US Eastern Time (UT −5:00), on Daylight Saving Time. In the record, she specifies that the problem was opened at 04/27/2001 21:00.

  The date and time are collected as follows:

  ```
  Universal time             2001/04/28
                             01:00

  Original local             2001/04/27
                             21:00

  Current user's local       04/27/2001
                             21:00 (external format seen by the
                                   user in Atlanta)
  ```

  The user files the record and the external format is removed, leaving only the following:

  ```
  Universal time             2001/04/28
                             01:00

  Original local             2001/04/27
                             21:00
  ```

- A user in Frankfurt displays that record just after it was entered. From the US perspective, Germany is 6 hours ahead of US Eastern time. The user is on Central European Time (UT +1:00), Daylight Saving Time, and has selected the DD-MM-YYYY external format.

  Values in the user's local time are automatically added to the record, yielding the following:

```
Universal time            2001/04/28
                          01:00
Original local            2001/04/27
                          21:00
Current user's local      28-04-2001
                          03:00 (external format seen by the
                                user in Frankfurt)
```

He sees that the problem record was opened on 28-04-2001 03:00. In effect, the system translated the date and time from universal time into a format instantly recognizable by the user in Frankfurt.

Searching Related Field Data

Related field data can be searched using freeform or structured search methods. A description of how to perform freeform searches and structured searches in general is provided in the *Tivoli Information Management for z/OS User's Guide*.

Freeform Searches

Users can perform freeform searches two ways:

- Search original local values.

  To search original local values, enter normal search arguments that specify date or time data in internal format. This type of search will cause the values as they are originally entered by a user to be searched. For example, to search on problem records opened on April 27, 2001 at 3:30 p.m.:

  ```
  se dato/2001/04/27 timo/15:30
  ```

- Search universal time values.

  Note: Because of the complexity of freeform search arguments for universal time values, it is suggested (but not required) that you use structured searches instead of freeform searches when searching for universal time values. If you choose to use a freeform search for universal time values, you should be aware of the following.

  To search universal time values, enter the search argument with a special ending character for the p-word of the related field:

  ```
  se dato&2001/04/27 timo&15:30
  ```

  In this example, the same p-words (`dato` and `timo`) are entered but an ampersand character is used instead of a slash. (If your p-words use the underscore character instead of the slash, substitute the percent sign % for the underscore character to search universal time values.) The ampersand tells Tivoli Information Management for z/OS to search the universal time values (not the original local values) stored in the database.

  If you are performing a freeform search on universal time fields, be aware that records are only cognized in universal time and the local time of the person who entered the data. This has the following implications:

  - A search for DATE/2001/02/20 TIME/12:00 will find all records entered on 02/20/2001 at noon in the local time of the person entering the data. This means you might find a record from noon Sydney time and another record from noon Paris time, even though these records were entered many hours apart.

- Time ranges that cross the universal time date boundary (such as 16:00 to 22:00 Eastern Time) require a complex universal time search argument. For this reason, it is suggested that you use structured searches when searching universal time values, because the structured search performs the universal time conversions automatically.

Structured Searches

Users can enter values in the user's local date format or user's local time format. Internally, the search is performed using the equivalent universal time values. For example, if you enter a structured search containing a date value of `03/18/2001` and time value of `11:15` for the related fields, Tivoli Information Management for z/OS uses the universal time equivalent values to find any matching records.

By default, a structured search performed against related field data performs a universal time search. It does *not* cause original local values to be searched.

If you perform a structured search on universal time fields, you should be aware of the following limitations:

- For all users:
  - Wildcard (*) and truncation (.) characters are not allowed in search arguments.
- For users in time zones that observe Daylight Saving Time (DST):
  - A search which specifies a time or time range must also specify a date or date range in the related date field.
  - A date range specified with a related time value cannot include all or part of more than 5 calendar years. For example, a search from 01/01/2001 to 12/31/2005 is acceptable, whereas a search from 12/31/2001 to 01/01/2006 is not acceptable because it includes part of 6 calendar years.
  - A search will fail if its parameters fall entirely within the skipped hour when DST begins.
  - Search times that occur during the repeated hour when DST ends are assumed to occur during the first hour (the hour before DST ends). For example, a search range for 10/29/2000 01:30–03:30 using US Eastern Time as the time zone will find all records with a value between the ″first″ 1:30 and 3:30, a range of three hours that would include the 01:15 that occurs after DST ends.

Scenario Showing Search of Universal Time Values

In this scenario, users in three different time zones in standard time simultaneously experience a problem and enter a problem record with the date and time the problem occurred as follows:

- A user in Toronto enters 03/18/2001 11:15. The user is in the ET time zone (US, Canada Eastern time zone with daylight savings).
- A user in St. Louis enters 03/18/2001 10:15. The user is in the CT time zone (US, Canada, Mexico Central time zone with daylight savings).
- A user in Phoenix enters 03/18/2001 09:15. The user is in the MST time zone (US, Canada, Mexico Mountain time zone without daylight savings). (Phoenix never goes on Daylight Saving Time.)

Because the month is March, none of these time zones are on Daylight Saving Time.

Internally, the times entered by the Toronto, St. Louis, and Phoenix users are converted into universal time:

```
Toronto     11:15       5 hr offset from UT  =   16:15
St. Louis   10:15       6 hr offset from UT  =   16:15
Phoenix     09:15       7 hr offset from UT  =   16:15
```

A user in New York (same time zone as Toronto) searches for those records. She enters a structured search and types `03/18/2001` and `11:15` as the date and time the problem occurred. The search is performed using the equivalent universal time values.

She could also have entered the following freeform search to find the same records:

```
se dato&2001/03/18 timo&16:15
```

All three records will be found by the search because the problems really did occur at the same time, even though the original local time values entered by the users were different. The ampersand character tells Tivoli Information Management for z/OS to find the data using universal time equivalents. This type of search can be useful if you need to find records associated with events occurring around the world.

For example, suppose your network support team installs a new network router, and the router was not installed successfully and problems occurred at many of your company's geographic locations. You may want to judge the impact of that installation by finding all the problems that occurred at a particular time (or time range) at all your company's locations. If you used the regular slash character in the freeform search instead (as shown below), the search would yield no matching records because 16:15 does not match the original local values of 11:15, 10:15, or 09:15.

```
se dato/2001/03/18 timo/16:15
```

Note: Your users may not have a need to enter searches that narrow in focus, but if they do, they should understand how freeform and structured searches are performed when working with related field data. If you choose to implement universal time processing, your Tivoli Information Management for z/OS administrator should be prepared to educate users on how to perform searches of date and time fields. The *Tivoli Information Management for z/OS User's Guide* also describes how to perform searching when universal time processing is enabled.

More About Related Fields

To perform the universal time-to-local time conversion, Tivoli Information Management for z/OS requires data for both fields in a related pair. If a date is entered and there is no value for the related time field, the universal time date is set to the same value as the local date. If the time is entered but there is no value for the related date, the time zone is adjusted by the base universal time offset of the local time zone. Since the date is not known, no adjustment is performed for Daylight Saving Time. In both of these cases, the date and time values are adjusted if necessary when a value is entered for the related time or date field.

There may be situations where fields that were collected as related fields are processed as independent fields. If any of the conditions below are true, then the fields are not defined as related in that environment:

- The session does not have the TIMEZONE keyword on BLGPARMS, and is therefore not processing universal time dates.

- The database does not contain a DATETIME record (perhaps because it was deleted or the record accessed was UNFLATTENed into a database without a DATETIME record).
- The DATETIME record does not contain a row for the field (perhaps because it was deleted or the record accessed was UNFLATTENed into a database with different definitions in the DATETIME record).

In these cases, the field is treated as an independent field and the user's local date and time values are set to the same values as the original local date and time values. This enables sessions without the TIMEZONE keyword to access records created by sessions with the TIMEZONE keyword and still see the date and time values as they were entered. If the fields are updated, they will be stored as independent fields and the universal time values will be lost. This can result in incorrect values being stored. To avoid this problem, it is recommended that:

- All databases have the same definitions in the DATETIME record.
- All sessions have the TIMEZONE keyword specified. (The TIMEZONE keyword values can be different.)
- Definitions in the DATETIME record are never changed or deleted.

For list processor data, the related field is the one with the related root and the same row index. For example, the date in row 3 of a date list is related to the time in row 3 of a time list.

Only a single value can be entered in a related date or time field. If a user tries to enter multiple responses in one or both related time fields, an error message is displayed and the data is not accepted.

Changing one field's value can cause the universal time value in the related field to change also. For example, if you change the time (in Eastern daylight time, ET) from 18:00 to 21:00, the universal date changes. That is, the addition of 3 hours (from 6 p.m. to 9 p.m.) means that the next day has already started in universal time.

The use of related fields may be associated with some minor performance impact. Records with a large number of related pairs may take longer to read because each related date and time pair is converted to local time and external format when the record is read.

Relating Date and Time Fields

The tasks described in this section can be performed by your program administrator or the person responsible for customizing your system.

To relate a date field and a time field, create a DATETIME reference record and add a row that specifies the s-word index of the two fields that are related. You can create a DATETIME record through the entry panel of the System application of Tivoli Information Management for z/OS.

A DATETIME record including most of the related fields is provided with the base Tivoli Information Management for z/OS product.

On BLG00010, the System Record Entry panel, type **2** and press Enter.

```
+ BLG00010 ------------ SYSTEM RECORD ENTRY --------------- 1 OF 1-+
|                                                                  |
| USE....Identify the type of description (record) to be entered.  |
|                                                                  |
| 1.CLASS.............Define authority and users in a privilege    |
|                     class record.                                |
| 2.REFERENCE.........Define reference information.                |
| 3.LOGSAVE...........Define information used by the Automatic     |
|                     Log Save and DB2 Extract Facilities.         |
| 4.MAP...............Define PostProcessor data mapping.           |
| 5.INDEX.............Define index for text search.                |
|                                                                  |
|                                                                  |
+------------------------- SELECT ITEM ----------------------------+




===> 2
```

On the Reference Entry panel, select option **5** and press Enter.

```
+ BLG00020 ------------- REFERENCE ENTRY ----------------- 1 OF 1 -+
|                                                                  |
| USE....Identify the type of reference information to be entered. |
|                                                                  |
| 1.USERS...............Define users and their logon IDs.          |
| 2.ALIAS...............Define alias for panels.                   |
| 3.COMMAND.............Define command aliases and authorization.  |
| 4.TIMEZONE............Define time zones.                         |
| 5.DATE/TIME...........Define date/time field relationships.      |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
+------------------------ SELECT ITEM -----------------------------+




===> 5
```

Programming Interface information

On the Date/Time Field Relationship Entry panel, enter the s-word index values for the date and time fields, the prefix word for the time field, and, optionally, a comment that describes the pair of fields (for administrative purposes).

```
BLGLDTTM           Date/Time Field Relationship Entry           LINE 1 OF 12

USE....List relationships between date and time fields.  Fields listed here
       will be processed internally using Universal Time (UT).
                                                             RECORD: DATETIME
     --S-word indexes-   Time
     Record Date  Time  Prefix                        Comment
''''  ____  0C34  0C61  TIME/_   Date/time entered_____________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
''''  ____  ____  ____  ______   ______________________________________________
 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.

===>
```

Leave the **Record** field blank unless you have the situation where a particular date and time combination is used differently across record types (see the following note).

If you are using list processor data, specify the root s-word of the two related lists. The related date and time s-words must both be list processor roots or must both be regular field s-words. If a list processor root is related to a regular field s-word, neither will be processed correctly.

Press Enter after entering the s-word index values. Type **end** on the command line and press Enter to file the record.

Note: Generally, relationships are one date to one time, with an s-word existing in the table only once. For example, you would not typically have the **Date entered** field listed once with a **Time entered** field and again with a **Time opened** field. However, you may have unique circumstances where the s-word for a date is associated with more than one time s-word across record types, or the time is associated with more than one date field. In this case you should specify each date and time field combination on a separate line and also specify the s-word index value for the record type. For example, if S0C34 is usally related to S0C61, but in change records only, S0C34 is instead related to S8123, then you would include one line `S0B06 0C34 8123` to define the relationship for change records, and another line `0C34 0C61` to define the relationship for all other records.

All record-specific pairs must be listed before any general pairs that use the same s-word. See the following panel BLGLDTTM for an example.

```
BLGLDTTM           Date/Time Field Relationship Entry           LINE 1 OF 12

USE....List relationships between date and time fields.  Fields listed here
       will be processed internally using Universal Time (UT).
                                                                RECORD: DATETIME
     --S-word indexes-   Time
     Record Date  Time  Prefix                    Comment
''''  0B06  0C43  0C70  TIMX/_   Actual date/time started (Change)____________
''''  ____  0C34  0C61  TIME/_   Date/time entered____________________________
''''  ____  0C35  0C62  TIMM/_   Date/time modified___________________________
''''  ____  0C37  0C64  TIMA/_   Date/time assigned___________________________
''''  ____  0C3D  0C6A  TIMO/_   Date/time occurred___________________________
''''  ____  0C38  0C65  TIMR/_   Date/time closed_____________________________
''''  ____  0C43  0C6C  TIMB/_   Actual date/time started_____________________
''''  ____  101F  105E  ______   PMF date/time modified_______________________
''''  ____  ____  ____  ______   _____________________________________________
''''  ____  ____  ____  ______   _____________________________________________
''''  ____  ____  ____  ______   _____________________________________________
 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.

===>
```

After the DATETIME record is filed, the changes take effect as follows:

- For interactive 3270 users: Changes take effect when a new session is started, or, if logical database partitions are used, when the user changes partitions.
- For API applications (including MRESs and API server applications used for Tivoli Information Management for z/OS Desktop and web interfaces): Changes take effect when a new session initialization transaction is performed, or, in the case of logical database partitions, when the partition is changed. In general this means Desktop users will not see these changes until the IBM HTTP server is restarted or program interface data tables are refreshed.

You can change the contents of an existing DATETIME record by updating the record (`UPD R DATETIME`) and filing the changes.

Warning

You must be very careful when changing existing rows. Deleting or changing s-words in a row can cause data with those s-words to be lost or corrupted.

End of Programming Interface information

Defining and Using Time Zones

A set of commonly used time zone definitions for many geographic locations is available in the TIMEZONE record provided with Tivoli Information Management for z/OS. The TIMEZONE record is a list processor table containing a list of time zone names (symbols) and their corresponding definitions.

You may have unique needs that are not met by these definitions. If you need to use other time zones not provided in the TIMEZONE record, or need to change the time zone

A. Alternative Date and Time-of-Day Formats

definitions provided, you can update the record to add your own definitions. When defining the TIMEZONE record, any symbol can be defined, but session members and user profiles must use a symbol that has been defined in the TIMEZONE record.

The time zone definitions provided are shown in Table 17 on page 267. An asterisk (*) in the offset column indicates that the time zone observes Daylight Saving Time. You can install the default TIMEZONE record provided with Tivoli Information Management for z/OS by running the BLHRCDSL TSX, which is also used to install data model records. (Complete instructions on running the BLHRCDSL TSX are provided in "Loading Records Provided with Tivoli Information Management for z/OS" on page 219.) The TIMEZONE record is included in the basic record set defined in the BLHLRBAS list.

Users can view the list of defined time zones by selecting the appropriate option in their user profile. The time zone for a particular user can be set in the following ways:

- In the session parameters (TIMEZONE keyword in the BLGPARMS macro). The setting of this keyword is required for time zone processing.
- In the user profile (User and database defaults option). A user profile selection overrides the TIMEZONE session parameter.
- By an API transaction (TIME_ZONE PDB). A value passed on the PDB overrides the TIMEZONE session parameter.

Users can view the time zone currently in effect by issuing the HELP STATUS command. The following panel is an example of the HELP STATUS output:

```
BLG1THLP               STATUS AND AVAILABLE COMMANDS              LINE 1 OF 35

VERSION = VxRxMx  RECORD ID = NONE  CLASS = MASTER  SUSPENSION LEVEL = 0
MODES = INQUIRY   SEQUENTIAL RECORD
DATABASE = 5   LOGICAL FILES = *
SESSION MEMBER = DD   WINDOW NAME = BLGISPFD   WINDOW LEVEL = JOYxxx
PRIMARY PARTITION = NONE   GLOBAL PARTITION = NONE
TIME ZONE FORMATS:    DEFAULT = ET      OLD = NONE
EXTERNAL DATE FORMATS:    DEFAULT = MM/DD/YYYY    OLD = MM/DD/YY
TRIGGER CHARACTERS = & #   DATA MODEL DATABASE = 5

COMMAND NAME      COMMAND DESCRIPTION

ARGUMENT        ADD/DELETE/MODIFY FREEFORM ARGUMENTS
BACK            BACK UP TO LAST DISPLAYED OR PROCESSED PANEL
CANCEL          TERMINATE THE CURRENT DIALOG WITHOUT SAVE
CHANGE          CHANGE SEARCH ARGUMENT OR AN SRC RECORD
COPY            COPY RECORD FROM THE CURRENT DATABASE
DELETE          DELETE RECORD FROM THE CURRENT DATABASE
DISPLAY         DISPLAY RECORD FROM THE CURRENT DATABASE
DROP            DROP USER-DEFINED LINE COMMAND
END             TERMINATE THE CURRENT DIALOG WITH SAVE
===>
```

Note: Normally, the equal sign (=) processing function, in which a user can type an equal sign into a date or time field to get the current system date or time, is based on the local time of the CPU of the system running Tivoli Information Management for z/OS. If universal time processing is enabled by coding the TIMEZONE keyword (and, optionally, the OTIMEZON keyword) in the session-parameters member, then equal sign processing for date and fields is determined differently. The current universal time is extracted from the system and adjusted by the offset for the local time zone. For example, the system may be physically based in Germany, but a user in California would receive the equivalent of California date and time when typing the equal sign into a date or time field. Equal sign processing automatically adjusts for the start and end of Daylight Saving Time, even if the local time on the z/OS system running Tivoli Information Management for z/OS is not changed.

If the TIMEZONE keyword is not specified, the current local time from the system is extracted and adjusted by the TIMEDEL value, if any TIMEDEL value was specified.

Table 17. Time Zone Definitions in the TIMEZONE Reference Record

Time Zone Symbol	Offset from Universal Time (UT)	Description of areas covered
NZT	+ 12:00 *	New Zealand (ex.Chatham)
EAT	+ 10:00 *	Australia Eastern with DST (Sydney, Melbourne, Canberra)
EAST	+ 10:00 *	Australia Eastern without DST (Brisbane)
TASMANIA	+ 10:00 *	Australia-Tasmania
CAT	+ 09:30 *	Australia Central with DST (Adelaide)
CAST	+ 09:30	Australia Central without DST (Darwin)
KST	+ 09:00	Korea
JST	+ 09:00	Japan, E. Indonesia
CHINA	+ 08:00	China, Taiwan, Hong Kong, Singapore, C. Indonesia, Malaysia
WAST	+ 08:00	Australia West without DST (Perth)
INDIA	+ 05:30	India
MSK	+ 03:00 *	Moscow, Russia
EET	+ 02:00 *	Eastern Europe with DST
SAFRICA	+ 02:00	South Africa
CET	+ 01:00 *	Central Europe with DST
WET	+ 00:00 *	UK, Western Europe
UT	+ 00:00	UT, GMT
NDT	– 03:30 *	Newfoundland with DST
AT	– 04:00 *	Canada Atlantic with DST
CHILE	– 04:00 *	Chile
ET	– 05:00 *	US/Canada Eastern with DST
EST	– 05:00	US/Canada Eastern without DST (Indiana)
CT	– 06:00 *	US/Canada/Mexico Central with DST

Table 17. Time Zone Definitions in the TIMEZONE Reference Record (continued)

Time Zone Symbol	Offset from Universal Time (UT)	Description of areas covered
MT	– 07:00 *	US/Canada/Mexico Mountain with DST
MST	– 07:00	US/Canada/Mexico without DST (Arizona)
PT	– 08:00 *	US/Canada/Mexico Pacific with DST
AKT	– 09:00 *	Alaska
HST	– 10:00 *	Hawaii
*Indicates the time zone observes Daylight Saving Time.		

The procedure to define a time zone varies depending on whether or not the time zone observes Daylight Saving Time (DST). If it does, you should define the Daylight Saving Time rules (when daylight savings starts and stops) before adding a time zone definition.

Defining Daylight Saving Time Rules

To define Daylight Saving Time rules for a new time zone, or to change the existing Daylight Saving Time rules for an existing time zone entry in the TIMEZONE record, follow these procedures.

Update the TIMEZONE record (`UPD R TIMEZONE`). The Time Zone Definition Summary panel is displayed.

```
 BLG0Z200              TIME ZONE DEFINITION SUMMARY         RECORD: TIMEZONE


  Choose 1 to define rules for Daylight Saving Time start and stop date/time.

  Choose 2 to define time zones, including symbol, offset from UT, Daylight
           Saving Time adjustment amount and which start/stop rules apply.


  Select one of the following, type END to save your changes, or type CANCEL
  to discard your changes.

             1. Daylight Saving rules.    2. Time zone definitions.

                                9. File record.



===> 1
```

Type **1** and press Enter. An entry panel is displayed for you to enter the rules for starting and stopping Daylight Saving Time. After entering the data on the Daylight Saving Time Schedule Entry panel, type **end** and press Enter to save the changes.

```
BLGLDSTS         Daylight Saving Time Schedule Entry          LINE 1 OF 13

USE....Define rules for start and stop of Daylight Saving Time.
                                                                RECORD: TIMEZONE
                          -------- Start --------  --------- End ---------
                                   Day  Day                  Day  Day
     Schedule  From  To            of   of  Local            of   of  Local
        ID     Year Year  Mon Week Week Mon Time    Mon Week Week Mon Time
'''' MYREGION  ____ ____  04   L    5   __  02:00  09   L    4   __  02:00
'''' NAMERICA  ____ ____  04   1    7   __  02:00  10   L    7   __  02:00
'''' SYRIA___  ____ ____  04   _    _   01  02:00  10   _    _   01  02:00
'''' CHILE___  ____ ____  10   1    7   09  02:00  03   1    7   09  02:00
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
'''' ________  ____ ____  __   _    _   __  _____  __   _    _   __  _____
 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.

===> end
```

The panel example shows the entry of a Daylight Saving Time rule called MYREGION. In this example, Daylight Saving Time starts at 2:00 a.m. local time on the last Friday of April and ends on the last Thursday of September at 2:00 a.m.

The second row shows the rule for North America. Daylight savings time starts on the first Sunday of April at 02:00 local time and ends on the last Sunday of October at 02:00. In Syria, it starts on April 1 and ends on October 1. In Chile, it starts on the first Sunday on or after October 9 and ends on the first Sunday on or after March 9.

The **From Year** and **To Year** fields are used only if rules are different in different years. For example, some part of a country may begin Daylight Saving Time earlier than other years due to special events. For instance, in 2000 part of Australia began Daylight Saving Time earlier than usual due to the Sydney Olympics. In this case, three lines are required: one for years up to and including 1999, one for the special rules in 2000, and a third for 2001 and beyond. For example:

```
                          -------- Start --------  --------- End ---------
                                   Day  Day                  Day  Day
     Schedule  From  To            of   of  Local            of   of  Local
        ID     Year Year  Mon Week Week Mon Time    Mon Week Week Mon Time
'''' AUSTRAL1  ____ 1999  10   L    7   __  02:00  03   L    7   __  03:00
'''' AUSTRAL1  2000 2000  08   _    _   27  02:00  03   L    7   __  03:00
'''' AUSTRAL1  2001 ____  10   L    7   __  02:00  03   L    7   __  03:00
```

A description of the entry fields follows:

Schedule ID
: A unique 1 to 8–character identifier used to represent the Daylight Saving Time start and stop rules. (This identifier is also used later on the Time Zone Definition panel to specify which rule should apply to a time zone.)

From Year
The first year the Daylight Saving Time rule should apply. Normally, if the rules do not change from year to year, this field is blank or 0.

To Year
The last year the Daylight Saving Time rule should apply. If the rule is expected to remain in effect with no expiration, leave this field blank or enter 9999.

Start Mon
A number (01–12) representing the month during which the Daylight Saving Time starts (for example, 04 for April).

Start Week
A value (1– 4, or L) indicating which week of the month (if you were to look at a Gregorian calendar) Daylight Saving Time starts:

```
1    -    First week of the month
2    -    Second week of the month
3    -    Third week of the month
4    -    Fourth week of the month
L    -    Last week of the month
```

Start Day of Week
A number (1–7) defining which day of the week Daylight Saving Time starts. For example:

```
1    -    Monday
2    -    Tuesday
3    -    Wednesday
4    -    Thursday
5    -    Friday
6    -    Saturday
7    -    Sunday
```

Start Day of Mon
A number (01–31) that defines the start day of the month as one of the following:

- If no values are specified for **Start Week** and **Start Day of Week**, the actual date on which Daylight Saving Time starts.
- If values are specified for **Start Week** and **Start Day of Week**, the earliest date on which Daylight Saving Time can start. For example, if the following values are entered, Daylight Saving Time starts on the first Sunday on or after the 10th of April:

```
-------- Start ---
         Day  Day
         of   of
Mon Week Week Mon
04   1    7   10
```

The default value is 01.

Start Local Time
The local time (00:00 to 23:59) when Daylight Saving Time starts. For example, 02:00 represents 2 a.m. The default value is 00:00.

End Mon
A number (01–12) representing the month during which Daylight Saving Time ends (for example, 04 for April).

End Week
A value (1–4, or L) that defines which week of the month Daylight Saving Time ends (for example, 1 for the first week, L is for the last week).

End Day of Week
A number (1–7) that defines which day of the week Daylight Saving Time ends (for example, 1 for Monday, 5 for Friday, 7 for Sunday).

End Day of Mon
A number (01–31) that defines the end day of the month as one of the following:

- If no values are specified for **End Week** and **End Day of Week**, the actual date on which Daylight Saving Time ends.
- If values are specified for **End Week** and **End Day of Week**, the earliest date on which Daylight Saving Time can end. For example, if the following values are entered, Daylight Saving Time ends on the first Sunday on or after the 10th of November:

```
-------- End-----
         Day  Day
         of   of
Mon Week Week Mon
11   1    7   10
```

The default value is 01.

End Local Time
The local time (00:00–23:59) when Daylight Saving Time ends (for example, 02:00 represents 2:00 a.m.). The default value is 00:00.

After updating this panel and typing end to save your changes, you are returned to the summary panel where you can type **9** and press Enter to file the TIMEZONE record or select another option.

```
BLG0Z200               TIME ZONE DEFINITION SUMMARY         RECORD: TIMEZONE


 Choose 1 to define rules for Daylight Saving Time start and stop date/time.

 Choose 2 to define time zones, including symbol, offset from UT, Daylight
          Saving Time adjustment amount and which start/stop rules apply.


 Select one of the following, type END to save your changes, or type CANCEL
 to discard your changes.

            1. Daylight Saving rules.    2. Time zone definitions.

                                 9. File record.



===> 2
```

Programming Interface information

Defining a Time Zone

To add a time zone definition, or to change an existing time zone definition in the TIMEZONE record, update the TIMEZONE record to display the Time Zone Definition Summary panel. Select option **2. Time Zone Definitions** and press Enter.

```
 BLG0Z200             TIME ZONE DEFINITION SUMMARY         RECORD: TIMEZONE


  Choose 1 to define rules for Daylight Saving Time start and stop date/time.

  Choose 2 to define time zones, including symbol, offset from UT, Daylight
           Saving Time adjustment amount and which start/stop rules apply.


  Select one of the following, type END to save your changes, or type CANCEL
  to discard your changes.

             1. Daylight Saving rules.    2. Time zone definitions.

                                9. File record.



===> 2
```

On the Time Zone Definition panel, enter a new row with the appropriate data if you are adding a time zone definition. Specify a time zone symbol, the offset from universal time, any Daylight Saving Time adjustment if applicable, and any optional comments to describe the time zone for administrative purposes. (If you are updating an existing time zone definition, scroll to the definition desired and overtype the values with the new values.)

```
BLGLTZDF                    Time Zone Definition                   LINE 1 OF 13

USE....Define time zones and their characteristics.
                                                                RECORD: TIMEZONE
       Time Zone   Offset    DST       DST
        Symbol     From UT   Adj.    Schedule                Comment
''''' MYZONE__    -08:00    1:00    MYREGION    Turquoise Isle
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
''''' ________    ______    ____    ________    _____________________________
 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.

===> end
```

Time Zone Symbol
: A unique 1 to 8–character symbol that identifies the time zone (for example, ET for Eastern Time with daylight savings, EST for Eastern Time with no daylight saving).

The symbol can be of your choice; it does not have to match any predetermined symbol provided in the TIMEZONE record. You can change the time zone symbols if desired or add new ones.

Note: If you change the symbol for a time zone already in use, you must also change everything that uses the time zone, such as session members, user profiles, and API applications.

Offset From UT
The offset from universal time (−14:00 to +14:00) of this time zone while not on Daylight Saving Time. For example, +01:00 for Germany, or −05:00 for Eastern United States.

DST Adj.
The adjustment in hours and minutes applied when Daylight Saving Time is in effect. The adjustment is 1:00 in most cases. If the time zone does not observe Daylight Saving Time, then enter 0 or leave blank.

DST Schedule
The schedule ID for one or more rows in the table that describe when Daylight Saving Time starts and stops for this time zone. The schedule ID is defined on the Daylight Saving Time Schedule Entry panel, BLGLDSTS. If the time zone does not observe Daylight Saving Time, then leave blank.

Comment
An optional description of the time zone, which may possibly include the names of major cities. Users see this description when viewing the list of available time zones in the user profile.

Type **end** and press Enter to save your changes and to return to the Time Zone Definition Summary panel. On the Time Zone Definition Summary panel, type **9** and press Enter to file the changes to the TIMEZONE record.

After the TIMEZONE record is filed, the changes take effect as follows:

- **For interactive 3270 users**: Changes take effect when a new session is started, or when the user modifies the time zone in his or her user profile (that is, when the user does a permanent or temporary save or a reset). If logical database partitions are used, the change takes place when the user changes partitions.
- **For API applications** (including MRESs and API server applications used for Tivoli Information Management for z/OS Desktop and web interfaces): Changes take effect when a new session initialization transaction is performed, when the time zone is changed to a different value using the TIME_ZONE PDB on a transaction, or in the case of logical database partitions, when the partition is changed. In general this means that Desktop users will see the changes as soon as any Desktop user on the server with a different time zone in the user profile performs a transaction. If all users have the same time zone specified in their user profiles and use the same logical database partition, the HTTP Server must be restarted to pick up the changes.

Specifying Time Zones

A specific time zone value can be specified through session-parameters members that define the operating characteristics of Tivoli Information Management for z/OS, through an API application interacting with the database, or by a user through a user profile selection.

Specifying a Time Zone with the BLGPARMS TIMEZONE Keyword

To enable universal time processing and to specify a default time zone for all users of a session, include the TIMEZONE keyword on the BLGPARMS macro and specify the appropriate time zone symbol (for example, TIMEZONE=ET) as described on page 332. After updating your session-parameters members with this keyword, you must reassemble them for universal time processing to take place. If session-parameters members are not reassembled, the default date and time support processing provided with Tivoli Information Management for z/OS will be used.

If the TIMEZONE keyword is not specified, no universal time processing will occur, even if the DATETIME and TIMEZONE records are present in the Tivoli Information Management for z/OS database. This means that if the TIMEZONE keyword is not specified, the values of date and time fields will represent the data as it was entered into the record. A list of time zone symbols provided with Tivoli Information Management for z/OS is available in Table 17 on page 267; however, other time zone symbols can be defined in the TIMEZONE reference record and specified in the TIMEZONE keyword. The value for TIMEZONE must be defined in the TIMEZONE reference record. Time zone definitions include rules for Daylight Saving Time adjustments.

The TIMEZONE keyword, if specified, replaces the TIMEDEL keyword. If you already have the TIMEDEL keyword specified, you should remove it. If both the TIMEZONE and TIMEDEL keywords are specified, TIMEDEL is ignored.

The processing of the equal sign function for date and time fields will determine the current local date and time by taking the current universal time from the system and adjusting it by the offset defined for the current time zone. Equal sign processing will automatically adjust for Daylight Saving Time start and end, even if the local time on the computer system is not changed. (If TIMEZONE is not specified, equal sign processing uses the current local time from the computer system, adjusted by any TIMEDEL value if specified.)

The time zone currently in effect can be viewed in the output generated by the HELP STATUS command.

Specifying Time Zones with API Applications

If you are using API applications to interact with the Tivoli Information Management for z/OS database, the TIMEZONE value specified in the session-parameters member is used. To use a time zone other than that specified in the session-parameters member, your API application programmer must do the following:

- For HLAPI applications: Specify the TIME_ZONE control PDB with the desired time zone symbol as the data in any HLAPI transaction.
- For LLAPI applications: Set the PICATZON field with the desired time zone symbol for the LLAPI.

The specified value remains in effect until a subsequent transaction changes it. The time zone symbol selected should be defined in the TIMEZONE record.

End of Programming Interface information

Specifying Time Zones through the User Profile

Users can select a time zone that overrides the time zone set by the BLGPARMS TIMEZONE keyword by specifying an option (**User and database defaults**) in their user profile. For more information about setting values in user profiles, see "Specifying Preferences in the User Profile" on page 230 or refer to the *Tivoli Information Management for z/OS User's Guide*.

By default, all output such as panel displays and report output for the date and time fields collected displays the external format of the user accessing the data. The date format displayed can be changed by a user through the user profile.

If no value is specified in the user profile, then the time zone specified on the TIMEZONE keyword in the session member is used.

B

Defining Tivoli Information Management for z/OS Data Sets

This appendix describes the Tivoli Information Management for z/OS VSAM data sets and tells how to define them using the Access Method Services (AMS) DEFINE CLUSTER command. The following data sets are described:
- SDDS
- SDIDS
- SDLDS
- DICTDS
- RPANLDS
- WPANLDS

This appendix assumes that you have a working knowledge of VSAM data sets and the DEFINE CLUSTER command. Refer to *DFSMS/MVS Access Method Services for VSAM Catalogs* for a complete description of the parameters for the AMS DEFINE CLUSTER command. Refer to the following documents for additional information about VSAM:
- *DFSMS/MVS Access Method Services for the Integrated Catalog Facility*
- *DFSMS/MVS Using Data Sets*
- *DFSMS/MVS Macro Instructions for Data Sets*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*

This appendix is designed to help you create the VSAM data sets that are used by Tivoli Information Management for z/OS. Most installations initially create small data sets; that is, 10 000 records or smaller. Therefore, the sizes given in this chapter are intended for generating small data sets.

Notes:

1. If you are using sysplex support, VSAM data sets must be managed by SMS (the Storage Management Subsystem facility of DFSMS/MVS).
2. Any key-sequenced data set that follows may be defined as an extended format data set.

Sample JCL

Tivoli Information Management for z/OS provides sample JCL defining the Tivoli Information Management for z/OS VSAM data sets. The following samples can be found in the SBLMSAMP library:
- BLGDATAB (defines an SDDS, SDIDS, and SDLDS for a Tivoli Information Management for z/OS database)
- BLGDICT (defines a DICTDS)
- BLGRPNL (defines an RPANLDS)
- BLMWPNL (defines a WPANLDS)

Understanding the AMS DEFINE CLUSTER Command Syntax Description

The syntax diagrams for the AMS DEFINE CLUSTER commands in this chapter show some parameters with brackets *[ ]*, braces *{ }*, or vertical bars |. These are for clarification purposes only; they are not part of the commands.

- Brackets indicate optional data
- Braces indicate that you must select exactly one of the choices enclosed within the braces
- Vertical bars indicate selectable items. You must choose one of these items.

In addition, a plus sign + is used at the end of the line on which continuation is required. You can use any valid character that you want to identify continuation.

Using REUSE or NOREUSE

The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

Defining the SDDS

The syntax diagram for the DEFINE CLUSTER command for defining an SDDS follows:

```
DEFINE                                 +
  CLUSTER (                            +
    NAME(sddsname)                     +
    [INDEXED]                          +
    KEYS(n 0)                          +
    [LOG(NONE)]                        +
    SHAREOPTIONS(1 3)                  +
    [NOREUSE]                            +
    VOLUMES(volser [volser...])        +
    [UNIQUE])                          +
  DATA(                                +
    NAME(sddsname.data)                +
    [CONTROLINTERVALSIZE(cisize)]      +
    RECORDSIZE(avg max)                +
    {RECORDS(primary secondary) | TRACKS(primary secondary) |
       CYLINDERS(primary secondary)} +
    [FREESPACE(freeci freeca)]         +
    SPEED)                             +
  INDEX(                               +
    [NAME(sddsname.index)]             +
    [NOIMBED]                          +
    [NOREPLICATE])
```

CLUSTER
Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole. Enclose these parameters in parentheses.

NAME(sddsname)
Specifies the name of the VSAM cluster. You must specify NAME. The *sddsname* is a standard data set name.

You can use up to 100 VSAM clusters in an SDDS. If you use more than one cluster for a Tivoli Information Management for z/OS database, you must define each cluster, and they must conform to the following naming conventions:

- Each cluster must have the same name, except for a 3-character sequence:

- The first character of the 3-character sequence is a trigger character. You can use any valid data-set-name character as the trigger character. You define the trigger character in the BLGCLUST macro, as described under "BLGCLUST Macro — Defining a Database" on page 333. The purpose of this character is to indicate that you are using multiple VSAM clusters in an SDDS.

  The trigger character does not have to be a unique character in a cluster name. However, since the trigger character indicates the start of the sequence, it must be the first occurrence of that character in a cluster name.

  The trigger character must be identical in each cluster name.
- The remaining 2 characters in the sequence must be numeric. In the first cluster, these characters must range from 00 to 95. In the remaining clusters, these characters must range from 01 to 99.
- The numeric characters following the trigger must be in consecutive numerical order for the database that contains the multiple VSAM clusters.
- The sequence must occur in the same position in each SDDS cluster name.

The following is an example of multiple VSAM clusters that form a 10-cluster SDDS. The number sign (#) is the trigger character.

```
BLM.SDDS#27
BLM.SDDS#28
BLM.SDDS#29
BLM.SDDS#30
BLM.SDDS#31
BLM.SDDS#32
BLM.SDDS#33
BLM.SDDS#34
BLM.SDDS#35
BLM.SDDS#36
```

Note: If you define # as the trigger character and you use more than one # within a name, Tivoli Information Management for z/OS treats only the first # as a trigger character. In this case, Tivoli Information Management for z/OS treats the remaining #s like any other character.

INDEXED
Indicates that the SDDS is a key-sequenced data set. You do not have to specify this keyword because INDEXED is the default.

KEYS(n 0)
The first value, *n*, indicates the length of the key field in the SDDS VSAM record. Tivoli Information Management for z/OS identifies the format of a record stored in the SDDS. You can specify a value of 7 or 8 for *n* when you are defining an SDDS for a Tivoli Information Management for z/OS format database. Generally, a key format of 7 is recommended.

Note: SDDSs for databases 0, 1, 2, and 3 must have a key length of 8.

- A value of 7 indicates that the stored record has key 7 format.
- A value of 8 indicates that the stored record has key 8 format.

The second value, 0, specifies the offset of the key field in the SDDS.

You must specify this keyword and its values.

LOG(NONE)

Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)

Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE

The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

VOLUMES(volser [volser...])

Specifies the volumes to contain the SDDS. *Volser* is the volume serial number of the device. You must specify one or more devices. This keyword is not required for SMS-managed data sets.

UNIQUE

Indicates you are allocating the SDDS as separate DASD space. This keyword is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space.

Note: If you are using the integrated catalog facility (ICF) in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA

Specifies attributes of the data component of the SDDS. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(sddsname.data)

Specifies the name of the data component of the SDDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *sddsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)

Specifies the control interval size for the data component of the SDDS. The sizes you can specify must be:

- A multiple of 512 for values from 512 to 8192
- A multiple of 2048 for values from 8192 to 32768

RECORDSIZE(avg max)
Specifies the average and maximum record sizes for the SDDS. The maximum record size must be:
- For a key 7 format SDDS, equal to the control interval size minus 7 (max=cisize−7).
- For a key 8 format SDDS, at least 7 bytes less than the control interval size (max≤cisize−7) It is recommended that you set the maximum record size at 7 bytes less than the control interval size so that Tivoli Information Management for z/OS can use the maximum available space for the blocking of the logical records. For example, for the control interval size of 2048, use a maximum record size of 2041.

If you are using a USERS record for notification, use a maximum record size greater than 10 000 because the USERS record is at least this large.

RECORDS(primary secondary)

TRACKS(primary secondary)

CYLINDERS(primary secondary)
Specifies the amount of space to allocate to the SDDS. *Primary* specifies the initial space allocation; *secondary* specifies the increments of the allocation when new extents are necessary to extend the SDDS. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required.

FREESPACE(freeci freeca)
Specifies the percentage of freespace to allocate in the data set. *Freeci* is the percentage to allocate for each control interval and *freeca* is the percentage to allocate for each control area. The recommended values for *freeci* and *freeca* depend, to a large degree, on how the SDDS is used. If it is used as a repository, relatively little freespace is required. For a high level of update activity before deletion, the use of freespace can be advantageous.

SPEED
Indicates that the data component's space is not preformatted during initial load. The SPEED keyword is required.

INDEX
Specifies attributes of the index component of the cluster. The attributes follow the INDEX keyword and you must enclose them in parentheses.

NAME(sddsname.index)
Specifies the name of the index component of the SDDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *sddsname.INDEX*, the name is more recognizable, for example, in a LISTVTOC listing.

Using IMBED or NOIMBED
IMBED indicates that the lowest level index records (sequence set) of the data set are written as many times as they fit on the first track adjacent to the related control area. This can shorten the time it takes to retrieve records when physical access to the DASD is required. With NOIMBED, the sequence set is written once and resides with the rest of the index levels.

NOIMBED may be the better choice for the SDDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the SDDS index control intervals (CIs) recommended, then after the first access there are no physical reads of the index from the DASD, so IMBED has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- If you are using a controller cache function, IMBED can be counterproductive. Because entire tracks are buffered into the controller cache, IMBED can cause cache memory in the controller to be used to unnecessarily store the replicated index records.
- If you are using RLS (sysplex mode is enabled), you must use NOIMBED. RLS does not support use of IMBED.
- Starting with DFSMS/MVS Version 1.5, IMBED is no longer supported, and is ignored if specified on the IDCAMS DEFINE. If sysplex mode is not enabled, you may continue to use the old VSAM data sets that were defined with the IMBED attribute.

For these reasons, NOIMBED is recommended for the SDDS index. Consider using IMBED for the SDDS index only if you are not buffering most of the SDDS index through LSR and you are not using a controller cache function for the SDDS.

Using REPLICATE or NOREPLICATE

REPLICATE indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, REPLICATE can reduce the rotational delay of the disk.

However, NOREPLICATE may be the better choice for the SDDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the SDDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so REPLICATE has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- If you are using a controller cache function, REPLICATE may be counterproductive. Because entire tracks are buffered into the controller cache, REPLICATE can cause cache memory in the controller to be used to unnecessarily store the replicated index records.

For these reasons, NOREPLICATE is recommended for the SDDS index. Consider using REPLICATE for the SDDS index only if you are not buffering most of the SDDS index through LSR and you are not using a controller cache function for the SDDS.

Defining the SDIDS

The syntax diagram for the DEFINE CLUSTER command for defining an SDIDS follows:

```
DEFINE                              +
  CLUSTER(                          +
    NAME(sdidsname)                 +
    VOLUMES(volser [volser... ])    +
    [INDEXED]                       +
    KEYS(n 0)                       +
```

```
|                 [LOG(NONE)]                     +
|                 SHAREOPTIONS(1 3)               +
|                 [NOREUSE]                         +
                  [UNIQUE])                       +
                DATA(                             +
                  NAME(sdidsname.data)            +
                  [CONTROLINTERVALSIZE(cisize)]   +
                  RECORDSIZE(avg max)             +
                  {RECORDS(primary secondary) | TRACKS(primary secondary) |
                     CYLINDERS(primary secondary)} +
                  [FREESPACE(freeci freeca)]      +
                  [SPEED])                        +
                INDEX(                            +
                  [NAME(sdidsname.index)]         +
                  [NOIMBED]                       +
                  [NOREPLICATE])
```

CLUSTER

Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole. Enclose these parameters in parentheses.

NAME(sdidsname)

Specifies the name of the SDIDS. You must specify NAME. The *sdidsname* is a standard data set name, which is specified on the NAME parameter.

You can use up to 100 VSAM clusters in an SDIDS. If you use more than one cluster for a Tivoli Information Management for z/OS database, you must define each cluster, and they must conform to the following naming conventions. These conventions are the same as those for naming the SDDS.

- Each cluster must have the same name, except for a 3-character sequence:
  - The first character of the 3-character sequence is a trigger character. You can use any valid data-set-name character as the trigger character. You define the trigger character in the BLGCLUST macro, as described under “BLGCLUST Macro — Defining a Database” on page 333. The purpose of this character is to indicate that you are using multiple VSAM clusters in an SDIDS.

    The trigger character does not have to be a unique character in a cluster name. However, since the trigger character indicates the start of the sequence, it must be the first occurrence of that character in a cluster name.

    The trigger character must be identical in each cluster name.
  - The remaining 2 characters in the sequence must be numeric. In the first cluster, these characters must range from 00 to 98. In the remaining clusters, these characters must range from 01 to 99.
  - The numeric characters following the trigger must be in consecutive numerical order for the database that contains the multiple VSAM clusters.
  - The sequence must occur in the same position in each SDIDScluster name.

The following is an example of multiple VSAM clusters that form a 10-cluster SDIDS. The number sign (#) is the trigger character.

```
BLM.SDIDS#01
BLM.SDIDS#02
BLM.SDIDS#03
BLM.SDIDS#04
BLM.SDIDS#05
BLM.SDIDS#06
```

```
BLM.SDIDS#07
BLM.SDIDS#08
BLM.SDIDS#09
BLM.SDIDS#10
```

Note: If you define # as the trigger character and you use more than one # within a name, Tivoli Information Management for z/OS treats only the first # as a trigger character. In this case, Tivoli Information Management for z/OS treats the remaining #s like any other character.

VOLUMES(volser [volser...])
Specifies the volumes to contain the SDIDS. *Volser* is the volume serial number of the device. You must specify one or more devices. This keyword is not required for SMS-managed data sets.

INDEXED
Specifies that the SDIDS is a key-sequenced data set. You do not have to specify this keyword because INDEXED is the default.

KEYS(n 0)
Specifies information about the key field of the SDIDS. Tivoli Information Management for z/OS requires that you store the key of the record in the first *n* bytes of the record. Therefore, the values *n* and 0 represent the length and offset, respectively, of the key. You must specify this keyword and its values. You can specify a key length of 18 or 34.

The 18-byte key is typically used for SBCS data. The 34-byte key is generally used for DBCS data (but can be used for SBCS data), and supports a longer database search argument.

The 18- and 34-byte sizes provide improved database performance and eliminate the need for VSAM spanned records, which are not supported. The 16- and 32-byte keys provided with earlier releases are also no longer supported.

If you use DBCS characters, the 34-byte key is the better choice. An 18-byte key may severely limit the number of searchable characters. With an 18-byte key, the maximum searchable DBCS characters is 7, as compared to 16 SBCS characters. If your searchable data is prefixed, the amount of DBCS data that can be searched is further limited by the length of the prefix. With a prefix of PERS/, the searchable DBCS data is limited to 4 DBCS characters.

If you use SBCS characters, the 18-byte key is the better choice. The smaller key results in a reduction in the size of the SDIDS data component. This saves on DASD space and more importantly, can increase the effectiveness of buffers for the data component by as much as 30% over a 34-byte key.

Both SBCS and DBCS users can use the 18 or 34 key lengths. If you are currently using an 18-byte key, you can redefine your SDIDS to VSAM and run the BLGUT1 utility program to rebuild your SDIDS, or the BLGUT1M utility program to change the key length. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on these utility programs.

LOG(NONE)
Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)
Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE
The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

UNIQUE
Indicates you are allocating the SDIDS as separate DASD space. This keyword is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space. It is recommended that you specify UNIQUE.

Note: If you are using ICF in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA
Specifies attributes of the data component of the SDIDS. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(sdidsname.data)
Specifies the name of the data component of the SDIDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *sdidsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)
Specifies the control interval size for the data component of the SDIDS. The sizes you can specify must be:
- A multiple of 512 for values from 512 to 8192
- A multiple of 2048 for values from 8192 to 32768.

Refer to the calculations for maximum record size above to determine the control interval size for your database.

RECORDSIZE(avg max)
Specifies the average and maximum record sizes for the SDIDS. The maximum record size must be at least 7 bytes less than the control interval size (max≤cisize−7). You must have already calculated the maximum (see “Working with SDIDSs” on page 176).

B. Defining Data Sets

RECORDS(primary secondary)

TRACKS(primary secondary)

CYLINDERS(primary secondary)
Specifies the amount of space to allocate to the SDIDS. *Primary* specifies the initial space allocation; *secondary* specifies the increments of the allocation when new extents are necessary to extend the SDIDS. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required.

FREESPACE(freeci freeca)
Specifies the percentage of freespace to allocate in the data set. *Freeci* is the percentage to allocate for each control interval and *freeca* is the percentage to allocate for each control area.

SPEED
Specifies that the data component's space is not preformatted during initial load. VSAM uses this keyword only when rebuilding the SDIDS through the BLGUT1 rebuild utility, or when reorganizing the SDIDS. The SDIDS rebuild utility performance is improved if you specify SPEED. The SPEED keyword is recommended.

INDEX
Specifies attributes of the index component of the cluster. The attributes follow the INDEX keyword and you must enclose them in parentheses.

NAME(sdidsname.index)
Specifies the name of the index component of the SDIDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *sdidsname.INDEX*, the name is more recognizable, for example, in a LISTVTOC listing.

Using IMBED or NOIMBED
IMBED indicates that the lowest level index records (sequence set) of the data set are written as many times as they fit on the first track adjacent to the related control area. This can shorten the time it takes to retrieve records when physical access to the DASD is required. With NOIMBED, the sequence set is written once and resides with the rest of the index levels.

NOIMBED may be the better choice for the SDIDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the SDIDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so IMBED has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- If you are using a controller cache function, IMBED can be counterproductive. Because entire tracks are buffered into the controller cache, IMBED can cause cache memory in the controller to be used to unnecessarily store the replicated index records.
- If you are using RLS (sysplex mode is enabled), you must use NOIMBED. RLS does not support use of IMBED.

- Starting with DFSMS/MVS Version 1.5, IMBED is no longer supported, and is ignored if specified on the IDCAMS DEFINE. If sysplex mode is not enabled, you may continue to use the old VSAM data sets that were defined with the IMBED attribute.

For these reasons, NOIMBED is recommended for the SDIDS index. Consider using IMBED for the SDIDS index only if you are not buffering most of the SDIDS index through LSR and you are not using a controller cache function for the SDIDS.

Using REPLICATE or NOREPLICATE

REPLICATE indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, REPLICATE can reduce the rotational delay of the disk.

However, NOREPLICATE may be the better choice for the SDIDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the SDIDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so REPLICATE has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- If you are using a controller cache function, REPLICATE may be counterproductive. Because entire tracks are buffered into the controller cache, REPLICATE can cause cache memory in the controller to be used to unnecessarily store the replicated index records.

For these reasons, NOREPLICATE is recommended for the SDIDS index. Consider using REPLICATE for the SDIDS index only if you are not buffering most of the SDIDS index through LSR and you are not using a controller cache function for the SDIDS.

Defining the SDLDS

The syntax diagram for the DEFINE CLUSTER command for defining an SDLDS follows:

```
DEFINE                             +
  CLUSTER(                         +
    NAME(sdldsname)                +
    VOLUMES(volser [volser... ])   +
    NUMBERED                       +
    [LOG(NONE)]                    +
    SHAREOPTIONS(1 3)              +
    [NOREUSE]                          +
    [UNIQUE])                      +
  DATA(                            +
    [NAME(sdldsname.data)]         +
    [CONTROLINTERVALSIZE(cisize)]  +
    RECORDSIZE(max max)            +
    {RECORDS(primary) | TRACKS(primary) |  CYLINDERS(primary)})
```

CLUSTER

Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole; you must enclose these parameters in parentheses.

NAME(sdldsname)
Specifies the name of the SDLDS. You must specify NAME. The *sdldsname* is a standard data set name, which is specified on the NAME parameter.

VOLUMES(volser [volser...])
Specifies the volumes to contain the SDLDS. *Volser* is the volume serial number of the device. You must specify one or more devices. Allocate the SDLDS on different devices from those on which you allocated the SDDS so that, if the disk for the SDDS becomes damaged, the SDLDS is not damaged at the same time. This keyword is not required for SMS-managed data sets.

NUMBERED
Specifies that the SDLDS is a relative-record data set. This keyword is required.

LOG(NONE)
Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)
Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE
The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

UNIQUE
Indicates you are allocating the SDLDS as separate DASD space. This keyword is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space. It is recommended that you specify UNIQUE.

Note: If you are using ICF in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA
Specifies attributes of the data component of the SDLDS. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(sdldsname.data)
Specifies the name of the data component of the SDLDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set

name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *sdldsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)
Specifies the control interval size for the data component of the SDLDS. The sizes you can specify must be:

- A multiple of 512 for values from 512 to 8192
- A multiple of 2048 for values from 8192 to 32768.

The recommended size is the same as the control interval size specified for the SDDS.

RECORDSIZE(max max)
Specifies the size for the SDLDS records. Because this is a NUMBERED data set, the average and maximum record sizes must be the same. The maximum record size must be at least 7 bytes less than the control interval size (max ≤ cisize–7).

The recommended size to specify is cisize–7. This size must match the maximum record size specified for the SDDS.

RECORDS(primary)

TRACKS(primary)

CYLINDERS(primary)
Specifies the amount of space to allocate to the SDLDS. Do not specify a secondary allocation because the secondary extents are allocated immediately when the SDLDS formatting utility is processed. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required.

After you define the SDLDS, you must format the data set by running the BLGUTR utility. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*. You cannot format the data set until the BLX-SP procedure is started. If you format the data set before the BLX-SP procedure is started, you will receive an ABEND.

Defining the DICTDS

The dictionary data set (DICTDS) contains s-words and p-words that you use to create and modify panels with PMF and also to generate reports.

The syntax diagram for the DEFINE CLUSTER command for defining a DICTDS follows:

```
DEFINE                          +
  CLUSTER(                      +
    NAME(dictdsname)            +
    [INDEXED]                   +
    KEYS(3 0)                   +
    [LOG(NONE)]                 +
    SHAREOPTIONS(1 3)           +
    [NOREUSE]                        +
    VOLUMES(volser [volser... ])     +
    [UNIQUE])                   +
  DATA(                         +
    [NAME(dictdsname.data)]     +
    [CONTROLINTERVALSIZE(cisize)]    +
    RECORDSIZE(115 115)         +
    {RECORDS(primary secondary) | TRACKS(primary secondary) |
       CYLINDERS(primary secondary)} +
    [FREESPACE(freeci freeca)]       +
    [SPEED])                    +
```

B. Defining Data Sets

```
INDEX(                            +
  [NAME(dictdsname.index)]        +
  [NOIMBED]                       +
  [NOREPLICATE])
```

CLUSTER
Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole; you must enclose these parameters in parentheses.

NAME(dictdsname)
Specifies the name of the DICTDS. You must specify NAME. The dictdsname is a standard data set name.

INDEXED
Specifies that the DICTDS is a key-sequenced data set. You do not have to specify this keyword because INDEXED is the default.

KEYS(3 0)
Specifies information about the key field of the DICTDS. Tivoli Information Management for z/OS requires that you store the key of the record in the first three bytes of the record. Therefore, the values 3 and 0 represent the length and offset, respectively, of the key. You must specify this keyword and its values.

LOG(NONE)
Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)
Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE
The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

VOLUMES(volser [volser...])
Specifies the volumes to contain the DICTDS. *Volser* is the volume serial number of the device. You must specify one or more devices. This keyword is not required for SMS-managed data sets.

UNIQUE
Indicates you are allocating the dictionary data set as separate DASD space. This

keyword is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space. It is recommended that you specify UNIQUE.

Note: If you are using ICF in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA
Specifies attributes of the data component of the dictionary data set. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(dictdsname.data)
Specifies the name of the data component of the DICTDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *dictdsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)
Specifies the control interval size for the data component of the DICTDS. The sizes you can specify must be:
- A multiple of 512 for values from 512 to 8192
- A multiple of 2048 for values from 8192 to 32768.

The recommended size is 4096.

RECORDSIZE(115 115)
Specifies the average and maximum record sizes for the DICTDS. The size of the dictionary data set records is fixed at 115 bytes. Therefore, the value that you should specify for both average and maximum is 115.

RECORDS(primary secondary)

TRACKS(primary secondary)

CYLINDERS(primary secondary)
Specifies the amount of space to allocate to the DICTDS. *Primary* specifies the initial space allocation; *secondary* specifies the increments of the allocation when new extents are necessary to extend the DICTDS. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required. The dictionary data set as distributed by Tivoli uses approximately 30 tracks for Tivoli Information Management for z/OS on a 3380 DASD device.

FREESPACE(freeci freeca)
Specifies the percentage of freespace to allocate in the data set. *Freeci* is the percentage to allocate for each control interval and *freeca* is the percentage to allocate for each control area. The dictionary has two sections (s-words and p-words). Any s-word or p-word that you add always appends to the end of its respective section. Therefore, it is recommended that you specify *no* freespace for either the control intervals or the control areas. When you add your first s-word or p-word, a control area split occurs for the last control area for the section. Thereafter, when you add more s-words or p-words, they fit in the area that was split.

SPEED
Specifies that the data component's space is not preformatted during initial load. VSAM uses this keyword only when you use the BLGUT5 utility program to

initially load DICTDS or when you are reorganizing the DICTDS. The BLGUT5 utility program performance improves if you specify SPEED. This applies to only the first DICTDS load; subsequent DICTDS loads do not improve performance. The SPEED keyword is recommended.

INDEX
Specifies attributes of the index component of the cluster. The attributes follow the INDEX keyword and you must enclose them in parentheses.

NAME(dictdsname.index)
Specifies the name of the index component of the DICTDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *dictdsname.INDEX*, the name is more recognizable, for example, in a LISTVTOC listing.

Using IMBED or NOIMBED
IMBED indicates that the lowest level index records (sequence set) of the data set are written as many times as they fit on the first track adjacent to the related control area. This can shorten the time it takes to retrieve records when physical access to the DASD is required. With NOIMBED, the sequence set is written once and resides with the rest of the index levels.

NOIMBED may be the better choice for the DICTDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the DICTDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so IMBED has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- Normally, you would not use a controller cache function for the DICTDS, but if you were, IMBED can be counterproductive. Because entire tracks are buffered into the controller cache, IMBED can cause cache memory in the controller to be used to unnecessarily store the replicated index records.
- If you are using RLS (sysplex mode is enabled), you must use NOIMBED. RLS does not support use of IMBED.
- Starting with DFSMS/MVS Version 1.5, IMBED is no longer supported, and is ignored if specified on the IDCAMS DEFINE. If sysplex mode is not enabled, you may continue to use the old VSAM data sets that were defined with the IMBED attribute.

For these reasons, NOIMBED is recommended for the DICTDS index. Consider using IMBED for the DICTDS index only if you are not buffering most of the DICTDS index through LSR.

Using REPLICATE or NOREPLICATE
REPLICATE indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, REPLICATE can reduce the rotational delay of the disk.

However, NOREPLICATE may be the better choice for the DICTDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the DICTDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so REPLICATE has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- Normally, you would not use a controller cache function for the DICTDS, but if you were, REPLICATE may be counterproductive. Because entire tracks are buffered into the controller cache, REPLICATE can cause cache memory in the controller to be used to unnecessarily store the replicated index records.

For these reasons, NOREPLICATE is recommended for the DICTDS index. Consider using REPLICATE for the DICTDS index only if you are not buffering most of the DICTDS index through LSR.

After you define the dictionary (DICTDS), you must initialize the data in the dictionary by copying the supplied dictionary from a PDS into the VSAM dictionary. Use the BLGUT5 utility program to do this. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for information on using BLGUT5.

Defining the RPANLDS

You can use any number of read panel data sets, but, with more data sets, Tivoli Information Management for z/OS must search the multiple data sets for the panels. This activity can affect performance, depending upon the value used in the PNLBCNT parameter on the BLGPARMS macro. See "Using One or More Read Panel Data Sets" on page 198 for additional information.

The size of your RPANLDS depends on the number of panels that you store in it and the average size of a panel. You can choose whatever record size and control interval size you want.

The panels that you create or modify are probably similar to the Tivoli distributed panels. Therefore, you should use the same control interval and record sizes when defining your read panel data sets.

The syntax diagram for the DEFINE CLUSTER command for defining an RPANLDS follows:

```
DEFINE                              +
  CLUSTER(                          +
    NAME(rpanldsname)               +
    [INDEXED]                       +
    KEYS(10 0)                      +
|   [LOG(NONE)]                     +
|   SHAREOPTIONS(1 3)               +
|   [NOREUSE]                            +
    VOLUMES(volser [volser... ])    +
    [UNIQUE])                       +
  DATA(                             +
    [NAME(rpanldsname.data)]        +
    [CONTROLINTERVALSIZE(cisize)]   +
    RECORDSIZE(avg max)             +
    {RECORDS(primary secondary) | TRACKS(primary secondary) |
       CYLINDERS(primary secondary)} +
    [FREESPACE(freeci freeca)]      +
    [SPEED])                        +
```

```
INDEX(                          +
  [NAME(rpanldsname.index)]     +
  [NOIMBED]                     +
  [NOREPLICATE])
```

CLUSTER
Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole; you must enclose these parameters in parentheses.

NAME(rpanldsname)
Specifies the name of the RPANLDS. You must specify NAME. The rpanldsname is a standard data set name.

INDEXED
Specifies that the RPANLDS is a key-sequenced data set. You do not have to specify this keyword because INDEXED is the default.

KEYS(10 0)
Specifies information about the key field of the RPANLDS. Tivoli Information Management for z/OS requires that you store the key of the record in the first 10 bytes of the record. Therefore, the values 10 and 0 represent the length and offset, respectively, of the key. You must specify this keyword and its values.

LOG(NONE)
Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)
Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE
The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

VOLUMES(volser [volser...])
Specifies the volumes to contain the RPANLDS. *Volser* is the volume serial number of the device. You must specify one or more devices. This keyword is not required for SMS-managed data sets.

UNIQUE
Indicates you are allocating the RPANLDS as separate DASD space. This keyword

is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space. It is recommended that you specify UNIQUE.

Note: If you are using ICF in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA

Specifies attributes of the data component of the RPANLDS. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(rpanldsname.data)

Specifies the name of the data component of the RPANLDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *rpanldsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)

Specifies the control interval size for the data component of the RPANLDS. The recommended size for the shipped Tivoli base panel data set is 4096.

RECORDSIZE(avg max)

Specifies the average and maximum record sizes for the RPANLDS.

- The average size of panels shipped by Tivoli is 900.
- The recommended maximum size for the shipped Tivoli base panels is 4089.

The maximum record size must be:

- At least 7 bytes less than the control interval size (max≤cisize–7)
- Greater than or equal to 80.

RECORDS(primary secondary)

TRACKS(primary secondary)

CYLINDERS(primary secondary)

Specifies the amount of space to allocate to the RPANLDS. *Primary* specifies the initial space allocation; *secondary* specifies the increments of the allocation when new extents are necessary to extend the RPANLDS. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required.

FREESPACE(freeci freeca)

Specifies the percentage of freespace to allocate in the data set. *Freeci* is the percentage to allocate for each control interval and *freeca* is the percentage to allocate for each control area.

Normally, freespace offers little value for panel data sets.

SPEED

Specifies that the data component's space is not preformatted during initial load. This keyword applies only when you are using the Tivoli Information Management for z/OS BLGUT6 utility program to initially load the RPANLDS, or when you are reorganizing the RPANLDS. The BLGUT6 utility program performance improves if you specify SPEED. This applies to only the first RPANLDS load; subsequent RPANLDS loads do not improve performance. The SPEED keyword is recommended.

INDEX
Specifies attributes of the index component of the cluster. The attributes follow the INDEX keyword and you must enclose them in parentheses.

NAME(rpanldsname.index)
Specifies the name of the index component of the RPANLDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *rpanldsname.INDEX*, the name is more recognizable, for example, in a LISTVTOC listing.

Using IMBED or NOIMBED
IMBED indicates that the lowest level index records (sequence set) of the data set are written as many times as they fit on the first track adjacent to the related control area. This can shorten the time it takes to retrieve records when physical access to the DASD is required. With NOIMBED, the sequence set is written once and resides with the rest of the index levels.

NOIMBED may be the better choice for the RPANLDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the RPANLDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so IMBED has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- Normally, you would not use a controller cache function for the RPANLDS, but if you were, IMBED can be counterproductive. Because entire tracks are buffered into the controller cache, IMBED can cause cache memory in the controller to be used to unnecessarily store the replicated index records.
- If you are using RLS (sysplex mode is enabled), you must use NOIMBED. RLS does not support use of IMBED.
- Starting with DFSMS/MVS Version 1.5, IMBED is no longer supported, and is ignored if specified on the IDCAMS DEFINE. If sysplex mode is not enabled, you may continue to use the old VSAM data sets that were defined with the IMBED attribute.

For these reasons, NOIMBED is recommended for the RPANLDS index. Consider using IMBED for the RPANLDS index only if you are not buffering most of the RPANLDS index through LSR.

Using REPLICATE or NOREPLICATE
REPLICATE indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, REPLICATE can reduce the rotational delay of the disk.

However, NOREPLICATE may be the better choice for the RPANLDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the RPANLDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so REPLICATE has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).

- Normally, you would not use a controller cache function for the RPANLDS, but if you were, REPLICATE may be counterproductive. Because entire tracks are buffered into the controller cache, REPLICATE can cause cache memory in the controller to be used to unnecessarily store the replicated index records.

For these reasons, NOREPLICATE is recommended for the RPANLDS index. Consider using REPLICATE for the RPANLDS index only if you are not buffering most of the RPANLDS index through LSR.

After you define the RPANLDS for your production database, you must copy the Tivoli panels from a PDS into this VSAM RPANLDS. You do this by processing the BLGUT6 utility program. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.

If you are defining an RPANLDS to contain your own panels, no such initialization is necessary.

Defining the WPANLDS

The write panel data set (WPANLDS) is similar to the RPANLDS in that the format of the data is the same. The size of your WPANLDS depends on the number of panels you expect it to contain at any one time. Remember that you must use your WPANLDS only for modifying and testing panels using the PMF facility of Tivoli Information Management for z/OS. After you fully test your new panels, copy the panels into your production RPANLDS. Then you can delete the panels in your WPANLDS.

The syntax diagram for the DEFINE CLUSTER command for defining an WPANLDS follows:

```
DEFINE                              +
  CLUSTER(                          +
    NAME(wpanldsname)               +
    [INDEXED]                       +
    KEYS(10 0)                      +
    [LOG(NONE)]                     +
    SHAREOPTIONS(1 3)               +
    [NOREUSE]                           +
    VOLUMES(volser [volser...])     +
    [UNIQUE])                       +
  DATA(                             +
    [NAME(wpanldsname.data)]        +
    [CONTROLINTERVALSIZE(cisize)]   +
    RECORDSIZE(avg max)             +
    {RECORDS(primary secondary) | TRACKS(primary secondary) |
       CYLINDERS(primary secondary)} +
    [FREESPACE(freeci freeca)])     +
  INDEX(                            +
    [NAME(wpanldsname.index)]       +
    [NOIMBED]                       +
    [NOREPLICATE])
```

CLUSTER
Indicates that you are defining a VSAM cluster. Follow CLUSTER with the parameters specified for the cluster as a whole; you must enclose these parameters in parentheses.

B. Defining Data Sets

NAME(wpanldsname)
Specifies the name of the WPANLDS. You must specify NAME. The wpanldsname is a standard data set name.

INDEXED
Specifies that the WPANLDS is a key-sequenced data set. You do not have to specify this keyword because INDEXED is the default.

KEYS(10 0)
Specifies information about the key field of the WPANLDS. Tivoli Information Management for z/OS requires that you store the key of the record in the first ten bytes of the record. Therefore, the values 10 and 0 represent the length and offset, respectively, of the key. You must specify this keyword and its values.

LOG(NONE)
Specifies that CICS should not log changes (the file is non-recoverable). You must include LOG(NONE) if you are using VSAM RLS in a sysplex. RLS requires use of a LOG parameter.

SHAREOPTIONS(1 3)
Specifies how your VSAM cluster can be shared among users. You must specify a value for this keyword. In a non-sysplex environment, the value must be SHAREOPTIONS(1 3).

If you specify SHAREOPTIONS(1 3), you specify that the cluster can be used by only the BLX-SP for read and write processing. VSAM ensures complete data integrity for the cluster.

If sysplex mode is enabled, a value of SHAREOPTIONS(2 3) is recommended. Although the value is not required and is ignored by VSAM RLS, SHAREOPTIONS(2 3) enables other programs such as Access Method Services (IDCAMS) to perform tasks such as opening the data set in NSR read-only mode.

Note: The SHAREOPTIONS value of existing Tivoli Information Management for z/OS VSAM clusters may be changed using the AMS ALTER command.

NOREUSE
The REUSE option is not supported. When defining clusters, either specify NOREUSE explicitly, or allow VSAM to default to NOREUSE by not specifying either option.

VOLUMES(volser [volser...])
Specifies the volumes to contain the WPANLDS. *Volser* is the volume serial number of the device. You must specify one or more devices. This keyword is not required for SMS-managed data sets.

UNIQUE
Indicates you are allocating the WPANLDS as separate DASD space. This keyword is optional. If you omit it, VSAM defaults to SUBALLOCATION, which allocates the space from existing VSAM data space. It is recommended that you specify UNIQUE.

Note: If you are using ICF in OS/390 MVS, the UNIQUE and SUBALLOCATION parameters are ignored.

DATA
Specifies attributes of the data component of the WPANLDS. The attributes follow the DATA keyword, and you must enclose them in parentheses.

NAME(wpanldsname.data)
Specifies the name of the data component of the WPANLDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *wpanldsname.DATA*, the name is more recognizable, for example, in a LISTVTOC listing.

CONTROLINTERVALSIZE(cisize)
Specifies the control interval size for the data component of the WPANLDS. The sizes you can specify must be:

- A multiple of 512 for values from 512 to 8192
- A multiple of 2048 for values from 8192 to 32768.

The recommended size is the same size that you used to define your read panel data sets.

RECORDSIZE(avg max)
Specifies the average and maximum record sizes for the WPANLDS. The maximum record size must be:

- At least 7 bytes less than the control interval size (max≤cisize–7)
- Greater than or equal to 80.

The recommended sizes are the same values that you used to define your read panel data sets.

RECORDS(primary secondary)

TRACKS(primary secondary)

CYLINDERS(primary secondary)
Specifies the amount of space to allocate to the WPANLDS. *Primary* specifies the initial space allocation; *secondary* specifies the increments of the allocation when new extents are necessary to extend the WPANLDS. You can specify the allocation units in terms of RECORDS, TRACKS, or CYLINDERS. One of these keywords is required.

FREESPACE(freeci freeca)
Specifies the percentage of freespace to allocate in the data set. *Freeci* is the percentage to allocate for each control interval and *freeca* is the percentage to allocate for each control area.

INDEX
Specifies attributes of the index component of the cluster. The attributes follow the INDEX keyword and you must enclose them in parentheses.

NAME(wpanldsname.index)
Specifies the name of the index component of the WPANLDS VSAM cluster. This keyword is optional. If you omit the name, VSAM automatically generates a data set name for you. Refer to *DFSMS/MVS Using Data Sets* for more information on system-generated names. If you specify a name like *wpanldsname.INDEX*, the name is more recognizable, for example, in a LISTVTOC listing.

Using IMBED or NOIMBED

IMBED indicates that the lowest level index records (sequence set) of the data set are written as many times as they fit on the first track adjacent to the related control area. This can shorten the time it takes to retrieve records when physical access to the DASD is required. With NOIMBED, the sequence set is written once and resides with the rest of the index levels.

Whether you use IMBED or NOIMBED for the WPANLDS index probably makes very little difference to performance, since the WPANLDS is usually not heavily used.

- If you are using LSR and have allotted enough buffers to contain all of the WPANLDS index CIs, then after the first access there are no physical reads of the index from the DASD, so IMBED has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- Normally, you would not use a controller cache function for the WPANLDS, but if you were, IMBED can be counterproductive. Because entire tracks are buffered into the controller cache, IMBED can cause cache memory in the controller to be used to unnecessarily store the replicated index records.
- Starting with DFSMS/MVS Version 1.5, IMBED is no longer supported, and is ignored if specified on the IDCAMS DEFINE. If sysplex mode is not enabled, you may continue to use the old VSAM data sets that were defined with the IMBED attribute.

NOIMBED is recommended for the WPANLDS index. Consider using IMBED for the WPANLDS index only if performance for PMF users is important and you are unable to buffer most of the WPANLDS index through LSR.

Using REPLICATE or NOREPLICATE

REPLICATE indicates that each index (for all levels, not just the sequence set) record is written on a track as many times as it fits. When VSAM reads the DASD to retrieve this lowest level of the index to locate a record, REPLICATE can reduce the rotational delay of the disk.

However, NOREPLICATE may be the better choice for the WPANLDS index for the following reasons:

- If you are using LSR and have allotted enough buffers to contain all of the WPANLDS index CIs (as recommended), then after the first access there are no physical reads of the index from the DASD, so REPLICATE has little value to offset its costs (extra DASD space, and extra time to write changes to the replicated indexes).
- Normally, you would not use a controller cache function for the WPANLDS, but if you were, REPLICATE may be counterproductive. Because entire tracks are buffered into the controller cache, REPLICATE can cause cache memory in the controller to be used to unnecessarily store the replicated index records.

NOREPLICATE is recommended for the WPANLDS index. Consider using REPLICATE for the WPANLDS index only if performance for PMF users is important and you are unable to buffer most of the WPANLDS index through LSR.

C

Defining VSAM Resources to the BLX-SP in a Non-Sysplex Environment

Non-Sysplex Environment

The information in this chapter applies to you if you are installing Tivoli Information Management for z/OS in a non-sysplex environment only. If you are installing Tivoli Information Management for z/OS in a sysplex, there is no need to define VSAM resources to the BLX-SP because the BLX-SP does not manage those resources in a sysplex.

This appendix provides the following:

- A procedure for defining a VSAM resource definition member
- Syntax and examples of the following Tivoli Information Management for z/OS macros:
  - BLXNSR
  - BLXDSN
  - BLXGEN
- An example of how to define an LSR buffer pool using the VSAM BLDVRP macro
- A sample VSAM resource definition member for Tivoli Information Management for z/OS

This appendix assumes that you have a working knowledge of the VSAM BLDVRP macro. For more information on this macro, refer to the *DFSMS/MVS Macro Instructions for Data Sets*.

This appendix is designed to help you define VSAM resources to the BLX-SP.

Defining a VSAM Resource Definition Member

Programming Interface information

Follow these steps to define VSAM resources to the BLX-SP in a non-sysplex environment:

1. Create a BLX-SP VSAM resource definition member. You must define a VSAM resource definition member for your BLX-SP whether you use NSR or LSR. Multiple BLX-SPs can share a VSAM resource definition member.
   a. Use the BLXNSR macro to identify the number of concurrent users (VSAM placeholders) required for the access of VSAM data sets using NSR. This must be the first macro coded in the VSAM resource definition member.
   b. Use the BLXDSN macro to identify the names of the VSAM data sets and:

- The shared resource pool that the data set uses (LSR keyword), or
- Default to NSR (omit LSR keyword).

c. Use the BLXGEN macro to identify the end of the BLX-SP VSAM resource definition. This must be the last BLX macro coded in the VSAM resource definition member. The BLXGEN macro must precede all BLDVRP macros.

d. Include your VSAM BLDVRP macro specifications that define the LSR pools. Do this step only if you are using LSR.

2. Assemble and link-edit the BLX-SP VSAM resource definition member. You must link-edit this member with the following attributes:
   RMODE=24
   NORENT
   REUS

   If you do not use these attributes, a failure occurs when you attempt to initialize the BLX-SP.

   You can link-edit the VSAM resource definition member into your program library that contains the rest of the BLX-SP executable code. You must assemble and link-edit all VSAM resource definition members before attempting to start the BLX-SP. Refer to the *Assembler H Version 2 Language Reference* or the *High Level Assembler Language Reference* for information about assembling modules.

   Note: If you are migrating from an earlier version of the product,, you do not have to assemble and link-edit the VSAM resource definition members again unless you had SHARE=YES specified on one or more BLXDSN macros. The SHARE keyword is no longer supported.

3. Specify the BLX-SP VSAM resource definition member name in the BLX-SP parameters member. The VSAM resource definition member name can be any valid data set member name. The BLX-SP parameters member is processed during initialization. The VSAMRESOURCES keyword identifies the member name containing the LSR pool definitions, NSR placeholders, and connections between these resources and VSAM data sets.

Note: See the sample in “Defining BLX-SP Parameters Members” on page 343.

Understanding the Macro Syntax Description

The syntax diagrams for the Tivoli Information Management for z/OS macros in this chapter show some parameters with brackets *[ ]*, braces *{ }*, or vertical bars |. These are for clarification purposes only; they are not part of the macros.

- Brackets indicate optional data.
- Braces indicate that you must select exactly one of the choices enclosed within the braces.
- Vertical bars indicate selectable items. You must choose one of these items.
- **Bold, underlined values** in these macro descriptions are the default values.

BLXNSR Macro — Specifying the Number of Nonshared Resource Placeholders

The BLXNSR macro specifies the number of VSAM placeholders to allocate for all VSAM data sets that use nonshared resources. The value you specify for the PLACES keyword defines the number of VSAM placeholders to allocate for these VSAM data sets. All of the VSAM data sets that use nonshared resources share these placeholders.

You cannot change the number of placeholders during processing of the BLX-SP. If the value you initially specified for the PLACES keyword is not adequate, you must stop the BLX-SP and restart it with an updated VSAM resource definition member.

The syntax for the BLXNSR macro is:

```
BLXNSR   PLACES=value
```

PLACES

Specifies the number of placeholders to allocate for all VSAM data sets that use nonshared resources. The value you specify must be numeric and in the range of 15–255. This includes data sets not defined in the VSAM definition member, which default to using NSR (see note, page 304).

NSR placeholders are required when data sets are accessed in load mode, even if you specify LSR for the data sets.

Coding the BLX-SP BLXNSR Macro

In the example that is described in “Defining LSR Buffer Pools” on page 80, three data sets use nonshared resources. These data sets are allocated for write access when used by the users’ address spaces. Because only one user’s address space can have write access to a data set at a time, only three placeholders are required (one position per data set). Because NSR placeholders are required to access data sets in load mode and when single user access is requested by some utilities, additional NSR placeholders may be required. Code the BLXNSR macro as shown in Figure 24.

```
*
*     DEFINE NON-SHARED RESOURCES TO BLX-SP
*
         BLXNSR  PLACES=20               Twenty places for NSR.
```

Figure 24. Coding the BLX-SP BLXNSR Macro

BLXDSN Macro — Connecting VSAM Resources to a VSAM Data Set

You can use the BLXDSN macro to connect VSAM resources to a VSAM data set and to assign a logical name for the data set. You can use this name to reference the data set by using operator commands.

You must specify one BLXDSN macro for each data set you want to make a connection to or assign a logical name to. If you want to use LSR with the clusters in a multiple-cluster SDDS, you must specify the BLXDSN macro for each cluster. You can specify a shared resource pool for the data set by using the LSR keyword. You must specify this keyword if you define LSRs for the data set and intend to use LSR to access the data set. If you omit the LSR keyword, nonshared resources are used to access the data set.

The maximum number of concurrent positions that you can have within the data set is determined either by the shared resource pool definition identified with the LSR keyword or by the value you specify for the PLACES keyword in the BLXNSR macro. If you do not specify the LSR keyword, the identified data set uses the nonshared resource placeholder pool defined by the BLXNSR macro.

Note: You do not need to code the BLXDSN macro for key-sequenced data sets that use nonshared resources for data set access. Data sets allocated by a user's address space for which there is no data set definition are automatically allocated by using nonshared resources for data set access. This automatic process also generates a logical name for the data set.

The syntax for the BLXDSN macro is:

```
[label]   BLXDSN   DSN=dsname
                   [,LSR={lsrpoolid|(lsrdataid,lsrindexid)}]
                   [,DSORG={NUM|KEY}]
                   [,USERBUF={n|0}]
```

Note: The SHARE keyword is no longer supported.

label A standard assembler language instruction label. The label you specify must not begin with the characters BL or SYS. This label is optional. If you specify this label, the BLX-SP assigns it as the logical name for the data set. If you do not specify this label, the BLX-SP generates a logical name for the data set. Although this label is optional, it is recommended that you use one. You can reference the data set using this label on operator commands.

DSN Name of the key-sequenced VSAM data set. You must use a cataloged data set and specify 1–44 characters for its name.

This keyword is required.

LSR Indicates whether this data set uses a VSAM LSR pool. You may not want to specify LSR for relative record data sets because it does not provide any performance improvements for them. If you do not specify this keyword, then NSR will be used with this data set.

lsrpoolid
Identifies the BLDVRP macro that defines the VSAM shared resource pool containing buffers to use for both the data and index components of this data set. The value you specify for *lsrpoolid* corresponds to the value that you specified for the SHRPOOL keyword in the VSAM BLDVRP macro that defines the shared resource pool for the data set index and data component buffers.

lsrdataid
Identifies the BLDVRP macro that defines the VSAM shared resource pool containing buffers to use for the data component of this data set. The value you specify for *lsrdataid* corresponds to the value that you specified for the SHRPOOL keyword in the VSAM BLDVRP macro that defines the shared resource pool for the data set data component buffers.

lsrindexid
Identifies the BLDVRP macro that defines the VSAM shared resource pool containing buffers to use for the index component of this data set. The value you specify for *lsrindexid* corresponds to the value that you specified for the

SHRPOOL keyword in the VSAM BLDVRP macro that defines the shared resource pool for the data set index component buffers.

DSORG

Identifies the organization of the VSAM data set. Key-sequenced organization is the default if you do not code the DSORG keyword.

NUM Identifies the organization of the data set as a VSAM relative-record data set. The SDLDS is a VSAM relative-record data set.

KEY Identifies the organization of the data set as a VSAM key-sequenced data set. The SDDS, SDIDS, dictionary, and panel data sets are VSAM key-sequenced data sets.

USERBUF=n

Specifies the number (*n*) of temporary buffers the BLX-SP should maintain for this data set. The value of *n* ranges from 0 to 32767. The default value is 0.

A value of 0 means that storage for a temporary user buffer is obtained and released from the system for every I/O processing request to this data set from a user. This can cause minor performance degradation.

A value greater than 0 means that the BLX-SP maintains up to the specified number of buffers allocated and assigns them to users, reducing system storage obtain/release requests, but potentially increasing the amount of storage required for the BLX-SP.

The size of each buffer is the maximum LRECL of the data set. Buffers are not allocated until they are needed, so even if 50 buffers are specified, if the maximum ever used is 10, only 10 are allocated. Because these buffers are generally only used for brief periods of time during an I/O processing request, the number of buffers needed is much smaller than the number of active users. If at any time, more buffers are needed than are available in the BLX-SP, additional buffers are obtained from the system. As I/O processing requests complete and buffers become available again, if the number of available buffers exceeds the value specified for this keyword, the excess buffer storage is returned to the system.

This keyword is optional.

Coding the BLX-SP BLXDSN Macros

Using the data from the example in “Defining LSR Buffer Pools” on page 80, code the BLXDSN macros as shown in Figure 25 on page 306.

```
***********************************************************************
*     DEFINE VSAM DATA SETS TO BLX-SP AND CONNECT VSAM RESOURCES      *
***********************************************************************
*
*     DEFINE DATA SETS USING LOCAL SHARED RESOURCES (LSR)
*
BASEPNS  BLXDSN DSN=BLM.IBMPNLS,                                        X
               LSR=(2,2)
READPNS  BLXDSN DSN=BLM.READ.PANELS,                                    X
               LSR=(2,2)
DICT     BLXDSN DSN=BLM.DICT,                                           X
               LSR=(2,2)
MGMTSDDS BLXDSN DSN=BLM.SDDS,                                           X
               LSR=(0,0)
MGMTINDX BLXDSN DSN=BLM.SDIDS,                                          X
               LSR=(1,1)
TESTPNS  BLXDSN DSN=BLM.TEST.PANELS,                                    X
               LSR=(2,2)
TESTSDDS BLXDSN DSN=BLM.TEST.SDDS,                                      X
               LSR=(3,3)
TESTINDX BLXDSN DSN=BLM.TEST.SDIDS,                                     X
               LSR=(4,4)
*
*     DEFINE DATA SETS USING NON-SHARED RESOURCES (NSR)
*
MGMTLOG  BLXDSN  DSN=BLM.SDLDS,DSORG=NUM
JILLS    BLXDSN  DSN=BLM.JILLS.PANELS
JACKS    BLXDSN  DSN=BLM.JACKS.PANELS
```

Figure 25. Coding the BLX-SP BLXDSN Macro

BLXGEN Macro — Generating the BLX-SP VSAM Resource Definition Member CSECT

The BLXGEN macro indicates the end of the BLX-SP VSAM resource definition macro set. This macro begins the code-generation phase of the VSAM resource definition member CSECT (assembler language). This macro must follow all other BLX-SP VSAM resource definition macros, but must precede the list forms of the VSAM BLDVRP macros used to define one or more LSR pools.

The syntax for the BLXGEN macro is:

```
BLXGEN
```

The BLXGEN macro has no parameters.

BLDVRP Macro—Defining Buffer Pools

You can define LSR buffer pools using the VSAM BLDVRP macro.

VSAM BLDVRP Macro Keyword Considerations

Consider the following points when coding the VSAM BLDVRP macro:

- You cannot use VSAM global shared resources (GSR) because the BLX-SP environment does not support them. If you specify GSR in the BLDVRP macro, the BLX-SP uses LSR for the data set.
- The MF keyword of the BLDVRP macro must be specified as MF=L to define a list form of the macro so that it can be run when the VSAM resource definition member is processed.

- The MODE keyword of the BLDVRP macro must be omitted or coded as MODE=24. If MODE=31 is specified, the VSAM resource definition member is rejected.
- The value of the RMODE31 keyword in the BLDVRP macro is ignored and replaced with a value of ALL when the BLDVRP macro is run.
- The VSAM resource definition member must be link-edited with the following attribute. Otherwise, the VSAM resource definition member is rejected:
  RMODE=24
  NORENT
  REUS

  Note: RMODE, NORENT, and REUS represent relocation mode, nonreentrant, and reusable, respectively.

  STRNO defines the number of placeholders for the buffer pool. See "Using VSAM Placeholders" on page 77 for guidelines.

Note: Refer to *DFSMS/MVS Macro Instructions for Data Sets* for a description of the VSAM BLDVRP macro. Also refer to *DFSMS/MVS Using Data Sets* for additional information about using the VSAM BLDVRP macro.

Coding the VSAM BLDVRP Macros

Figure 26 through Figure 30 are examples based on the analysis and assumptions made in the example given in "Defining LSR Buffer Pools" on page 80. The following guidelines are used in coding the BLDVRP macro:

Note: Remember, if a buffer of the exact control interval size is not available, VSAM uses the next larger buffer size available.

Use separate data and index pools for key-sequenced data sets so that later changes in data set definitions do not affect index buffer pools, and each buffer pool is used for its intended purpose.

In addition, use an LSR pool to define the placeholders required for the production database user group. Define the first LSR pool for the production database SDDS. This resource pool has both a data and an index pool defined. Define the second LSR pool for the production database SDIDS. This resource pool has both a data and index pool defined. Define the third LSR pool for the read panel and dictionary data sets. The data pool requires separate placeholders defined for the:

- Dictionary data components
- Production base panel data component
- Production read panel data component
- Development test panel data component

The index pool requires separate placeholders defined for the:
- Dictionary index components
- Production base panel index component
- Production read panel index component
- Development test panel index component

The minimum number of buffers in a buffer pool that can be allocated for the data component of any VSAM data set is the number of positions that are maintained in the shared resource pool plus one.

Define the first LSR pool for the production database SDDS. It defines the resources required for the production database user group. Figure 26 shows how to define this pool.

```
***********************************************************************
*     RESOURCE POOL ZERO (FOR PRODUCTION DATABASE SDDS)               *
***********************************************************************
*
*        RESOURCE POOL ZERO (SDDS DATA BUFFER POOLS)
*
LSRD0  BLDVRP  BUFFERS=(4096(31)),      SDDS DATA BUFFERS             X
               KEYLEN=7,                POOL USED BY SDDS             X
               STRNO=30,                MAINTAIN 30 POSITIONS         X
               SHRPOOL=0,               IDENTIFY POOL                 X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY             X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
*
*        RESOURCE POOL ZERO (SDDS INDEX BUFFER POOLS)
*
LSRI0  BLDVRP  BUFFERS=(2048(53)),      SDDS INDEX BUFFERS AND        X
               SHRPOOL=0,               POOL IDENTIFIER               X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY            X
               RMODE31=ALL,                                           X
               MODE=24,                                               X
               MF=L
```

Figure 26. Defining Resource Pool 0: Production Database SDDS

Define the second LSR pool for the production database SDIDS. It defines the resources required for the production database user group. The example uses a 34-byte key. You can use an 18-byte key. The key size you use depends on the key size you specified when you created your SDIDS. See “SDIDS Keys” on page 42 for information on SDIDS key size. Figure 27 on page 309 shows how to define this pool.

```
**********************************************************************
*     RESOURCE POOL ONE (FOR PRODUCTION DATABASE SDIDS)              *
**********************************************************************
*
*        RESOURCE POOL ONE (SDIDS DATA BUFFER POOLS)
*
LSRD1  BLDVRP  BUFFERS=(2048(500)),     SDIDS DATA BUFFERS + PAD       X
               KEYLEN=34,               POOL USED BY SDIDS             X
               STRNO=45,                MAINTAIN 45 POSITIONS          X
               SHRPOOL=1,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL ONE (SDIDS INDEX BUFFER POOLS)
*
LSRI1  BLDVRP  BUFFERS=(3584(34)),      SDIDS INDEX BUFFERS            X
               SHRPOOL=1,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
```

Figure 27. Defining Resource Pool 1: Production Database SDIDS

Define the third LSR pool for the dictionary and read panel data sets. It defines the resources required for both the production and the test and development user groups. The resource pool requires separate placeholders defined for the:

- Common base panel data set data component
- Common dictionary data set data component
- Production read panel data set data component
- Test and development test panel data set data component
- Common base panel data set index component
- Common dictionary data set index component
- Production read panel data set index component
- Test and development test panel data set index component.

Figure 28 on page 310 shows how to define this pool.

```
*********************************************************************
*     RESOURCE POOL TWO  (PANEL/DICTIONARY DATA SETS)               *
*********************************************************************
*
*        RESOURCE POOL TWO (PANEL/DICTIONARY DATA BUFFER POOLS)
*
LSRD2  BLDVRP  BUFFERS=(4096(36)),      DICT/PANEL DATA BUFFERS        X
               KEYLEN=10,               POOL USED BY DICT/PANELS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               STRNO=35,                COMBINED POSITIONS REQUIRED    X
               TYPE=(LSR,DATA),         INDEX AND DATA POOL            X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL TWO (PANEL/DICTIONARY INDEX BUFFER POOLS)
*
LSRI2  BLDVRP  BUFFERS=(1536(30)),      DICT/PANEL INDEX BUFFERS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
```

Figure 28. Defining Resource Pool 2: Panel and Dictionary Data Sets

Define the fourth LSR pool for the test and development database SDDS. It defines the resources required for the test and development database user group. Figure 29 shows how to define this pool.

```
*********************************************************************
*     RESOURCE POOL THREE (TEST/DEVELOPMENT DATABASE SDDS)          *
*********************************************************************
*
*        RESOURCE POOL THREE (SDDS DATA BUFFER POOLS)
*
LSRD3  BLDVRP  BUFFERS=(4096(6)),       SDDS DATA BUFFERS              X
               KEYLEN=7,                POOL USED BY SDDS              X
               STRNO=20,                MAINTAIN 20 POSITIONS          X
               SHRPOOL=3,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL THREE (SDDS INDEX BUFFER POOLS)
*
LSRI3  BLDVRP  BUFFERS=(512(5)),        SDDS INDEX BUFFERS +PAD        X
               SHRPOOL=3,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
```

Figure 29. Defining Resource Pool 3: Test and Development Database SDDS

Define the fifth LSR pool for the test and development database SDIDS. It defines the resources required for the test and development database user group. Figure 30 shows how to define this pool.

Note: If you are using a multiple-cluster SDDS or SDIDS, you need one BLXDSN macro per SDDS or SDIDS in a cluster. You cannot simply put the data set numbered #01 in the VSAM resource definition member.

```
***********************************************************************
*     RESOURCE POOL FOUR (TEST/DEVELOPMENT DATABASE SDIDS)            *
***********************************************************************
*
*        RESOURCE POOL FOUR (SDIDS DATA BUFFER POOLS)
*
LSRD4  BLDVRP  BUFFERS=(2048(50)),      SDIDS DATA BUFFERS + PAD       X
               KEYLEN=34,               POOL USED BY SDIDS             X
               STRNO=20,                MAINTAIN 20 POSITIONS          X
               SHRPOOL=4,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL FOUR (SDIDS INDEX BUFFER POOLS)
*
LSRI4  BLDVRP  BUFFERS=(512(3)),        SDIDS INDEX BUFFERS +PAD       X
               SHRPOOL=4,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
```

Figure 30. Defining Resource Pool 4: Test and Development Database SDIDS

Sample VSAM Resource Definition Member—BLXVDEF

Figure 31 on page 312 combines the previous steps to define the VSAM resource definition member to the BLX-SP for use in a non-sysplex environment. This example is called BLXVDEF in the SBLMSAMP library. Refer to your assembler language documentation for information on working with the assembler language (see the Bibliography).

```
//BLXVDEF JOB
//*
//*******************************************************************
//*                                                                 *
//* THIS JOB IS USED TO DEFINE, ASSEMBLE AND LINKEDIT THE VSAM      *
//* DEFINITION FOR BLX-SP.                                          *
//*                                                                 *
//* THE FUNCTION OF THIS JOB IS TO:                                 *
//*     1. SPECIFY VSAM DATA SETS USING LSR POOLS AND CONNECT THE   *
//*        RESOURCE POOL TO THE DATA SET.                           *
//*     2. SPECIFY VSAM DATA SETS USING NON-SHARED RESOURCES.       *
//*     3. SPECIFY THE NUMBER NON-SHARED RESOURCE PLACEHOLDERS.     *
//*     4. SPECIFY THE VSAM BLDVRP USED TO BUILD THE LSR POOLS.     *
//*     5. ASSEMBLE THIS VSAM RESOURCE DEFINITION.                  *
//*     6. LINKEDIT THIS VSAM RESOURCE DEFINITION NAMED BLXVDEF.    *
//*                                                                 *
//* SETUP REQUIREMENTS:                                             *
//*                                                                 *
//* 1. JOB CARD CHANGED TO MEET INSTALLATION REQUIREMENTS.          *
//* 2. INFORMATION MANAGEMENT for z/OS MACRO LIBRARY INSTALLED.             *
//* 3. VSAM MACROS ARE ASSUMED TO BE IN SYS1.MACLIB                 *
//* 4. DATA SET NAMES CHANGED TO MEET INSTALLATION REQUIREMENTS.    *
//* 5. RESOURCE POOL SIZES CHANGED TO MEET INSTALLATION REQUIREMENTS. *
//*    THE AMS LISTCAT COMMAND CAN BE USED TO VERIFY THE CONTROL    *
//*    INTERVAL SIZES FOR THE VSAM DATA SETS.                       *
//* 6. PLACEHOLDER NUMBERS CHANGED TO MEET INSTALLATION REQUIREMENTS. *
//* 7. LOAD MODULE NAME CHANGED TO MEET INSTALLATION REQUIREMENTS.  *
//*                                                                 *
//* EXAMPLE ASSUMPTIONS:                                            *
//*                                                                 *
//* 1. THREE NSR DATA SETS                                          *
//* 2. PRODUCTION AND DEVELOPMENT DATABASES REQUIRED                *
//*    A. NINETY PRODUCTION USERS                                   *
//*    B. TWENTY DEVELOPMENT USERS                                  *
//* 3. CONTROL INTERVAL SIZES ARE AS DESCRIBED IN THE PLANNING AND  *
//*    INSTALLATION GUIDE AND REFERENCE                             *
```

Figure 31. Example: VSAM Resource Definition Member—BLXVDEF (Part 1 of 5)

```
//**********************************************************************
//*
//*    ASSEMBLE VSAM DEFINITION
//*
//ASSEM    EXEC PGM=ASMA90,REGION=4024K,PARM='NODECK'
//SYSLIB   DD   DISP=SHR,DSN=BLM.SBLMMACS
//         DD   DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD   DUMMY
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT2   DD   UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT3   DD   UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSPRINT DD   SYSOUT=*
//SYSPUNCH DD  DISP=(NEW,PASS),DSN=&OBJLIB,UNIT=SYSDA,
//             SPACE=(80,(200,50))
//SYSIN    DD   *
BLXVDEF  CSECT
         SPACE 2
***********************************************************************
*     DEFINE NON-SHARED RESOURCES TO BLX-SP                           *
***********************************************************************
         BLXNSR  PLACES=20                TWENTY PLACES FOR NSR.
         SPACE 2
***********************************************************************
*     DEFINE VSAM DATA SETS TO BLX-SP AND CONNECT VSAM RESOURCES      *
***********************************************************************
*
*     DEFINE DATA SETS USING LOCAL SHARED RESOURCES (LSR)
*
BASEPNS  BLXDSN DSN=BLM.IBMPNLS,                                      X
               LSR=(2,2)
READPNS  BLXDSN DSN=BLM.READ.PANELS,                                  X
               LSR=(2,2)
DICT     BLXDSN DSN=BLM.DICT,                                         X
               LSR=(2,2)
MGMTSDDS BLXDSN DSN=BLM.SDDS,                                         X
               LSR=(0,0)
MGMTINDX BLXDSN DSN=BLM.SDIDS,                                        X
               LSR=(1,1)
TESTPNS  BLXDSN DSN=BLM.TEST.PANELS,                                  X
               LSR=(2,2)
TESTSDDS BLXDSN DSN=BLM.TEST.SDDS,                                    X
               LSR=(3,3)
TESTINDX BLXDSN DSN=BLM.TEST.SDIDS,                                   X
               LSR=(4,4)
*
*     DEFINE DATA SETS USING NON-SHARED RESOURCES (NSR)
*
MGMTLOG  BLXDSN  DSN=BLM.SDLDS,DSORG=NUM
JILLS    BLXDSN  DSN=BLM.JILLS.PANELS
JACKS    BLXDSN  DSN=BLM.JACKS.PANELS
         SPACE 2
```

Figure 31. Example: VSAM Resource Definition Member—BLXVDEF (Part 2 of 5)

```
***********************************************************************
*     GENERATE VSAM DEFINITION FOR BLX-SP                             *
***********************************************************************
         BLXGEN                           GENERATE DEFINITION
         SPACE 2
***********************************************************************
*     RESOURCE POOL ZERO (FOR PRODUCTION DATABASE SDDS)               *
***********************************************************************
*
*        RESOURCE POOL ZERO (SDDS DATA BUFFER POOLS)
*
LSRD0  BLDVRP  BUFFERS=(4096(31)),      SDDS DATA BUFFERS            X
               KEYLEN=7,                POOL USED BY SDDS            X
               STRNO=30,                MAINTAIN 30 POSITIONS        X
               SHRPOOL=0,               IDENTIFY POOL                X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY            X
               RMODE31=ALL,                                          X
               MODE=24,                                              X
               MF=L
*
*        RESOURCE POOL ZERO (SDDS INDEX BUFFER POOLS)
*
LSRI0  BLDVRP  BUFFERS=(2048(53)),      SDDS INDEX BUFFERS AND       X
               SHRPOOL=0,               POOL IDENTIFIER              X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY           X
               RMODE31=ALL,                                          X
               MODE=24,                                              X
               MF=L
         SPACE 2
***********************************************************************
*     RESOURCE POOL ONE (FOR PRODUCTION DATABASE SDIDS)               *
***********************************************************************
*
*        RESOURCE POOL ONE (SDIDS DATA BUFFER POOLS)
*
LSRD1  BLDVRP  BUFFERS=(2048(500)),     SDIDS DATA BUFFERS + PAD     X
               KEYLEN=34,               POOL USED BY SDIDS           X
               STRNO=45,                MAINTAIN 45 POSITIONS        X
               SHRPOOL=1,               IDENTIFY POOL                X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY            X
               RMODE31=ALL,                                          X
               MODE=24,                                              X
               MF=L
*
*        RESOURCE POOL ONE (SDIDS INDEX BUFFER POOLS)
*
LSRI1  BLDVRP  BUFFERS=(3584(34)),      SDIDS INDEX BUFFERS          X
               SHRPOOL=1,               POOL IDENTIFIER              X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY           X
               RMODE31=ALL,                                          X
               MODE=24,                                              X
               MF=L
         SPACE 2
```

Figure 31. Example: VSAM Resource Definition Member—BLXVDEF (Part 3 of 5)

```
***********************************************************************
*     RESOURCE POOL TWO  (PANEL/DICTIONARY DATA SETS)                 *
***********************************************************************
*
*        RESOURCE POOL TWO (PANEL/DICTIONARY DATA BUFFER POOLS)
*
LSRD2  BLDVRP  BUFFERS=(4096(36)),      DICT/PANEL DATA BUFFERS        X
               KEYLEN=10,               POOL USED BY DICT/PANELS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               STRNO=35,                COMBINED POSITIONS REQUIRED    X
               TYPE=(LSR,DATA),         INDEX AND DATA POOL            X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL TWO (PANEL/DICTIONARY INDEX BUFFER POOLS)
*
LSRI2  BLDVRP  BUFFERS=(1536(30)),      DICT/PANEL INDEX BUFFERS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
         SPACE 2
***********************************************************************
*     RESOURCE POOL THREE (TEST/DEVELOPMENT DATABASE SDDS)            *
***********************************************************************
*
*        RESOURCE POOL THREE (SDDS DATA BUFFER POOLS)
*
LSRD3  BLDVRP  BUFFERS=(4096(6)),       SDDS DATA BUFFERS              X
               KEYLEN=7,                POOL USED BY SDDS              X
               STRNO=20,                MAINTAIN 20 POSITIONS          X
               SHRPOOL=3,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL THREE (SDDS INDEX BUFFER POOLS)
*
LSRI3  BLDVRP  BUFFERS=(512(5)),        SDDS INDEX BUFFERS +PAD        X
               SHRPOOL=3,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
         SPACE 2
```

Figure 31. Example: VSAM Resource Definition Member—BLXVDEF (Part 4 of 5)

```
***********************************************************************
*     RESOURCE POOL FOUR (TEST/DEVELOPMENT DATABASE SDIDS)            *
***********************************************************************
*
*        RESOURCE POOL FOUR (SDIDS DATA BUFFER POOLS)
*
LSRD4  BLDVRP  BUFFERS=(2048(50)),      SDIDS DATA BUFFERS + PAD       X
               KEYLEN=34,               POOL USED BY SDIDS             X
               STRNO=20,                MAINTAIN 20 POSITIONS          X
               SHRPOOL=4,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
*
*        RESOURCE POOL FOUR (SDIDS INDEX BUFFER POOLS)
*
LSRI4  BLDVRP  BUFFERS=(512(3)),        SDIDS INDEX BUFFERS +PAD       X
               SHRPOOL=4,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,                                            X
               MODE=24,                                                X
               MF=L
         SPACE 2
         END   BLXVDEF
/*
//*********************************************************************
//*    LINK VSAM DEFINITION ON GOOD ASSEMBLY                          *
//*********************************************************************
//LKED     EXEC PGM=IEWL,COND=(1,LT,ASSEM),
//           PARM='XREF,LET,LIST,MAP,REUS,NCAL,AMODE=24,RMODE=24'
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSLMOD  DD   DSN=BLM.SBLMMOD1,DISP=SHR
//SYSLIN   DD   DSN=&OBJLIB,DISP=(OLD,DELETE)
//         DD   *
      ENTRY   BLXVDEF
      NAME    BLXVDEF(R)
//
```

Figure 31. Example: VSAM Resource Definition Member—BLXVDEF (Part 5 of 5)

End of Programming Interface information

D

Defining Tivoli Information Management for z/OS Session-Parameters Members

This appendix provides:
- The naming convention for session-parameters members
- The syntax and descriptions for the following Tivoli Information Management for z/OS macros:
  - BLGPARMS
  - BLGCLUST
  - BLGCLDSN
  - BLGGEN
- Sample JCL and session-parameters CSECT for creating a session-parameters member.

This appendix is designed to help you create a session-parameters member.

Programming Interface information

Defining Session-Parameters Members (BLGSESaa)

You specify session parameters by using the assembler language macros described in this appendix to define a control section (CSECT). The macros are delimited by a CSECT statement and an END statement. You must store each set of session parameters as a member of a PDS.

Tivoli Information Management for z/OS processes the session parameters during its initialization phase.

- You must link-edit session-parameters members as nonreentrant and nonreusable.
- To work in Tivoli Information Management for z/OS Version 7.1, all currently existing session-parameters members must be reassembled. See Figure 32 on page 340 for an example. Also, refer to your assembler language documentation for information on assembling modules.
- Review the changes to the session parameters to see what changes might be required or that you might want to apply.
- As in Information/Management Version 5.1, VSAM macros are no longer used in the session-parameters member.
- If you define multiple session-parameters members for your installation, be sure that a data set defined as a read panel data set in one session-parameters member is not defined as a write panel data set in a different session-parameters member.

Defining an Initialization Load Module

You create an initialization load module by assembling and link-editing a session-parameters member CSECT into the appropriate load library. Because the BLGSESaa modules are nonreentrant, you must use NORENT as one of your linkage editor parameters. See Figure 32 on page 340 for a sample job stream to assemble and link-edit a session-parameters member. The load module must reside in one of the load libraries accessible to Tivoli Information Management for z/OS:

- A task library (that is, ISPF's ISPLLIB DD statement)
- A step or a job library
- A system link library (such as SYS1.LINKLIB)

The load library contains one or more load modules that you define; you can put different load modules in different load libraries. The load module must contain a single assembler language CSECT that you create. You can name this CSECT anything you want; however, for convenience, you can use the load module's name.

Naming the Session-Parameters Member

The member name you use must follow these naming rules:

- The name can be 7 or 8 characters long.
- The first 6 characters of the name must be *BLGSES*.
- The remaining 1 or 2 characters can be alphabetic, numeric, or national (#, $, @).

If you use a single numeric digit, it is right-justified with a leading zero. If you use a single alphabetic or national character, it is left-justified with a trailing blank. If you omit the SESS keyword when you start Tivoli Information Management for z/OS, a default suffix of 00 is used.

Understanding the Macro Syntax Description

In the lists of parameters for the macros and session-parameters members described in this chapter, some parameters are shown with brackets *[ ]*, braces *{ }*, or vertical bars |. These are for clarification purposes only; they are not part of any keyword in the lists.

- Brackets indicate optional data.
- Braces indicate that you must select exactly one of the choices enclosed within the braces.
- Vertical bars indicate selectable items. You must choose one of these items.
- **Bold, underlined values** in these macro and session-parameters member descriptions are the default values.

BLGPARMS Macro — Defining Tivoli Information Management for z/OS's Operating Characteristics

The BLGPARMS assembler macro defines the data sets that the users' address spaces share. It also defines other operating characteristics for a session, such as the report tailoring parameters and the maximum amount of program storage Tivoli Information Management for z/OS uses.

The syntax of the BLGPARMS macro is:

```
[label]   BLGPARMS   RPANLDS=(maclabel,...)
                     ,DATEFMT=extfmt
                     [,APIENQ={WAIT|NOWAIT}]
                     [,APIRETRY={n|25}]
                     [,ATTNKEY={DISABLE|ENABLE}]
                     [,CAS={ssss|BLX1}]
```

```
[,DATECNV={BLGCDATS|modname}]
[,DICTDS=maclabel]
[,EXTSORT={YES|NO}]
[,GBLPID=gpartname]
[,LINECNT={number|0}]
[,LPLNCMD={ALL|V61}]
[,MODELDB=([{4|5|6|7|8|9}],
          [,dartchr])]
[,MODSIZE={bytes|MAX}]
[,ODATEFMT=oldextfmt]
[,OTIMEZON=oldtzsymt]
[,PANEL={pnlname|BLG0ENTR}]
[,PNLBCNT={number|50}]
[,RFTDS=maclabel]
[,SIDEVC=([{unit|SYSDA}]
          [,{trksize|3120}])]
[,SODEVC=([{unit|SYSDA}]
          [,{trksize|3120}])]
[,SORT={modname|SORT}]
[,SORTPFX=([{n1|2147483647}]
           [,{n2|32767}]
           [,{n3|500}])]
[,SWDEVC=([{unit|SYSDA}]
          [,{trksize|3120}])]
[,TEXTAUD={YES|NO}]
[,TIMECNV=timertn]
[,TIMEDEL=[-]{HH[:MM]|00:00}]
[,TIMEZONE=tzsym]
[,TSXTASKS=([max],[idle])]
[,WPANLDS=maclabel]
```

Note: Before Version 5 of Information/Management, the LSR, ENQLVL, BREPORT, and IREPORT keywords were valid operands on the BLGPARMS macro. These keywords are not required in Tivoli Information Management for z/OS Version 7.1, but they can still be used for compatibility with old session parameters. However, if you specify these keywords, the Tivoli Information Management for z/OS code does not process them. Before Information/Management Version 6.2, SRCHENQ was a valid operand on the BLGPARMS macro. If you set this operand in Information/Management Version 5.1, you do not have to change it unless you want to change the value. If you change the value, you must use the SRCHLIMIT operand.

label A standard assembler language instruction label. This label is optional.

maclabel,...
Specifies the label of a BLGCLDSN macro, that identifies a RPANLDS that you are using. You must specify a minimum of one label. You can specify RPANLDS=maclabel if there is only one label. This label is used only for the RPANLDS.

RPANLDS
Identifies the BLGCLDSN macros that describe the read panel data sets.

This keyword is required. You can specify the labels of one or more BLGCLDSN macros in a list. The user's address spaces share the RPANLDS, which are normally accessed in read-only mode. If you do not specify RDONLY=YES in the BLGCLDSN macro for an RPANLDS, that RPANLDS is automatically accessed in read/write mode. In read/write mode, you can issue a request to copy a panel into the data set.

The order in which you specify the BLGCLDSN labels determines the order in which Tivoli Information Management for z/OS searches for a panel during normal processing. For example, assume that you specify RPANLDS(A,B,C) and the BLGCLDSN macros A, B, and C define data sets X, Y, and Z, respectively. When Tivoli Information Management for z/OS retrieves a panel, it first looks for the panel in data set X. If Tivoli Information Management for z/OS does not find the panel in data set X, it looks in data set Y, and finally in data set Z. The result is that Tivoli Information Management for z/OS provides logical data set concatenation for these VSAM data sets.

Specifying more than one read panel data set degrades performance. The amount of performance degradation varies, depending on the following factors:

- The number of panel buffers specified on the PNLBCNT keyword (see page 327)
- The number of BLX-SP user buffers maintained for each read panel data set (specified by the USERBUF keyword on the BLXDSN macro in the VSAM definition member for a non-sysplex environment)
- The number of read panel data sets that you must access to retrieve a panel
- The number of panels in each data set
- Whether the VSAM index components for these data sets are completely maintained in virtual storage through shared resource pools
- Whether the VSAM data components for these data sets are partially or completely maintained in virtual storage through shared resource pools.

You must decide whether to accept the reduced performance to gain the advantage of maintaining your modified panels in a separate data set from the Tivoli base panel definitions and any service that may be applied to these definitions.

DATEFMT

Enables you to specify a default external date format. This keyword is required unless a value other than BLGCDATS is specified for the DATECNV keyword. The value specified for the external date format is the default value for the session.

The external date formats in use by a session can be displayed by entering the HELP STATUS command.

extfmt

Specifies the external date format that you select from the list below. The external date format is the default format for the session. Users can override this format through a user profile selection. If no user profile selection of external date format is made, users will see this format when entering or working with records on panels or when viewing data in reports.

EXTFMT can be any one of the following external date formats:

```
MM/DD/YY      DD/MM/YY      YY/MM/DD      DDMMMYY
MM/DD/YYYY    DD/MM/YYYY    YYYY/MM/DD    DDMMMYYYY
MM-DD-YY      DD-MM-YY      YY-MM-DD      YYDDD
MM-DD-YYYY    DD-MM-YYYY    YYYY-MM-DD    YYYYDDD
MM.DD.YY      DD.MM.YY      YY.MM.DD
MM.DD.YYYY    DD.MM.YYYY    YYYY.MM.DD
```

Only one format can be specified. If none of the formats above satisfy your external date format requirements, you can specify your own date conversion routine in the **datertn** parameter of the DATECNV keyword.

The default external date format specified on the DATEFMT keyword is the format used for all new records created in Tivoli Information Management for z/OS. If you have records in your database that were created or updated with versions of Tivoli Information Management for z/OS prior to Version 7.1 (including Information/Management and Tivoli Service Desk for OS/390), and those records used an external date format which was different from the format specified on DATEFMT, you can specify the ODATEFMT keyword to enable those older records to be processed by Tivoli Information Management for z/OS. The ODATEFMT keyword causes Tivoli Information Management for z/OS to automatically migrate records in the older external date format as they are being accessed.

More information about date formats is provided in “Enabling Alternative Date and Time-of-Day Formats” on page 227.

APIENQ

Specifies whether to place an API in a wait condition after it fails to enqueue on a record.

- If you specify *WAIT*, the API is placed in a wait condition until the record is available.
- If you specify *NOWAIT*, the API either bypasses the record or retries the enqueue depending on the APIRETRY value and how many times it has already tried. *NOWAIT* is the default value.

APIRETRY

Specifies the number of times a Tivoli Information Management for z/OS API that is performing add record relations, update, and check-in tasks retries an enqueue on a record. Tivoli Information Management for z/OS waits approximately 1 second between retries. If the API cannot enqueue on the record after this number of attempts, the user receives a message, or a return code, or both, saying that the record is not available.

You can use any number from *0* through *255*. The default value is 25.

Note: Use this parameter only if you specified a value of *NOWAIT* for the APIENQ parameter.

ATTNKEY

Indicates how you use the attention key during sessions with Tivoli Information Management for z/OS. The valid values for ATTNKEY are:

ENABLE

Indicates that Tivoli Information Management for z/OS does not attempt to control the effect of an attention interrupt. ATTNKEY=ENABLE can end your Tivoli Information Management for z/OS session. This value is the default.

DISABLE

Indicates that Tivoli Information Management for z/OS code attempts to intercept attention interrupts and causes them to be ignored.

Notes:

1. During a file operation, pressing the attention key to end an Tivoli Information Management for z/OS session can result in a corrupted database record.

2. If you code ATTNKEY=DISABLE, Tivoli Information Management for z/OS issues the STAX macro. This STAX macro remains in effect until Tivoli Information Management for z/OS is ended or the STAX macro is issued again. Refer to *TSO/E Programming Services* for further details about the STAX macro.
3. DISABLE is the recommended value when spanned records are defined in the SDIDS.

CAS Specifies the name of the BLX-SP subsystem that this session-parameters member is used with. This name must be the same as the one in the BLXSSINM module for the subsystem. Otherwise, an ABEND will occur. See "Defining a Subsystem to MVS" on page 144 for information about creating the BLXSSINM module.

The name you use must follow these naming rules:
- The name must have 4 characters.
- The first character must be an alphabetic or national (#, $, @) character.
- The remaining characters can be alphabetic, numeric, or national characters.

This keyword is optional. If you do not specify it, the default value, *BLX1* is used. If you use the default value, it is recommended that you not specify a keyword.

DATECNV

Specifies the name of a date conversion routine. This keyword is optional, and the default is BLGCDATS. BLGCDATS is the name of the routine supplied with Tivoli Information Management for z/OS. As an alternative, you can specify the name of a date-conversion-exit load module written by personnel at the installation site. This exit routine transforms the format of a date from internal to external form and from external to internal form. If you specify this load module, Tivoli Information Management for z/OS loads it during its initialization phase. The exit routine must exist in a load library accessible to Tivoli Information Management for z/OS.

The date conversion routine BLGCDATS must be specified for users to choose their own external date format interactively through the user profile or through an API program.

If you omit this parameter, the BLGCDATS routine runs. BLGCDATS handles all the external date formats that are supported by the DATEFMT keyword which are listed on page 320.

If you plan to supply your own installation exit, evaluate its effect on an existing database. When you install your user exit, enter dates in your external format for all new records that you create.

For more information on using your own date conversion routine, see "Implementing an External Date Format through User Exits" on page 242.

DICTDS

Identifies the BLGCLDSN macro that describes the dictionary data set (DICTDS). This keyword is required if you use the session-parameters member for PMF or for reports (including record printing).

The DICTDS is accessed in read/write mode if you do not specify RDONLY=YES in the BLGCLDSN macro for this data set.

EXTSORT

Specifies whether the sort routine specified by the SORT parameter should also be used to sort list data and search results lists (sort-on-prefix data). This parameter is ignored if the table panel used to display the list data or search results do not specify sort-on-prefix.

Note: You can only sort cognized data.

YES Specifies that Tivoli Information Management for z/OS use the sort routine specified by the SORT parameter to sort search results lists and list data.

Notes:

1. If EXTSORT=YES is specified and the SORT parameter is not specified, the default sort routine for the SORT parameter is used.
2. If EXTSORT=YES is specified, the *n3* value for the SORTPFX parameter is ignored because the *n3* value applies only to sorts done by the Tivoli Information Management for z/OS internal sort routine.
3. Using an external sort routine for sorting is not as efficient as using the Tivoli Information Management for z/OS internal sort routine, so it is recommended that you specify an external sort routine only if you find that the Tivoli Information Management for z/OS internal sort routine does not sort the list data or search results list in the right order for your language.

NO Specifies that Tivoli Information Management for z/OS use its internal sort routines to sort list data and search results lists. This is the default value.

GBLPID

Specifies the Global Partition Name. If you have a set of records which you would like to be accessible to all users, put all such records into a single partition which is designated “Global”. For information about database partitioning, refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.

These are the rules for specifying a Global Partition Name:

- The value can be up to nine alphanumeric characters.
- If a value is not provided on the **GBLPID=** parameter and a Global Partition Name is not specified via program exit **BLG01448**, then no Global Partition exists.
- If a value is supplied on the **GBLPID=** parameter, that value is used instead of the value set via program exit **BLG01448**.
- If a value of **GBLPID=''** is supplied, the default **Global Partition Name** established via program exit **BLG01448** is ignored.
- If **GBLPID=** is omitted, the **Global Partition Name** set by, program exit **BLG01448** will be used.

LINECNT

Indicates the maximum number of lines to allocate in virtual storage for the display of Tivoli Information Management for z/OS search results. The specified number applies to each active or pending display of the search results list. The size of the display screen determines the minimum number of lines allocated.

Specifying a number between 0 and the number of lines that the screen can display, implies that no additional lines are available beyond the number required to display a single screen.

If you specify a small number, such as 3, each active or pending display of the search results list uses less virtual storage. However, when you scroll up or down on that display, the database must supply the new information. This occurs for each scroll operation, thereby increasing response time because of the required I/O operation.

If you specify a large number, such as 100, each active or pending display of the search results list uses more virtual storage. However, if you scroll up or down into previously displayed information that was not yet deleted internally, additional database I/O is unnecessary, thus improving response time.

For example, assume that your workstation can display 20 search results matches and you do a search that produces a large number of matches. Assume that you scroll down one page, scroll down another page, and finally scroll up the maximum number of lines to return to the top of the list.

Regardless of the value you specify for the LINECNT keyword, the database must supply information for the first 20 matches in order to generate the first display of the search results list. When you scroll down one page, the database must supply information for the second set of 20 matches. When you scroll down another page, the database must supply information for the third set of 20 matches. However, when you scroll up the maximum number of lines (to display the first 20 matches), the database may or may not have to supply information.

If you specify 40 or less for this keyword, the database must supply information for all of the first 20 matches. If you specify 60 or more, the database does not have to supply information for the redisplays of the first 20 matches. If you choose 55, the database supplies information for only the first 5 matches; the other 15 matches remain in virtual storage.

Each logical record in a search results list uses a certain number of bytes of virtual storage. One line (one logical record) in a search results list for database 5 (the Tivoli Information Management for z/OS database) uses 127 bytes. Therefore, if you specify LINECNT=50 and you obtain 50 or more search matches from the Tivoli Information Management for z/OS database, the search results list uses 6350 (that is, 50 x 127) bytes of virtual storage.

The default value for this keyword is 0.

LPLNCMD

Specifies whether to use the list processor line commands as they existed prior to Information/Management Version 6.2 or to use the line commands as shipped in Tivoli Information Management for z/OS Version 7.1. For information about the current line commands for list processor panels, refer to the BLG01396 program exit in the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.

ALL Specifies the use of Tivoli Information Management for z/OS Version 7.1 list processor line commands. This value is the default.

V61 Specifies the use of list processor line commands as they existed in Information/Management Version 6.1 with the following additions:

- **A**dd and **R**epeat can be used on any line, not just the last line.
- The block forms of **D**elete (DD) and **R**epeat (RR) are supported.
- The block forms with a repetition factor of **D**elete and **R**epeat (for example, DD5 or RR3) are supported.

MODELDB

Enables you to specify a data model database and a trigger character to identify data attribute records. The database specified is also the database where the validation records (and data view and data attribute records) reside.

{4|5|6|7|8|9}

Specifies the number of the database which contains the data model records. The default value is 5.

dartchr

Specifies the data attribute record trigger character. The trigger character can be an alphabetic, numeric, or national (@, #, $) character. The ampersand (&) character is reserved for use by data attribute records provided with Tivoli Information Management for z/OS. The trigger character is used to distinguish a panel name from a data attribute record identifier when running Tivoli Information Management for z/OS interactively. If the value specified for **dartchr** is the same as the character in the fourth position of a panel name being loaded, the name is assumed to be that of a data attribute and the record is loaded from the data model database. There is no default value for **dartchr**.

MODSIZE

Identifies the maximum virtual storage size (in bytes) that Tivoli Information Management for z/OS can attempt to use for its load modules. Tivoli Information Management for z/OS tries to maintain load module storage at or below the value you specify. However, if you request several nested functions (for example, SUSPEND, DISPLAY, and UPDATE commands), the actual amount of virtual storage used for load modules can exceed the value you specify until you end the requested functions. Tivoli Information Management for z/OS uses a least-recently-used (LRU) algorithm to determine which load modules to delete. The default value for this keyword is *MAX*.

If you specify a very low value, such as 0, Tivoli Information Management for z/OS maintains low storage use, with increased overhead, by continually deleting modules and loading the requested modules. However, certain Tivoli Information Management for z/OS load modules are always in virtual storage, such as BLGINIT, BLGINIT1, BLGMAIMS, because they are required for every Tivoli Information Management for z/OS session.

If you specify a large value, such as 750 000, Tivoli Information Management for z/OS keeps load modules in virtual storage even though they are currently not in use. This can result in more overhead on the z/OS swapping subsystem, but it reduces the overhead of deleting and loading modules.

When you specify any number, Tivoli Information Management for z/OS continually maintains internal tables so that it can always verify the following:

- When the number exceeds the one you specified
- When a load module is currently in use, and, hence, it is not a candidate for deletion
- When that module is freed, and, hence, it is a candidate for deletion.

Tivoli Information Management for z/OS requires a small amount of processing time to maintain its internal tables. This processing time is in addition to loading a module from one of the load libraries, if it becomes necessary.

When you specify a number for MODSIZE, Tivoli Information Management for z/OS counts the size of each load module regardless of whether a load library or the link pack area supplied that module. Instead of specifying a numeric value for MODSIZE, you can specify the word MAX. If you specify MAX, currently loaded modules remain in storage for the duration of the Tivoli Information Management for z/OS session. Tivoli Information Management for z/OS need not to update its internal tables continually, and thus avoids the previously mentioned performance degradation. Also, because each load module is loaded only once from a load library, the performance of an Tivoli Information Management for z/OS session improves.

A value of MAX is suggested for this keyword:

- If you decide to place reentrant Tivoli Information Management for z/OS load modules in the link pack area (LPALIB).
- If you are running one of the Tivoli Information Management for z/OS utilities that uses this session-parameters member.

ODATEFMT

Enables you to specify an external date format for older records in your database. This keyword is optional. The default value is the value specified for *extfmt* on the DATEFMT keyword.

The default external date format on the DATEFMT keyword is the date format used for all new records created in Tivoli Information Management for z/OS. The ODATEFMT keyword parameter (*oldextfmt*) is used to process records that were created with versions of Tivoli Information Management for z/OS prior to Version 7.1 (including Information/Management and Tivoli Service Desk for OS/390). This keyword causes Tivoli Information Management for z/OS to automatically migrate older records as they are being accessed so that users can process the records and optionally take advantage of a user profile option which enables them to specify a preferred date format when working with records.

If you specify a value for ODATEFMT, ensure that:

- The value specified for ODATEFMT is one of the 22 supported external date formats recognized by Tivoli Information Management for z/OS.
- A default external date format is also specified on the DATEFMT keyword.
- The default external date format is different from the format specified for ODATEFMT. Specifically, the lengths of the two external date formats are different.

  An error condition will occur during assembly of the session-parameters member if the lengths of the two external date formats are the same. For example, the following statement will result in an error because both entries have a length of 10 even though the format MM/DD/YYYY is different from DD/MM/YYYY:

  ```
  DATEFMT=MM/DD/YYYY
  ODATEFMT=DD/MM/YYYY
  ```

For databases containing two different old formats, one should be specified on DATEFMT and the other on ODATEFMT.

OTIMEZON

Specifies the time zone symbol associated with older records in the database if the time zone symbol is different from the default time zone symbol specified through the TIMEZONE keyword. This keyword is optional. If you have records in the database that were created with versions of Tivoli Information Management for z/OS prior to Version 7.1 (including Information/Management and Tivoli Service Desk for OS/390), and the dates and times in those records are for a time zone which is different from the default time zone being used, you should specify the OTIMEZON keyword and supply the time zone symbol associated with the older records as the value for OTIMEZON. The value is used to convert dates and times in the old records to universal time.

If you include a value for OTIMEZON, you must also specify a default time zone symbol with the TIMEZONE keyword. If TIMEZONE is not specified, OTIMEZON is ignored. If OTIMEZON is not specified, the value for TIMEZONE is used as the default. Only one time zone can be specified with the OTIMEZON keyword.

The time zone currently in effect can be viewed in the output generated by the HELP STATUS command. For more information about setting time zones, see “Defining a Time Zone” on page 271.

A list of time zone symbols provided with Tivoli Information Management for z/OS is available in Table 17 on page 267; however, other time zone symbols can be defined in the TIMEZONE reference record and specified here. Values specified as time zone symbols must be defined in the TIMEZONE reference record.

PANEL

Specifies the name of the control panel to use upon entry to Tivoli Information Management for z/OS. This control panel selects the Primary Options menu to display. It is also used as the target of the ;INITIALIZE command.

This keyword is optional. If you omit it, the default is BLG0ENTR.

PNLBCNT

Indicates the maximum number of local panel buffers to allocate in virtual storage. This number must be a positive integer from *1* to *32 767*. The default value for this keyword is 50. Panels are automatically eliminated from this set of buffers as necessary, using a least-recently-used (LRU) algorithm.

Specifying a small number of buffers increases the chance that the BLX-SP must provide a panel. This reduces system performance, but less storage is required in the user’s address space.

Maintaining panel buffers in the user’s address space reduces the load on the BLX-SP and reduces the time for panel-to-panel transition. Message BLG22556, which is saved in the BLX-SP log data set, indicates the number of panel buffers a session used. After the session ends, this information can be obtained from the BLX-SP LOGSYSOUT queue and used to determine the optimal value to specify for this keyword.

Notes:

1. While testing for the optimal number of panel buffers to specify, specify some extra buffers to enable infrequently used panels to be loaded without displacing panels used in the typical work session.
2. See page 345, for information about the LOGSYSOUT parameter.

RFTDS

Identifies the BLGCLDSN macro that describes the data set containing the Report Format Tables (RFTs) or application program interface (API) data and pattern tables.

The RFTs control the format of reports generated by Tivoli Information Management for z/OS. Unlike the other data sets, this is a PDS.

Users' address spaces share the RFTDS and access it in read-only mode.

This keyword is optional, but you must specify it if you run reports against the database, if you print records, or if you call an API using the session-parameters member.

SIDEVC

Indicates the SORTIN data set device characteristics that the Tivoli Information Management for z/OS report function uses.

unit Indicates the type of unit on which the data set is to reside. The unit type can be a unit address (entered as 3 hexadecimal characters), an IBM-supplied device type (entered as either 1 to 8 decimal digits or the character hyphen (-)), or a user-assigned group name (entered as 1 to 8 alphanumeric characters). If you omit the unit type, the data set is automatically allocated using SYSDA as the default dynamic allocation unit name.

Consider using a virtual input/output (VIO) unit type for your SORTIN data set.

trksize

Indicates a decimal number less than or equal to the minimum track size, in bytes, of the types of devices covered by the unit type. If you omit the *trksize*, the default number is 3120 bytes.

If the *trksize* value you specify is less than the logical record size that you write to the SORTIN data set, use the logical record size instead. The maximum value for *trksize* is 32 752.

Notes on the SIDEVC keyword:

See "SORTIN, SORTOUT, and SORTWK01 Data Sets" on page 165 for more information on this data set.

SODEVC

Indicates the SORTOUT data set device characteristics that the Tivoli Information Management for z/OS report function uses.

unit Indicates the type of unit on which the data set is to reside. The unit type can be a unit address (entered as 3 hexadecimal characters), an IBM-supplied device type (entered as either 1 to 8 decimal digits or the character hyphen), or a user-assigned group name (entered as 1 to 8 alphanumeric characters). If you omit the unit type, the data set is automatically allocated using SYSDA as the default dynamic allocation unit name.

Consider using a VIO unit type for your SORTOUT data set.

trksize

Indicates a decimal number less than or equal to the minimum track size, in bytes, of the types of devices covered by the unit type. If you omit the *trksize*, the default number is 3120 bytes.

If the *trksize* value you specify is less than the logical record size that you write to the SORTOUT data set, use the logical record size instead. The maximum value for *trksize* is 32 752.

Notes on the SODEVC keyword:
See "SORTIN, SORTOUT, and SORTWK01 Data Sets" on page 165 for more information on this data set.

SORT Entry-point name of the program Tivoli Information Management for z/OS uses to perform sorts. This keyword is optional. The default is SORT (IBM OS/VS Sort/Merge - 5740-SM1).

Note: This keyword only affects sorting for reports, list data, and search results lists if the EXTSORT keyword value is YES. If EXTSORT is NO, sorting is done internally by Tivoli Information Management for z/OS.

Figure 15 on page 167 contains the parameter lists that Tivoli Information Management for z/OS passes to the SORT routine that you indicate. Tivoli Information Management for z/OS calls the SORT routine twice, and Figure 15 shows both parameter lists. Refer to the *DFSORT Application Programming Guide* for the standard interface to the IBM Sort/Merge program.

You can allocate a data set with the ddname of BLGSMSG in your TSO logon procedure to receive any Sort/Merge program product messages generated during your use of Tivoli Information Management for z/OS. This data set is optional.

SORTPFX
Enables you to specify parameters (*n1*, *n2*, *n3*) to control both the building of record results and the Tivoli Information Management for z/OS sort-on-prefix function.

Note: You can sort cognized data only.

You may not need to sort your search results list by RNID if you chose not to reuse SDDS position numbers. Refer to the description of the BLGUT9 utility and the *Tivoli Information Management for z/OS Program Administration Guide and Reference* for more information on storing records in system-assigned RNID order.

n1 Specifies the maximum number of lines that you want a search results list to contain. If the number of records that satisfy the search exceeds this value, Tivoli Information Management for z/OS does not build a search results list; however, the number of search results appears. The valid range for *n1* is from *1* to *2 147 483 647*. The default value is *2 147 483 647*.

Note: If the number of matches in the search results list is less than *n1* but greater than 32 767, Tivoli Information Management for z/OS displays a maximum of 32 767 matches.

Through an API, you can also use this parameter to limit the number of search results matches returned from inquiry processing. Because the APIs do not sort on search results lists, this field can be used to limit the number of results returned in the programming interface results table (PIRT). Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information.

n2 Specifies the maximum number of matches in the search results list that controls the sort-on-prefix function used in building record results for display. The Tivoli Information Management for z/OS code ignores the *n2* parameter if the table panel used to display the search results does not specify sort-on-prefix. If the number of records that satisfy the search does not exceed this number and the table panel used to display the results specifies sort-on-prefix, the sort-on-prefix operation occurs before the record results display. If the number of records that satisfy the search exceeds this value, no sort-on-prefix operation occurs even though you specified it in the table panel that displays the results. The search results are displayed in the order in which the SDIDS finds them. The valid range for *n2* is from *1* to *32 767*. The default value is *32 767*.

n3 Specifies the maximum number of lines you want sorted using the sort routine that is more efficient for sorting a smaller number of lines. This sort technique reads all the matched SDDS records, extracts the sort field value, sorts a table of all those values, and then retrieves the records to be displayed. The Tivoli Information Management for z/OS code ignores the *n3* parameter if the table panel used to display the search results does not specify sort-on-prefix. If the number of your search matches is greater than this number, the SDIDS is read and the records are placed in the search results list in order. This technique can be very fast for sorting large search results lists, but it becomes a performance problem when the sort field contains large numbers of different values. For more information on avoiding massive-range reads from the SDIDS, see “Sorting Search Results Lists” on page 61. The valid range for *n3* is from *1* to *32 767*. The default value is *500*.

SWDEVC

Indicates the SORTWK01 data set device characteristics that the Tivoli Information Management for z/OS report function uses.

unit Indicates the type of unit on which the data set is to reside. The unit type can be a unit address (entered as three hexadecimal characters), an IBM-supplied device type (entered either as one to eight decimal digits or the character hyphen), or a user-assigned group name (entered as one to eight alphanumeric characters). If you omit the unit type, the data set is automatically allocated using SYSDA as the default dynamic allocation unit name.

Consider using a VIO unit type for your SORTWK01 data set.

trksize

Indicates a decimal number less than or equal to the minimum track size, in bytes, of the types of devices covered by the unit type. If you omit the *trksize*, a default number of 3120 bytes is used.

If the *trksize* value you specify is less than the logical record size that must be written to the SORTWK01 data set, use the logical record size instead. The maximum value for *trksize* is 32 752.

Notes on the SWDEVC keyword:

See “SORTIN, SORTOUT, and SORTWK01 Data Sets” on page 165 for more information on this data set.

TEXTAUD

Indicates whether a freeform text audit trail (date, time, and user who last altered the text) is to be maintained when the ISPF/PDF editor initiates processing for a record that already contains the requested type of text.

YES Indicates that you want a text audit trail maintained when users add to or change existing text. When a user selects the type of text, if the record already contains that type of text, a prompt asks the user to choose whether to add new text or update existing text.

- If the response is to add, the ISPF/PDF editor is used to add text. When the user saves the additions, the new text with its audit information is appended to the existing text.
- If the user chooses update, the Tivoli Information Management for z/OS editor is used to update existing text.

NO Indicates that the text audit trail is not important for this particular session-parameters member. If a user chooses PDF in the profile editor selection, the ISPF/PDF editor is accessed. All previous control information for the requested type of text in this record is replaced by the date and time that the ISPF/PDF editor was started. This is the default value.

Tivoli Information Management for z/OS ignores the TEXTAUD keyword when either of the following is true:
- The user profile specifies the Tivoli Information Management for z/OS editor.
- The record the user is updating contains no text of the requested type.

TIMECNV
Specifies the name of a time-of-day conversion-exit load module written by personnel at the installation site. This exit routine transforms the format of a time from internal to external form and from external to internal form. If you specify this load module, Tivoli Information Management for z/OS loads it during its initialization phase. The exit routine must exist in a load library accessible to Tivoli Information Management for z/OS. The AMODE of your time conversion routine must be 31.

If you omit this keyword, the external form of time defaults to 24-hour clock HH:MM, where HH is the hour of the day and MM is the minutes within the hour. If you plan to supply your own installation exit, you must evaluate its effect on an existing database. When you install your user exit, you must enter times in your external format for all new records that you create. However, a display of an old record shows any times in the old format.

Note: When you are running Tivoli Information Management for z/OS utilities that require session-parameters members, be sure you are using the same time conversion routine that you specified in the session-parameters member to create records. Otherwise, unpredictable results can occur. Updating old times in existing records damages your SDIDS unless your time conversion routine can convert your user external format and can also support the HH:MM Tivoli Information Management for z/OS format. Your time conversion routine does need not be able to convert the Tivoli Information Management for z/OS format, but it does need to recognize the format so that it does not attempt to convert it (this could damage your SDIDS).

TIMEDEL
A number specified as [−]{HH[:MM]}. It indicates that a remote terminal is in a different time zone from that of the processor. The current date or time supplied by Tivoli Information Management for z/OS (using the equal sign during record create/update or by a predefined system date or time variable within an RFT) is the

same as the processor's, although you receive the date and time appropriate for the time zone in which the terminal resides. That is, the date or time is stored in the record as modified by the TIMEDEL keyword value.

The TIMEDEL keyword value specifies the number of hours and minutes ,positive or negative, to add to the processor's current date and time to arrive at the terminal's current date and time. For instance, if a terminal is in the Pacific Standard Time zone and is remotely connected to a processor in the Eastern Standard Time zone, use a TIMEDEL keyword value of -3. Two a.m. Eastern Standard Time on December 31, 2000, for the processor is adjusted to 11:00 PM Pacific Standard Time on December 30, 2000, for the workstation.

Note: This time zone adjustment does not include communication delays.

The default value for the TIMEDEL keyword is 00:00.

Note: If universal time processing is enabled (the TIMEZONE keyword is specified), TIMEDEL should not be specified.

TIMEZONE

Specifies the default time zone symbol for all users of the session (for example TIMEZONE=ET for United States/Canada Eastern with daylight savings time). This keyword is required if you are implementing universal time processing. If TIMEZONE is not specified, no universal time processing will occur, even if the DATETIME and TIMEZONE records are present in the Tivoli Information Management for z/OS database.

A list of time zone symbols provided with Tivoli Information Management for z/OS is available in Table 17 on page 267; however, other time zone symbols can be defined in the TIMEZONE reference record and specified here. Values specified as time zone symbols must be defined in the TIMEZONE reference record.

The time zone currently in effect can be viewed in the output generated by the HELP STATUS command. For more information about setting time zones, see "Defining a Time Zone" on page 271.

Only one time zone symbol should be specified on the TIMEZONE keyword. If you need to include a different time zone symbol for older records in your database, see the description of the OTIMEZON keyword on page 326. OTIMEZON is required if you have records in the database that contain date and time fields that were created before those date and time fields were defined as universal time fields and were entered in a time zone different than the one specified on the TIMEZONE keyword. If you have records in the database that were created with versions of Tivoli Information Management for z/OS prior to Version 7.1 (including Information/Management and Tivoli Service Desk for OS/390), and the dates and times in those records are for a time zone which is different from the default time zone being used, you should specify the OTIMEZON keyword.

The TIMEZONE keyword replaces the TIMEDEL keyword if specified. If you already have the TIMEDEL keyword specified, you should remove it. If the TIMEDEL keyword is specified, it will be ignored.

The time zone symbol specified with this keyword can be overridden by API applications using the TIME_ZONE PDB.

Notes:

1. Universal time processing is not supported in the Integration Facility. Therefore, you should not code the TIMEZONE keyword if you are using the Integration Facility.
2. To avoid having users experiment with the User's time zone field in the user profile if you are not implementing universal time processing through use of this keyword, it is recommended that your Tivoli Information Management for z/OS administrator use PMF to remove the field from panel BLG0P700, User and Database Defaults.

TSXTASKS

Specifies the maximum number of nested TSXs allowed and the number of subtasks that should be kept idle in storage.

max The maximum number of concurrent (nested) TSXs allowed. The default is 7. The maximum is 255.

idle The maximum number of subtasks, which were attached for previous TSXs, will be kept attached and idle when all TSXs have finished processing. The default is 2. The maximum is 255.

When you run a TSX, a TSX subtask is created. If that TSX calls another TSX (which is considered nested under the calling TSX), another TSX subtask is created if one does not already exist. A nested TSX can be started with a LINK or PROCESS control line or with the RUN command.

WPANLDS

Identifies the BLGCLDSN macro that describes the read/write panel data set. Use this data set to write or copy panels in PMF. If you plan to use PMF, you must specify this keyword.

If you create or modify a panel using PMF, the new or changed panel is stored in the write panel data set (WPANLDS). After the panel is stored in the WPANLDS, you can display it or copy it to another panel data set using PMF. You can also copy panels to the WPANLDS from another panel data set using PMF.

BLGCLUST Macro — Defining a Database

The BLGCLUST assembler macro defines the databases that are available during a Tivoli Information Management for z/OS session. Use this macro once for each database you want to access during a Tivoli Information Management for z/OS session. You can specify that the SDDS for the Tivoli Information Management for z/OS read/write database consist of one VSAM cluster or up to 100 VSAM clusters. You can also specify that the SDDS for the Tivoli Information Management for z/OS-format read-only databases (4, 6, 7, 8, and 9) consist of one VSAM cluster or up to 100 VSAM clusters. However, the user-defined format read-only databases (0, 1, 2, and 3) can have only one cluster.

One BLGCLUST macro must specify NAME=5; that is, the number of the Tivoli Information Management for z/OS read/write database.

You can also specify that the SDIDS for the Tivoli Information Management for z/OS index database consist of one to 100 VSAM clusters.

The syntax of the BLGCLUST macro is:

```
[label]   BLGCLUST   NAME=n
                     ,SDDS=label
                     ,SDIDS=label
                     [,COGENQ={n|10}]
                     [,IDSKEYP=key]
                     [,SDLDS=label]
                     [,SRCHLIMIT=([{enque|0}]
                                  [,{srchwarn|0}]
                                  [,{srchend|0}]))]
                     [,TRIGGER=(t,[xx][,yy])
```

Note: Before Information/Management Version 5, the PRODUCT keyword was required. This keyword is not required in this version but you can still use it to maintain compatibility with old session-parameters members. However, if you specify the PRODUCT keyword, the Tivoli Information Management for z/OS code ignores it.

label A standard assembler language instruction label. This label is optional.

NAME
External name to assign to this database. You cannot use the same name twice. The name must be a single numeric digit (*n*), defined as follows:

0 A user database in a user-defined format
1 A user database in a user-defined format
2 A user database in a user-defined format
3 A user database in a user-defined format
4 A Tivoli Information Management for z/OS read-only database
5 The Tivoli Information Management for z/OS read/write database
6 A Tivoli Information Management for z/OS read-only database (reserved for use with Tivoli Inventory)
7 A Tivoli Information Management for z/OS read-only database
8 A Tivoli Information Management for z/OS read-only database
9 A Tivoli Information Management for z/OS read-only database

SDDS Identifies the label on the BLGCLDSN macro that defines the SDDS. The read/write SDDS (NAME=5) data set is used to store the records created through Tivoli Information Management for z/OS. A read-only SDDS (NAME=4, 6, 7, 8, or 9) data set contains records created when Tivoli Information Management for z/OS accesses the read/write data set (NAME=5) through another session-parameters member.

If you specify a multiple-cluster SDDS for the Tivoli Information Management for z/OS read/write database, you specify only the first SDDS cluster in SDDS=label. Tivoli Information Management for z/OS determines the names of the remaining clusters as explained under “Defining the SDDS” on page 278.

If the session-parameters member is used for defining a user database referenced by the BLGOZUD utility, databases 0, 1, 2, and 3 require a one-cluster SDDS. Otherwise, you receive an error message when you attempt to assemble the session-parameters member.

The databases cannot share an SDDS; that is, you can use only one BLGCLUST macro to reference any BLGCLDSN macro that defines an SDDS. You must specify this keyword for all database definitions.

SDIDS
Identifies the label on the BLGCLDSN macro that defines the SDIDS. The SDIDS data set contains an index for the SDDS. The index is used when searching for or sorting records in the SDDS. A read-only SDIDS (NAME=4, 6, 7, 8, or 9) data set

contains indexes created when Tivoli Information Management for z/OS accesses the read/write data set (NAME=5) through another session-parameters member.

The databases cannot share an SDIDS; that is, you can use only one BLGCLUST macro to reference any BLGCLDSN macro that defines an SDIDS. You must specify this keyword for all database definitions.

COGENQ

Specifies the number of fields to be cognized when a record is filed while holding an enqueue. The enqueue is released after this number of fields are cognized to allow other transactions access to the database. If all the identified fields were not cognized during the first enqueue, the enqueue is requested again. When the enqueue is obtained, the remaining fields (up to the specified number for the COGENQ parameter) are cognized. The cycle continues until all the fields marked to be cognized in the record are cognized.

This keyword is optional and used only in a non-sysplex (non-RLS) environment. If sysplex mode is enabled and you specify this keyword, it will be ignored. However, if you do not specify a keyword value in a non-sysplex environment, a default value of *10* is used. You can use any number from *0* through *32 767*.

0 Holds the enqueue until all fields marked to be cognized in the record are cognized.

1 Releases the enqueue after cognizing one field in a record.

2 - 32 767

Holds the enqueue until the specified number of fields are cognized (or uncognized) when a record is filed, or until all the fields in the record are cognized.

idskeyp=*key*

Identifies a label prefix which relates key range values to the SDIDS component of a database. The *idskeyp* keyword is used with multiple-cluster SDIDS databases and is optional. It should consist of 1 to 7 characters. If you specify it, you must also specify the TRIGGER keyword to define the trigger character. The label prefix you enter for *key* is used as a partial label identifier on BLGCLKEY macro statements. In the following assembler example, MYKEY is used as the label prefix. MYKEY must also be found as part of the BLGCLKEY macro label.

```
BLGCLUST NAME=5,TRIGGER=(#,1,6),
         SDDS=DBS5DDS,
         SDIDS=DBSIDS5,IDSKEYP=MYKEY,
...
MYKEY1 BLGCLKEY KEY=BA,TYPE=HEX
MYKEY2 BLGCLKEY KEY=BC,TYPE=HEX
MYKEY3 BLGCLKEY KEY=A
MYKEY4 BLGCLKEY KEY=RN
MYKEY5 BLGCLKEY KEY=RO
```

SDLDS

Identifies the label on the BLGCLDSN macro that defines the SDLDS. The SDLDS data set is used to recover a damaged read/write SDDS.

The databases cannot share an SDLDS; that is, you can use only one BLGCLUST macro to reference any BLGCLDSN macro that defines an SDLDS. You must include this keyword if you want stored records recovered in the event the read/write SDDS becomes unusable.

You can use this keyword only if you specify NAME=5.

SRCHLIMIT

Enables you to specify parameters that deal with the SDIDS enqueue and the searching of the SDIDS:

enque Specifies the number of records that a search or report can read from the SDIDS before the SDIDS is released to other users who are performing updates. Define one session-parameters member for interactive users and another session-parameters member for batch reports. The batch reports should have a lower value specified for *enque* than the interactive user.

The enqueue value is ignored when sysplex mode is enabled because data set enqueues are not necessary. Therefore, in a sysplex, you do not have to specify a value for this keyword.

You can use values from *0* to *99 999 999*. Leading zeros are ignored. This keyword is optional, but if you do not specify it, a default value of *0* is used.

If you specify 0 or you do not enter a value for this keyword, Tivoli Information Management for z/OS holds the database as long as it takes to do the search. In some cases, this could take a longer time than you want, and other users cannot access the database until the search is completed.

Note: *Enque* replaces the SRCHENQ keyword. However, the SRCHENQ keyword is still accepted. The two provide the same function. However, if you want to change the value of SRCHENQ, then you must use the SRCHLIMIT parameter.

srchwarn

Specifies the maximum number of SDIDS records that Tivoli Information Management for z/OS reads before it displays a message to warn the user that a search or sort has caused an excessive number of reads.

You can use values from *0* to *99 999 999*; leading zeros are ignored. This keyword is optional and if you do not specify it, a default value of *0* is used.

A value of *0* specifies that no warning message is issued. If the value you specify for *srchend* is greater than 0, then the value you specify for *srchwarn* must be less than or equal to the value you specify for *srchend*. Otherwise, you cannot assemble your session-parameters member. This restriction prevents you from setting a warning limit that can never be reached, because the search or sort ends when *srchend* is reached.

srchend

Specifies the maximum number of SDIDS records that Tivoli Information Management for z/OS reads before ending a search or sort, and then issues a message that tells you the search or sort ended. Specifying a maximum value can help keep a search or sort from performing excessive reads of the SDIDS.

You can use values from *0* to *99 999 999*; leading zeros are ignored. This keyword is optional, and if you do not specify it, a default value of *0* is used.

A value of *0* means that Tivoli Information Management for z/OS can perform an unlimited number of reads of the SDIDS, and the search or sort will run to completion. If this value is greater than 0, then it must be equal to or greater than the value you specified for *srchwarn*. Otherwise, you cannot to assemble your session-parameters member. This restriction prevents you from setting a warning limit that can never be reached, because the search or sort ends when *srchend* is reached.

The following are examples and descriptions of various combinations of the SRCHLIMIT values:

SRCHLIMIT=(0,0,0)
The enqueue on the SDIDS is not released, and no messages are issued.

SRCHLIMIT=(100,0,300)
The enqueue on the SDIDS is released after every 100 SDIDS reads, and the search or sort ends after 300 SDIDS reads. This value can also be specified as SRCHLIMIT=(100,,300).

SRCHLIMIT=(0,300,0)
The enqueue on the SDIDS is not released, and a warning message is issued after 300 SDIDS reads. This value can also be specified as SRCHLIMIT=(,300).

SRCHLIMIT=(100,300,300)
The enqueue on the SDIDS is released after every 100 SDIDS reads, and both warning and ending messages are issued after 300 SDIDS reads.

SRCHLIMIT=(100,300,200)
Not valid.

TRIGGER
Indicates that you are using a multiple-cluster database for the SDDS, SDIDS, or both.

T represents the trigger character in the names of the database. You must use this same trigger character when naming the clusters. If you are using multiple clusters for both the SDDS and SDIDS, the same trigger character must be used for both. The trigger character can be an alphabetic, numeric, or national (@, #, $) character.

XX specifies how many SDDS clusters are to be used for the SDDS database components.

YY specifies how many SDIDS clusters are to be used for cognized word bit lists.

For example, `TRIGGER=(#,1,6)` indicates the pound sign will be used as the trigger character for all the clusters, which consist of one SDDS cluster and six SDIDS clusters.

The value of *xx* or *yy* must have a numeric value in the range from *1* through *99*. If the trigger character and the values are omitted, then there is no trigger character and all of the data is stored in one VSAM cluster. If the trigger character is specified, but no value is specified for *xx* or *yy*, the specified trigger character is used and the SDDS data is stored in 5 VSAM clusters and the SDIDS data is stored in 1 VSAM cluster. See the naming conventions for a multiple-cluster SDDS given on page 278 for rules on specifying this character.

Note: Do not specify this parameter for databases 0, 1, 2, and 3 because you cannot use multiple-cluster SDDSs or multiple-cluster SDIDSs for those databases.

BLGCLDSN Macro — Specifying the Name and Attributes of a Data Set

The BLGCLDSN assembler macro assigns a name to a Tivoli Information Management for z/OS data set (for example, a read panel data set) and defines the data set's attributes. Use one BLGCLDSN macro for each data set you name.

The syntax of the BLGCLDSN macro is:

```
[label]   BLGCLDSN   [DSN=dsname]
                     [,FILE=ddname]
                     [,RDONLY={YES|NO}]
```

Note: Before Information/Management Version 5, the LSR keyword was required. This keyword is not required in this version but you can still use it to maintain compatibility with old session-parameters members. However, if you specify the LSR keyword, the Tivoli Information Management for z/OS code ignores it.

label A standard assembler language instruction label. The label you specify must not begin with the characters BL or SYS. This label is required and you must specify it as a parameter in either the BLGPARMS or BLGCLUST macro. Labels in the form *SDDSx#y* are reserved for generated names. If you specify a label in this form, an assembly error occurs when you attempt to generate your sessions.

DSN Name of the data set that you are specifying. You must use a cataloged data set and specify 1–44 characters for its name. This keyword is optional for the RFTDS. Otherwise, this keyword is required.

If you are using a multiple-cluster SDDS, specify only the first cluster name.

FILE Identifies the ddname for RFTDSs. This keyword is optional; it is only applicable for the BLGCLDSN macro that defines the RFTDS. If you want more than one RFTDS, you must use the FILE keyword to identify a ddname. To use multiple RFTDSs, concatenate the RFTDSs to this ddname before starting a Tivoli Information Management for z/OS session. If you concatenate RFTDSs to this ddname, the block sizes of all the concatenated data sets must be identical.

Notes:

1. If you concatenate RFTDSs to this ddname, Tivoli Information Management for z/OS does not use the RFTDS that is specified in the session-parameters member.
2. Tivoli Information Management for z/OS first tries to use the DD statement for the RFTDS. If you omit the FILE keyword or you did not preallocate the DD statement, Tivoli Information Management for z/OS uses the data set name (DSN).

RDONLY

Indicates the access mode for the data set. This keyword applies to panel and dictionary data sets only; it cannot be used for the SDLDS. If you omit the RDONLY keyword, NO is the default.

If you omit the RDONLY keyword or specify a value of NO, you access the data set in read/write mode so that you can write to the data set during the Tivoli Information Management for z/OS session.

If you specify a value of YES, you access the data set in read-only mode.

BLGCLKEY Macro — Defining an SDIDS Key Range String

The BLGCLKEY macro is used when setting up a multiple-cluster SDIDS environment to define the beginning key value for an SDIDS cluster.

Each SDIDS cluster, except the first SDIDS cluster, must have a starting key value defined for it. Do not define a starting key value for the first SDIDS cluster; X'00' is predefined as the starting key value for this cluster. The remaining key values determine which index information is contained in each cluster. Key values can be entered using the BLGCLKEY macro. The control record in each cluster contains the key range information for that cluster.

Each BLGCLKEY macro label must be unique. A single BLGCLKEY macro label may be referenced by multiple *idskeyp* keywords in the BLGCLUST macro. You can put the BLGCLKEY macro anywhere in the session-parameters member, as long as it appears after the CSECT statement and before the BLGGEN macro.

The syntax of the BLGCLKEY macro is:

```
[label]   BLGCLKEY   KEY=value,[TYPE=HEX]
```

label A standard assembler language instruction label. The label you specify must not begin with the characters BL or SYS. This label is required and it must contain the label prefix you defined as the *idskeyp* value in the BLGCLUST macro for the SDIDS. The label prefix can exist anywhere in the label.

```
Example:

BLGCLUST NAME=5,TRIGGER=(#,1,6),
         SDDS=DBS5DDS,
         SDIDS=DBSIDS5,IDSKEYP=MYKEY,
...
MYKEY1 BLGCLKEY KEY=BA,TYPE=HEX
MYKEY2 BLGCLKEY KEY=BC,TYPE=HEX
MYKEY3 BLGCLKEY KEY=A
MYKEY4 BLGCLKEY KEY=RN
MYKEY5 BLGCLKEY KEY=RO
```

KEY Defines the beginning key value for an SDIDS cluster. Specify a string from 1 to 32 characters. If you enter less than 32 characters, Tivoli Information Management for z/OS pads the string on the right with hexadecimal zeros. This keyword is required.

The key of the first record in the first cluster is always assumed to be hexadecimal zeros. This keyword enables you to define the key values for the first record in the remaining clusters. The key value of the last record in the last cluster is assumed to be less than 32X'FF'.

TYPE=HEX

Specifies that the value specified by the KEY keyword should be treated as a hexadecimal value. HEX is the only value permitted for TYPE. The keyword TYPE=HEX is optional. If it is specified, the first two characters of the key value are treated as hexadecimal values. If it is not specified, the first character of the key value is treated as a character value. If you are entering an s-word value for the SDIDS key, you should specify a key type of HEX.

BLGGEN Macro — Beginning the Code-Generation Phase

The BLGGEN macro indicates the end of the Tivoli Information Management for z/OS macro set. This macro indicates that you specified all the Tivoli Information Management for z/OS macro instructions. In addition, this macro begins the code-generation phase for your session-parameters assembler language CSECT. This macro must follow all other Tivoli Information Management for z/OS definition macros, and it must precede the assembler END statement.

The syntax of the BLGGEN macro is:

```
BLGGEN [LIST={YES|NO}]
```

LIST Specifies whether or not the generated control block structures are shown in the assembler listing output. The default is LIST=NO, which indicates the control block structures are not listed.

Sample JCL for a Session-Parameters Member

Figure 32 is a sample job stream that you can use to assemble and link-edit a session-parameters member (BLGSESaa). The JCL member is called BLGALSPM; it is provided in the SBLMSAMP library.

```
//BLGALSPM JOB
//*********************************************************************
//*
//* SAMPLE JCL TO ASSEMBLE AND LINK-EDIT SESSION MODULE USING SESSION
//* MODEL BLGSES00.  IN THIS EXAMPLE, THE SOURCE IS LOCATED IN
//* BLM.SBLMSAMP(BLGSES00), THE INFORMATION MANAGEMENT for z/OS MACROS
//* ARE LOCATED IN BLM.SBLMMACS AND THE MODULE IS PLACED IN
//* BLM.SBLMMOD1
//*
//*********************************************************************
//ASM       EXEC PGM=ASMA90,REGION=1024K,PARM='NODECK'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIB    DD DISP=SHR,DSN=BLM.SBLMMACS
//          DD DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN    DD DISP=(MOD,PASS),DSN=&LOADSET,UNIT=SYSDA,
//             SPACE=(80,(200,50))
//SYSIN     DD DISP=SHR,DSN=BLM.SBLMSAMP(BLGSES00)
//LINK1     EXEC PGM=IEWL,PARM='LIST,XREF,LET,NORENT'
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DISP=SHR,DSN=BLM.SBLMMOD1
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DISP=(OLD,DELETE),DSN=&LOADSET
//          DD *
            ENTRY BLGSES00
            NAME BLGSES00(R)
/*
```

Figure 32. Example: JCL to assemble and link-edit a Session-Parameters Member

Sample Session-Parameters CSECT

Figure 33 on page 341 shows a sample session-parameters member. The sample session-parameters member is called BLGSES00; it is provided in the SBLMSAMP library.

```
************************************************************************
*   SAMPLE SESSION-PARAMETERS MEMBER WITH ONE READ PANEL DATA SET,
*   ONE WRITE PANEL DATA SET, AND A SINGLE-CLUSTER SDDS.
*************************************************************************
         TITLE 'BLGSES00 - SESSION PARAMETERS'
BLGSES00 CSECT
         BLGPARMS  DICTDS=DICTDS,       NAME THE DICTIONARY            X
               RFTDS=RFTS,              NAME THE REPORT FORMAT TABLES  X
               DATEFMT=MM/DD/YYY,       INT/EXT DATE FORMAT            X
               RPANLDS=RPANEL1,         NAME THE READ PANEL DATA SET   X
               WPANLDS=WPANELS          NAME THE WRITE PANEL DATA SET
********       CAS=BLX1,                SPECIFY TARGET BLX-SP SERVER
* UNCOMMENT THE ABOVE LINE AND ADD IT TO THE BLGPARMS MACRO INVOCATION
* IF YOU WANT TO USE THIS MEMBER WITH A BLX-SP SERVER OTHER THAN THE
* DEFAULT BLX-SP SERVER
*
MGMT     BLGCLUST NAME=5,               READ/WRITE CLUSTER             X
               SDDS=MGTSDDS,            NAME THE SDDS                  X
               SDIDS=MGTSDIDS,          NAME THE SDIDS                 X
               SDLDS=MGTSDLDS           NAME THE SDLDS
MGTSDDS  BLGCLDSN DSN=BLM.SDDS
MGTSDIDS BLGCLDSN DSN=BLM.SDIDS
MGTSDLDS BLGCLDSN DSN=BLM.SDLDS
DICTDS   BLGCLDSN DSN=BLM.DICT
RFTS     BLGCLDSN DSN=BLM.SBLMFMT,FILE=RFTDD
RPANEL1  BLGCLDSN DSN=BLM.IBMPNLS,RDONLY=YES
WPANELS  BLGCLDSN DSN=BLM.WPANELS
         BLGGEN
         END
```

Figure 33. Example: Session-Parameters Member with Macros

End of Programming Interface information

D. Session-Parameters Members

E

Defining BLX-SP Parameters Members

This appendix provides:
- The syntax and descriptions of the BLX-SP parameters
- A sample BLX-SP parameters member

This appendix is designed to help you create a BLX-SP parameters member.

Programming Interface information

Defining BLX-SP Parameters Members

BLX-SP parameters members control the operation of the BLX-SP for all user sessions.

Naming a BLX-SP Parameters Member

These members are standard text files contained as members of a PDS. The member name you use must follow these naming rules:
- The name can be 5 or 6 characters long.
- The name must begin with an alphabetic or national (#, $, @) character.
- The remaining characters can be alphabetic, numeric, or national characters.
- The first 4 characters must be the same as the 4 characters of the BLX-SP subsystem that you want to use.

The BLX-SP procedure must specify the PDS in which the specified member resides through a DD statement with the ddname BLXPRM. See "Defining a BLX-SP Procedure" on page 146 for information about the BLX-SP procedure.

Understanding the BLX-SP Parameters Member Syntax Description

In the list of parameters described, some parameters are shown with brackets *[ ]*, braces *{ }*, or vertical bars |. These are for clarification purposes only; they are not part of any keyword in the lists.
- Brackets indicate optional data.
- Braces indicate that you must select one of the choices enclosed within the braces.
- Vertical bars indicate selectable items. You must choose one of these items.
- **Bold, underlined values** in these BLX-SP parameter descriptions are the default values.

Using the BLX-SP Parameters

The BLX-SP parameters members contain the following parameters:

```
APISECURITY={ON|OFF}
[APICHKOUTLIM=hhmmssth]
[DBCS={NO|YES}]
[LOG={ON|OFF}]
[LOGLINES={n|0}]
[LOGSYSOUT={class|A}]
```

```
[LOGTOD={x|(x[,y[,z]])}]
[MAILQ=(queuename,warning_limit,maximum_limit)]
[MAILQWAITTM=hhmmssth]
SHUTDOWNWT=hhmmssth
SHUTDOWNTFY=hhmmssth
[SYSPLEX={YES|NO}]
[TRACE={ON|OFF}]
[TRACELINES={n|0}]
[TRACEPOINTS={x|(x[,y[,z[,...]]]])]
[TRACESYSOUT={class|A}]
[TRACETOD={x|(x[,y[,z]])}]
VSAMRESOURCES=VSAM_resource_load_module_name
[WRITEOPER=code]
```

Notes:

1. The DESTNAMES parameter which was used previously in a shared database environment to specify names of BLX-SP that should receive buffer invalidation messages is no longer supported.
2. You can modify the log and trace parameters through the TRACE/LOG (TL) command. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference* for a description of and instructions for using this command.

APICHKOUTLIM

Specifies a period of time that a record can be checked out by an API application user. When this time period is exceeded for a checked out record, the record is made available to other users.

You must specify this time in the following format: HHMMSSTH. HH, MM, SS, T, and H represent hours, minutes, seconds, tenths of seconds, and hundredths of seconds, respectively. The default is 0 (record is checked out forever until it is checked in).

This keyword enables you to release a checked-out record after the check out period has expired. If the check out period is exceeded, the record is no longer checked out.

When a record is checked out by an API application (through the HLAPI transaction HL04 or LLAPI transaction T104), the following occurs:

- If the record is not already checked out, or if it is checked out and the check out period has expired, the check out period is added to the current clock time and stored in the record.
- If the record is already checked out to a different application ID and the check out period has not expired, an error is returned to the API application to indicate the record is already in use.
- If the record is already checked out to the same application ID, the check out period is reset to a full check out period and saved in the record.

The check out period is checked when the API application performs an Update Record or Delete Record transaction. The check out period is also checked when an interactive user updates or deletes a record.

This keyword is optional. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide* for more information on API transactions for checking out records.

APISECURITY

Specifies whether to activate the BLX-SP security checking for the APIs. The security check validates that the *MVS application user ID* is allowed to access a Tivoli Information Management for z/OS database with the application ID specified in PICAUSRN. Additional information on API security can be found in *Tivoli Information Management for z/OS Application Program Interface Guide*. This is a required parameter, and there is no default value. The parameter member is shipped with a setting of ON, but it can be set to OFF to disable this security checking feature. Because **APISECURITY** is a required parameter, the BLX-SP initialization process will fail with a parsing error if it is not added to a parameter member from a previous version before use.

DBCS Specifies whether the BLX-SP supports DBCS data. If you do not use DBCS data, specify *NO* or omit the parameter (the default value is *NO*) so your installation's performance is not affected by the overhead required to process DBCS data.

LOG Specifies whether to activate (*ON*) or not to activate (*OFF*) the BLX-SP log data set.

If you enter the LOG keyword, you must specify one of the valid keyword values. If you omit this keyword, the default LOG value is *OFF*.

LOGLINES

Instructs the BLX-SP to close and free the BLX-SP log data set and to allocate and activate a new BLX-SP log data set whenever the number of lines (records) in the data set meets or exceeds the limit value (*n*) that you specify.

Valid values for *n* are:

- *0*, to indicate that you do not want data set switching to occur, regardless of the number of lines (records) in the data set.
- Any whole number from *50* to *200 000* (inclusive).

If you omit this keyword, the default LOGLINES value is *0*.

LOGSYSOUT

Specifies the SYSOUT class (*c*) to which the BLX-SP log data set is allocated.

Valid values for *c* are *A–Z* and *0–9*.

If you omit this keyword, the default SYSOUT class value is *A*.

LOGTOD

Specifies from one to three time-of-day values. At the specified time-of-day, BLX-SP closes and frees the BLX-SP log data set, and allocates and activates a new log data set (regardless of any value entered on LOGLINES). If you specify more than one time-of-day value, you must enclose the values in parentheses. You must specify each time-of-day value as HH:MM:SS. HH, MM, and SS represent hours, minutes, and seconds, respectively, and the three values are separated by colons (:).

A 24-hour, 60-minute, 60-second clock is the unit of measure for all time-of-day values. When you specify a time-of-day value, you must include hours, minutes, and seconds. Hours must range from *00* to *23*, minutes must range from *00* to *59*, and seconds must range from *00* to *59*. To specify midnight, you must use *00:00:00*.

If you specify this parameter with a null value or with any other value that contains fewer than eight characters, you receive an error. If you omit this keyword, there is no timed log switching.

MAILQ

Specifies the BLX-SP queue to use for e-mail notification. To use the queue, you must set a warning limit and define the maximum number of mail items that can be placed on the queue. For example, MAILQ=(PROBLEM,5,10) specifies that the problem queue will issue a warning when the sixth mail notice is queued. The queue is full after the 10th piece of mail is queued.

queuename
Specifies the mail queue name defined to the BLX-SP. The queue name can be 1 to 8 alphabetic or numeric characters. Up to three names can be defined as separate MAILQ parameter entries. Default queue names are MAILQ1, MAILQ2, and MAILQ3.

warning_limit
Specifies the number of items that can be placed on the BLX-SP mail queue before a message is issued to the operator stating that the limit has been reached. Valid values are between 0 (no mail queuing) and 32767. The default is 0.

maximum_limit
Specifies the number of items that can be on the BLX-SP mail queue before a message is issued to the operator stating that the queue is full. Any new mail items sent to the queue beyond this limit are discarded. Valid values are between 0 (no mail queueing) and 32767. The default is 0.

All three values are required. If a parameter value is missing, invalid, or out of range, the BLX-SP service provider will not start. Currently no check is made to ensure the maximum limit is greater than or equal to the warning limit.

MAILQWAITTM
Specifies a period of time that the BLX-SP is to wait for mail to clear off the mail queue(s) before shutting down. The specified wait time is broken down into 10 second intervals, and the BLX-SP checks every 10 seconds. Once the queues are cleared, the BLX-SP continues the shutdown. If the mail queues are not cleared by the time specified in this parameter, the queues are flushed (mail items are lost) and shutdown processing continues.

After shutdown is started, you cannot query the mail queues to determine the queue status (number of items on the queues). The only way to clear the queues is to have one or more TSXs using the DEQMAIL control line to extract items from the queues until they are empty. Use care in choosing the MAILQWAITTM value so that it supports the minimum amount of time that you can wait for the BLX-SP to shut down.

You must specify this time in the following format: HHMMSSTH. HH, MM, SS, T, and H, represent hours, minutes, seconds, tenths of seconds, and hundredths of seconds, respectively. The default is 0 (no wait time).

SHUTDOWNWT
Specifies a period of time that the BLX-SP is to continue processing after receiving a STOP command. This time permits Tivoli Information Management for z/OS users to complete any processing that was active when the operator issued the STOP command. Users cannot start Tivoli Information Management for z/OS during this time, but those already running Tivoli Information Management for z/OS can continue until the time expires. If no users are active when the MVS STOP command is issued, the BLX-SP stops processing immediately regardless of the SHUTDOWNWT interval.

You must specify this time in the following format: HHMMSSTH. HH, MM, SS, T, and H, represent hours, minutes, seconds, tenths of seconds, and hundredths of seconds, respectively.

This keyword is required.

SHUTDOWNTFY
Specifies the amount of time between operator notification messages. After the operator issues the STOP command, Tivoli Information Management for z/OS sends messages to the operator periodically to indicate the number of users who still have processing active.

You must specify this interval in the following format: HHMMSSTH. HH, MM, SS, T, and H, represent hours, minutes, seconds, tenths of seconds, and hundredths of seconds, respectively.

This keyword is required.

SYSPLEX
Specifies whether to enable sysplex mode.

SYSPLEX=YES enables sysplex mode for the BLX-SP and all its users and is required for sysplex data sharing. If you are not sharing VSAM data sets, SYSPLEX=YES is optional. When sysplex mode is enabled, all VSAM data sets are accessed using VSAM record-level sharing (RLS). Data sets are not accessed using local shared resource (LSR) or nonshared resource (NSR) buffer pools.

Note: If SYSPLEX=YES is specified, the BLX-SP parameter VSAMRESOURCES keyword is ignored.

NO specifies that the BLX-SP should not run in sysplex mode. VSAM data sets are accessed using local shared resource (LSR) buffer pools and cross-memory support, or nonshared resources (NSR), depending on what is defined for your VSAM resources. Sharing of VSAM data sets is not allowed when SYSPLEX=NO.

This keyword is optional. If you omit this keyword, the default SYSPLEX value is *NO*.

TRACE
Specifies whether to activate (*ON*) or not to activate (*OFF*) the BLX-SP trace data set.

If you enter the TRACE keyword, one of the valid keyword values is required. If you omit this keyword, the default TRACE value is *OFF*.

Note: The trace data set collects information for diagnostic purposes. Do not activate the trace data set unless you experience problems with Tivoli Information Management for z/OS and your Tivoli service representative supplies you with trace points.

TRACELINES
Instructs the BLX-SP to close and free the BLX-SP trace data set and to allocate and activate a new BLX-SP trace data set whenever the number of lines (records) in the data set meets or exceeds the limit value (*n*) that you specify.

This keyword has no default. Valid values for *n* are:

- *0*, to indicate that you do not want data set switching to occur, regardless of the number of lines (records) in the data set
- Any whole number from *50* to *200 000* (inclusive).

If you omit this keyword, the default TRACELINES value is *0*.

TRACEPOINTS
Specifies one or more trace points to set. The values *x*, *y*, and *z*, are specified as decimal digits within the range of 1 to 64. Each number represents a trace point to turn on. If you omit the TRACEPOINTS keyword, all trace points are initially set to OFF.

TRACESYSOUT
Specifies the SYSOUT class to which the BLX-SP trace data set is allocated.

If you omit this keyword, the default SYSOUT class value is *A*.

TRACETOD
Specifies from one to three time-of-day values. At the specified time-of-day, the BLX-SP closes and frees the BLX-SP trace data set, and allocates and activates a new BLX-SP trace data set (regardless of any value entered on TRACELINES). If you specify more than one time-of-day value, you must enclose the values in parentheses. You must specify each time-of-day value as HH:MM:SS. HH, MM, and SS represent hours, minutes, and seconds, respectively, and the three values are separated by colons (:).

A 24-hour, 60-minute, 60-second clock is the unit of measure for all time-of-day values. When you specify a time-of-day value, you must include hours, minutes, and seconds. Hours must range from *00* to *23*, minutes must range from *00* to *59*, and seconds must range from *00* to *59* To specify midnight, you must use *00:00:00*.

VSAMRESOURCES
Specifies the name of the VSAM resource definition module constructed with the VSAM BLDVRP and BLX-SP BLXDSN, BLXNSR, and BLXGEN macros. You can provide additional VSAM resource definitions to the BLX-SP by using the BLX-SP ADDVDEF command. However, when you restart the BLX-SP, you lose the data that these definitions contain. Therefore, you may want to include these additional resource definitions in VSAMRESOURCES.

This keyword is required when sysplex mode is not enabled (BLX-SP parameter is SYSPLEX=NO) or when the BLX-SP SYSPLEX keyword is omitted. This keyword is ignored when SYSPLEX=YES.

See “BLXDSN Macro — Connecting VSAM Resources to a VSAM Data Set” on page 303, “BLXNSR Macro — Specifying the Number of Nonshared Resource Placeholders” on page 303, and “BLXGEN Macro — Generating the BLX-SP VSAM Resource Definition Member CSECT” on page 306, which describe the macros used to specify VSAM resources.

WRITEOPER
Specifies the default write to operator (WTO) routing code. Valid values are from *1* to *128*. All WTOs that are not a result of command responses are automatically routed to this code. To determine the routing codes for a console, you can do one of the following:

- Display console characteristics by issuing the `DISPLAY CONSOLES,A` command from a console

- Review the ROUTCODE parameter of the CONSOLE statements in the CONSOLxx member of SYS1.PARMLIB.

Refer to the *OS/390 MVS: System Commands* for more information about consoles and routing codes.

Coding a BLX-SP Parameters Member

There are several basic rules you must follow when you code a BLX-SP parameters member.

- Comments must begin with **/*** and end with ***/**.
- Comments can be in any column between 1 and 72, inclusive.
- Nothing can be present in column 1 (except comments).
- Nothing can be present past column 72.
- Begin the parameters member with a statement identifier of BLXPRM.
- You can separate parameters with commas. Parameters do not have to appear on separate lines.
- End the parameters member with a semicolon (**;**) after the last parameter.

Sample BLX-SP Parameters Member

Figure 34 on page 350 shows a sample BLX-SP parameters member. See member BLX100 in the SBLMSAMP library. For a sample BLX-SP parameters member that uses sysplex data sharing, see member BLX1SH in SBLMSAMP or refer to Figure 14 on page 160.

```
/***********************************************************************/
/*         BLX-SP OPERATING PARAMETERS                                 */
/***********************************************************************/

 BLXPRM                              /* SPECIFY BLX-SP PARAMETERS     */
     /************************************************************/
     /*   BLX-SP TRACE OPTIONS                                   */
     /************************************************************/
        TRACE=OFF,                   /* DON'T PRODUCE TRACE OUTPUT    */
     /************************************************************/
     /*   BLX-SP LOG OPTIONS                                     */
     /************************************************************/
        LOG=ON,                      /* PRODUCE LOG INFORMATION       */
        LOGSYSOUT=A,                 /* JES SYSOUT CLASS FOR LOG DS   */
        LOGLINES=0,                  /* MAX # OF LINES IN A LOG DS    */
     /************************************************************/
     /*   BLX-SP SHUT DOWN OPTIONS                               */
     /************************************************************/
        SHUTDOWNWT=00050000,         /* SHUTDOWN WAIT TIME HHMMSSTH   */
        SHUTDOWNTFY=00001000,        /* SHUTDOWN NOTIFY WT HHMMSSTH   */
     /************************************************************/
     /*   BLX-SP MESSAGE ROUTING OPTIONS                         */
     /************************************************************/
        WRITEOPER=1,                 /* DEFAULT WTO ROUTING CODE      */
     /************************************************************/
     /*   API OPTIONS                                            */
     /************************************************************/
        APISECURITY=XXX,             /* Replace XXX with ON or OFF    */
        APICHKOUTLIM=00000000,       /* HHMMSSSTH - NO LIMIT SET      */
     /************************************************************/
     /*   MAILQ OPTIONS                                          */
     /************************************************************/
     /* MAILQ=(PROBLEM,5,10)         /* DEFINE FIRST MAIL QUEUE       */
     /* MAILQ=(CHANGE,5,10)          /* DEFINE SECOND MAIL QUEUE      */
     /* MAILQ=(ACTIVITY,5,10)        /* DEFINE THIRD MAIL QUEUE       */
     /* MAILQWAITTM=00050000,        /* SUSPEND SHUTDOWN FOR HHMMSSTH */
     /*                              /* TO PROCESS MAIL QUEUES.       */
     /************************************************************/
     /*   BLX-SP VSAM RELATED OPTIONS                            */
     /************************************************************/
        VSAMRESOURCES=BLXVDEF;       /* BLXDSN, BLXNSR, BLXGEN AND    */
                                     /* THE VSAM BLDVRP MACROS        */
```

Figure 34. Example: BLX-SP Parameters Member (for use in a non-sysplex environment)

End of Programming Interface information

F

Resource Names That Tivoli Information Management for z/OS Enqueues On

The following table lists the resource names that Tivoli Information Management for z/OS enqueues on. This information may be useful to you for diagnostic purposes, or if you are sharing data sets across z/OS systems through use of a component other than global resource serialization.

When setting up GRS or any other enqueue manager, note that the resources enqueued with a scope of SYSTEMS must be enqueued across systems; those with a scope of SYSTEM should only be enqueued within a single system.

Major Resource Names

Tivoli Information Management for z/OS enqueues on the *major* resource names shown in following table. Also listed are the enqueue scopes. The minor resource names are described in the Notes column.

Major Resource Name	Enqueue Scope – Non-Sysplex	Enqueue Scope – Sysplex Enabled	Notes
BLGAPI	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.
BLGDICTN	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.
BLGMAIL	SYSTEM	SYSTEM	Minor name identifies the mail queue from which mail is currently being dequeued.
BLGPMFPU	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.
BLGRNID	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.
BLGSEQN	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.
BLGSDIDS	–	SYSTEMS	Minor name identifies a data set name. Resource name is used only when sysplex support is enabled.
BLGUT4	SYSTEM	SYSTEMS	Minor name is the VSAM data set name of the SDLDS.
BLGVCGZR	SYSTEM	SYSTEMS	Minor name identifies a record in a data set.

Major Resource Name	Enqueue Scope – Non-Sysplex	Enqueue Scope – Sysplex Enabled	Notes
BLXCAS	SYSTEM	SYSTEM	Minor name is *xxxx* where *xxxx* is the 4-character BLX-SP subsystem name.
BLXDASDS	SYSTEM	–	Minor name identifies a data set name. Resource name is used only when sysplex support is not enabled.
BLXDSCTL	SYSTEMS	SYSTEMS	Minor name identifies a data set name.
BLXLCDS	SYSTEM	SYSTEM	Minor name identifies a data set name.
BLXSNAP	SYSTEM	SYSTEM	Minor name identifies a data set name.
BLXSPCAS	SYSTEM	SYSTEM	Minor name is STOP *xxxx* where *xxxx* is the 4-character BLX-SP subsystem name.

G

Translate Tables

This appendix displays the Latin and non-Latin alphabet translate tables, used for display, and the Latin and non-Latin blank substitution translate tables. The Latin and non-Latin alphabet translation tables are used for displaying and indexing (cognizing) data in the database. Each table denotes the first hexadecimal digit in the first column on the left (for example, 0x) and the second hexadecimal digit across the top (for example, x7).

Latin and Non-Latin Translate Tables

Table 18. Latin Alphabet Translate Table for Displaying Data

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1x	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2x	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
3x	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
4x	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
5x	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
6x	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
7x	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
8x	80	C1	C2	C3	C4	C5	C6	C7	C8	C9	8A	8B	8C	8D	8E	8F
9x	90	D1	D2	D3	D4	D5	D6	D7	D8	D9	9A	9B	9C	9D	9E	9F
Ax	A0	A1	E2	E3	E4	E5	E6	E7	E8	E9	AA	AB	AC	AD	AE	AF
Bx	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
Cx	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
Dx	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ex	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
Fx	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Table 19. Non-Latin Alphabet Translate Table for Displaying Data

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1x	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2x	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
3x	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
4x	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Table 19. Non-Latin Alphabet Translate Table for Displaying Data (continued)

5x	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
6x	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
7x	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
8x	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
9x	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
Ax	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
Bx	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
Cx	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
Dx	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ex	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
Fx	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Table 20. Latin Blank Substitution Translate Table, for Cognizing Data

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
1x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
2x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
3x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
4x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
5x	50	40	40	40	40	40	40	40	40	40	40	5B	5C	40	40	40
6x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
7x	40	40	40	40	40	40	40	40	40	40	40	7B	7C	40	40	40
8x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
9x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Ax	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Bx	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Cx	40	C1	C2	C3	C4	C5	C6	C7	C8	C9	40	40	40	40	40	40
Dx	40	D1	D2	D3	D4	D5	D6	D7	D8	D9	40	40	40	40	40	40
Ex	40	40	E2	E3	E4	E5	E6	E7	E8	E9	40	40	40	40	40	40
Fx	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	40	40	40	40	40	40

Table 21. Non-Latin Blank Substitution Translate Table, for Cognizing Data

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
1x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
2x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
3x	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
4x	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
5x	50	51	52	53	54	55	56	40	58	40	5A	5B	5C	5D	5E	5F
6x	60	61	40	40	40	40	40	40	40	40	6A	6B	6C	6D	6E	6F
7x	40	40	40	40	40	40	40	40	40	79	7A	7B	7C	7D	7E	7F

Table 21. Non-Latin Blank Substitution Translate Table, for Cognizing Data (continued)

8x	40	81	82	83	84	85	86	87	88	89	8A	40	8C	8D	8E	8F
9x	90	91	92	93	94	95	96	97	98	99	9A	40	40	9D	9E	9F
Ax	40	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	40	AC	AD	AE	AF
Bx	40	40	40	40	40	40	40	40	40	40	40	BB	BC	BD	BE	BF
Cx	40	C1	C2	C3	C4	C5	C6	C7	C8	C9	40	40	40	40	40	40
Dx	40	D1	D2	D3	D4	D5	D6	D7	D8	D9	40	40	40	40	40	40
Ex	E0	40	E2	E3	E4	E5	E6	E7	E8	E9	40	40	40	40	40	40
Fx	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	40	40	40	40	40	40

H

Programming Interface Macros

This appendix lists the macros that Tivoli Information Management for z/OS provides as programming interfaces for customers.

Note: Do not use any Tivoli Information Management for z/OS macros other than those identified in this section as programming interfaces.

Executable Macros

BLGCKDSN
BLGCKNUM
BLGCLDSN
BLGCLGV
BLGCLKEY
BLGCLMSG
BLGCLSDS
BLGCLSEP
BLGCLSKY
BLGCLUST
BLGGEN
BLGGENPT
BLGPARMS
BLGREPDB
BLGREPDD
BLXCKDSN
BLXCKNUM
BLXCLGV
BLXCLMSG
BLXDMSGG
BLXDSN
BLXGEN
BLXMODID
BLXNSR

Mapping Macros

BLGUCCB
BLGUHICA
BLGUHICC
BLGUHIDM
BLGUHIVP
BLGUPALT
BLGUPDB
BLGUPIAT
BLGUPICA
BLGUPICC
BLGUPIDT
BLGUPIHT
BLGUPIMB
BLGUPIPT
BLGUPIRT
BLGUSLB
BLGUSRB
BLGUSUB
BLGUTSCA
BLMVATSR

I

Tivoli Information Management for z/OS Program Data Sets and Sample Members

This appendix lists the Tivoli Information Management for z/OS program data sets and the members of the Tivoli Information Management for z/OS sample library, SBLMSAMP.

Program Data Sets

All the members of the Tivoli Information Management for z/OS data set libraries supplied with Tivoli Information Management for z/OS Version 7.1 have member names beginning with BLG, BLH, BLM, BLX, BTN, EYL, or EYM.

The Tivoli Information Management for z/OS program is distributed on tape (with some features provided on separate tapes), and you must install it using SMP/E, as outlined in the *Program Directory*. During SMP/E installation, several data sets are created. Table 22 lists and describes these data sets.

Table 22. Target Data Set Libraries

File Name	Contents
SBLMCMOD	The Tivoli Enterprise Console Adapter feature
SBLMDICT	The Tivoli Information Management for z/OS dictionary
SBLMEXEC	Installation Tailoring Facility REXX EXECs
SBLMFMT	Report format tables (RFTs) that are used to generate reports, relational data mapping tables (RDMTs) used by the DB2 Extract Facility, and program interface data tables (PIDTs) and program interface pattern tables (PIPTs) that define problem, change, and configuration data models for use with the application program interfaces (APIs).
SBLMHTMV	Data set used with the web connector feature
SBLMINST	Installation Tailoring Facility files and panels
SBLMMACS	Macros that you use to assemble session-parameters members and API control block mapping macros
SBLMMOD1	The Tivoli Information Management for z/OS load modules
SBLMPNLS	The Tivoli Information Management for z/OS panels
SBLMRCDS	Records shipped with Tivoli Information Management for z/OS for special purposes (such as data model records)
SBLMREXD	Data set used to support using DBCS data with the Tivoli Information Management for z/OS Desktop
SBLMREXX	Data set used with the web connector feature

Table 22. Target Data Set Libraries (continued)

File Name	Contents
SBLMSAMP	Sample job control language (JCL), ISPF panel style members, ISPF PDF edit macro BLGISMAC, and source statements used to build the PIDTs and PIPTs contained in BLM..SBLMFMT. Also contains interface materials for the Integration Facility and API and NetView Bridge Adapter examples.
SBLMSRC1	Message CSECT source that you can use to customize your messages
SBLMSTUB	Stub modules for resolving external references during SMP/E installation
SBLMTSX	TSX REXX EXECs
SBLMTXT1	Common parts needed for installation of optional features

Sample Members

Table 23 describes the samples that are supplied with Tivoli Information Management for z/OS. These members are located in the SBLMSAMP sample library. Samples are listed for the base product and optional features.

Table 23. Description of Sample Members in SBLMSAMP Library

Member Name	Purpose
BLGABDBG	GML source for Debug action bar
BLGABDIA	GML source for Dialog action bar
BLGABENV	GML source for Environment action bar
BLGABHLP	GML source for Help action bar
BLGABOPT	GML source for Options action bar
BLGABPMF	GML source for PMF action bar
BLGABREC	GML source for Record action bar
BLGABSCR	GML source for Scroll action bar
BLGABSER	GML source for Search action bar
BLGABSRC	GML source for SRC action bar
BLGABWIN	GML source for Window action bar
BLGALSPM	Assembles and link-edits a BLG session-parameters member. See Figure 32 on page 340.
BLGALTER	Migrates existing VSAM data sets so that they are enabled for VSAM RLS
BLGARCJ	Runs the Archiver
BLGARCP	Provides Archiver input parameters
BLGBSPCX	Procedure to start the NetView Bridge Adapter. Refer to the *Tivoli Information Management for z/OS Guide to Integrating with Tivoli Applications*
BLGCCHK	Obtains information for the control PDBs for the HLAPI’s delete, check-in, and check-out transactions
BLGCCRT	Obtains information for the control PDBs for the HLAPI’s create transaction
BLGCINQ	Obtains information for the control PDBs for the HLAPI’s inquiry transaction
BLGCOBT	Obtains information for the control PDBs for the HLAPI’s obtain record transaction
BLGCPDB	Definition of the HLAPI’s PDB control block in C; used by BLGNBSRC
BLGCRET	Obtains information for the control PDBs for the HLAPI’s retrieve transaction
BLGCTPCA	Definition of the NetView Bridge Adapter’s TPCS control block in C; used by BLGNBSRC

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLGCTRAN	Primary selection panel
BLGCTSP	Obtains information for the control PDBs for the HLAPI's invoke TSP transaction
BLGCUPD	Obtains information for the control PDBs for the HLAPI's update record transaction
BLGDATAB	Defines the Tivoli Information Management for z/OS database
BLGDATE8	JCL to convert dates from 10 to 8 characters on panels
BLGDG1	JCL that shows a sample of defining a GDG data set
BLGDICT	Defines the dictionary data set (DICTDS). Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGD2000	Sample date conversion routine used when records have dates in two different external date formats
BLGEXPND	Expands a database. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGHFCRT	Field help for the create transaction control PDBs
BLGHFDEL	Field help for the delete transaction control PDBs
BLGHFFFT	Help panel for freeform text entry panel
BLGHFINQ	Field help for the inquiry transaction control PDBs
BLGHFLDA	Help panel for HLAPI input PDBs
BLGHFLDB	Help panel for HLAPI input PDBs
BLGHFLD1	Help panel for HLAPI input PDBs
BLGHFLD2	Help panel for HLAPI input PDBs
BLGHFOBT	Field help for the obtain record transaction control PDBs
BLGHFRET	Field help for the retrieve transaction control PDBs
BLGHFTSP	Field help for the invoke TSP transaction control PDBs
BLGHFUPD	Field help for the update transaction control PDBs
BLGHMAIN	Help panel for main selection panel
BLGHPCRT	Help panel explaining PF key function for create transaction
BLGHPINQ	Help panel explaining PF key function for inquiry transaction
BLGHPMSG	Help panel explaining PF key function on messages panel
BLGHPORT	PF key help after an inquiry transaction
BLGHPPF1	General PF key help
BLGHPPF2	General PF key help
BLGHPREC	Help panel explaining the results of an inquiry transaction
BLGHPRET	Help panel explaining PF key function for retrieve transaction
BLGHPSTA	PF key help on input PDB panels
BLGHPUPD	Help panel explaining PF key function for update transaction
BLGIATTR	GML source for panel)ATTR section
BLGICRT	Panel used to obtain input PDB information for create transaction
BLGIFFT	Panel used to obtain freeform text data for inquiry transaction
BLGIGMLA	GML source for panel BLGISPFA
BLGIGMLD	GML source for panel BLGISPFD
BLGIGMLE	GML source for panel BLGISPFE

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLGIGMLI	GML source for panel BLGISPFI
BLGIINIT	GML source for panel)INIT section
BLGIINQ	Panel used to obtain input PDB information for inquiry transaction
BLGIPROC	GML source for panel)PROC section
BLGIRET	Panel used to obtain input PDB information for retrieve transaction
BLGISGUI	Runs Tivoli Information Management for z/OS as a background job under ISPF with the GUI parameter specified
BLGISMAC	ISPF PDF edit macro
BLGISPFA	ISPF panel containing the INFO window for enhanced panel style. Action bar contains “Environment Dialog Record Window ISPF Help” plus administrator actions for “PMF SRC Debug”.
BLGISPFD	ISPF panel shipped with V5 that contains the window for standard panel style, that is no action bars
BLGISPFE	ISPF panel containing the INFO window for enhanced panel style. Action bar contains “Environment Dialog Record Window ISPF Help”.
BLGISPFI	ISPF panel containing the INFO window for enhanced panel style. Action bar contains “Environment Dialog Record Window ISPF Help” plus inquiry actions for “Search Scroll”.
BLGISTAT	Panel used to obtain additional input PDB information
BLGIUPD	Panel used to obtain input PDB information for update transaction
BLGKDATA	Used to link edit non-Latin alphabet table
BLGLDICT	Loads the DICTDS
BLGLRPNL	Loads the read panel data set (RPANLDS). Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGNBLNK	JCL to link-edit BLGNBSRC with BLGBUSR
BLGNBREX	REXX EXEC, runs under NetView and calls the NetView Bridge Adapter
BLGNBSRC	C source code for a user-written program that can be called by the NetView Bridge Adapter
BLGOFFT	Panel for obtaining freeform text data set for retrieve transaction
BLGOMSG	Messages panel for inquiry transaction
BLGORECS	Display record IDs that met search criteria on inquiry transaction
BLGORET	Display panel for a record retrieved by the inquiry transaction
BLGPANEL	The list of new, changed, and deleted panels for Tivoli Information Management for z/OS Version 7.1. This list does not include message and help panels.
BLGPVARS	Contains profile variables. Refer to the *Tivoli Information Management for z/OS Program Administration Guide and Reference*.
BLGRECV	CLIST to receive the offload data set sent by the SEND component of the Automatic Log Save function
BLGRPNL	Defines the read panel data set (RPANLDS).
BLGSES00	Sample session-parameters member. See Figure 33 on page 341.
BLGSRNID	Assembler user exit that returns the RNID of a record to the calling TSP for record manipulation. Refer to the *Tivoli Information Management for z/OS Panel Modification Facility Guide*.
BLGTECAD	Sample JCL for running the TEC adapter TSX
BLGTOAMS	Produces SAM reports

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLGUSERJ	Sample JCL used with the Archiver for user relationships
BLGUTIMC	External time-conversion routine
BLGUTRJ	Formats a recovery log data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT1J	Rebuilds SDIDS from corresponding SDDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT1MJ	Initializes/migrates the SDIDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT10J	JCL used with the case converter utility, BLGUT10
BLGUT10P	Input parameters for use with the BLGUT10 utility
BLGUT20J	Obtains statistics about SDDS data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT21J	Obtains statistics about SDIDS data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT22J	Obtains statistics about VSAM panel data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT3J	Restores database from log data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT4J	Offloads log data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT5FJ	Offloads VSAM dictionary data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT5J	Loads or maintains VSAM dictionary data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT6FJ	Offloads panels from a VSAM panel data set to a PDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT6J	Loads or maintains VSAM panel data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT6MJ	Migrates field lengths and validation patterns on panels. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT7J	Creates 7-byte key SDDS from 8-byte key SDDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT8J	Builds PIDT and PIPT tables. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGUT9J	Sets database option of reusing SDDS position numbers for new records. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGU23BJ	Backs up the SDDS. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGU23PJ	Prunes and sorts offloaded log data. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGU23RJ	Restores the SDDS from a master backup data set. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGU23UJ	Updates master backup data with pruned log data. Refer to the *Tivoli Information Management for z/OS Operation and Maintenance Reference*.
BLGYACCS	Produces a create type PIDT for activity records

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLGYACIS	Produces an inquiry type PIDT for activity records
BLGYACLS	Produces an inquiry type PIDT for activity records related to change records
BLGYACRS	Produces a retrieve type PIDT for activity records
BLGYACUS	Produces an update type PIDT for activity records
BLGYCAIS	Produces an inquiry type PIDT for call records
BLGYCHAS	Produces an add record relations PIDT for change records
BLGYCHCS	Produces a create type PIDT for change records
BLGYCHIS	Produces an inquiry type PIDT for change records
BLGYCHRS	Produces a retrieve type PIDT for change records
BLGYCHUS	Produces an update type PIDT for change records
BLGYDAIS	Produces an inquiry type PIDT for data attribute records
BLGYDCCS	Produces a create type PIDT for data center records
BLGYDCIS	Produces an inquiry type PIDT for data center records
BLGYDCRS	Produces a retrieve type PIDT for data center records
BLGYDCUS	Produces an update type PIDT for data center records
BLGYDYNS	Source PIDT for the Archiver
BLGYHCCS	Produces a create type PIDT for configuration hardware component records
BLGYHCIS	Produces an inquiry type PIDT for configuration hardware component records
BLGYHCRS	Produces a retrieve type PIDT for configuration hardware component records
BLGYHCUS	Produces an update type PIDT for configuration hardware component records
BLGYHFAS	Produces PIDT to add feature relations to hardware component records
BLGYHFCS	Produces a create type PIDT for configuration hardware component feature records
BLGYHFIS	Produces an inquiry type PIDT for configuration hardware component feature records
BLGYHFLS	Produces an inquiry type PIDT used to list hardware features of a component record
BLGYHFRS	Produces a retrieve type PIDT for configuration hardware component feature records
BLGYHFUS	Produces an update type PIDT for configuration hardware component feature records
BLGYHNCS	Produces a create type PIDT for financial hardware records
BLGYHNIS	Produces an inquiry type PIDT for financial hardware records
BLGYHNRS	Produces a retrieve type PIDT for financial hardware records
BLGYHNUS	Produces an update type PIDT for financial hardware records
BLGYHSCS	Produces a create type PIDT for configuration hardware subcomponent records
BLGYHSIS	Produces an inquiry type PIDT for configuration hardware subcomponent records
BLGYHSRS	Produces a retrieve type PIDT for configuration hardware subcomponent records
BLGYHSUS	Produces an update type PIDT for configuration hardware subcomponent records
BLGYHXAS	Produces PIDT to add connection relations to hardware component records
BLGYHXCS	Produces a create type PIDT for configuration hardware component connection records
BLGYHXIS	Produces an inquiry type PIDT for configuration hardware component connection records
BLGYHXLS	Produces an inquiry type PIDT for hardware component-related connection list
BLGYHXRS	Produces a retrieve type PIDT for configuration hardware component connection records
BLGYHXUS	Produces an update type PIDT for configuration hardware component connection records

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLGYPRCS	Produces a create type PIDT for problem records
BLGYPRIS	Produces an inquiry type PIDT for problem records
BLGYPRRS	Produces a retrieve type PIDT for problem records
BLGYPRUS	Produces an update type PIDT for problem records
BLGYSCCS	Produces a create type PIDT for configuration software component records
BLGYSCIS	Produces an inquiry type PIDT for configuration software component records
BLGYSCRS	Produces a retrieve type PIDT for configuration software component records
BLGYSCUS	Produces an update type PIDT for configuration software component records
BLGYSFAS	Produces PIDT to add feature relations to software component records
BLGYSFCS	Produces a create type PIDT for configuration software component feature records
BLGYSFIS	Produces an inquiry type PIDT for configuration software component feature records
BLGYSFLS	Produces an inquiry type PIDT used to list software features of a component record
BLGYSFRS	Produces a retrieve type PIDT for configuration software component feature records
BLGYSFUS	Produces an update type PIDT for configuration software component feature records
BLGYSNCS	Produces a create type PIDT for financial software records
BLGYSNIS	Produces an inquiry type PIDT for financial software records
BLGYSNRS	Produces a retrieve type PIDT for financial software records
BLGYSNUS	Produces an update type PIDT for financial software records
BLGYSVCS	Produces a create type PIDT for service records
BLGYSVIS	Produces an inquiry type PIDT for service records
BLGYSVRS	Produces a retrieve type PIDT for service records
BLGYSVUS	Produces an update type PIDT for service records
BLGYSXAS	Produces PIDT to add connection relations to software component records
BLGYSXCS	Produces a create type PIDT for configuration software component connection records
BLGYSXIS	Produces an inquiry type PIDT for configuration software component connection records
BLGYSXLS	Produces an inquiry type PIDT used to list software connections for a component
BLGYSXRS	Produces a retrieve type PIDT for configuration software component connection records
BLGYSXUS	Produces an update type PIDT for configuration software component connection records
BLGYSYCS	Produces a create type PIDT for system records
BLGYSYIS	Produces an inquiry type PIDT for system records
BLGYSYRS	Produces a retrieve type PIDT for system records
BLGYSYUS	Produces an update type PIDT for system records
BLGYVLIS	Produces an retrieve type PIDT for validation records
BLGYVLRS	Produces an inquiry type PIDT for validation records
BLG0CMDS	ISPF table containing aliases for all Tivoli Information Management for z/OS commands to be issued using function keys and action bars on enhanced panel style windows.
BLG0EPSC	GML source for command table
BLG0EPSK	GML source for keylists
BLG0KEYS	ISPF table containing INFO context-specific function key definitions
BLHRCDSJ	JCL to load data model records into the database

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLM@EDIT	Panel used to display field data on the PDF editor panel
BLMACCPT	Performs SMP/E ACCEPT of Tivoli Information Management for z/OS
BLMALALC	Allocates target and distribution data sets for all FMIDs
BLMALDEF	Defines DDDEFs to SMP/E for all FMIDs
BLMALLOC	Allocates target and distribution data sets
BLMAPPLY	Performs SMP/E APPLY of Tivoli Information Management for z/OS
BLMDDDEF	Defines DDDEFS to SMP/E
BLMESCAL	Runs problem escalation
BLMHLCJ	JCL to compile, link, and run the 370 HLAPI sample program BLMHLCS. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLCS	C sample programs for the HLAPI. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLCU	C include file that defines HLAPI structures. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLPJ	JCL to compile, link, and run the PL/I HLAPI sample program BLMHLPS. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLPS	PL/I to access Tivoli Information Management for z/OS through the HLAPI. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLXJ	JCL to run sample REXX/HLAPI interface EXEC HLMHLXS. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMHLXS	REXX sample program for the REXX/HLAPI interface. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMISMKD	Creates installation directories in HFS
BLMKACPT	JCL used with SMP/E to accept the HLAPI/CICS client
BLMKALOC	JCL to allocate product data sets for HLAPI/CICS client
BLMKAPLY	JCL used with SMP/E to apply the HLAPI/CICS client
BLMKDDEF	JCL to define the DDDEF statements for the HLAPI/CICS client
BLMLLCS	C sample programs for the LLAPI. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMLLCU	C include file that defines the LLAPI structures. Refer to the *Tivoli Information Management for z/OS Application Program Interface Guide*.
BLMMRES	Procedure to start the MRES
BLMMRESP	Provides parameters to start the MRES
BLMNINDJ	JCL to run the BLMNINDX batch job that updates the text search indexes used by OS/390 Text Search.
BLMNDDEF	JCL to define the DDDEF statements for the CALLIBs for the NetView Bridge Adapter.
BLMSAL1	Second part of JCL that performs Automatic Log Save Receive processing
BLMSARV	First part of JCL that performs Automatic Log Save Receive processing
BLMSASD	JCL that performs Automatic Log Save Send processing
BLMSASDA	JCL that performs Automatic Log Save Send processing—automatically submitted after a successful Automatic Log Save Receive
BLMSASDE	JCL to process an existing Automatic Log Save Send data set
BLMSDCS	JCL to copy Automatic Log Save Send data set to a generation of a generation data group

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLMSDSD	JCL to perform DB2 Extract Send processing
BLMSL1	Partial JCL to load DB2 Extract SQL information into DB2
BLMSL2	JCL to load DB2 Extract SQL information into DB2 for recovery
BLMSRCV	Partial JCL to load DB2 Extract information into DB2
BLMSSND	JCL that processes the DB2 Extract SQL data set
BLMSTEP2	JCL to compile, link, and bind a new plan for DSNTEP2 for use with the DB2 Extract Facility
BLMTALOC	JCL to allocate the Tivoli Enterprise Console (TEC) Event Adapter data sets
BLMTDDEF	JCL to define the DDDEF statements for the TEC Event Adapter
BLMTECL	Job to link-edit code for the TEC Event Adapter
BLMTMDIR	JCL to execute the BLMTMKDR EXEC to create directories for the TEC Event Adapter
BLMTMKDR	Creates the Tivoli Information Management for z/OS TEC Event Adapter hierarchical file system structure for the target libraries
BLMTSRCJ	JCL to complete customization of the OS/390 Text Search Engine environment for Tivoli Information Management for z/OS
BLMTSRPJ	JCL to start the text search request processor (TSRP) used with OS/390 Text Search
BLMTSRPP	Parameters member for the text search request processor
BLMUALOC	JCL to allocate the HLAPI/USS data sets
BLMUDDEF	JCL to define the DDDEF statements for the HLAPI/USS client
BLMUMDIR	JCL to execute the BLMUMKDR EXEC to create the directories for the HLAPI/USS
BLMUMKDR	Creates the Tivoli Information Management for z/OS HLAPI/USS hierarchical file system structure for the target libraries
BLMWJCL	JCL to start the web connector server as an MVS batch job
BLMWMIME	Media type table used with the Web connector
BLMWPNL	Defines the write panel data set (WPANLDS)
BLMYKCRE	Sample HLAPI/CICS program (HL08 create record transaction)
BLMYKCTL	Sample HLAPI/CICS program (start and end logical session)
BLMYKDEL	Sample HLAPI/CICS program (HL13 delete record transaction)
BLMYKLNK	Job to link-edit the HLAPI/CICS sample programs
BLMYKMAP	Maps for the HLAPI/CICS sample program
BLMYKMNU	Sample HLAPI/CICS program (menu map example)
BLMYKRTV	Sample HLAPI/CICS program (HL06 retrieve record transaction)
BLXABMSG	ISPF panel for displaying messages if a user attempts to start Tivoli Information Management for z/OS when the BLX-SP is not running or when the BLX-SPis restarted
BLXCFSTR	IXCMIAPU job for coupling facility structures
BLXIPCSP	IPCS data and exit definition statements
BLXRACFT	Sample entry in the RACF started procedures table (ICHRIN03) for BLXSHR
BLXRLSCD	Defines RLS share control data sets (IEFBR14 job)
BLXSHR	BLX-SP procedure for sharing data sets
BLXTAAPM	Sample NetView automation table entry for INFOMGMT APM instrumentation
BLXVDEF	Defines a VSAM resource definition member to the BLX-SP. See Figure 31 on page 312.

Table 23. Description of Sample Members in SBLMSAMP Library (continued)

Member Name	Purpose
BLX1PROC	Procedure to start the BLX-SP
BLX1SH	BLX-SP parameters member for sharing data sets
BLX100	BLX-SP parameters member; it contains BLX-SP operating parameters. See Figure 34 on page 350.
BTN@ISP	Selection menu screen
BTNAMS	Starts SAM problem entry
BTNAMSRP	Creates and prints RMF/SAM reports
BTNBATCH	Establishes libraries needed for batch interface to Integration Facility
BTNCNMBD	Integration Facility record access for NPDA
BTNCNMSU	Updates NPDA records
BTNCNMTS	Updates NPDA records
BTNIIF	Allocates files required by ISPF, PDF, and systems management program
BTNISPH0	Integration Facility tutorial screen
BTNMDATE	Converts dates to user format
BTNNPDA	Starts NPDA alert processing
BTNNPDPF	Profiles in description field
BTNOPCPF	Profiles in description field
BTNSAM	SAM record update
BTNSAMBD	Integration Facility record access for SAM
BTNSAMH0	SAM tutorial screen
BTNSAMP	RMF and SAM selection panel
BTNSAMPF	Profiles in description field
BTNSDATE	Converts dates from user format
BTNSLRCC	Report of closed changes
BTNSLRCP	Report of closed problems
BTNSLRJ1	Report of closed problems
BTNSLRJ2	Report of closed problems
BTNSLRPF	Profiles in description field
BTNTDATE	Tests date-conversion interface routine
BTNTOAMS	Produces SAM reports
BTNUDATE	Converts a date format to a different format
BTNX6JOB	Creates problem record using Integration Facility
BTNX7JOB	Builds command list (CLIST) when batch job ends in error
EYLALIAS	Produces an ALIAS table for the HLAPI
EYLSJ002	Allocates the input and output data set for use by the NetView Bridge Adapter
EYMPOST	Starts the NetView AutoBridge PostProcessor

J

New, Changed, and Removed Panels

This appendix contains lists of new, changed, and deleted panels. Panels added or changed because of authorized program analysis reports (APARs) are included. However, these lists do not include help or message panels for versions prior to Information/Management Version 6.3. The new, changed, and removed panels for Tivoli Information Management for z/OS Version 7.1 are also listed in member BLGPANEL of the SBLMSAMP library.

These panels are grouped according to version number, as follows:

- Panels that are new for Tivoli Information Management for z/OS Version 7.1
- Panels that are changed for Tivoli Information Management for z/OS Version 7.1
- Panels that were removed for Tivoli Information Management for z/OS Version 7.1
- Panels that are new for Tivoli Service Desk for OS/390 Version 1.2
- Panels that are changed for Tivoli Service Desk for OS/390 Version 1.2
- Panels that were removed for Tivoli Service Desk for OS/390 Version 1.2
- Panels that are new for TME 10 Information/Management Version 1.1
- Panels that are changed for TME 10 Information/Management Version 1.1
- Panels that were removed for TME 10 Information/Management Version 1.1
- Panels that are new for Information/Management Version 6.3
- Panels that changed in Information/Management Version 6.3
- Panels that were removed in Information/Management Version 6.3
- Panels that are new for Information/Management Version 6.2
- Panels that changed in Information/Management Version 6.2
- Panels that were removed in Information/Management Version 6.2
- Panels that are new for Information/Management Version 6.1
- Panels that changed in Information/Management Version 6.1
- Panels that were removed in Information/Management Version 6.1

Refer to the *Tivoli Information Management for z/OS Terminal Simulator Guide and Reference* and the *Tivoli Information Management for z/OS Panel Modification Facility Guide* for details about panels.

New Panels – Tivoli Information Management for z/OS Version 7.1

BLGESC14	BLG1AELD	BLG6TBAL	BLH1N003
BLGESC16	BLG1AELU	BLG6TBLD	BLH1N004
BLGLAPST	BLG1APST	BLG6TBLP	BLH1N008
BLGLAPVR	BLG1ARED	BLG6TBNM	BLH1N009
BLGLDSSD	BLG1AREU	BLG6TBSI	BLH1N010
BLGLDSTS	BLG1ASTS	BLG6TBSW	BLH1N011
BLGLDTMD	BLG1ATSA	BLG6URN5	BLH1N012
BLGLDTTM	BLG1AZDF	BLG6UTZN	BLH1N013
BLGLJ300	BLG1A11B	BLHLNTXT	BLH1N019
BLGLQ300	BLG1C015	BLHLX0BC	BLH1N020
BLGLREVD	BLG1ROLE	BLHLX0CC	BLH1N040
BLGLREVR	BLG1TDEH	BLHLX0TO	BLH1S000

BLGLSUPD	BLG1TEHD	BLHLX1BC	BLH1S003
BLGLSUPU	BLG1TTSA	BLHLX1CC	BLH1S004
BLGLTSTD	BLG1VQ50	BLHLX1TO	BLH1S006
BLGLTSTK	BLG1VQ70	BLH0N000	BLH1S008
BLGLTSTL	BLG1VU51	BLH0N001	BLH1S009
BLGLTZDD	BLG1VU61	BLH0N010	BLH1S010
BLGLTZDF	BLG1VU71	BLH0N090	BLH1S011
BLGLVLSB	BLG2DSTS	BLH0N095	BLH1S012
BLGRFCHG	BLG2DTTM	BLH0N096	BLH1S013
BLGRFPRO	BLG2GNRC	BLH0N110	BLH1S014
BLGRFSOL	BLG2J400	BLH0N200	BLH1S040
BLGTFDAT	BLG2Q400	BLH0S000	BLH1TMTT
BLGTFZON	BLG2SUPP	BLH0S001	BLH1TTXT
BLGTROLE	BLG2TBDD	BLH0S010	BLH1X000
BLGTSPXX	BLG2TBDL	BLH0S011	BLH1X004
BLG0C015	BLG2TSTD	BLH0S090	BLH1X006
BLG0J2C0	BLG2TSTL	BLH0S091	BLH1X009
BLG0J400	BLG2TTSA	BLH0S100	BLH1X010
BLG0J401	BLG2TZDF	BLH0S110	BLH1X012
BLG0K400	BLG2ZSC7	BLH0S111	BLH1X040
BLG0Q2C0	BLG2Z200	BLH0S201	BLH2N110
BLG0Q400	BLG2Z250	BLH0X010	BLH2S010
BLG0Z150	BLG6APSL	BLH0X011	BLH2S110
BLG0Z151	BLG6APST	BLH0X015	BLH2X110
BLG0Z152	BLG6APVR	BLH0X090	BLH2X111
BLG0Z153	BLG6DTRC	BLH0X091	BLH6INDX
BLG0Z200	BLG6DVCH	BLH0X110	BLHM2FD6B
BLG0Z250	BLG6EFND	BLH0X111	BLM6DTDF
BLG1AAPD	BLG6PELI	BLH1N000	BLM6EFPO
BLG1AAPU	BLG6REVD	BLH1N001	BTN1C126

Changed Panels – Tivoli Information Management for z/OS Version 7.1

BLGESC04	BLG0G183	BLG0V600	BLM2CUA0
BLGESC06	BLG0G184	BLG0V610	BLM2CU00
BLGLDATT	BLG0G190	BLG0V700	BLM2C000
BLGLTBDD	BLG0G191	BLG0V710	BLM2C030
BLGLTBDK	BLG0G192	BLG0V720	BLM2FD17
BLGLTBDL	BLG0G193	BLG0V730	BLM2FD32
BLGLTSFD	BLG0G290	BLG0W200	BLM2FD37
BLGLTSFK	BLG0G293	BLG0W210	BLM2FD45
BLGLTSFL	BLG0G337	BLG0W250	BLM2FD46
BLGMIG01	BLG0G338	BLG0W500	BLM2FD48
BLGTPSET	BLG0G357	BLG0W510	BLM2FD77
BLGTSPAE	BLG0G358	BLG0X100	BLM2FD89
BLGTSPAU	BLG0G359	BLG0Y200	BLM2FD9A
BLGTSPCE	BLG0G390	BLG0Y300	BLM2FD94
BLGTSPCU	BLG0G391	BLG0ZD50	BLM2FD99
BLGTSPPE	BLG0G392	BLG0ZD51	BLM2PC01
BLGTSPPU	BLG0G490	BLG0ZU00	BLM2PC02
BLG0AAE0	BLG0G493	BLG0ZU50	BLM2PC1B
BLG0AC01	BLG0G590	BLG0Z100	BLM2PC1D
BLG0AC04	BLG0G593	BLG0Z500	BLM2PC1G
BLG0AC09	BLG0G727	BLG0Z510	BLM2PC1H
BLG0AC22	BLG0G728	BLG0Z530	BLM2PC10
BLG0AC23	BLG0G737	BLG00000	BLM2PC19
BLG0AC27	BLG0G738	BLG00001	BLM2PS05
BLG0AC31	BLG0G739	BLG00002	BLM2PT09
BLG0AC44	BLG0G742	BLG00010	BLM2P020
BLG0AC45	BLG0G747	BLG00011	BLM2TPDS
BLG0BU00	BLG0G748	BLG00020	BLM2YE00
BLG0BU10	BLG0G749	BLG1AACP	BLM2YE50
BLG0BU11	BLG0G751	BLG1AACP	BLM2Y250
BLG0BU90	BLG0G770	BLG1AAUP	BLM2Y750

BLG0B100	BLG0G771	BLG1ACOP	BLM2Y770
BLG0B200	BLG0G772	BLG1AF68	BLM2Y780
BLG0B300	BLG0G780	BLG1ARVC	BLM2ZIMN
BLG0B400	BLG0G781	BLG1A000	BLM6ALDA
BLG0B402	BLG0G782	BLG1A001	BLM6CRTA
BLG0B500	BLG0G787	BLG1A097	BLM6FATA
BLG0B502	BLG0G788	BLG1A1DD	BLM6INTA
BLG0B901	BLG0G790	BLG1A1G1	BLM6NPNL
BLG0B902	BLG0G791	BLG1A1G2	BLM6PANA
BLG0B903	BLG0G792	BLG1A1G5	BLM6PARA
BLG0CU00	BLG0G890	BLG1A100	BLM6PLMD
BLG0CU01	BLG0HU10	BLG1A121	BLM6PREX
BLG0CU10	BLG0HU11	BLG1A504	BLM6TRTA
BLG0CU11	BLG0H000	BLG1A511	BLM6VENA
BLG0CU21	BLG0IU00	BLG1A604	BLM6Y00E
BLG0CU30	BLG010E0	BLG1A704	BLM6Y007
BLG0CU31	BLG0I100	BLG1A711	BLM8CCA0
BLG0C051	BLG0I200	BLG1TADD	BLM8CC00
BLG0C052	BLG0JU00	BLG1TBAC	BLM8CC10
BLG0C053	BLG0JU10	BLG1TBAT	BLM8CD00
BLG0C054	BLG0JU11	BLG1TDAD	BLM8CF00
BLG0C055	BLG0J100	BLG1TDBA	BLM8CF10
BLG0C056	BLG0J200	BLG1TDDE	BLM8CL10
BLG0C061	BLG0J210	BLG1TDES	BLM8CL20
BLG0C062	BLG0J220	BLG1TDJU	BLM8CL30
BLG0C063	BLG0J230	BLG1TDNO	BLM8CL50
BLG0C064	BLG0J240	BLG1TDSR	BLM8CL60
BLG0C065	BLG0J250	BLG1TDST	BLM8CT00
BLG0C066	BLG0J260	BLG1TJUS	BLM8CU00
BLG0C071	BLG0J270	BLG1TNOT	BLM8CU1A
BLG0C072	BLG0J280	BLG1TRES	BLM8CU1B
BLG0C073	BLG0J290	BLG1TSLX	BLM8CU1E
BLG0C100	BLG0J295	BLG1TSLY	BLM8CU1H
BLG0C151	BLG0J300	BLG1TSTA	BLM8CU11
BLG0C152	BLG0K190	BLG1TSTT	BLM8CU12
BLG0C153	BLG0K291	BLG1UT01	BLM8CU13
BLG0C154	BLG0K292	BLG1UT02	BLM8CU17
BLG0C155	BLG0K293	BLG1UT07	BLM8CU2A
BLG0C156	BLG0L100	BLG1UT20	BLM8CU2B
BLG0C161	BLG0L200	BLG1UT31	BLM8CU2E
BLG0C162	BLG0L300	BLG1UT32	BLM8CU2H
BLG0C163	BLG0L400	BLG2ALIS	BLM8CU21
BLG0C164	BLG0L500	BLG2CMND	BLM8CU22
BLG0C165	BLG0L600	BLG2EN10	BLM8CU23
BLG0C166	BLG0L700	BLG2EN20	BLM8CU27
BLG0C171	BLG0L901	BLG2FGFD	BLM8CU31
BLG0C172	BLG0L902	BLG2FGFL	BLM8CU37
BLG0C173	BLG0L903	BLG2FGID	BLM8CU43
BLG0C200	BLG0M100	BLG2FGIN	BLM8CU47
BLG0C300	BLG0M200	BLG2G115	BLM8CU5A
BLG0C500	BLG0M300	BLG2G125	BLM8CU5B
BLG0C600	BLG0M500	BLG2G126	BLM8CU5D
BLG0C620	BLG0M600	BLG2G130	BLM8CU5E
BLG0C630	BLG0M620	BLG2G135	BLM8CU5I
BLG0C700	BLG0M630	BLG2G145	BLM8CU51
BLG0C800	BLG0M700	BLG2G155	BLM8CU53
BLG0DFSM	BLG0M800	BLG2G160	BLM8CU55
BLG0DU01	BLG0N100	BLG2G165	BLM8CU57
BLG0DU02	BLG0N101	BLG2G175	BLM8CU6A
BLG0DU03	BLG0N105	BLG2G180	BLM8CU6B
BLG0DU04	BLG0N110	BLG2G185	BLM8CU61
BLG0DU05	BLG0N113	BLG2G195	BLM8CU63
BLG0DU06	BLG0N120	BLG2G315	BLM8CU64
BLG0DU07	BLG0N121	BLG2G335	BLM8CU67
BLG0DU08	BLG0N122	BLG2G355	BLM8CU7A
BLG0DU09	BLG0N130	BLG2G400	BLM8CU7B
BLG0DU10	BLG0N132	BLG2G500	BLM8CU7E

BLG0DU11	BLG0N133	BLG2G710	BLM8CU7H
BLG0DU12	BLG0N134	BLG2G725	BLM8CU71
BLG0DU13	BLG0N140	BLG2G730	BLM8CU72
BLG0DU14	BLG0N141	BLG2G735	BLM8CU73
BLG0DU15	BLG0N142	BLG2G740	BLM8CU76
BLG0DU16	BLG0N150	BLG2G745	BLM8CU77
BLG0DU17	BLG0N151	BLG2G750	BLM8CU78
BLG0DU18	BLG0N153	BLG2G775	BLM8CU79
BLG0DU19	BLG0N154	BLG2G785	BLM8CU8A
BLG0DU20	BLG0N157	BLG2G795	BLM8CU8B
BLG0DU21	BLG0N160	BLG2G850	BLM8CU81
BLG0DU22	BLG0N164	BLG2JU00	BLM8CU83
BLG0DU23	BLG0N170	BLG2L700	BLM8CU85
BLG0DU24	BLG0N172	BLG2PU00	BLM8CU87
BLG0DU25	BLG0N200	BLG2Q100	BLM8CU88
BLG0DU26	BLG0N201	BLG2THLP	BLM8CU9A
BLG0DU27	BLG0N205	BLG2TPRL	BLM8CU9B
BLG0DU28	BLG0N210	BLG2TSFD	BLM8CU9C
BLG0DU29	BLG0N220	BLG2TSFL	BLM8CU9D
BLG0DU30	BLG0N221	BLG2TSID	BLM8CU9E
BLG0DU31	BLG0N222	BLG2TSIN	BLM8CU9F
BLG0DU32	BLG0N230	BLG2TSTA	BLM8CU9G
BLG0DU33	BLG0N231	BLG2TVID	BLM8CU9H
BLG0DU34	BLG0N232	BLG2VQ50	BLM8CU9I
BLG0DU35	BLG0N250	BLG2VQ60	BLM8CU9K
BLG0DU36	BLG0N251	BLG2VQ70	BLM8CU9L
BLG0DU37	BLG0N253	BLG2W500	BLM8CU9M
BLG0DU38	BLG0N254	BLG2ZCLA	BLM8CU9N
BLG0DU39	BLG0N257	BLG2ZCMD	BLM8CU9O
BLG0DU40	BLG0N300	BLG2ZDAT	BLM8CU9P
BLG0DU42	BLG0N301	BLG2ZD51	BLM8CU9Q
BLG0DU43	BLG0N305	BLG2ZGEN	BLM8CU9R
BLG0DU44	BLG0N306	BLG2ZMSG	BLM8CU9S
BLG0DU45	BLG0N307	BLG2ZOSV	BLM8CU9T
BLG0DU46	BLG0N310	BLG2ZPAN	BLM8CU9U
BLG0DU47	BLG0N320	BLG2ZPRO	BLM8CU9V
BLG0DU48	BLG0N350	BLG2ZREP	BLM8CU9W
BLG0DU49	BLG0N352	BLG2ZSRH	BLM8CU9X
BLG0DU50	BLG0N353	BLG2Z500	BLM8CU9Y
BLG0DU51	BLG0N355	BLG20002	BLM8CU9Z
BLG0DU52	BLG0N357	BLG6ALTD	BLM8CU91
BLG0DU53	BLG0N360	BLG6ASSD	BLM8CU97
BLG0DU54	BLG0N400	BLG6ATSN	BLM8BU00
BLG0DU55	BLG0N410	BLG6CDFR	BLM8BU10
BLG0DU56	BLG0N500	BLG6CDTO	BTN0BU11
BLG0DU57	BLG0N510	BLG6CLUS	BTN0B100
BLG0DU58	BLG0N600	BLG6CMPD	BTN0B101
BLG0DU59	BLG0N610	BLG6CPUR	BTN0B102
BLG0DU60	BLG0N700	BLG6CRDT	BTN0B200
BLG0DU61	BLG0N701	BLG6CRTA	BTN0B300
BLG0DU62	BLG0N710	BLG6DABA	BTN0B500
BLG0DU63	BLG0N720	BLG6DARR	BTN0CETX
BLG0DU64	BLG0N721	BLG6DATX	BTN0CU00
BLG0DU65	BLG0N722	BLG6DNOT	BTN0CU01
BLG0DU66	BLG0N740	BLG6DSTA	BTN0C100
BLG0DU67	BLG0N741	BLG6HBSR	BTN0C101
BLG0DU68	BLG0N742	BLG6HHFN	BTN0C200
BLG0DU69	BLG0N750	BLG6INSD	BTN0C300
BLG0DU70	BLG0N751	BLG6INTA	BTN0DU01
BLG0DU71	BLG0N753	BLG6ISPF	BTN0DU03
BLG0DU72	BLG0N754	BLG6ITYP	BTN0DU08
BLG0DU73	BLG0N757	BLG6JPEX	BTN0DU18
BLG0DU74	BLG0N758	BLG6LOFI	BTN0DU28
BLG0DU75	BLG0N759	BLG6OCCD	BTN0DU30
BLG0DU76	BLG0N760	BLG6PREX	BTN0DU31
BLG0D100	BLG0N764	BLG6PURD	BTN0D100
BLG0D102	BLG0N770	BLG6PU01	BTN0D102

BLG0D105	BLG0N772	BLG6REQD	BTN0D200
BLG0D107	BLG0N780	BLG6SCHD	BTN0D500
BLG0D113	BLG0N781	BLG6TARD	BTN0ENCC
BLG0D120	BLG0N782	BLG6TBAT	BTN0ENCF
BLG0D130	BLG0N783	BLG6TBDE	BTN0ENHD
BLG0D140	BLG0N784	BLG6TSTA	BTN0ENNO
BLG0D150	BLG0N787	BLG6URN3	BTN0ENOM
BLG0D151	BLG0N790	BLG6VPAD	BTN0ENPC
BLG0D152	BLG0N791	BLG600CD	BTN0ENSU
BLG0D153	BLG0N792	BLG8PWIN	BTN0ENSY
BLG0D154	BLG0PU00	BLG8RWIN	BTN0EN20
BLG0D157	BLG0P100	BLH0IU20	BTN0EPPC
BLG0D160	BLG0P101	BLH0IU21	BTN0E000
BLG0D164	BLG0P200	BLH0I000	BTN0E090
BLG0D170	BLG0P300	BLH0I001	BTN0E190
BLG0D172	BLG0P410	BLH0I002	BTN0E290
BLG0D200	BLG0P420	BLH0I004	BTN0E390
BLG0D204	BLG0P430	BLH0I010	BTN0E890
BLG0D205	BLG0P510	BLH0I011	BTN0F000
BLG0D220	BLG0P511	BLH0I012	BTN0F090
BLG0D230	BLG0P520	BLH0I022	BTN0F190
BLG0D250	BLG0P521	BLH0I023	BTN0F290
BLG0D251	BLG0P530	BLH0I12A	BTN0F390
BLG0D253	BLG0P531	BLH0I120	BTN0G000
BLG0D254	BLG0P700	BLH0I121	BTN0G1A1
BLG0D257	BLG0P810	BLH0I123	BTN0G1A3
BLG0D300	BLG0P811	BLH0I132	BTN0G190
BLG0D302	BLG0P820	BLH0I141	BTN0G191
BLG0D305	BLG0P821	BLH0I22A	BTN0H000
BLG0D306	BLG0P830	BLH0I220	BTN0JU00
BLG0D307	BLG0P831	BLH0I221	BTN0J100
BLG0D350	BLG0QUP0	BLH0I222	BTN0J200
BLG0D353	BLG0Q100	BLH0I223	BTN0J300
BLG0D355	BLG0Q200	BLH0I224	BTN0L100
BLG0D356	BLG0Q210	BLH0I225	BTN0L200
BLG0D357	BLG0Q213	BLH0I226	BTN0L300
BLG0D400	BLG0Q214	BLH0I227	BTN0L500
BLG0D500	BLG0Q215	BLH0I228	BTN0M010
BLG0D600	BLG0Q216	BLH0I229	BTN0M100
BLG0D700	BLG0Q220	BLH0I230	BTN0M200
BLG0D703	BLG0Q230	BLH0I231	BTN0N100
BLG0D720	BLG0Q240	BLH0I232	BTN0N101
BLG0D740	BLG0Q250	BLH6ACCN	BTN0N134
BLG0D750	BLG0Q260	BLH6ITMR	BTN0N200
BLG0D751	BLG0Q270	BLM0AC02	BTN0N201
BLG0D752	BLG0Q280	BLM0B000	BTN0N500
BLG0D753	BLG0Q290	BLM0B001	BTN0N510
BLG0D754	BLG0Q300	BLM0B002	BTN0Q100
BLG0D757	BLG0RU00	BLM0B003	BTN0Q200
BLG0D758	BLG0R010	BLM0B050	BTN0Q300
BLG0D759	BLG0R011	BLM0B051	BTN0S010
BLG0D760	BLG0R020	BLM0B052	BTN0S020
BLG0D764	BLG0R100	BLM0B053	BTN0S021
BLG0D770	BLG0SU50	BLM0B060	BTN0W020
BLG0D772	BLG0S010	BLM0B100	BTN0W030
BLG0D780	BLG0S011	BLM0B101	BTN0XAUT
BLG0D783	BLG0S020	BLM0B102	BTN0XDES
BLG0D784	BLG0S021	BLM0B103	BTN0XHID
BLG0D787	BLG0S060	BLM0B108	BTN0XLPC
BLG0D790	BLG0S061	BLM0B150	BTN0XSUM
BLG0EN10	BLG0S500	BLM0B151	BTN0XUSR
BLG0EN20	BLG0S510	BLM0CU10	BTN1CTGY
BLG0E090	BLG0S530	BLM0CU11	BTN1TDDE
BLG0E190	BLG0TU01	BLM0S010	BTN1TDST
BLG0E290	BLG0TU02	BLM0YE00	BTN2ENCC
BLG0E390	BLG0T100	BLM0YE50	BTN2ENCF
BLG0E490	BLG0T101	BLM0YU00	BTN2ENHD

BLG0E590
BLG0E690
BLG0E790
BLG0E890
BLG0F090
BLG0F190
BLG0F270
BLG0F271
BLG0F272
BLG0F273
BLG0F278
BLG0F280
BLG0F281
BLG0F282
BLG0F283
BLG0F288
BLG0F290
BLG0F390
BLG0F590
BLG0F890
BLG0GU10
BLG0GU11
BLG0G1A1
BLG0G1A2
BLG0G1A3
BLG0G1A4
BLG0G1A5
BLG0G1B1
BLG0G1B2
BLG0G1B3
BLG0G1B4
BLG0G1B5
BLG0G137
BLG0G138
BLG0G167
BLG0G168
BLG0G170
BLG0G171
BLG0G172
BLG0G182
BLG0T110
BLG0T200
BLG0U000
BLG0U001
BLG0U100
BLG0U120
BLG0U200
BLG0VD50
BLG0VD51
BLG0VD60
BLG0VD61
BLG0VD70
BLG0VD71
BLG0VE60
BLG0VE61
BLG0VE62
BLG0VE70
BLG0VE71
BLG0VE72
BLG0VE73
BLG0VE74
BLG0VE75
BLG0VQ50
BLG0VQ52
BLG0VQ60
BLG0VQ61
BLG0VQ62
BLG0VQ70
BLG0VQ71
BLG0VQ72
BLG0VQ73
BLG0VQ74
BLG0VU50
BLG0VU60
BLG0VU70
BLG0V090
BLG0V190
BLG0V500
BLG0V502
BLG0V510
BLM0YU10
BLM0YU11
BLM0YU50
BLM0YU51
BLM0YU60
BLM0YU61
BLM0YU80
BLM0YU90
BLM0YU91
BLM0Y100
BLM0Y203
BLM0Y251
BLM0Y261
BLM0Y300
BLM0Y310
BLM0Y330
BLM0Y600
BLM0Y650
BLM0Y703
BLM0Y713
BLM0Y751
BLM0Y761
BLM0Y771
BLM0Y781
BLM0Y793
BLM0Y800
BLM0Y810
BLM0Y830
BLM0Y850
BLM0Y860
BLM0Y880
BLM1S201
BLM1S203
BLM1S204
BLM1TDDE
BLM1Y115
BLM1Y615
BLM2B152
BLM2CN00
BLM2CR10
BTN2ENNO
BTN2ENOM
BTN2ENPC
BTN2ENSU
BTN2ENSY
BTN6ALTD
BTN6ASSD
BTN6CMDP
BTN6CMPD
BTN6CRDT
BTN6DATE
BTN6DATX
BTN6DNOT
BTN6DSTA
BTN6DSTT
BTN6ENDD
BTN6OCCD
BTN6REQD
BTN6SCHD
BTN6TARD
BTN600CD
BTN7CMPD
BTN7TARD
EYMAM100
EYMBM100
EYM5MARK
EYM5M100
EYM5M110
EYM5M200
EYM5M250
EYM6ALTD
EYM6CRDT
EYM9MAIN

Removed Panels – Tivoli Information Management for z/OS Version 7.1

BLG3S223
BLH0I241
BLM0B004
BLM0B008
BLM1B070
EYMTSPPE

New Panels — Tivoli Service Desk for OS/390 Version 1.2

BLGLFGFD
BLGLFGFK
BLGLFGFL
BLGLFGID
BLGLFGIN
BLGLTSFD
BLGLTSFK
BLGLTSFL
BLGLTSID
BLGLTSIN
BLG0C800
BLG0E790
BLG4T200
BLG4T210
BLG4T211
BLG4T213
BLG4T215
BLG4T216
BLG4T217
BLG4T218
BLG4T303
BLG4T304
BLG4T305
BLG4T306
BLHTSI21
BLHTSI99
BLH0IU20
BLH0IU21
BLH0I000
BLH0I001
BLH0I002
BLH0I004
BLH0I010
BLH0I011
BLH0I012
BLH0I022
BLH9A002
BLH9A003
BLH9A004
BLH9A005
BLH9A006
BLH9A007
BLH9A008
BLH9A009
BLH9I001
BLMLPROB
BLM0B000
BLM0B001

BLG0J280
BLG0J295
BLG0K850
BLG0L700
BLG0M800
BLG0Q280
BLG0Y300
BLG1AJ29
BLG1AJ80
BLG1ARFR
BLG1ASSO
BLG1ATBO
BLG1ATTI
BLG1A11Z
BLG1A550
BLG1A551
BLG1A770
BLG1A771
BLG1CSWT
BLG1C800
BLG1FGFL
BLG1FGLX
BLG1FGLY
BLG1TBAT
BLG1TBIN
BLG1TIID
BLG1TPRL
BLG1TSLX
BLG1TSLY
BLG2FGFD
BLG2FGFL
BLG2FGID
BLG2FGIN
BLG2L700
BLG2TPRL
BLG2TSFD
BLG2TSFL
BLG2TSID
BLG2TSIN
BLG2Y300
BLG4C063
BLG4C069
BLG4C070
BLG4C083
BLG4C146
BLG4C147
BLG4C148
BLG4C149
BLG4C150
BLG4C151
BLG4E067
BLG4F034
BLG4J034
BLG4N021
BLG4P037
BLG4T307
BLG4T308
BLG4T309
BLG4T310
BLG4T311
BLG4T312
BLG4T313
BLG4V529
BLG6ATSN
BLG6ATTI
BLG6CDCA
BLG6CGMX
BLG6CSVL
BLG6DFAT
BLG6FATT
BLG6FAUT
BLG6FDSC
BLG6FGID
BLG6FGLX
BLG6FGNM
BLG6HFNM
BLG6HHFN
BLG6ITYP
BLG6JPEX
BLG6PRMT
BLG6SLCT
BLG6SWFM
BLG6SWPN
BLG6TIID
BLG6TSLX
BLG6TSNM
BLG6UPSD
BLG9C083
BLG9C146
BLG9C147
BLG9C148
BLG9C149
BLG9C150
BLG9C151
BLG9E067
BLG9F034
BLG9J034
BLG9P037
BLHLI122
BLHLI124
BLHLI125
BLHLI126
BLHLI127
BLHLI128
BLHLI129
BLHLI130
BLHLI131
BLHTSI04
BLHTSI10
BLHTSI20
BLH0I023
BLH0I12A
BLH0I120
BLH0I121
BLH0I123
BLH0I132
BLH0I141
BLH0I22A
BLH0I220
BLH0I221
BLH0I222
BLH0I223
BLH0I224
BLH0I225
BLH0I226
BLH0I227
BLH0I228
BLH0I229
BLH0I230
BLH0I231
BLH0I232
BLH1AI01
BLH1AI02
BLH1AI04
BLH1AI10
BLH1AI12
BLH1AI14
BLH1AI15
BLH1AI18
BLH1AI19
BLH1AI20
BLH1AI21
BLH1AI99
BLH1AZ11
BLH1AZ12
BLH1AZ13
BLH1AZ14
BLH1AZ41
BLH1A097
BLH1THWD
BLH2I001
BLH2I002
BLH4A001
BLH4A002
BLH4A003
BLH4A004
BLH4A005
BLH4A006
BLH4A007
BLH4A008
BLH4A009
BLH4I001
BLH6ACCN
BLH6ITMR
BLH9A001
BLM0B002
BLM0B049
BLM0B050
BLM0B051
BLM0B052
BLM0B060
BLM0B100
BLM0B101
BLM0B102
BLM0B108
BLM0B150
BLM0B151
BLM0B152
BLM0CU10
BLM0CU11
BLM0S010
BLM1A1C4
BLM1A1C5
BLM1A1C8
BLM1A16A
BLM1B000
BLM1B010
BLM1B020
BLM1B04Z
BLM1B040
BLM1B060
BLM1B072
BLM1B110
BLM1B114
BLM1B119
BLM1B150
BLM1B151
BLM1TDDE
BLM1TDES
BLM2B000
BLM2B050
BLM2B152
BLM2FD0E
BLM2FD0F
BLM2S010
BLM4C301
BLM4C302
BLM4C303
BLM4D250
BLM6GWID
BLM6MXRW
BLM6SLCO
BLM6SLRF
BLM6TCID
BLM6UPSD
BLM9C301
BLM9C302
BLM9C303
BLM9D250

Changed Panels – Tivoli Service Desk for OS/390 Version 1.2

BLGAPIDI
BLGAPI00
BLGAPI05
BLGDUMP1
BLGLDATT
BLGLVLST
BLGLVSEL
BLGMIG01
BLGTPSET
BLG0AC01
BLG0AC04
BLG0AC09
BLG0AC12
BLG0AC22
BLG0AC23
BLG0AC27
BLG0AC31
BLG0AC44
BLG0AC45
BLG0BU00
BLG0BU10
BLG0BU11
BLG0B100
BLG0B200
BLG0B300
BLG0B500
BLG0CU00
BLG0CU01
BLG0CU10
BLG0CU11
BLG0CU21
BLG0CU30
BLG0CU31
BLG0C100
BLG0C200
BLG0C300
BLG0C600
BLG0C620
BLG0C630
BLG0DATT
BLG0DU01
BLG0DU02
BLG0DU03
BLG0DU04
BLG0DU05
BLG0DU06
BLG0DU07
BLG0DU08
BLG0DU09
BLG0DU10
BLG0DU11
BLG0DU12
BLG0DU13
BLG0DU14
BLG0DU15
BLG0DU16
BLG0DU17
BLG0DU18
BLG0DU19
BLG0DU20
BLG0DU21
BLG0DU22
BLG0DU23
BLG0DU24
BLG0DU25
BLG0DU26
BLG0DU27
BLG0DU28
BLG0DU29
BLG0DU30
BLG0DU31
BLG0DU32
BLG0DU33
BLG0DU34
BLG0DU35
BLG0DU36
BLG0DU37
BLG0DU38
BLG0DU39
BLG0DU40
BLG0DU42
BLG0DU43
BLG0DU44
BLG0DU45
BLG0DU46
BLG0DU47
BLG0DU48
BLG0DU49
BLG0DU50
BLG0DU51
BLG0DU52
BLG0DU53
BLG0DU54
BLG0DU55
BLG0DU56
BLG0DU57
BLG0DU58
BLG0DU59
BLG0DU60
BLG0DU61
BLG0DU62
BLG0DU63
BLG0DU64
BLG0DU65
BLG0DU66
BLG0DU67
BLG0DU68
BLG0DU69
BLG0DU70
BLG0DU71
BLG0DU72
BLG0DU73
BLG0DU74
BLG0DU75
BLG0DU76
BLG0D100
BLG0D102
BLG0D120
BLG0D130
BLG0D140
BLG0D152
BLG0D153
BLG0D154
BLG0D164
BLG0D200
BLG0D220
BLG0D230
BLG0D253
BLG0D254
BLG0D307
BLG0D357
BLG0D700
BLG0D703
BLG0D720
BLG0D740
BLG0D780
BLG0D790
BLG0ENTR
BLG0EN10
BLG0EN20
BLG0E090
BLG0E190
BLG0E290
BLG0E390
BLG0E890
BLG0F090
BLG0F190
BLG0F270
BLG0F271
BLG0F272
BLG0F273
BLG0F278
BLG0F280
BLG0F281
BLG0F282
BLG0F283
BLG0F288
BLG0F290
BLG0F390
BLG0F890
BLG0G1A1
BLG0G1A2
BLG0G1A3
BLG0G1A4
BLG0G1A5
BLG0G1B1
BLG0G1B2
BLG0G1B3
BLG0G1B4
BLG0G1B5
BLG0G125
BLG0G126
BLG0G130
BLG0G135
BLG0G137
BLG0G138
BLG0G155
BLG0G160
BLG0G165
BLG0G167
BLG0G168
BLG0G170
BLG0G171
BLG0G172
BLG0G175
BLG0G180
BLG0G182
BLG0G183
BLG0G184
BLG0G185
BLG0G190
BLG0G191
BLG0G193
BLG0G195
BLG0G290
BLG0G293
BLG0G315
BLG0G335
BLG0G337
BLG0G338
BLG0G355
BLG0G359
BLG0G390
BLG0G391
BLG0G400
BLG0G490
BLG0G493
BLG0G500
BLG0G590
BLG0G593
BLG0G710
BLG0G725
BLG0G727
BLG0G728
BLG0G730
BLG0G735
BLG0G737
BLG0G738
BLG0G739
BLG0G740
BLG0G742
BLG0G745
BLG0G747
BLG0G748
BLG0G749
BLG0G750
BLG0G751
BLG0G770
BLG0G771
BLG0G772
BLG0G775
BLG0G780
BLG0G781
BLG0G782
BLG0G785
BLG0G787
BLG0G788
BLG0G790
BLG0G791
BLG0G792
BLG0G795
BLG0G850
BLG0G890
BLG0G893
BLG0HU10

BLG0HU11
BLG0H000
BLG0IU00
BLG0I200
BLG0JU00
BLG0JU10
BLG0JU11
BLG0J210
BLG0J212
BLG0K091
BLG0K190
BLG0K210
BLG0K291
BLG0K292
BLG0L100
BLG0L200
BLG0L300
BLG0L500
BLG0M100
BLG0M200
BLG0M300
BLG0M500
BLG0M600
BLG0M620
BLG0M630
BLG0N100
BLG0N101
BLG0N110
BLG0N120
BLG0N121
BLG0N122
BLG0N130
BLG0N132
BLG0N133
BLG0N140
BLG0N141
BLG0N142
BLG0N153
BLG0N154
BLG0N164
BLG0N200
BLG0N201
BLG0N210
BLG0N220
BLG0N221
BLG0N222
BLG0N230
BLG0N231
BLG0N232
BLG0N253
BLG0N254
BLG0N300
BLG0N307
BLG0N310
BLG0N350
BLG0N357
BLG0N360
BLG0N400
BLG0N410
BLG0N500
BLG0N600
BLG0N610
BLG0N700
BLG0N701
BLG0N710
BLG0N720
BLG0N721
BLG0N722
BLG0N740
BLG0N741
BLG0N742
BLG0N780
BLG0N781
BLG0N782
BLG0N790
BLG0N791
BLG0N792
BLG0PU00
BLG0P420
BLG0P520
BLG0P521
BLG0P820
BLG0P821
BLG0QUP0
BLG0Q100
BLG0Q210
BLG0Q213
BLG0Q214
BLG0Q215
BLG0RU00
BLG0R010
BLG0R011
BLG0R020
BLG0SU50
BLG0S010
BLG0S011
BLG0S020
BLG0S021
BLG0S060
BLG0S061
BLG0S510
BLG0TU01
BLG0TU02
BLG0T100
BLG0T101
BLG0U001
BLG0VD50
BLG0VD51
BLG0VD60
BLG0VD61
BLG0VD70
BLG0VD71
BLG0VE60
BLG0VE70
BLG0VE72
BLG0VE75
BLG0VQ50
BLG0VQ60
BLG0VQ70
BLG0VQ71
BLG0VQ72
BLG0VQ73
BLG0VQ74
BLG0VU50
BLG0VU60
BLG0VU70
BLG0V090
BLG0V190
BLG0V510
BLG0V700
BLG0V710
BLG0V720
BLG0V730
BLG0W000
BLG0W104
BLG0W105
BLG0W106
BLG0W107
BLG0W108
BLG0W109
BLG0W110
BLG0W111
BLG0W112
BLG0W113
BLG0W114
BLG0W115
BLG0W116
BLG0W117
BLG0W118
BLG0W119
BLG0W120
BLG0W200
BLG0W210
BLG0W300
BLG0W301
BLG0W302
BLG0W500
BLG0Y100
BLG0ZD50
BLG0ZD51
BLG0ZU00
BLG0ZU50
BLG0Z510
BLG0Z530
BLG00000
BLG00001
BLG00002
BLG00011
BLG00030
BLG1AACP
BLG1AAUP
BLG1ACOP
BLG1AF68
BLG1A094
BLG1A097
BLG1A1DD
BLG1A210
BLG1A511
BLG1A711
BLG1TADD
BLG1TBAC
BLG1TCOL
BLG1TCON
BLG1TCOS
BLG1TDAD
BLG1TDBA
BLG1TDDE
BLG1TDES
BLG1TDHD
BLG1TDJU
BLG1TDNO
BLG1TDRE
BLG1TDSR
BLG1TDST
BLG1TJUS
BLG1TNOT
BLG1TRES
BLG1TSTT
BLG1TVID
BLG1TVI2
BLG1UT20
BLG2ALIS
BLG2CMND
BLG2CU00
BLG2DATT
BLG2EN10
BLG2EN20
BLG2G115
BLG2G125
BLG2G126
BLG2G130
BLG2G135
BLG2G145
BLG2G155
BLG2G160
BLG2G165
BLG2G175
BLG2G180
BLG2G185
BLG2G195
BLG2G315
BLG2G335
BLG2G355
BLG2G400
BLG2G500
BLG2G710
BLG2G725
BLG2G730
BLG2G735
BLG2G740
BLG2G745
BLG2G750
BLG2G775
BLG2G785
BLG2G795
BLG2G850
BLG2J210
BLG2J211
BLG2J212
BLG2K210
BLG2Q210

BLG2Q211
BLG2S020
BLG2THLP
BLG2TVID
BLG2VQ50
BLG2VQ70
BLG2VU70
BLG2W500
BLG2ZCLA
BLG2ZCMD
BLG2ZCRE
BLG2ZD51
BLG2ZGEN
BLG2ZMSG
BLG2ZOSV
BLG2ZPAN
BLG2ZPRO
BLG2ZREP
BLG2ZSC6
BLG2ZSRH
BLG2Z500
BLG2O002
BLG4A002
BLG4A500
BLG4C001
BLG4C002
BLG4C004
BLG4C005
BLG4C006
BLG4C008
BLG4C017
BLG4C018
BLG4C019
BLG4C020
BLG4C021
BLG4C022
BLG4C023
BLG4C024
BLG4C025
BLG4C026
BLG4C029
BLG4C030
BLG4C032
BLG4C034
BLG4C039
BLG4C043
BLG4C045
BLG4C048
BLG4C052
BLG4C053
BLG4C054
BLG4C055
BLG4C056
BLG4C057
BLG4C058
BLG4C059
BLG4C062
BLG4C064
BLG4C066
BLG4C067
BLG4C068
BLG4C074
BLG4C075
BLG4C077
BLG4C078
BLG4C085
BLG4C086
BLG4C087
BLG4C089
BLG4C095
BLG4C098
BLG4C104
BLG4C113
BLG4C114
BLG4C122
BLG4C124
BLG4C125
BLG4C127
BLG4C128
BLG4C129
BLG4C140
BLG4C141
BLG4C500
BLG4C824
BLG4C835
BLG4C838
BLG4C839
BLG4C864
BLG4C870
BLG4C871
BLG4C918
BLG4D006
BLG4D008
BLG4D009
BLG4E003
BLG4E007
BLG4E008
BLG4E011
BLG4E023
BLG4E058
BLG4E060
BLG4E061
BLG4E062
BLG4E063
BLG4E064
BLG4F024
BLG4F026
BLG4F503
BLG4F506
BLG4F507
BLG4F508
BLG4F509
BLG4F510
BLG4H052
BLG4H056
BLG4H072
BLG4J003
BLG4J012
BLG4J022
BLG4J024
BLG4J028
BLG4J030
BLG4J031
BLG4J032
BLG4J033
BLG4K003
BLG4K005
BLG4K006
BLG4K007
BLG4K008
BLG4L001
BLG4L002
BLG4M005
BLG4N001
BLG4O007
BLG4O008
BLG4O009
BLG4O063
BLG4O064
BLG4O066
BLG4O074
BLG4O089
BLG4O104
BLG4O147
BLG4O148
BLG4O193
BLG4O194
BLG4O195
BLG4O196
BLG4O197
BLG4O199
BLG4O200
BLG4O201
BLG4O202
BLG4O203
BLG4O204
BLG4O205
BLG4O206
BLG4O207
BLG4O208
BLG4O209
BLG4O210
BLG4O211
BLG4O212
BLG4O213
BLG4O214
BLG4O215
BLG4O216
BLG4O217
BLG4O218
BLG4O220
BLG4O221
BLG4O222
BLG4O223
BLG4O224
BLG4O225
BLG4O226
BLG4O227
BLG4O228
BLG4O229
BLG4O230
BLG4O231
BLG4O242
BLG4O243
BLG4O244
BLG4O246
BLG4O460
BLG4O463
BLG4O464
BLG4O467
BLG4O468
BLG4O470
BLG4P006
BLG4P007
BLG4P015
BLG4P017
BLG4P018
BLG4P025
BLG4P027
BLG4P029
BLG4P030
BLG4P032
BLG4P502
BLG4P504
BLG4P505
BLG4P507
BLG4P508
BLG4P509
BLG4P510
BLG4P511
BLG4R504
BLG4R505
BLG4S000
BLG4S001
BLG4S100
BLG4S101
BLG4S102
BLG4S103
BLG4S104
BLG4S105
BLG4S107
BLG4S108
BLG4S109
BLG4S110
BLG4S202
BLG4S214
BLG4S217
BLG4S223
BLG4S224
BLG4S225
BLG4S226
BLG4S506
BLG4S509
BLG4T003
BLG4T029
BLG4U004
BLG4V502
BLG4V503
BLG4V504
BLG4V507
BLG4V508

BLG4V511
BLG4V512
BLG4V513
BLG4V550
BLG4V557
BLG4V558
BLG4W011
BLG4W027
BLG4W032
BLG4Z002
BLG6ALTD
BLG6ALTT
BLG6ASSD
BLG6BKSZ
BLG6CDFR
BLG6CDTO
BLG6CFMI
BLG6CLUS
BLG6CMPD
BLG6CPED
BLG6CPSD
BLG6CPUR
BLG6CRDT
BLG6DABA
BLG6DARC
BLG6DARR
BLG6DATX
BLG6DCBZ
BLG6DNOT
BLG6DSTA
BLG6EDSW
BLG6FMID
BLG6FMIE
BLG6HBSR
BLG6INSD
BLG6ISPF
BLG6JOIT
BLG6LOFI
BLG6NPED
BLG6NPSD
BLG6OCCD
BLG6PFKY
BLG6PPED
BLG6PPSD
BLG6PREX
BLG6PRTP
BLG6PURD
BLG6PWIN
BLG6REQD
BLG6REQN
BLG6RERE
BLG6RWIN
BLG6SCHD
BLG6TARD
BLG6VDTQ
BLG6VPAD
BLG600AA
BLG600CD
BLG9C022
BLG9C023
BLG9C024
BLG9C025
BLG9C026
BLG9C028
BLG9C029
BLG9C030
BLG9C033
BLG9C074
BLG9C077
BLG9C078
BLG9C080
BLG9C081
BLG9C086
BLG9C088
BLG9C095
BLG9C096
BLG9C097
BLG9C100
BLG9C101
BLG9C102
BLG9C103
BLG9C104
BLG9C105
BLG9C106
BLG9C107
BLG9C108
BLG9C109
BLG9C110
BLG9C111
BLG9C112
BLG9C113
BLG9C114
BLG9C115
BLG9C117
BLG9C118
BLG9C129
BLG9C130
BLG9C131
BLG9C132
BLG9C133
BLG9C134
BLG9C135
BLG9C136
BLG9C137
BLG9C919
BLG9D001
BLG9D012
BLG9E026
BLG9E050
BLG9E051
BLG9E052
BLG9E053
BLG9E054
BLG9E057
BLG9E058
BLG9F002
BLG9F003
BLG9F004
BLG9F005
BLG9F020
BLG9F021
BLG9F022
BLG9F023
BLG9F024
BLG9F025
BLG9F026
BLG9F027
BLG9G015
BLG9H058
BLG9H059
BLG9H072
BLG9I008
BLG9J014
BLG9J028
BLG9K003
BLG9K006
BLG9K007
BLG9K008
BLG9L001
BLG9L002
BLG9N001
BLG9N002
BLG9O052
BLG9O055
BLG9O061
BLG9O062
BLG9O063
BLG9O064
BLG9O074
BLG9P018
BLG9P024
BLG9P025
BLG9P031
BLG9P032
BLG9P034
BLG9R002
BLG9S216
BLG9S223
BLG9T001
BLG9V003
BLG9W009
BLG9W010
BLG9W013
BLG9W015
BLG9W033
BLG9W034
BLG9Z002
BLM0AC02
BLM0YE00
BLM0YU00
BLM0YU10
BLM0YU11
BLM0YU50
BLM0YU51
BLM0YU60
BLM0YU61
BLM0YU80
BLM0YU90
BLM0YU91
BLM0Y100
BLM0Y203
BLM0Y251
BLM0Y261
BLM0Y300
BLM0Y310
BLM0Y330
BLM0Y600
BLM0Y650
BLM0Y703
BLM0Y713
BLM0Y751
BLM0Y761
BLM0Y771
BLM0Y781
BLM0Y793
BLM0Y800
BLM0Y810
BLM0Y830
BLM0Y850
BLM0Y860
BLM0Y880
BLM1AC05
BLM1T218
BLM1T221
BLM1T223
BLM1T224
BLM1T225
BLM1T226
BLM1T228
BLM1T229
BLM1T260
BLM1Y115
BLM1Y615
BLM1Y655
BLM2CG00
BLM2CN00
BLM2CR10
BLM2CUA0
BLM2CU00
BLM2C000
BLM2C030
BLM2FD17
BLM2FD37
BLM2FD45
BLM2FD46
BLM2FD48
BLM2FD55
BLM2FD58
BLM2FD77
BLM2FD89
BLM2FD9A
BLM2FD91
BLM2FD94
BLM2FD99
BLM2PC01
BLM2PC04
BLM2PC05
BLM2PC1B
BLM2PC1D
BLM2PC1G
BLM2PC1H

BLM2PC1M
BLM2PC10
BLM2PC19
BLM2PS05
BLM2PT09
BLM2P020
BLM2TPDS
BLM2YE00
BLM2YE50
BLM2Y250
BLM2Y750
BLM2Y770
BLM2Y780
BLM2ZIMN
BLM4C077
BLM4C078
BLM4C090
BLM4D001
BLM4D004
BLM4D005
BLM4D008
BLM4D015
BLM4D016
BLM4D038
BLM4D039
BLM4D059
BLM4D066
BLM4D072
BLM4D082
BLM4D096
BLM4D098
BLM4D115
BLM4D140
BLM4D142
BLM4D152
BLM4D153
BLM4D154
BLM4D237
BLM4D300
BLM4I011
BLM4I014
BLM4I015
BLM4I016
BLM4I020
BLM4I023
BLM4I024
BLM4I025
BLM4I027
BLM4J009
BLM4J020
BLM4J021
BLM4M002
BLM4R005
BLM4R006
BLM4R010
BLM4R025
BLM4R059
BLM4R070
BLM4R071
BLM4R072
BLM4R073
BLM4R076
BLM4R077
BLM4R102
BLM4R105
BLM4S001
BLM6CDCA
BLM6CGMX
BLM6CRTA
BLM6CSVL
BLM6NPNL
BLM6PANA
BLM6PARA
BLM6PLMD
BLM6PREX
BLM6VENA
BLM6Y00E
BLM6Y007
BLM8CL10
BLM8CL20
BLM8CL50
BLM8CL60
BLM8CUA0
BLM8CU17
BLM8CU27
BLM8CU37
BLM8CU47
BLM8CU5D
BLM8CU57
BLM8CU67
BLM8CU77
BLM8CU85
BLM8CU87
BLM8CU97
BLM9C001
BLM9D002
BLM9D003
BLM9D004
BLM9D005
BLM9D006
BLM9D009
BLM9D010
BLM9D012
BLM9D020
BLM9D021
BLM9D023
BLM9D024
BLM9D025
BLM9D031
BLM9D032
BLM9D034
BLM9D037
BLM9D039
BLM9D040
BLM9D055
BLM9D058
BLM9D066
BLM9D067
BLM9D077
BLM9D078
BLM9D079
BLM9D098
BLM9D101
BLM9D152
BLM9D153
BLM9D237
BLM9P015
BLM9P100
BLM9P101
BLM9R069
BLM9R070
BLM9R071
BLM9R072
BLM9R073
BLM9R076
BLM9S001
BLM9S002
BLM9S003
BLM9S004
BLX4C104
BLX4C113
BLX4C120
BLX4C127
BLX4C128
BLX4C129
BLX4C140
BLX4C141
BLX4C142
BLX4C170
BLX4C172
BLX4C173
BLX4C195
BLX4C196
BLX4C197
BLX4C198
BLX4S022
BLX4T101
BLX4T102
BTN0BU00
BTN0BU10
BTN0BU11
BTN0B100
BTN0B102
BTN0B200
BTN0B300
BTN0B500
BTN0CU00
BTN0CU01
BTN0C100
BTN0C101
BTN0C200
BTN0C300
BTN0DU01
BTN0DU03
BTN0DU08
BTN0DU18
BTN0DU28
BTN0DU30
BTN0DU31
BTN0D100
BTN0D102
BTN0D200
BTN0D500
BTN0ENOM
BTN0EN20
BTN0EPPC
BTN0E000
BTN0E090
BTN0E190
BTN0E290
BTN0E390
BTN0E890
BTN0F000
BTN0F090
BTN0F190
BTN0F290
BTN0F390
BTN0G000
BTN0G1A1
BTN0G1A3
BTN0G190
BTN0G191
BTN0JU00
BTN0L100
BTN0L200
BTN0L300
BTN0L500
BTN0M100
BTN0M200
BTN0N100
BTN0N101
BTN0N200
BTN0N201
BTN0N500
BTN0N510
BTN0Q100
BTN0S010
BTN0S020
BTN0S021
BTN0W020
BTN0XSUM
BTN00000
BTN00001
BTN1TDDE
BTN1TDST
BTN1TSC1
BTN1TSC4
BTN1TSF3
BTN1TSF6
BTN1TSRN
BTN1TSSE
BTN1TSST
BTN2ENCC
BTN2ENCF
BTN2ENHD
BTN2ENNO
BTN2ENOM
BTN2ENPC
BTN2ENSU
BTN2ENSY
BTN20000

BTN20001	BTN6DNOT	BTN600CD	EYM6ALTD
BTN4J001	BTN6DSTA	BTN7CMPD	EYM6CRDT
BTN6ALTD	BTN6DSTT	BTN7TARD	EYM9MAIN
BTN6ASSD	BTN6ENDD	BTN9C009	EYM9M901
BTN6CMDP	BTN6FMID	EYMAM100	EYM9M902
BTN6CMPD	BTN6OCCD	EYMBM100	EYM9M903
BTN6CRDT	BTN6REQD	EYM5M110	EYM9M904
BTN6DATE	BTN6SCHD	EYM5M200	EYM9POST
BTN6DATX	BTN6TARD	EYM5M250	

Removed Panels – Tivoli Service Desk for OS/390 Version 1.2

BLG3S223
BLG4S106
EYMTSPPE

New Panels – Information/Management Version 1.1

BLGTCMIG	EYMAM100	EYM2M510	EYM6RNOD
BLG0DFSM	EYMBM100	EYM4M901	EYM6SPFX
BLG1DFSM	EYMBM500	EYM4M902	EYM6TFLD
BLG2ZSC6	EYMTSPPE	EYM4M903	EYM6TPAN
BLG6COLN	EYM1MBAK	EYM4M904	EYM6TPNI
BLG6DVCN	EYM1MCOP	EYM5MARK	EYM6URN0
BLG6ODBC	EYM1M500	EYM5M100	EYM6USER
BLM1T260	EYM1M501	EYM5M110	EYM6USRE
BLM2FD0B	EYM1M92A	EYM5M200	EYM9MAIN
BLM2FD0C	EYM1M92B	EYM5M250	EYM9MAPB
BLM2FD0D	EYM1M921	EYM5M500	EYM9MAPE
BLM2FD6A	EYM1M925	EYM6ACCN	EYM9M500
BLM4D049	EYM1M926	EYM6ALTD	EYM9M7XC
BLM6CDCA	EYM1M928	EYM6ALTT	EYM9M7XI
BLM6CGMX	EYM1M929	EYM6CLAE	EYM9M901
BLM6CSCM	EYM2M100	EYM6CRDT	EYM9M902
BLM6CSVL	EYM2M110	EYM6CRTM	EYM9M903
BLM6INCT	EYM2M120	EYM6DSAB	EYM9M904
BLM6PLMD	EYM2M130	EYM6MARK	EYM9POST
BLM6PLMT	EYM2M200	EYM6POWN	EYM9RSET
BLM6PLMU	EYM2M250	EYM6RIDN	EYM9XMIT
BLM9D049	EYM2M500		

Changed Panels – Information/Management Version 1.1

The following panels had internal or external changes in Version 1.1. For a list of the panels associated with expanded (10-digit) date fields, see the members listed in the BLM.V1R1M0.SBLMPNLY data set.

BLGAPI00	BLG0V720	BLG6VDTA	BLM6NEWD
BLGAPI05	BLG0V730	BLG7ARG1	BLM6PVAL
BLGAPI10	BLG00001	BLG7DSAB	BLM6SODI
BLGLDATT	BLG00002	BLG9C062	BLM6SOPF
BLGTDBXM	BLG00010	BLM1AT02	BLM8CL00
BLGTEXST	BLG00020	BLM1A40E	BLM8CL10
BLG0AC12	BLG00011	BLM1A40F	BLM8CL20
BLG0AC46	BLG1ACOP	BLM1A40J	BLM8CL50
BLG0DATT	BLG1A000	BLM1A40M	BLM8CL60
BLG0EN10	BLG1A001	BLM1A806	BLM8CU5D
BLG0EN20	BLG1A1G1	BLM1TSU4	BLM8CU6B
BLG0VD70	BLG1A1G2	BLM1T209	BLM8CU64
BLG0VD71	BLG1A1G8	BLM1T255	BLM8CU85
BLG0VE61	BLG1A1G9	BLM2CR10	BLM8CU9E
BLG0VE71	BLG1A210	BLM2CR20	BLM8CU9F
BLG0VE72	BLG1A604	BLM2FD17	BLM8CU9J
BLG0VQ52	BLG1A704	BLM2FD59	BLM8CU9U
BLG0VQ61	BLG1DARC	BLM2FD60	BLM8CU9W
BLG0VQ70	BLG1TVID	BLM2FD61	BLM8CU9Y
BLG0VQ71	BLG2TVID	BLM2PC04	BLM9D119
BLG0VQ72	BLG4C026	BLM2PC1L	BTN0EN20
BLG0VQ73	BLG4C062	BLM2PC1N	BTN0Q300
BLG0VQ74	BLG40195	BLM2PC1P	EYMAPI00
BLG0VU70	BLG4V517	BLM2PS07	EYM00011
BLG0V502	BLG6DSAB	BLM2ZIMN	EYM00020
BLG0V600	BLG6SCVD	BLM4D119	EYM1ACOP
BLG0V700	BLG6SOVD	BLM6LIDA	EYM1A210
BLG0V710	BLG6VDES		

Removed Panels – Information/Management Version 1.1

No panels were removed in Version 1.1.

New Panels – Information/Management Version 6.3

BLGAPIDI
BLGAPIPX
BLGLCMDD
BLGLCMDU
BLGLDATT
BLGLJHPL
BLGLJSPL
BLGLQHPL
BLGLQSPL
BLGLUSER
BLGTFCMD
BLGTSALI
BLGTSCMD
BLGTSDAR
BLGTSUSR
BLGTSVAL
BLG0DATT
BLG0HU10
BLG0HU11
BLG0USER
BLG0VD60
BLG0VD61
BLG0VD70
BLG0VD71
BLG0VE60
BLG0VE61
BLG0VE62
BLG0VE70
BLG0VE71
BLG0VE72
BLG0VE73
BLG0VE74
BLG0VE75
BLG0VQ52
BLG0VQ60
BLG0VQ61
BLG0VQ62
BLG0VQ70
BLG0VQ71
BLG0VQ72
BLG0VQ73
BLG0VQ74
BLG0VU60
BLG0VU70
BLG0V502
BLG0V600
BLG0V601
BLG0V610
BLG0V700
BLG0V701
BLG0V710
BLG0V720
BLG0V730
BLG00040
BLG00041
BLG1AJH0
BLG1AJ50
BLG1AQ52
BLG1A502
BLG1A503
BLG1A600
BLG1A604
BLG1A606
BLG1A607
BLG1A608
BLG1A609
BLG1A611
BLG1A700
BLG1A704
BLG1A706
BLG1A707
BLG1A708
BLG1A709
BLG1A711
BLG1DARC
BLG1TPND
BLG1TPNL
BLG1TSRP
BLG1VU50
BLG1VU60
BLG2APNO
BLG2CAEP
BLG2CMDD
BLG2CMND
BLG2DATT
BLG2JHPL
BLG2JSPL
BLG2QHPL
BLG2QSPL
BLG2SEPC
BLG2TPND
BLG2TPNL
BLG2USER
BLG2VQ60
BLG2VQ70
BLG2VU60
BLG2VU70
BLG20040
BLG4C000
BLG4E062
BLG4E063
BLG4E064
BLG4E065
BLG4E066
BLG4L002
BLG4N002
BLG4N003
BLG4N004
BLG4N005
BLG4N006
BLG4N007
BLG4N008
BLG4N009
BLG4N010
BLG4T028
BLG4T029
BLG6ADDR
BLG6APNO
BLG6AUCO
BLG6AUTC
BLG6AUUA
BLG6BOOP
BLG6CACT
BLG6CAEP
BLG6CALI
BLG6CAUT
BLG6CHPN
BLG6COGN
BLG6CRTA
BLG6DADA
BLG6DADD
BLG6DAFI
BLG6DARC
BLG6DARR
BLG6DATH
BLG6DIEN
BLG6DIOV
BLG6DMDR
BLG6DTPE
BLG6FANY
BLG6FOEN
BLG6GNRC
BLG6HETA
BLG6HPID
BLG6INTA
BLG6JOIT
BLG6JOI1
BLG6KSPL
BLG6LIST
BLG6LPRN
BLG6NRTA
BLG6PATY
BLG6PCGZ
BLG6PPID
BLG6PREN
BLG6PREX
BLG6PRFX
BLG6PRIN
BLG6PRIX
BLG6PWIN
BLG6RACC
BLG6RECA
BLG6RELE
BLG6RENO
BLG6RERE
BLG6RWIN
BLG6SARE
BLG6SCVD
BLG6SEPC
BLG6SODI
BLG6SOVD
BLG6SPCH
BLG6SPLD
BLG6SWIN
BLG6TEXT
BLG6TSPT
BLG6UNA1
BLG6VDDS
BLG6VDIN
BLG6VRNM
BLG6VTSP
BLG6VVIN
BLG7AUCO
BLG8DARC
BLG8SWIN
BLG8VDDS
BLG8VDIN
BLG8VVIN
BLG9C000
BLG9E062
BLG9E063
BLG9E064
BLG9E065
BLG9E066
BLG9L002
BLG9N002
BLG9N003
BLG9N004
BLG9N005
BLG9N006
BLG9N007
BLG9N008
BLG9N009
BLG9N010
BLG9T028
BLG9T029
BLM2FD0A
BLM2VDDS
BLM4D145
BLM4D146
BLM4D150
BLM4D151
BLM4D152
BLM4D153
BLM4D154
BLM6TSNM
BLM6VDDS
BLM9D145
BLM9D146
BLM9D150
BLM9D151
BLM9D152
BLM9D153
BLM9D154
BTN1AJH0
BTN1MSN1
BTN1MSN2
BTN4N002
BTN9N002

Changed Panels – Information/Management Version 6.3

BLGAPI02
BLGAPI05
BLGLALIS
BLGLALSD
BLGLVLST
BLGLVSEL
BLG0AAE0
BLG0AC12
BLG0AC42
BLG0B901
BLG0EN10
BLG0EN20
BLG0GU10
BLG0JU00
BLG0JU10
BLG0JU11
BLG0J100
BLG0J212
BLG0J250
BLG0K091
BLG0K100
BLG0K190
BLG0K212
BLG0K292
BLG0L901
BLG0Q100
BLG0Q213
BLG0Q215
BLG0S010
BLG0VE71
BLG0VLST
BLG0VQ50
BLG0VU50
BLG0V510
BLG00000
BLG00001
BLG00002
BLG00020
BLG1ATBL
BLG1A000
BLG1A001
BLG1A1G1
BLG1A1G2
BLG1A1G8
BLG1A1G9
BLG1A511
BLG1A911
BLG2ABND
BLG2ALIS
BLG2APNU
BLG2BONU
BLG2CNAM
BLG2CRDN
BLG2DEVF
BLG2DEVL
BLG2DFEA
BLG2DMSN
BLG2DNUM
BLG2DOCN
BLG2DTYP
BLG2DVNA
BLG2DVSR
BLG2ECID
BLG2ECLV
BLG2ECNU
BLG2ENVF
BLG2ENVS
BLG2ERCE
BLG2FEAT
BLG2FLDF
BLG2FMIE
BLG2FNAM
BLG2INUM
BLG2IPID
BLG2JU00
BLG2J212
BLG2K000
BLG2K212
BLG2LCRM
BLG2LIDE
BLG2LOCO
BLG2LSPE
BLG2LTYE
BLG2LVLF
BLG2LVLP
BLG2MAUF
BLG2MIUF
BLG2MNAM
BLG2MODF
BLG2MSGC
BLG2NPRO
BLG2NUPR
BLG2OPCF
BLG2OPCS
BLG2PCSF
BLG2PIDF
BLG2PRID
BLG2PTFF
BLG2PTNU
BLG2PUBF
BLG2PUBS
BLG2PZFF
BLG2Q100
BLG2REAN
BLG2REGF
BLG2RMID
BLG2ROUF
BLG2ROUS
BLG2RPQN
BLG2SLEV
BLG2SPID
BLG2SQMP
BLG2STMT
BLG2SUID
BLG2SWCC
BLG2SWRC
BLG2SYMC
BLG2THLP
BLG2UMID
BLG2UTID
BLG2VLST
BLG2VSEL
BLG2ZOSV
BLG2ZSC2
BLG2ZSC3
BLG2ZSC4
BLG2ZSC5
BLG2ZSOC
BLG20000
BLG20020
BLG4C012
BLG4E060
BLG4E061
BLG4N001
BLG4P030
BLG6ALIS
BLG6HPID
BLG6KSPL
BLG6PANL
BLG6PPID
BLG6SPLD
BLG6URN3
BLG6VALR
BLG6VDES
BLG6VDSQ
BLG6VDTA
BLG9C012
BLG9C064
BLG9C141
BLG9E060
BLG9E061
BLG9N001
BLG9P030
BLM1AA03
BLM1T254
BLM2FD0A
BLM2FD71
BLM2PC04
BLM2PC06
BLM2ZIMN
BLM4D107
BLM4D113
BLM4D114
BLM4D115
BLM4D121
BLM4D127
BLM4D142
BLM4D144
BLM4D202
BLM6FANY
BLM6PETN
BLM8CU5B
BLM8CU89
BLM9D107
BLM9D113
BLM9D114
BLM9D115
BLM9D121
BLM9D127
BLM9D142
BLM9D202
BLM9D244
BTNT00P1
BTN0EN20
BTN0JU00
BTN0J100
BTN0J200
BTN0J300
BTN0Q100
BTN0Q200
BTN0XAUT
BTN0XDES
BTN0XSUM
BTN0XUSR
BTN1XAUT

Removed Panels – Information/Management Version 6.3

BLG0AC37

New Panels – Information/Management Version 6.2

BLGLVLST
BLGLVSEL
BLG0AC46
BLG0VD50
BLG0VD51
BLG0V500
BLG0V501
BLG0V510
BLG1AFTX
BLG1A50A
BLG1A507
BLG1A508
BLG1A509
BLG1A511
BLG6VALR
BLG6VDTQ
BLM1T254
BLM6JUST
BLM6SUPC
BLM6VDIN

BLG0VLST	BLG1A500	BLG6VDES	BLM6VRNM
BLG0VQ50	BLG1A504	BLG6VDSQ	BLM6VVIN
BLG0VU50	BLG1A506	BLG6VDTA	

Changed Panels – Information/Management Version 6.2

The following panels had both internal and external changes in Version 6.2:

BLGTDBXM	BLG1A1CB	BLM1AA03	BLM8CU7B
BLGTPSET	BLG1A1DD	BLM1AC0D	BLM8CU8B
BLG0F173	BLG1A19B	BLM1A205	BTNMIGD2
BLG0J100	BLG6DEVL	BLM1S201	BTNTA112
BLG00000	BLG6ISPF	BLM1T208	BTNTPE02
BLG00001	BLG6PNAM	BLM1T253	BTN1A153
BLG1A1AB	BLG7ALTT	BLM8CU5B	BTN1A173

The following panels had only external changes made for Version 6.2:

BLGAPI02	BLGLECN4	BLGLPRID	BLGLUMI5
BLGDUMP0	BLGLECN5	BLGLPTFF	BLGLUTID
BLGDUMP1	BLGLENVF	BLGLPTNU	BLGLUTI1
BLGESCAL	BLGLENVS	BLGLPTN1	BLGLUTI2
BLGESC02	BLGLERCE	BLGLPTN2	BLGLUTI3
BLGLABND	BLGLFEAT	BLGLPTN3	BLGLUTI4
BLGLALIS	BLGLFEA1	BLGLPTN4	BLGLUTI5
BLGLALSD	BLGLFEA2	BLGLPTN5	BLGMIG3D
BLGLAPNU	BLGLFEA3	BLGLPUBF	BLGMIG3E
BLGLAPN1	BLGLFEA4	BLGLPUBS	BLGMIG31
BLGLAPN2	BLGLFEA5	BLGLPZFF	BLGMIG32
BLGLAPN3	BLGLFLDF	BLGLREAN	BLGMIG33
BLGLAPN4	BLGLFMIE	BLGLREA1	BLGMIG35
BLGLAPN5	BLGLFNAM	BLGLREA2	BLGMIG37
BLGLBONU	BLGLFNA1	BLGLREA3	BLGMIG39
BLGLBON1	BLGLFNA2	BLGLREA4	BLGMIG49
BLGLBON2	BLGLFNA3	BLGLREA5	BLGMIG62
BLGLBON3	BLGLFNA4	BLGLREGF	BLGMIG91
BLGLBON4	BLGLFNA5	BLGLRMID	BLGMIG92
BLGLBON5	BLGLINUM	BLGLRMI1	BLGMIG93
BLGLCNAM	BLGLIPID	BLGLRMI2	BLGMIG94
BLGLCNA1	BLGLIPI1	BLGLRMI3	BLGRFALS
BLGLCNA2	BLGLIPI2	BLGLRMI4	BLGRFHBK
BLGLCNA3	BLGLIPI3	BLGLRMI5	BLGRFUSR
BLGLCNA4	BLGLIPI4	BLGLROUF	BLGSSQMP
BLGLCNA5	BLGLIPI5	BLGLROUS	BLGTAC1R
BLGLCRDN	BLGLLCRM	BLGLRPQN	BLGTDBX1
BLGLDEVF	BLGLLIDE	BLGLRPQ1	BLGTZSTA
BLGLDEVL	BLGLLOC0	BLGLRPQ2	BLG0AC45
BLGLDFEA	BLGLLOC1	BLGLRPQ3	BLG0BU00
BLGLDMSN	BLGLLOC2	BLGLRPQ4	BLG0BU10
BLGLDMS1	BLGLLOC3	BLGLRPQ5	BLG0BU11
BLGLDMS2	BLGLLOC4	BLGLSLEV	BLG0BU90
BLGLDMS3	BLGLLOC5	BLGLSLE1	BLG0B100
BLGLDMS4	BLGLLSPE	BLGLSLE2	BLG0B200
BLGLDMS5	BLGLLTYE	BLGLSLE3	BLG0B300
BLGLDNUM	BLGLLVLF	BLGLSLE4	BLG0B400
BLGLDNU1	BLGLLVLP	BLGLSLE5	BLG0B402
BLGLDNU2	BLGLMAUF	BLGLSPID	BLG0B500
BLGLDOCN	BLGLMIUF	BLGLSPI1	BLG0B502
BLGLDOC1	BLGLMNAM	BLGLSPI2	BLG0B901
BLGLDOC2	BLGLMNA1	BLGLSPI3	BLG0B902
BLGLDTYP	BLGLMNA2	BLGLSPI4	BLG0B903
BLGLDVNA	BLGLMNA3	BLGLSPI5	BLG0CU00
BLGLDVN1	BLGLMNA4	BLGLSQMP	BLG0CU01
BLGLDVN2	BLGLMNA5	BLGLSTMT	BLG0CU10
BLGLDVN3	BLGLMODF	BLGLSUID	BLG0CU11
BLGLDVN4	BLGLMSGC	BLGLSUI1	BLG0CU21
BLGLDVN5	BLGLNPR0	BLGLSUI2	BLG0CU30
BLGLDVSR	BLGLNPR1	BLGLSUI3	BLG0CU31
BLGLECID	BLGLNPR2	BLGLSUI4	BLG0C040
BLGLECI1	BLGLNUPR	BLGLSUI5	BLG0C051
BLGLECI2	BLGLNUP1	BLGLSWCC	BLG0C052
BLGLECI3	BLGLNUP2	BLGLSWRC	BLG0C053
BLGLECI4	BLGLOAD0	BLGLSYMC	BLG0C054
BLGLECI5	BLGLOAD1	BLGLUMID	BLG0C055
BLGLECLV	BLGLOPCF	BLGLUMI1	BLG0C056
BLGLECNU	BLGLOPCS	BLGLUMI2	BLG0C061
BLGLECN1	BLGLPCSF	BLGLUMI3	BLG0C062
BLGLECN2	BLGLPIDF	BLGLUMI4	BLG0C063
BLGLECN3			

BLG0C064
BLG0C065
BLG0C066
BLG0C071
BLG0C072
BLG0C073
BLG0C100
BLG0C151
BLG0C152
BLG0C153
BLG0C154
BLG0C155
BLG0C156
BLG0C161
BLG0C162
BLG0C163
BLG0C164
BLG0C165
BLG0C166
BLG0C171
BLG0C172
BLG0C173
BLG0C200
BLG0C300
BLG0C400
BLG0C422
BLG0C500
BLG0C600
BLG0C620
BLG0C630
BLG0C640
BLG0C700
BLG0DU01
BLG0DU02
BLG0DU03
BLG0DU04
BLG0DU05
BLG0DU06
BLG0DU07
BLG0DU08
BLG0DU09
BLG0DU10
BLG0DU11
BLG0DU12
BLG0DU13
BLG0DU14
BLG0DU15
BLG0DU16
BLG0DU17
BLG0DU18
BLG0DU19
BLG0DU20
BLG0DU21
BLG0DU22
BLG0DU23
BLG0DU24
BLG0DU25
BLG0DU26
BLG0DU27
BLG0DU28
BLG0DU29
BLG0DU30
BLG0DU31
BLG0DU32
BLG0DU33
BLG0DU34
BLG0DU35
BLG0DU36
BLG0DU37
BLG0DU38
BLG0DU39
BLG0DU40
BLG0DU42
BLG0DU43
BLG0DU44
BLG0DU45
BLG0DU46
BLG0DU47
BLG0DU48
BLG0DU49
BLG0DU50
BLG0DU51
BLG0DU52
BLG0DU53
BLG0DU54
BLG0DU55
BLG0DU56
BLG0DU57
BLG0DU58
BLG0DU59
BLG0DU60
BLG0DU61
BLG0DU62
BLG0DU63
BLG0DU64
BLG0DU65
BLG0DU66
BLG0DU67
BLG0DU68
BLG0DU69
BLG0DU70
BLG0DU71
BLG0DU72
BLG0DU73
BLG0DU74
BLG0DU75
BLG0DU76
BLG0D040
BLG0D100
BLG0D102
BLG0D104
BLG0D105
BLG0D107
BLG0D113
BLG0D120
BLG0D130
BLG0D140
BLG0D141
BLG0D150
BLG0D151
BLG0D152
BLG0D153
BLG0D154
BLG0D157
BLG0D160
BLG0D164
BLG0D170
BLG0D172
BLG0D200
BLG0D204
BLG0D205
BLG0D220
BLG0D230
BLG0D250
BLG0D251
BLG0D253
BLG0D254
BLG0D257
BLG0D300
BLG0D302
BLG0D305
BLG0D306
BLG0D307
BLG0D350
BLG0D353
BLG0D355
BLG0D356
BLG0D357
BLG0D400
BLG0D401
BLG0D500
BLG0D501
BLG0D520
BLG0D600
BLG0D601
BLG0D620
BLG0D700
BLG0D701
BLG0D702
BLG0D703
BLG0D704
BLG0D705
BLG0D706
BLG0D707
BLG0D708
BLG0D709
BLG0D720
BLG0D740
BLG0D750
BLG0D751
BLG0D752
BLG0D753
BLG0D754
BLG0D757
BLG0D758
BLG0D759
BLG0D760
BLG0D764
BLG0D770
BLG0D772
BLG0D780
BLG0D783
BLG0D784
BLG0D787
BLG0D790
BLG0ENTR
BLG0EN10
BLG0EN20
BLG0E090
BLG0E102
BLG0E190
BLG0E202
BLG0E290
BLG0E301
BLG0E390
BLG0E402
BLG0E490
BLG0E590
BLG0E690
BLG0E700
BLG0E890
BLG0F090
BLG0F111
BLG0F122
BLG0F131
BLG0F150
BLG0F174
BLG0F175
BLG0F190
BLG0F211
BLG0F222
BLG0F231
BLG0F270
BLG0F271
BLG0F272
BLG0F273
BLG0F278
BLG0F280
BLG0F281
BLG0F282
BLG0F283
BLG0F288
BLG0F290
BLG0F311
BLG0F322
BLG0F331
BLG0F390
BLG0F411
BLG0F440
BLG0F445
BLG0F450
BLG0F590
BLG0F890
BLG0GU10
BLG0GU11
BLG0G000
BLG0G1A0
BLG0G1A1
BLG0G1A2
BLG0G1A3
BLG0G1A4

BLG0G1A5
BLG0G1B0
BLG0G1B1
BLG0G1B2
BLG0G1B3
BLG0G1B4
BLG0G1B5
BLG0G1C0
BLG0G1D0
BLG0G100
BLG0G113
BLG0G121
BLG0G128
BLG0G130
BLG0G131
BLG0G135
BLG0G136
BLG0G137
BLG0G138
BLG0G141
BLG0G142
BLG0G151
BLG0G161
BLG0G166
BLG0G167
BLG0G168
BLG0G170
BLG0G171
BLG0G172
BLG0G175
BLG0G176
BLG0G181
BLG0G182
BLG0G183
BLG0G184
BLG0G186
BLG0G190
BLG0G191
BLG0G192
BLG0G193
BLG0G196
BLG0G290
BLG0G293
BLG0G313
BLG0G316
BLG0G336
BLG0G337
BLG0G338
BLG0G351
BLG0G357
BLG0G358
BLG0G359
BLG0G362
BLG0G363
BLG0G390
BLG0G391
BLG0G392
BLG0G411
BLG0G490
BLG0G493
BLG0G511
BLG0G590
BLG0G593
BLG0G700
BLG0G713
BLG0G726
BLG0G727
BLG0G728
BLG0G730
BLG0G731
BLG0G735
BLG0G736
BLG0G737
BLG0G738
BLG0G739
BLG0G741
BLG0G742
BLG0G746
BLG0G747
BLG0G748
BLG0G749
BLG0G751
BLG0G762
BLG0G770
BLG0G771
BLG0G772
BLG0G777
BLG0G780
BLG0G781
BLG0G782
BLG0G786
BLG0G787
BLG0G788
BLG0G790
BLG0G791
BLG0G792
BLG0G796
BLG0G800
BLG0G851
BLG0G890
BLG0G893
BLG0IU00
BLG0I0E0
BLG0I030
BLG0I040
BLG0I100
BLG0I200
BLG0JU00
BLG0JU10
BLG0JU11
BLG0J200
BLG0J210
BLG0J211
BLG0J212
BLG0J220
BLG0J230
BLG0J240
BLG0J250
BLG0J260
BLG0J270
BLG0J290
BLG0J300
BLG0K091
BLG0K100
BLG0K190
BLG0K2A0
BLG0K200
BLG0K210
BLG0K211
BLG0K212
BLG0K220
BLG0K230
BLG0K240
BLG0K250
BLG0K260
BLG0K270
BLG0K290
BLG0K291
BLG0K292
BLG0K293
BLG0K800
BLG0L100
BLG0L200
BLG0L300
BLG0L400
BLG0L500
BLG0L600
BLG0L901
BLG0L902
BLG0L903
BLG0M100
BLG0M200
BLG0M300
BLG0M500
BLG0M600
BLG0M620
BLG0M630
BLG0M700
BLG0N040
BLG0N100
BLG0N101
BLG0N105
BLG0N110
BLG0N113
BLG0N120
BLG0N121
BLG0N122
BLG0N130
BLG0N132
BLG0N133
BLG0N134
BLG0N140
BLG0N141
BLG0N142
BLG0N150
BLG0N151
BLG0N153
BLG0N154
BLG0N157
BLG0N160
BLG0N164
BLG0N170
BLG0N172
BLG0N200
BLG0N201
BLG0N205
BLG0N210
BLG0N220
BLG0N221
BLG0N222
BLG0N230
BLG0N231
BLG0N232
BLG0N250
BLG0N251
BLG0N253
BLG0N254
BLG0N257
BLG0N300
BLG0N301
BLG0N305
BLG0N306
BLG0N307
BLG0N310
BLG0N320
BLG0N350
BLG0N352
BLG0N353
BLG0N355
BLG0N357
BLG0N360
BLG0N400
BLG0N410
BLG0N500
BLG0N510
BLG0N600
BLG0N610
BLG0N700
BLG0N701
BLG0N710
BLG0N720
BLG0N721
BLG0N722
BLG0N740
BLG0N741
BLG0N742
BLG0N750
BLG0N751
BLG0N753
BLG0N754
BLG0N757
BLG0N758
BLG0N759
BLG0N760
BLG0N764
BLG0N770
BLG0N772
BLG0N780
BLG0N781
BLG0N782
BLG0N783

BLG0N784
BLG0N787
BLG0N790
BLG0N791
BLG0N792
BLG0P100
BLG0P101
BLG0P200
BLG0P300
BLG0P410
BLG0P420
BLG0P430
BLG0P510
BLG0P511
BLG0P520
BLG0P521
BLG0P530
BLG0P531
BLG0P700
BLG0P810
BLG0P811
BLG0P820
BLG0P821
BLG0P830
BLG0P831
BLG0QUP0
BLG0Q100
BLG0Q200
BLG0Q210
BLG0Q211
BLG0Q212
BLG0Q213
BLG0Q214
BLG0Q215
BLG0Q216
BLG0Q220
BLG0Q230
BLG0Q240
BLG0Q250
BLG0Q260
BLG0Q270
BLG0Q290
BLG0Q300
BLG0RU00
BLG0R010
BLG0R011
BLG0R020
BLG0R100
BLG0SU50
BLG0S010
BLG0S011
BLG0S020
BLG0S021
BLG0S060
BLG0S061
BLG0S500
BLG0S510
BLG0S530
BLG0TU01
BLG0TU02
BLG0T100
BLG0T101
BLG0T110
BLG0T120
BLG0T200
BLG0U000
BLG0U001
BLG0U100
BLG0U120
BLG0U200
BLG0V090
BLG0V100
BLG0V190
BLG0V800
BLG0W104
BLG0W105
BLG0W106
BLG0W107
BLG0W108
BLG0W109
BLG0W110
BLG0W111
BLG0W112
BLG0W113
BLG0W114
BLG0W115
BLG0W116
BLG0W117
BLG0W118
BLG0W119
BLG0W120
BLG0W121
BLG0W123
BLG0W200
BLG0W210
BLG0W250
BLG0W300
BLG0W301
BLG0W500
BLG0W510
BLG0X100
BLG0Y200
BLG0ZD50
BLG0ZU00
BLG0ZU50
BLG0Z100
BLG0Z500
BLG0Z510
BLG0Z530
BLG00001
BLG00002
BLG1AACP
BLG1AAUP
BLG1ACOP
BLG1ACTL
BLG1AC0A
BLG1AC00
BLG1AC01
BLG1AC02
BLG1AC04
BLG1AC05
BLG1AC06
BLG1AC07
BLG1AC08
BLG1AC09
BLG1AC1R
BLG1AC1S
BLG1AC1T
BLG1AC1U
BLG1AC1V
BLG1AC1W
BLG1AC10
BLG1AC11
BLG1AC12
BLG1AC13
BLG1AC14
BLG1AC15
BLG1AC16
BLG1AC17
BLG1AC18
BLG1AC19
BLG1AC2S
BLG1AC2W
BLG1AC20
BLG1AC21
BLG1AC22
BLG1AC23
BLG1AC24
BLG1AC25
BLG1AC26
BLG1AC27
BLG1AC28
BLG1AC29
BLG1AC30
BLG1AC31
BLG1AC32
BLG1AC33
BLG1AC34
BLG1AC35
BLG1AC36
BLG1AC37
BLG1AC39
BLG1AC40
BLG1AD00
BLG1AD01
BLG1AEXT
BLG1AF46
BLG1AF48
BLG1AF49
BLG1AF68
BLG1AJ00
BLG1AJ01
BLG1AJ20
BLG1AJ21
BLG1AJ30
BLG1AJ40
BLG1AJ60
BLG1AJ70
BLG1AO01
BLG1ARVC
BLG1AW00
BLG1AW10
BLG1AY00
BLG1A000
BLG1A001
BLG1A002
BLG1A004
BLG1A005
BLG1A006
BLG1A007
BLG1A011
BLG1A013
BLG1A014
BLG1A015
BLG1A016
BLG1A017
BLG1A018
BLG1A019
BLG1A060
BLG1A061
BLG1A081
BLG1A082
BLG1A083
BLG1A084
BLG1A085
BLG1A088
BLG1A091
BLG1A092
BLG1A093
BLG1A097
BLG1A098
BLG1A099
BLG1A1AA
BLG1A1AC
BLG1A1A1
BLG1A1A3
BLG1A1A4
BLG1A1A5
BLG1A1A8
BLG1A1A9
BLG1A1BA
BLG1A1BB
BLG1A1BC
BLG1A1B1
BLG1A1B3
BLG1A1B4
BLG1A1B5
BLG1A1B8
BLG1A1B9
BLG1A1CA
BLG1A1CC
BLG1A1CD
BLG1A1CE
BLG1A1C1
BLG1A1C3
BLG1A1C4
BLG1A1C5
BLG1A1C8
BLG1A1C9
BLG1A1DA
BLG1A1DB

BLG1A1DC
BLG1A1DE
BLG1A1DF
BLG1A1DG
BLG1A1D1
BLG1A1D2
BLG1A1D3
BLG1A1D4
BLG1A1D5
BLG1A1D6
BLG1A1D7
BLG1A1D8
BLG1A1D9
BLG1A1EA
BLG1A1EF
BLG1A1EG
BLG1A1E0
BLG1A1E1
BLG1A1E2
BLG1A1E3
BLG1A1E4
BLG1A1E5
BLG1A1E6
BLG1A1E7
BLG1A1E8
BLG1A1E9
BLG1A1FA
BLG1A1FB
BLG1A1FC
BLG1A1FD
BLG1A1FE
BLG1A1FF
BLG1A1F0
BLG1A1F1
BLG1A1F2
BLG1A1F3
BLG1A1F4
BLG1A1F5
BLG1A1F6
BLG1A1F7
BLG1A1F8
BLG1A1G2
BLG1A1G5
BLG1A1G8
BLG1A1G9
BLG1A10A
BLG1A100
BLG1A101
BLG1A102
BLG1A103
BLG1A104
BLG1A105
BLG1A106
BLG1A108
BLG1A109
BLG1A11A
BLG1A110
BLG1A111
BLG1A113
BLG1A114
BLG1A115
BLG1A116
BLG1A117
BLG1A118
BLG1A119
BLG1A120
BLG1A121
BLG1A123
BLG1A124
BLG1A125
BLG1A128
BLG1A129
BLG1A131
BLG1A133
BLG1A134
BLG1A135
BLG1A136
BLG1A137
BLG1A138
BLG1A139
BLG1A14A
BLG1A141
BLG1A143
BLG1A144
BLG1A146
BLG1A147
BLG1A148
BLG1A149
BLG1A15A
BLG1A15B
BLG1A15C
BLG1A15D
BLG1A15E
BLG1A15F
BLG1A151
BLG1A152
BLG1A153
BLG1A154
BLG1A155
BLG1A156
BLG1A157
BLG1A158
BLG1A159
BLG1A16A
BLG1A16B
BLG1A16C
BLG1A161
BLG1A162
BLG1A163
BLG1A164
BLG1A166
BLG1A168
BLG1A169
BLG1A17A
BLG1A17C
BLG1A171
BLG1A173
BLG1A174
BLG1A178
BLG1A179
BLG1A18B
BLG1A18C
BLG1A181
BLG1A182
BLG1A183
BLG1A184
BLG1A188
BLG1A189
BLG1A19A
BLG1A190
BLG1A191
BLG1A193
BLG1A194
BLG1A195
BLG1A197
BLG1A198
BLG1A199
BLG1A2AA
BLG1A2AB
BLG1A2AC
BLG1A2A1
BLG1A2A2
BLG1A2A3
BLG1A2A4
BLG1A2A8
BLG1A2A9
BLG1A200
BLG1A201
BLG1A202
BLG1A203
BLG1A21A
BLG1A21B
BLG1A21C
BLG1A210
BLG1A211
BLG1A212
BLG1A213
BLG1A214
BLG1A215
BLG1A218
BLG1A219
BLG1A220
BLG1A221
BLG1A222
BLG1A223
BLG1A224
BLG1A225
BLG1A226
BLG1A227
BLG1A228
BLG1A23A
BLG1A23C
BLG1A231
BLG1A232
BLG1A233
BLG1A234
BLG1A235
BLG1A236
BLG1A237
BLG1A238
BLG1A239
BLG1A242
BLG1A257
BLG1A258
BLG1A259
BLG1A260
BLG1A261
BLG1A262
BLG1A27A
BLG1A27B
BLG1A27C
BLG1A271
BLG1A273
BLG1A274
BLG1A275
BLG1A278
BLG1A279
BLG1A28B
BLG1A28C
BLG1A281
BLG1A282
BLG1A283
BLG1A284
BLG1A285
BLG1A288
BLG1A289
BLG1A29A
BLG1A29B
BLG1A29C
BLG1A291
BLG1A293
BLG1A294
BLG1A295
BLG1A298
BLG1A299
BLG1A300
BLG1A410
BLG1A411
BLG1A413
BLG1A414
BLG1A415
BLG1A416
BLG1A418
BLG1A419
BLG1A515
BLG1A911
BLG1A915
BLG1A918
BLG1PDS1
BLG1PDS2
BLG1PDS3
BLG1TADD
BLG1TADO
BLG1TARG
BLG1TAUP
BLG1TBAC
BLG1TCAC
BLG1TCCD
BLG1TCFE
BLG1TCMF
BLG1TCOL

BLG1TCON
BLG1TCOS
BLG1TCRL
BLG1TDAD
BLG1TDBA
BLG1TDDE
BLG1TDES
BLG1TDHD
BLG1TDJU
BLG1TDNO
BLG1TDRE
BLG1TDSD
BLG1TDSR
BLG1TDST
BLG1TESC
BLG1TFEL
BLG1TGLD
BLG1THLP
BLG1THVD
BLG1TJUS
BLG1TMOD
BLG1TMSG
BLG1TMSQ
BLG1TMSR
BLG1TMST
BLG1TMVS
BLG1TNOT
BLG1TPCL
BLG1TPTH
BLG1TREL
BLG1TRES
BLG1TSFE
BLG1TSMD
BLG1TSMF
BLG1TSQL
BLG1TSRC
BLG1TSRL
BLG1TSTA
BLG1TSTT
BLG1TSUB
BLG1TSYH
BLG1TTBL
BLG1TVCD
BLG1TVID
BLG1TVI1
BLG1TVI2
BLG1TVRD
BLG1T007
BLG1T022
BLG1T039
BLG1UT01
BLG1UT02
BLG1UT07
BLG1UT20
BLG1UT24
BLG1UT25
BLG1UT26
BLG1UT31
BLG1UT32
BLG1XMP1
BLG1XMP2
BLG1XMP3
BLG1XMP4
BLG1XMP5
BLG1XMP6
BLG1ZSTA
BLG2ABND
BLG2ALIS
BLG2APNU
BLG2CNAM
BLG2CRDN
BLG2DEVF
BLG2DEVL
BLG2DFEA
BLG2DMSN
BLG2DNUM
BLG2DOCN
BLG2DTYP
BLG2DU01
BLG2DU33
BLG2DU39
BLG2DVNA
BLG2DVSR
BLG2D000
BLG2D104
BLG2D701
BLG2ECLV
BLG2ECNU
BLG2ENVF
BLG2ENVS
BLG2ERCE
BLG2FLDF
BLG2FMIE
BLG2FNAM
BLG2G000
BLG2G100
BLG2G100
BLG2G130
BLG2G135
BLG2G175
BLG2G700
BLG2G730
BLG2G730
BLG2G735
BLG2IPID
BLG2LCRM
BLG2LIDE
BLG2LOCO
BLG2LSPE
BLG2LTYE
BLG2LVLF
BLG2LVLP
BLG2MNAM
BLG2MODF
BLG2MSGC
BLG2NPRO
BLG2NUPR
BLG2N140
BLG2N160
BLG2N740
BLG2OPCF
BLG2OPCS
BLG2PCSF
BLG2PIDF
BLG2PRID
BLG2PTFF
BLG2PTNU
BLG2PUBF
BLG2PUBS
BLG2PZFF
BLG2REAN
BLG2RMID
BLG2ROUF
BLG2ROUS
BLG2RPQN
BLG2SLEV
BLG2SPID
BLG2STMT
BLG2SUID
BLG2SWCC
BLG2SWRC
BLG2SYMC
BLG2TGLD
BLG2THVD
BLG2THVD
BLG2TMSQ
BLG2TSQL
BLG2TSRC
BLG2TSUB
BLG2UMID
BLG2UTID
BLG2UT07
BLG2UT20
BLG2W510
BLG2ZCMD
BLG2ZSC3
BLG20000
BLG6ABND
BLG6ABNQ
BLG6ACCN
BLG6ACNM
BLG6ACTN
BLG6AIMP
BLG6ALIS
BLG6ALMD
BLG6ALTO
BLG6APAR
BLG6APNQ
BLG6APNU
BLG6APPL
BLG6APQ1
BLG6APQ2
BLG6APQ3
BLG6APQ4
BLG6APQ5
BLG6ARID
BLG6ASID
BLG6ASMN
BLG6ASNT
BLG6AUPA
BLG6BLID
BLG6BLOC
BLG6CARD
BLG6CFMI
BLG6CHAN
BLG6CICI
BLG6CITR
BLG6CLAE
BLG6CLAR
BLG6CLAX
BLG6CLSN
BLG6CNAM
BLG6CNAQ
BLG6CNDP
BLG6CNQ1
BLG6CNQ2
BLG6CNQ3
BLG6CNQ4
BLG6CNQ5
BLG6COAC
BLG6CODE
BLG6COM2
BLG6CORQ
BLG6CPNM
BLG6CRDN
BLG6CRDQ
BLG6CRES
BLG6CSCN
BLG6DCID
BLG6DCNM
BLG6DCNO
BLG6DEPM
BLG6DEQN
BLG6DEVA
BLG6DEVF
BLG6DEVN
BLG6DEVQ
BLG6DFEA
BLG6DFEQ
BLG6DIMP
BLG6DISL
BLG6DLOC
BLG6DMQ1
BLG6DMQ2
BLG6DMQ3
BLG6DMQ4
BLG6DMQ5
BLG6DMSN
BLG6DMSQ
BLG6DNQ1
BLG6DNQ2
BLG6DNUM
BLG6DNUQ
BLG6DOCN
BLG6DOCQ
BLG6DOQ1
BLG6DOQ2
BLG6DPCR
BLG6DSRL
BLG6DTYP
BLG6DTYQ

BLG6DUPN
BLG6DVNA
BLG6DVNQ
BLG6DVQ1
BLG6DVQ2
BLG6DVQ3
BLG6DVQ4
BLG6DVQ5
BLG6DVSN
BLG6DVSQ
BLG6DVSR
BLG6DVTP
BLG6ECNQ
BLG6ECNU
BLG6ELGU
BLG6ENQV
BLG6ENQ1
BLG6ENQ2
BLG6ENQ3
BLG6ENQ4
BLG6ENQ5
BLG6ENVF
BLG6ENVQ
BLG6ENVS
BLG6ERCD
BLG6ERCE
BLG6ERCQ
BLG6FEAD
BLG6FEAN
BLG6FEAS
BLG6FILE
BLG6FINN
BLG6FIST
BLG6FLDF
BLG6FLDQ
BLG6FLVL
BLG6FMID
BLG6FMIE
BLG6FNAM
BLG6FNAQ
BLG6FNID
BLG6FNQ1
BLG6FNQ2
BLG6FNQ3
BLG6FNQ4
BLG6FNQ5
BLG6FNUM
BLG6FTYP
BLG6FXPN
BLG6HDCL
BLG6HDWL
BLG6IDAP
BLG6IMTR
BLG6INSR
BLG6IPID
BLG6IPIQ
BLG6IPQ1
BLG6IPQ2
BLG6IPQ3
BLG6IPQ4
BLG6IPQ5
BLG6KIAF
BLG6LCRM
BLG6LCRN
BLG6LCRQ
BLG6LGID
BLG6LIDE
BLG6LIDQ
BLG6LINT
BLG6LOCO
BLG6LOCQ
BLG6LOPF
BLG6LOPS
BLG6LOPT
BLG6LOQ1
BLG6LOQ2
BLG6LOQ3
BLG6LOQ4
BLG6LOQ5
BLG6LPID
BLG6LPRT
BLG6LSPE
BLG6LSPQ
BLG6LSTP
BLG6LTER
BLG6LTYE
BLG6LTYP
BLG6LTYQ
BLG6LVLA
BLG6LVLF
BLG6LVLP
BLG6LVLQ
BLG6LVLS
BLG6LVQF
BLG6MDID
BLG6MFGR
BLG6MLVL
BLG6MNAM
BLG6MNAQ
BLG6MNGR
BLG6MNQ1
BLG6MNQ2
BLG6MNQ3
BLG6MNQ4
BLG6MNQ5
BLG6MNSP
BLG6MODF
BLG6MODL
BLG6MODQ
BLG6MSGC
BLG6MSGQ
BLG6NETN
BLG6NIMP
BLG6NODN
BLG6NPQ1
BLG6NPQ2
BLG6NPRO
BLG6NPRQ
BLG6NUPQ
BLG6NUPR
BLG6NUQ1
BLG6NUQ2
BLG6ODES
BLG6OPCF
BLG6OPCQ
BLG6OPCS
BLG6OPQC
BLG6OPTH
BLG6OSHT
BLG6OUTC
BLG6OWDP
BLG6PANL
BLG6PART
BLG6PATH
BLG6PCSF
BLG6PCSQ
BLG6PERS
BLG6PFDS
BLG6PFIN
BLG6PHAC
BLG6PIDF
BLG6PIDQ
BLG6PIDS
BLG6PLOC
BLG6POWN
BLG6PRCA
BLG6PRCL
BLG6PRCO
BLG6PRID
BLG6PRIQ
BLG6PRIV
BLG6PRNM
BLG6PRNO
BLG6PRON
BLG6PRRQ
BLG6PRTI
BLG6PRTN
BLG6PRVS
BLG6PTCL
BLG6PTFF
BLG6PTFQ
BLG6PTFS
BLG6PTNQ
BLG6PTNU
BLG6PTQ1
BLG6PTQ2
BLG6PTQ3
BLG6PTQ4
BLG6PTQ5
BLG6PTRF
BLG6PTYP
BLG6PUBF
BLG6PUBQ
BLG6PUBS
BLG6PUQF
BLG6PU01
BLG6PZFF
BLG6PZFQ
BLG6PZFS
BLG6RDST
BLG6REAN
BLG6REAQ
BLG6REN1
BLG6REN2
BLG6REN3
BLG6REQN
BLG6REQ1
BLG6REQ2
BLG6REQ3
BLG6REQ4
BLG6REQ5
BLG6REVR
BLG6RFTN
BLG6RIDN
BLG6RIDS
BLG6RKIA
BLG6RLOC
BLG6RLVN
BLG6RMID
BLG6RMIQ
BLG6RMQ1
BLG6RMQ2
BLG6RMQ3
BLG6RMQ4
BLG6RMQ5
BLG6RNCT
BLG6RNOD
BLG6RNOR
BLG6RNU1
BLG6ROQU
BLG6ROUF
BLG6ROUQ
BLG6ROUS
BLG6RPQN
BLG6RPQQ
BLG6RPTY
BLG6RQDP
BLG6RQU1
BLG6RQU2
BLG6RQU3
BLG6RQU4
BLG6RQU5
BLG6RSKA
BLG6RUSR
BLG6RVUR
BLG6SA01
BLG6SDM1
BLG6SDM2
BLG6SDM3
BLG6SDM4
BLG6SDM5
BLG6SDM6
BLG6SDM7
BLG6SDM8
BLG6SERN
BLG6SFID
BLG6SHFT
BLG6SIMP
BLG6SLAN
BLG6SLEQ
BLG6SLEV

BLG6SLQ1
BLG6SLQ2
BLG6SLQ3
BLG6SLQ4
BLG6SLQ5
BLG6SMID
BLG6SPED
BLG6SPID
BLG6SPIQ
BLG6SPQ1
BLG6SPQ2
BLG6SPQ3
BLG6SPQ4
BLG6SPQ5
BLG6SP01
BLG6SP02
BLG6SP03
BLG6SP04
BLG6SP05
BLG6SP06
BLG6SP07
BLG6SP08
BLG6SP09
BLG6SP10
BLG6SP11
BLG6SP12
BLG6SP13
BLG6SP14
BLG6SP15
BLG6SP16
BLG6SP17
BLG6SP18
BLG6SQAU
BLG6SQMP
BLG6SRID
BLG6SRNA
BLG6SR01
BLG6SSGP
BLG6SSID
BLG6STMQ
BLG6STMT
BLG6STPA
BLG6STYP
BLG6SUBS
BLG6SUBT
BLG6SUBT
BLG6SUID
BLG6SUIQ
BLG6SUQ1
BLG6SUQ2
BLG6SUQ3
BLG6SUQ4
BLG6SUQ5
BLG6SVOR
BLG6SWCC
BLG6SWCQ
BLG6SWRC
BLG6SWRQ
BLG6SYMC
BLG6SYMQ
BLG6SYSN
BLG6UCHN
BLG6UMID
BLG6UMIQ
BLG6UMOD
BLG6UMQ1
BLG6UMQ2
BLG6UMQ3
BLG6UMQ4
BLG6UMQ5
BLG6UNAM
BLG6UPID
BLG6UPRN
BLG6URN0
BLG6URN1
BLG6USDE
BLG6USER
BLG6USNA
BLG6UTID
BLG6UTIQ
BLG6UTQ1
BLG6UTQ2
BLG6UTQ3
BLG6UTQ4
BLG6UTQ5
BLG6VCON
BLG6VNAM
BLG6VNDN
BLG6VNST
BLG6VNUM
BLG6VSEQ
BLG6VSER
BLG6XCLS
BLG60APN
BLG60ECA
BLG60REA
BLG60RPQ
BLG60SCN
BLG60VPN
BLG600AA
BLG600AG
BLG600CC
BLG600CG
BLG600CN
BLG600CR
BLG7ACNM
BLG7ALTO
BLG7CARD
BLG7CLAX
BLG7CSCN
BLG7DCID
BLG7DEVN
BLG7DUPN
BLG7DVSN
BLG7FEAN
BLG7FINN
BLG7LVLS
BLG7LVL1
BLG7PART
BLG7PFDS
BLG7PLOC
BLG7PNAM
BLG7PRIV
BLG7PTRF
BLG7PU01
BLG7PZFS
BLG7REVR
BLG7SA01
BLG7SP01
BLG7SRNA
BLG7SR01
BLG7SUBS
BLG7UMOD
BLG7UMOD
BLG70ECA
BLG70OIP
BLG70REA
BLG70RPQ
BLG70SCN
BLG70UP1
BLG8CSCN
BLG8DUPN
BLG8FINN
BLG8PNAM
BLG8PTRF
BLG8SP01
BLMTYU51
BLM0YE00
BLM0YE50
BLM0YU00
BLM0YU10
BLM0YU11
BLM0YU50
BLM0YU51
BLM0YU60
BLM0YU61
BLM0YU80
BLM0YU90
BLM0YU91
BLM0Y100
BLM0Y202
BLM0Y203
BLM0Y251
BLM0Y260
BLM0Y261
BLM0Y300
BLM0Y310
BLM0Y330
BLM0Y600
BLM0Y650
BLM0Y702
BLM0Y703
BLM0Y712
BLM0Y713
BLM0Y751
BLM0Y760
BLM0Y761
BLM0Y771
BLM0Y781
BLM0Y792
BLM0Y793
BLM0Y800
BLM0Y810
BLM0Y830
BLM0Y850
BLM0Y860
BLM0Y880
BLM1AA01
BLM1AA02
BLM1AA04
BLM1AC0A
BLM1AC0B
BLM1AC0C
BLM1AC0E
BLM1AC02
BLM1AC03
BLM1AC04
BLM1AC05
BLM1AC07
BLM1AC08
BLM1AC09
BLM1AC45
BLM1AC46
BLM1AC47
BLM1AC48
BLM1AD0A
BLM1AD0B
BLM1AD0C
BLM1AD0I
BLM1AD02
BLM1AD03
BLM1AD04
BLM1AD05
BLM1AD06
BLM1AD09
BLM1AF00
BLM1AH01
BLM1AH02
BLM1AMCP
BLM1AP0B
BLM1AP0C
BLM1AP0D
BLM1AP0E
BLM1AP0G
BLM1AP01
BLM1AP02
BLM1AP03
BLM1AP04
BLM1AP05
BLM1AP06
BLM1AP07
BLM1AP08
BLM1AP09
BLM1AP10
BLM1AP11
BLM1AS0B
BLM1AS0D
BLM1AS0E
BLM1AS0F
BLM1AS01
BLM1AS02

BLM1AS04
BLM1AS05
BLM1AS06
BLM1AS07
BLM1AS08
BLM1AS09
BLM1AS11
BLM1AT02
BLM1AU00
BLM1AU01
BLM1AU10
BLM1AU11
BLM1AU12
BLM1AU13
BLM1AU20
BLM1AU21
BLM1AU22
BLM1AU23
BLM1AU24
BLM1AU30
BLM1AU40
BLM1AU5A
BLM1AU5B
BLM1AU5C
BLM1AU5D
BLM1AU50
BLM1AU51
BLM1AU52
BLM1AU53
BLM1AU54
BLM1AU55
BLM1AU56
BLM1AU57
BLM1AU58
BLM1AU59
BLM1AU60
BLM1AU70
BLM1AU80
BLM1AU90
BLM1AV00
BLM1AV10
BLM1A008
BLM1A010
BLM1A020
BLM1A030
BLM1A050
BLM1A060
BLM1A201
BLM1A202
BLM1A213
BLM1A214
BLM1A215
BLM1A400
BLM1A801
BLM1A802
BLM1A804
BLM1A805
BLM1A806
BLM1A807
BLM1A808
BLM1A811
BLM1A812
BLM1A815
BLM1A817
BLM1A831
BLM1A832
BLM1A833
BLM1A834
BLM1A835
BLM1A836
BLM1A837
BLM1A838
BLM1A839
BLM1A851
BLM1TDP0
BLM1TDP1
BLM1TDS0
BLM1TDS1
BLM1THSC
BLM1TPAL
BLM1TPDS
BLM1TSAC
BLM1TSCU
BLM1TSU1
BLM1TSU2
BLM1TSU4
BLM1TSU5
BLM1TSU6
BLM1TSU7
BLM1TSU8
BLM1TSU9
BLM1TUCU
BLM1T207
BLM1YCCH
BLM1YU51
BLM1Y109
BLM1Y11A
BLM1Y111
BLM1Y113
BLM1Y114
BLM1Y115
BLM1Y116
BLM1Y118
BLM1Y119
BLM1Y120
BLM1Y61A
BLM1Y61B
BLM1Y611
BLM1Y613
BLM1Y614
BLM1Y615
BLM1Y616
BLM1Y618
BLM1Y619
BLM1Y620
BLM1Y630
BLM1Y640
BLM1Y641
BLM1Y653
BLM1Y654
BLM1Y655
BLM1Y656
BLM1Y657
BLM1Y658
BLM1Y659
BLM2FD17
BLM2FD39
BLM2FD45
BLM2FD46
BLM2FD78
BLM6APPC
BLM6BDRP
BLM6CRTA
BLM6FATA
BLM6FINA
BLM6FNAM
BLM6FRFN
BLM6FRPA
BLM6FUEX
BLM6HEPA
BLM6HETA
BLM6INLE
BLM6INTA
BLM6KEPH
BLM6LANA
BLM6NCTA
BLM6NITA
BLM6NPNL
BLM6NRTA
BLM6PAID
BLM6PANA
BLM6PARA
BLM6PARM
BLM6PETN
BLM6PGUS
BLM6STPN
BLM6SWDI
BLM6TFNA
BLM6TOFN
BLM6TOPA
BLM6TRTA
BLM6VENA
BLM6YCHI
BLM6Y00A
BLM6Y00B
BLM6Y00C
BLM6Y00D
BLM6Y00F
BLM6Y00H
BLM6Y00J
BLM6Y00L
BLM6Y001
BLM6Y002
BLM6Y003
BLM6Y006
BLM8CCA0
BLM8CC00
BLM8CC10
BLM8CD00
BLM8CF00
BLM8CF10
BLM8CL10
BLM8CL20
BLM8CL30
BLM8CL50
BLM8CL60
BLM8CT00
BLM8CU00
BLM8CU1A
BLM8CU1B
BLM8CU1E
BLM8CU1H
BLM8CU11
BLM8CU12
BLM8CU13
BLM8CU17
BLM8CU2A
BLM8CU2B
BLM8CU2E
BLM8CU2H
BLM8CU21
BLM8CU22
BLM8CU23
BLM8CU27
BLM8CU31
BLM8CU37
BLM8CU41
BLM8CU43
BLM8CU47
BLM8CU5A
BLM8CU5D
BLM8CU5E
BLM8CU5I
BLM8CU51
BLM8CU53
BLM8CU55
BLM8CU57
BLM8CU6A
BLM8CU6B
BLM8CU61
BLM8CU63
BLM8CU64
BLM8CU67
BLM8CU7A
BLM8CU7E
BLM8CU7H
BLM8CU71
BLM8CU72
BLM8CU73
BLM8CU77
BLM8CU79
BLM8CU8A
BLM8CU81
BLM8CU83
BLM8CU85
BLM8CU87
BLM8CU9A
BLM8CU9B
BLM8CU9C
BLM8CU9D
BLM8CU9E

BLM8CU9F
BLM8CU9G
BLM8CU9H
BLM8CU9I
BLM8CU9J
BLM8CU9K
BLM8CU9L
BLM8CU9M
BLM8CU9N
BLM8CU9O
BLM8CU9P
BLM8CU9Q
BLM8CU9R
BLM8CU9S
BLM8CU9T
BLM8CU9U
BLM8CU9V
BLM8CU9W
BLM8CU9X
BLM8CU9Y
BLM8CU9Z
BLM8CU91
BLM8CU97
BTNMIGD1
BTNMIGSA
BTNNOTCE
BTNRFUSR
BTNTACLS
BTNTAC1R
BTNTAC1S
BTNTAC1U
BTNTAC1V
BTNTAC1W
BTNTAM01
BTNTAM02
BTNTAPRV
BTNTARPP
BTNTARUP
BTNTASAD
BTNTASAN
BTNTASCI
BTNTASCP
BTNTASCR
BTNTASCS
BTNTASGP
BTNTASPN
BTNTASPT
BTNTASRP
BTNTASUP
BTNTCAAP
BTNTCADC
BTNTCAHW
BTNTCANW
BTNTCAPC
BTNTCASW
BTNTCA21
BTNTCEAE
BTNTCEAP
BTNTCEA2
BTNTCEDC
BTNTCEF2
BTNTCEHW
BTNTCENW
BTNTCEPC
BTNTCESW
BTNTCE10
BTNTCE11
BTNTCN01
BTNTCN02
BTNTCN03
BTNTCN04
BTNTCN05
BTNTCN06
BTNTCN07
BTNTCN08
BTNTCN09
BTNTCONT
BTNTCSLK
BTNTCSTT
BTNTCST1
BTNTCS01
BTNTCS02
BTNTCS03
BTNTCS04
BTNTCS07
BTNTC125
BTNTDRAW
BTNTDU08
BTNTENAS
BTNTENCC
BTNTENCF
BTNTENDS
BTNTENES
BTNTENHD
BTNTENNS
BTNTENNV
BTNTENOM
BTNTENPC
BTNTENPS
BTNTENSA
BTNTENSS
BTNTENSY
BTNTENTR
BTNTEN20
BTNTFS01
BTNTFS02
BTNTFS03
BTNTFS04
BTNTFS05
BTNTFS06
BTNTFS07
BTNTFS08
BTNTG001
BTNTG002
BTNTHS01
BTNTHS02
BTNTHS03
BTNTHS05
BTNTHS06
BTNTHS07
BTNTLA01
BTNTLA02
BTNTLA03
BTNTLA04
BTNTLA05
BTNTLA06
BTNTLA07
BTNTLA08
BTNTLA09
BTNTLA10
BTNTLA11
BTNTLA12
BTNTLA13
BTNTLA14
BTNTLA15
BTNTNPDA
BTNTNRES
BTNTNS01
BTNTPC01
BTNTPC02
BTNTPC03
BTNTPC04
BTNTPC05
BTNTPC06
BTNTPC07
BTNTPC08
BTNTPC09
BTNTPC10
BTNTPC11
BTNTPC12
BTNTPC13
BTNTPC14
BTNTPC15
BTNTPE03
BTNTPE04
BTNTPE05
BTNTPE06
BTNTPE07
BTNTPE08
BTNTPE09
BTNTPE30
BTNTPF03
BTNTPF04
BTNTPF05
BTNTPF08
BTNTPF09
BTNTPRI1
BTNTPSNV
BTNTPSTT
BTNTPST1
BTNTPS01
BTNTPS04
BTNTPS05
BTNTPS07
BTNTSRC7
BTNTSR13
BTNTS027
BTNTS028
BTNTTOTT
BTNTW020
BTNTW030
BTNT00P1
BTN0BU00
BTN0BU10
BTN0BU11
BTN0B001
BTN0B002
BTN0B010
BTN0B101
BTN0B102
BTN0B200
BTN0B300
BTN0B500
BTN0CETX
BTN0CU00
BTN0CU01
BTN0C101
BTN0C200
BTN0C300
BTN0DU01
BTN0DU03
BTN0DU18
BTN0DU28
BTN0DU30
BTN0DU31
BTN0D100
BTN0D102
BTN0D104
BTN0D200
BTN0D500
BTN0EN20
BTN0EPPC
BTN0E090
BTN0E190
BTN0E290
BTN0E390
BTN0E890
BTN0F090
BTN0F190
BTN0F290
BTN0F390
BTN0G1A1
BTN0G1A3
BTN0G19A
BTN0G190
BTN0G191
BTN0H000
BTN0JU00
BTN0J100
BTN0J200
BTN0J300
BTN0L100
BTN0L200
BTN0L300
BTN0L500
BTN0M010
BTN0M100
BTN0M200
BTN0N100
BTN0N101
BTN0N134

BTN0N200
BTN0N201
BTN0Q100
BTN0Q200
BTN0Q300
BTN0S010
BTN0S020
BTN0S021
BTN0W001
BTN0W002
BTN0W003
BTN0W004
BTN0W005
BTN0W006
BTN0W007
BTN0W008
BTN0W009
BTN0W010
BTN0W011
BTN0W012
BTN0W013
BTN0W014
BTN0W015
BTN0W016
BTN0W017
BTN0W018
BTN0W019
BTN0W020
BTN0W021
BTN0W022
BTN0W023
BTN0W024
BTN0W025
BTN0W026
BTN0W027
BTN0W028
BTN0W029
BTN0W030
BTN0XAUT
BTN0XDES
BTN0XHID
BTN0XSUM
BTN0XUSR
BTN1AC0A
BTN1AC04
BTN1AC07
BTN1AC1R
BTN1AC1S
BTN1AC1U
BTN1AC1V
BTN1AC1W
BTN1ADRW
BTN1AECH
BTN1AECN
BTN1AM01
BTN1AM02
BTN1ARPP
BTN1ASAD
BTN1ASAN
BTN1ASCI
BTN1ASCP
BTN1ASCR
BTN1ASGP
BTN1ASPN
BTN1ASPT
BTN1ASRP
BTN1ASUP
BTN1AX03
BTN1AX06
BTN1AX11
BTN1AX12
BTN1AX13
BTN1A011
BTN1A015
BTN1A1CA
BTN1A1C3
BTN1A1C4
BTN1A1C8
BTN1A1C9
BTN1A1E2
BTN1A1E3
BTN1A1E7
BTN1A1E8
BTN1A102
BTN1A111
BTN1A112
BTN1A113
BTN1A114
BTN1A115
BTN1A118
BTN1A119
BTN1A120
BTN1A121
BTN1A123
BTN1A125
BTN1A128
BTN1A133
BTN1A141
BTN1A145
BTN1A146
BTN1A152
BTN1A158
BTN1A172
BTN1A178
BTN1A190
BTN1BPPP
BTN1CAT1
BTN1CCT3
BTN1CCT4
BTN1CEAE
BTN1CEF1
BTN1CENL
BTN1CETY
BTN1CE10
BTN1CE11
BTN1CSLK
BTN1CSTT
BTN1CST1
BTN1CS01
BTN1CS02
BTN1CS03
BTN1CS04
BTN1CS07
BTN1CT02
BTN1C125
BTN1DUPN
BTN1EE01
BTN1ENAS
BTN1ENCC
BTN1ENCF
BTN1ENDS
BTN1ENES
BTN1ENHD
BTN1ENHS
BTN1ENNS
BTN1ENNV
BTN1ENOM
BTN1ENPC
BTN1ENPS
BTN1ENSA
BTN1ENSS
BTN1ENSY
BTN1FS01
BTN1FS06
BTN1FS07
BTN1FS08
BTN1G001
BTN1G002
BTN1HS01
BTN1HS02
BTN1HS03
BTN1HS06
BTN1HS07
BTN1INIT
BTN1NE01
BTN1NPDA
BTN1NS01
BTN1PE01
BTN1PE29
BTN1PE30
BTN1PF01
BTN1PF02
BTN1PF03
BTN1PF04
BTN1PF06
BTN1PRII
BTN1PRI1
BTN1PSNV
BTN1PSTT
BTN1PST1
BTN1PS01
BTN1PS02
BTN1PS03
BTN1PS04
BTN1PS05
BTN1PS07
BTN1PTOT
BTN1P1E2
BTN1REQ1
BTN1SRC7
BTN1SR13
BTN1S008
BTN1S026
BTN1S027
BTN1S028
BTN1TADO
BTN1TAUP
BTN1TBAC
BTN1TDDE
BTN1TDES
BTN1TDSD
BTN1TDST
BTN1TINS
BTN1TJUS
BTN1TPS1
BTN1TSAD
BTN1TSAN
BTN1TSC1
BTN1TSC2
BTN1TSC3
BTN1TSC4
BTN1TSF1
BTN1TSF2
BTN1TSF3
BTN1TSF4
BTN1TSF5
BTN1TSF6
BTN1TSN1
BTN1TSRN
BTN1TSSE
BTN1TSST
BTN1TSTT
BTN1XREF
BTN100CN
BTN100CR
BTN100PA
BTN100PR
BTN100P1
BTN100TN
BTN20000
BTN6ACNM
BTN6BLOC
BTN6COM2
BTN6DEVN
BTN6DUPN
BTN6DUP1
BTN6DVNM
BTN6FMID
BTN6LCRN
BTN6LVLS
BTN6OCCT
BTN6OTIM
BTN6PLOC
BTN6PNAM
BTN6PU01
BTN6REQN
BTN6REQ1
BTN6RQDP
BTN6RQNM
BTN6STAT
BTN6TART

BTN6TIMX	BTN6XRF1	BTN600CR	BTN7ALTO
BTN6TNDP	BTN600AA	BTN600C1	BTN7CLOC
BTN6URN2	BTN600AG	BTN600PA	BTN7PLOC
BTN6URN4	BTN600CC	BTN600PR	BTN700TN
BTN6XREF	BTN600CN	BTN600TN	

Removed Panels – Information/Management Version 6.2

BLM6SNJU

New Panels – Information/Management Version 6.1

BLGDUMP0	BLG0S510	BLG6HBDN	BLM1AU5C
BLGDUMP1	BLG0S530	BLG6HBFD	BLM1AU5D
BLGLOAD0	BLG0ZD50	BLG6HBFP	BLM1AU58
BLGLOAD1	BLG0ZD51	BLG6HBQN	BLM1AU59
BLGLSQMP	BLG0ZU50	BLG6HBRL	BLM6DPFX
BLGRFHBK	BLG0Z500	BLG6HBSR	BLM6DTYP
BLGSSQMP	BLG0Z510	BLG6HBTI	BLM6NPNL
BLGTDBXM	BLG0Z530	BLG6ODES	BLM6PCGZ
BLGTDBX1	BLG1A091	BLG6PCPO	BLM6SVAL
BLGTSPN1	BLG1A515	BLG6RUSR	BLM6UCTL
BLGTZSTA	BLG1TDIS	BLG6SQAU	BLM6VDAO
BLG0AC37	BLG1ZSTA	BLG6SQMP	BLM6XFIN
BLG0AC43	BLG1Z500	BLG6ULCD	BLM8CU9U
BLG0AC44	BLG1Z511	BLG6URN4	BLM8CU9V
BLG0AC45	BLG1Z515	BLMTD247	BLM8CU9W
BLG0P101	BLG6HBAD	BLMTD248	BLM8CU9Y
BLG0SU50	BLG6HBBZ	BLM1AU5A	BTN7TARD
BLG0S500	BLG6HBCS	BLM1AU5B	

Changed Panels – Information/Management Version 6.1

BLGAPI02
BLGAPI05
BLGAPI10
BLGESCAL
BLGESC03
BLGLALSD
BLGNOTAE
BLGNOTAU
BLGNOTCE
BLGNOTCU
BLGNOTPE
BLGNOTPU
BLGTAC1R
BLGTAC1S
BLGTAC1T
BLGTAC1U
BLGTAC1V
BLGTAC1W
BLGTDFLT
BLGTENDL
BLGTENDS
BLGTEXST
BLGTPSET
BLGTPU00
BLGTSPAE
BLGTSPAU
BLGTSPCE
BLGTSPCU
BLGTSPPE
BLGTSPPU
BLG0AAE0
BLG0AC12
BLG0AC22
BLG0AC23
BLG0AC24
BLG0AC26
BLG0AC27
BLG0AC28
BLG0AC29
BLG0AC35
BLG0AC36
BLG0AC39
BLG0AC41
BLG0AC42
BLG0B402
BLG0B500
BLG0B502
BLG0C051
BLG0C052
BLG0C053
BLG0C054
BLG0C055
BLG0C056
BLG0C061
BLG0C062
BLG0C063
BLG0C064
BLG0C065
BLG0C066
BLG0C071
BLG0C072
BLG0C073
BLG0C485
BLG0DU14
BLG0DU15
BLG0DU24
BLG0DU25
BLG0DU51
BLG0DU67
BLG0D000
BLG0D320
BLG0EN10
BLG0EN20
BLG0F121
BLG0F122
BLG0F221
BLG0F222
BLG0H000
BLG0PU00
BLG0P100
BLG0P410
BLG0P510
BLG0P810
BLG0W000
BLG0W500
BLG00002
BLG00010
BLG1A1G5
BLG1A100
BLG1A210
BLG1TVID
BLG1TVI1
BLG1TVI2
BLG1T007
BLG6ACTN
BLG6ALTD
BLG6ASSD
BLG6CDFR
BLG6CDTO
BLG6CMPD
BLG6CPED
BLG6CPSD
BLG6CPUR
BLG6CRDT
BLG6DATX
BLG6DNOT
BLG6DSAB
BLG6DSTA
BLG6EDSW
BLG6INSD
BLG6LOFI
BLG6NPED
BLG6NPSD
BLG6OCCD
BLG6OUTY
BLG6PPED
BLG6PPSD
BLG6PRCO
BLG6PURD
BLG6REQD
BLG6SCHD
BLG6STAC
BLG6TARD
BLG6UNIT
BLG6VPAD
BLG600CD
BLG600CT
BLG7ACNM
BLG7DSAB
BLM0AC05
BLM0AC09
BLM0AC10
BLM0Y791
BLM0Y792
BLM1AA04
BLM1AC0A
BLM1AC00
BLM1AC02
BLM1AD0G
BLM1AD01
BLM1AD05
BLM1AD06
BLM1AD08
BLM1AP0A
BLM1AP0B
BLM1AS0B
BLM1AT01
BLM1AU02
BLM1AU30
BLM1AU50
BLM1AU51
BLM1AU52
BLM1AU53
BLM1AU54
BLM1AU55
BLM1AU56
BLM1AU57
BLM1A40J
BLM1A40M
BLM1A806
BLM1A807
BLM1S201
BLM1S203
BLM1T206
BLM1T207
BLM1T221
BLM1T223
BLM1T224
BLM1T225
BLM1T226
BLM1T228
BLM1T229
BLM1T255
BLM6APPT
BLM6INLE
BLM6NEWD
BLM6NOFO
BLM6PREX
BLM6STAN
BLM6STDA
BLM6TFNA
BLM6VADA
BLM6Y00E
BLM6Y007
BLM8CU17
BLM8CU27
BLM8CU31
BLM8CU37
BLM8CU39
BLM8CU47
BLM8CU5D
BLM8CU53
BLM8CU55
BLM8CU57
BLM8CU6B
BLM8CU61
BLM8CU67
BLM8CU69
BLM8CU7B
BLM8CU77
BLM8CU87
BLM8CU9J
BLM8CU9M
BLM8CU91
BLM8CU97
BLM8CU99
BTNTA112
BTNTBPPP
BTNTCE11
BTNTCN07
BTNTCS03
BTNTENTR
BTNTNPDA
BTNTPN02
BTNTPN03
BTNT00P1
BTN0B002
BTN0D000
BTN0D100
BTN0D102
BTN0D500
BTN0ENSY
BTN0EN20
BTN0E000
BTN0G191
BTN1A113
BTN1NPDA
BTN1PE01
BTN6ALTD
BTN6ASSD
BTN6CDES
BTN6CDE1
BTN6CMDP
BTN6CMPD
BTN6CRDT
BTN6CST1
BTN6CTGY
BTN6DATE
BTN6DATX
BTN6DNOT
BTN6DSAB

BTN6DSTA	BTN6OCCD	BTN6STAT	BTN7CMPD
BTN6DSTT	BTN6PRII	BTN6TARD	BTN7CTGY
BTN6ENDD	BTN6REQD	BTN600CD	BTN7STAC
BTN6JDES	BTN6SCHD	BTN600CT	

Removed Panels – Information/Management Version 6.1

BLG0P600	BLG0P610	BLG0P621	BLG1A009
BLG0P601	BLG0P611	BLG0P630	BLG6LDMD
BLG0P602	BLG0P620	BLG0P631	

K

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 407 for more information about each book.

Typical Tasks

Table 24. 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 24. 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 24. 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*

L

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.

M

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

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

P

R

S

T

U

V

W

X

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.

GC31-8751-00